Big data and remote sensing: A new software of ingestion

Badr-Eddine Boudriki Semlali[1], Chaker El Amrani[2]
[1,2]LIST Laboratory, Faculty of Sciences and Techniques of Tangier, Abdelmalek Essaâdi University, Morocco
[1]CommSensLab-UPC, Department of Signal Theory and Communications (TSC), UPC BarcelonaTech, Barcelona, Spain

Article Info

Article history:

Received Apr 5, 2020
Revised Jul 27, 2020
Accepted Sep 23, 2020

Keywords:

Big data
Data integration
Data preprocessing
Hadoop framework
Satellite sensors data

ABSTRACT

Currently, remote sensing is widely used in environmental monitoring applications, mostly air quality mapping and climate change supervision. However, satellite sensors occur massive volumes of data in near-real-time, stored in multiple formats and are provided with high velocity and variety. Besides, the processing of satellite big data is challenging. Thus, this study aims to approve that satellite data are big data and proposes a new big data architecture for satellite data processing. The developed software is enabling an efficient remote sensing big data ingestion and preprocessing. As a result, the experiment results show that 86 percent of the unnecessary daily files are discarded with a data cleansing of 20 percent of the erroneous and inaccurate plots. The final output is integrated into the Hadoop system, especially the HDFS, HBase, and Hive, for extra calculation and processing.

Corresponding Author:

Badr-Eddine Boudriki Semlali
Department of Computer Engineering
Abdelmalek Essaâdi University of Tangier
The old way of airport, Km 10, Ziaten, PO. Box 416, Tangier, Morocco
Email: badreddine.boudrikisemlali@uae.ac.ma, semlali.badro@gmail.com

1. INTRODUCTION

The considerable rise of industrial, transport, and agricultural activities has led to many environmental issues. Notably, the outdoor air pollution (AP) due to the emission of many anthropogenic pollutants such as the monoxide of carbon (CO), dioxide of carbon (CO_2), nitrogenous of oxide (NO_x), methane (CH_4), and so on [1]. Consequently, AP can seriously affect human health and catalyze climate change. For this reason, air quality (AQ) currently deserves special attention from several scientific communities. Indeed, continuous AQ monitoring is one of the proposed solutions helping to decision-makers [2]. It enables a near-real-time (NRT) monitoring of the aerosol optical depths (AOD), it provides a potential input data for AQ models, tracks the pollutant plumes emitted from industrial and agricultural areas, [2] and estimates the ozone (O_3) precursor.

Generally, the remote sensing (RS) technique refers to using the satellite sensors to measure the ocean, Earth, and atmospheric components thought electromagnetic energy without making physical contact with it [3]. Nowadays, there are more than 3,000 satellites in orbits [4] used in many purposes, such as military, earth observation weather, and forecasting support. All these satellites are instrumented with many sensors within different temporal (TMR), spatial (SPR), and spectral resolutions (STR) ranging from low to high [5]. Satellites sensors measure data, and then they transmit data into ground stations through downlink channels. In our survey, we collect data from the European Organization for The Exploitation of Meteorological Satellites (EUMETSAT) via the Mediterranean Dialogue Earth Observatory (MDEO) ground station installed at the Abdelmalek Essaâdi University (UAE) of Tangier in Morocco [6]. We also obtained RS data from the Earth Observation System Data and Information System (EOSDIS) of the National

Aeronautics and Space Administration (NASA), the Infusing Satellite Data into Environmental Applications (NESDIS) of the National Oceanic and Atmospheric Administration (NOAA), and the Copernicus Open Access Hub (COAH) platform operated by the European Space Agency (ESA). The RS data gathered come from many polar and geostationary satellites and various sensors.

Moreover, these data are stored in a complicated scientific file format precisely: The binary universal form for the representation (BUFR) of meteorological data [7], the network common data form (NetCDF) [8], the hierarchical data format (HDF5) [9] and so on. The daily volume of the received RS data reaches many gigabits (GB) and sumps up terabits (TB) per year. Furthermore, the velocity with which data is collected is fast, with a rate of thousands of files per day. RS data are considered as big data (BD) according to attribute definition (venue, volume, variety, veracity, velocity, value, vocabulary, and validity) [10]. Consequently, the processing is challenging and also take a vital execution time. For this aim, we have designed an original BD architecture to split and facilitate problem-solving the problems of RS BD [11].

In this manuscript, we will show the mechanisms and results of the developed ingestion layer. Thus, this phase is very critical because it is responsible for collecting unprocessed RS data, for handling an enormous volume of input data, and for extracting and filtering it. As a result, the ingestion layer has proceeded efficiently and obtained potential values with high accuracy, low volume, and reasonable execution time. This layer has performed all steps automatically and processed global RS global data in NRT [12]. Moreover, the Hadoop has stored the ingested RS data efficiently for extra processing and calculation.

2. BACKGROUND

RS term was invented in the 1960s by Evelyn Pruitt when working at the US office of Naval Research [13]. In 1849, was the first proposed attempt to make a topographic mapping over hundreds of the altitude. In 1858, French scientists endeavored to capture a ground image using a balloon equipped by a camera. In 1909, a German scientist attached a camera to pigeons to take images when flying over the fair [3]. Currently, RS generally refers to using the technology for measuring the specification of surface, ocean, and atmospheric components without making physical contact with them, by using electromagnetic energy [4]. RS techniques have become widely employed in several environmental applications, notably:

- AP occurs when harmful or extreme quantities of gases are emitted to the Earth's atmosphere. It may cause illnesses, including respiratory infections, heart disease, allergies, and death to humans; besides, it may also harm animals, vegetation, and the natural environment. AP could be generated both from human activity and natural processes [14].
- The O_3 hole and global warming the O_3 hole is a zone in the stratosphere overhead Antarctica where pollutant gases have destroyed O_3 particles. Global warming is the growth in the usual global surface temperature produced principally by the emission of greenhouses gases, frequently CO_2 and CH_4, which block heat in the lower levels of the atmosphere [15].
- Climate change and agriculture are interconnected processes. Global warming impacts agriculture in several ways, mainly through fluctuations in average temperatures, heat, and rainfall waves; changes in atmospheric CO_2 and ground-level O_3 density; and changes in sea level. Climate change is even now negatively affecting agriculture, instigating imminent land infertility, and the movement of local species [16].
- Forest fires have a big impact on the Earth's ecosystems and climate, for it considerably alters the landscape and vegetation zones and emits significant amounts of greenhouse gases and AOD. A forest fire can be regarded as a real environmental catastrophe, whether produced by natural causes or human activity. Hence, it is incredible to control nature, but it is possible to map forest fire risk regions and thereby reduce the incidence of fire and prevent damage [17].
- Natural disasters are a critical adverse event consequential from natural processes of the Earth, such as are floods, hurricanes, volcanic eruptions, earthquakes, storms, and so on. A natural disaster can kill lives or damage property and naturally cause some economic losses in its wake, which can need years of restoration [18].

In this research, we have used RS data to monitor the atmosphere, troposphere, sea, and surface component, as divided in Figure 1. We have collected satellite data of weather variables such as temperature, humidity, wind speed, and air pressure. Besides, we acquire data of clouds, trace gases (TG), AOD, land surface temperature (LST), forest fires, and so on, as shown in Figure 1.

Satellites are the principal instrument of the measurement. They use sensors within different TMR, SPR, STR [5]. Most of the used sensors have a Medium to high SPR. In contrast, there is 15 percent of the very high resolution, including the ASTER, VIIRS, and OLCI sensors. In addition to 11 percent of the low SPR, such as AMSU and ASCAT instruments. Moreover, the average TMR is for two days. Besides, the most used STR is the microwave (MW) and infrared (IR), the rest are visible (V), ultraviolet (UV), as shown in Figure 2.

Satellites always pass by unique orbits. This orbit can be polar or geostationary [19]. Concerning their latency, we notice that polar satellites of EUMETSAT delay about 35 min. The MSG provides data every 15 min [20], NASA and NOAA satellites sensors data take between one to three hours after their measurement, and the ESA satellites distribute data via the COAH after 2 hours of processing [21].

In this study, we apply RS techniques to supervise the AQ of Morocco and monitor climate changes in NRT. We acquired data from four organizations: the MDEO, NASA, the NOAA, and the ESA. Thus, a satellite the most used in our investigation are polar, excepting the meteosat second generation (MSG) is geostationary [20].

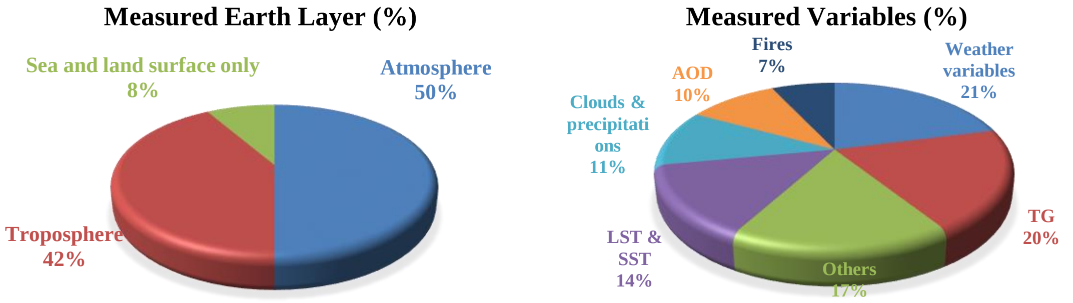


Figure 1. The measured earth components and environmental variables

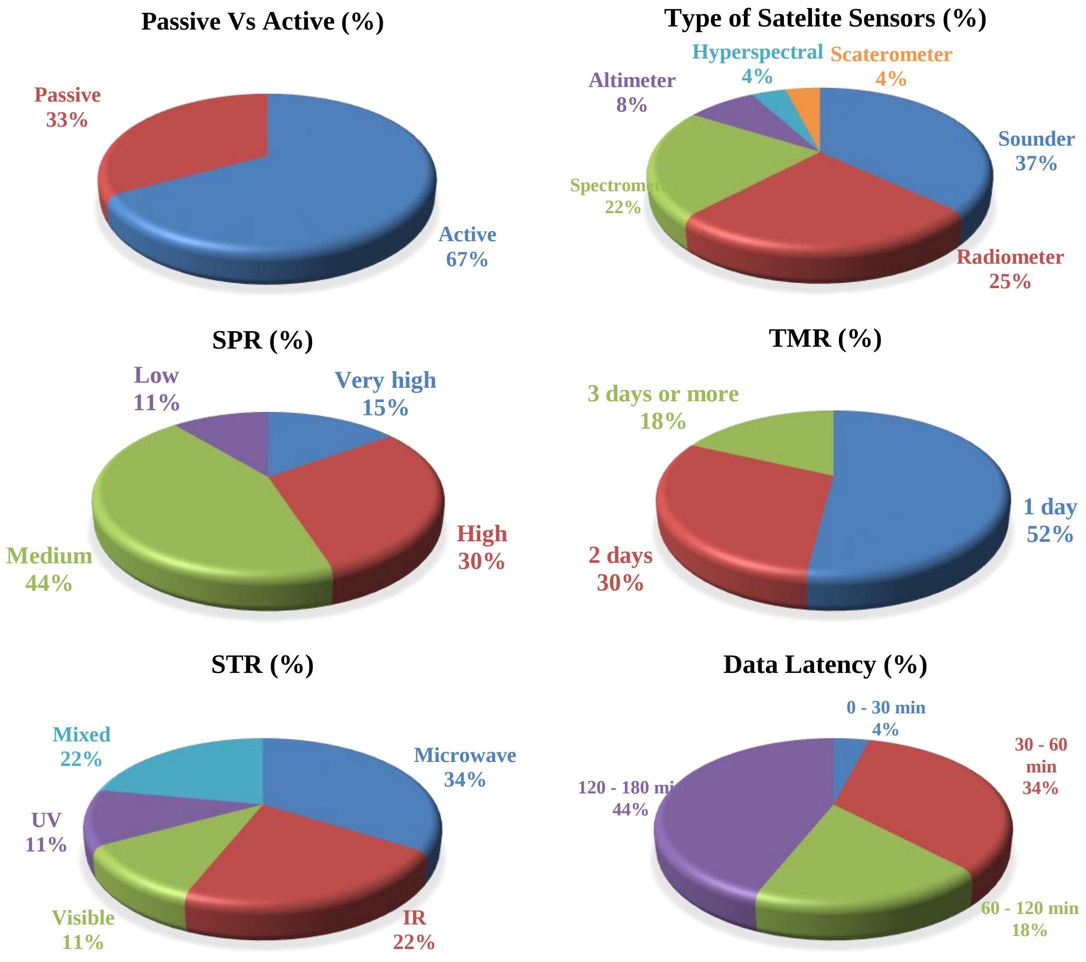


Figure 2. Used satellite sensors specifications

We have exploited many active and passive satellite sensors, such as IASI, AMSU, VIIRS, GOME-2, ASCAT, TROPOMI, and so on. We have used 67 percent of active sensors, and the rest is passive, and the most used sensors type is sounder, with 37 percent, as shown in Figure 2. In our work, RS data come with a high velocity reaching 40,000 files per day with an average latency of 30 minutes. These data continuously increase the storage space with 55 GB per day. The collected data are stored in scientific file format, particularly the NetCDF, HDF5, and BUFR as illustrates in Figure 3. Consequently, RSBD turns out to be challenging problems to be treated. The processing chain of the RSBD includes many challenges. Which are:

- The complexity and size of satellite data: RS data usually come with a high velocity reaching thousands of files per day, increasing the storage space continuously with TB per day. Generally, RS data are stored in scientific files format, particularly the NetCDF, HDF5, and BUFR. Consequently, RSBD turns out to be an extremely challenging problem to be treated.

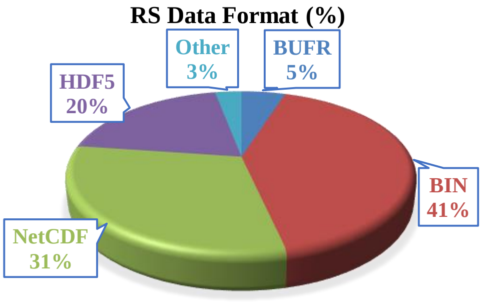


Figure 3. Used RS data format

- RS data velocity and freshness: Satellite sensors provide a fast and permanent stream of data. Thus, NRT monitoring requires rapid processing, which guarantees and keep RS data freshness.
- RS data quality and gaps: Satellite data sometimes comprise errors, gaps, and invalid datasets. It is recommended to remove them before the storage step.
- RS data processing: The existing architectures and solutions have some limitations and drawbacks in RS data ingestion and integration; it requires an ingenious paradigm of processing and scalable platform of processing.
- RSBD processing demands mathematical knowledge in probability and statistics to employ deep learning, machine learning, and neural network algorithms to unlock new insights.

Consequently, many kinds of research have been conducted on different architectures to solve RS data processing issues. These investigations aim mainly to employ parallel computing via the integration and the exploitation of the hardware's capacity [22] to store and process RSBD inside distributed clusters, notably the Hadoop [23]. To optimize algorithms and the processing patterns [24], and to process RS data in streaming [25].

3. METHOD

In this subsection, we illustrate the architecture of the ingestion layer and show the key parts of the developed software of RS data preprocessing, and explain how to integrate the output datasets into the Hadoop storage system especially, the Hadoop distributed file system (HDFS), HBase, and the Hive.

3.1. The data ingestion

We have proposed a BD architecture enabling an efficient RS processing. This architecture is composed of six layers which are: the data sources, the ingestion layer, the Hadoop storage, the processing and management layer, and finally, the visualization layer [26]. In this paper, we focus our clarifications on the ingestion layer. The ingestion layer is an inherent part of the proposed BD architecture. It is responsible for preprocessing RS data [27]. Thus, the acquired data are firstly stored and then processed. As a result, data staging is batch processing. Before beginning the design and the development of this layer, we have done many pieces of research to understand the nature and the characteristics of satellite data. The developed ingestion layer acquires, decompresses, filters, converts, and extracts refined information and datasets from enormous RSBD input [28]. JAVA, Python, and Shell were the principal programming languages used in development.

3.2. The data ingestion layer: SAT-ETL-integrator software

Figure 4 explains the different phases of the ingestion layer. The acquisition is the initial step in the satellite data processing. From Table 1, we collect datasets from four sources which are: the MDEO ground station through the EUMETCast protocol [29], the EOSDIS of NASA [30], the NESDIS of the NOAA [31], and the COAH [32]. These data come compressed from the ground station or datacenters. Also, we download data from ground stations, notably, the Meteorological Ground Stations (MGS) of Spain.

We have collected MGS data from other open-access platforms such as Wunderground, World weather online, and Accuweather measuring weather variables such as temperature, humidity, pressure, wind speed, and the precipitations. These MGS are located in 27 Moroccan cities. Which is Agadir, Azrou, Casablanca, Errachidia, Fez, Fnideq, Guelmim, Kenitra, Khenifra, Khouribga, Larache, Marrakech, Martil, Meknes, Midelt, Nador, Oujda, Ouazane, Rabat, Safi, Settat, Tangier, Taroudant, Taza, Tetouan, Tifelt and Tiznit. We have also downloaded data from the AERONET, which is a global alliance of ground-based sun-sky radiometers. The network enforces the standardization of instruments, calibration, and processing.

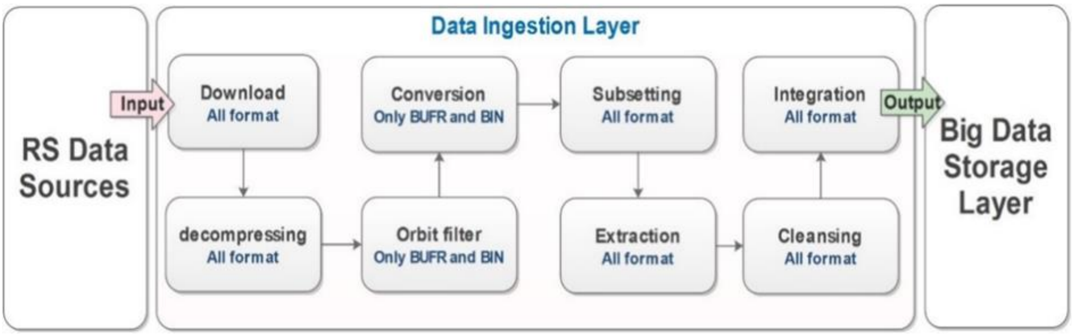


Figure 4. The general architecture of RS data preprocessing [28]

Table 1. RS data sources URLs

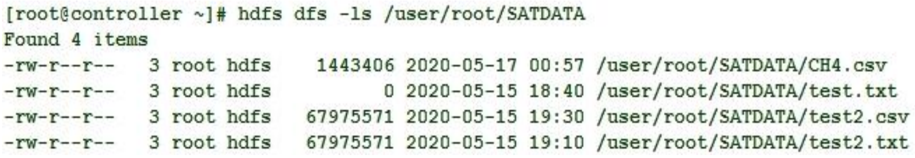
Data Source	Download Link
MDEO	https://www.eumetsat.int/website/home/Data/DataDelivery/EUMETCast/index.html
EOSDIS	https://earthdata.nasa.gov/earth-observation-data/near-real-time/download-nrt-data
NESDIS	ftp://ftp-npp.bou.class.noaa.gov/
COAH	https://scihub.copernicus.eu/
MGS Andalusia	Not open access
MGS Madrid	https://datos.madrid.es/portal/site/egob
Weather underground	www.weatherunderground.com
World weather online	www.worldweatheronline.com
Accuweather	https://www.accuweather.com/

The next step decompresses satellite data automatically to be ready to use. A Bash script decompresses all the extension of the received RSBD (Tar, Zip, and bz2). After the decompression, the number of files increases 240 times, and the size up to forty percent. According to these results, we confirm that RS data consume more storage space and become more complex to be processed after the decompression step. Generally, the HDF5, NetCDF, the BUFR, and the Bin file's format are used to store RS data. However, exploring this data demands a conversion from the scientific file's format to a text or the extensible markup language (XML). We have used two Python libraries, which are the BUFRextract (BUFREXC) [33] and the pybufr_ecmwf (ECMWF) [34].

After the conversion, datasets are ready to be extracted. Thus, the total size of data remains approximately the same after this operation. The collected data come from polar satellites flying in a low earth Orbit (LEO) with an altitude of 800 Km and making sixteen orbits daily [35]. Besides, processing all data of the globe will be challenging in terms of preprocessing. For this aim, we have developed a Python script that filters satellite data by countries using the longitude and latitude as the main parameters. Accordingly, we find that big countries such as the USA, China, and Australia have many files reaching more than seven hundred files per day. However, the smallest state, which is Qatar, contains only about fifty files, similarly in data size (MB). The final step is extraction. It allows selecting the needed variables; for instance, we were interested in twelve variables, including the temperature, humidity, pressure, wind speed, AOD, the vertical column density (VCD) of TG, etc. We developed a Python script performing an automatic data extraction, subset, and filters of inaccurate data.

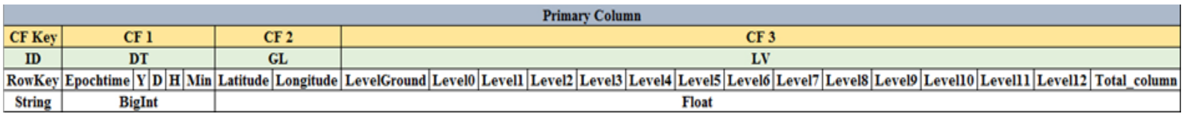
3.2. Data integration and storage: HDFS, HBase, and Hive

In this phase, we integrate the last preprocessed data to be stored inside the Hadoop system. The first stage creates the DFS and HDFS folder for storage, as shown in Figure 5. Then, pushing and copying the comma-separated values (CSV) file to HDFS, then importing the HDFS to HBase, which is built on the primary column (PC) and the column family (CF) as shown in Figure 6, and lastly, generating an external Hive table to store the HBase big columns. Thus, the CSV file is warehoused in Hadoop, which can be accessed and processed easily using HiveQL language-based and like to the structured query language (SQL) language queries 12]. We have chosen the Hadoop for RS data processing because it is one of the best tools at present that holding a massive volume of data, offers easier access, and ensures data redundancy to avoid data losses lest of failure or damage [16]. Still, the HDFS supports parallel data processing chief master/slave is topology [23].



```
[root@controller ~]# hdfs dfs -ls /user/root/SATDATA
Found 4 items
-rw-r--r--   3 root hdfs    1443406 2020-05-17 00:57 /user/root/SATDATA/CH4.csv
-rw-r--r--   3 root hdfs          0 2020-05-15 18:40 /user/root/SATDATA/test.txt
-rw-r--r--   3 root hdfs   67975571 2020-05-15 19:30 /user/root/SATDATA/test2.csv
-rw-r--r--   3 root hdfs   67975571 2020-05-15 19:10 /user/root/SATDATA/test2.txt
```

Figure 5. The HDFS folder for RS data storage



Primary Column																						
CF Key	CF 1					CF 2		CF 3														
ID	DT					GL		LV														
RowKey	Epochtime	Y	D	H	Min	Latitude	Longitude	LevelGround	Level0	Level1	Level2	Level3	Level4	Level5	Level6	Level7	Level8	Level9	Level10	Level11	Level12	Total_column
String	BigInt					Float																

Figure 6. The HBase architecture for RS data storage

4. EXPERIMENT

The ingestion software has automatically downloaded, decompressed, filtered, converted, subsetted, and extracted efficiency. The total daily data to manage is around 55 GB. The latency is averaging between 1 to 180 minutes. And the velocity sumps up 33 000 files per day, as details in Table 2. In this study, we have used a private cluster which runs with Intel(R) Core (TM) i7 central processing unit (CPU)@ 2.50 GHz and 16 GB random-access memory (RAM), running the Centos 7 (64 bit) and equipped with 1 TB of the hard disk drive (HDD).

Table 2. The specifications of the daily input data

Organization	Sensors	Products	File format	Size/day (GB)	Velocity/day	Latency (Minutes)
MDEO	8	27	NetCDF,	10	20000	1-35
NASA	8	14	HDF5,	12	7000	40-140
NOAA	6	6	BUFR,	14	8000	60-180
ESA	3	5	GRIB,	17	150	120-180
MGS	61	7	BIN and CSV	1	100	1-10

5. RESULTS AND DISCUSSIONS

As a result, and as shown in Table 2, we notice that the total daily size acquired was 55 GB. An increase of 40 percent happened after the decompression step because the ground station compresses RS data to facilitate data transmission. After the conversion step, the total size remained the same; however, after the subset operation, data decrease significantly due to the elimination of unneeded data. Globally, this ingestion layer helps to gains storage space by 80 percent. Accordingly, this result could be regarded as a solution to the satellite data perversity.

Generally, the preprocessing of the RSBD takes an important execution time. From Figure 7, we remark that the download phase took approximately five minutes. This number could be reduced by increasing the internet speed or/and changing the internet protocol (IP) protocol from the transmission control protocol (TCP) to the user datagram protocol (UDP). The decompressing's execution time took an average of 90 minutes. Thus, the conversion is the most prolonged operation reaching about 7 hours, and then the subset needs more than an hour. Finally, the extraction processes data took an average of 30 minutes. Accordingly, the total execution time is about 8 hours.

The extraction is the final step of the ingestion layer, and it is also a significant stage, as explained in the previous section. Figure 8 shows the daily total number of plots of the six countries. We notice that after the sub-setting, the number of plots decreases potentially to keep only values covering the countries' zone of interest. The quality, the minimum, and maximum filter eliminate about twenty percent of inaccurate and erroneous datasets. Finally, refined and final plots were stored into associated CSV output files, as shown in Table 3 that can be integrated into an HDFS. Accordingly, the extraction step reduces the number of uninteresting datasets up to twenty percent.

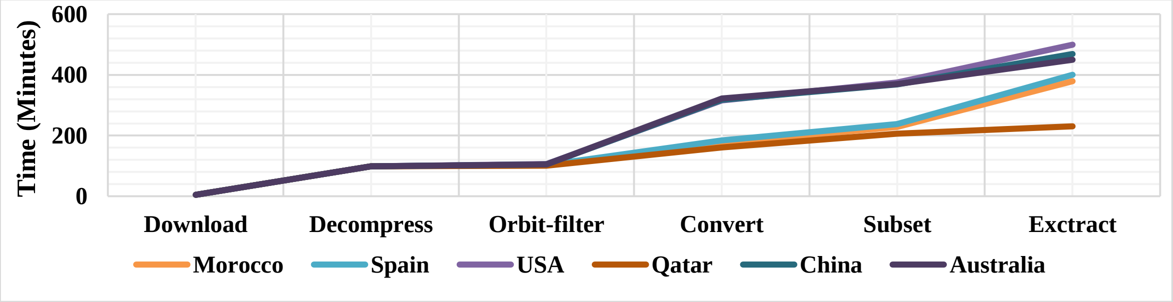


Figure 7. Time execution of the RSBD (minutes) during the processing in the ingestion layer

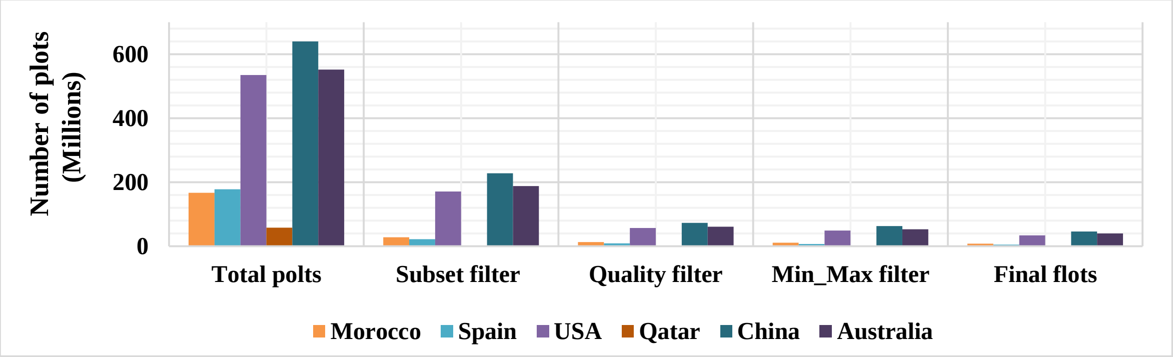


Figure 8. Total daily number of plots (millions) during the extraction step

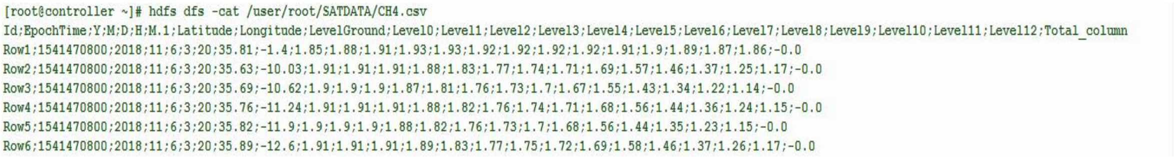
Table 3. The output CSV file containing the VMR (ppm) of the CO_2 of Morocco in 2018/11/06

H	Min	Lat	Long	LG	L0	L1	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	TC
3	17	43.8	3.1	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7	399.7
3	17	43.9	-2.5	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7	402.7
3	17	43.8	-1.9	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4
3	17	43.7	-1.3	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4	402.4
3	17	43.5	-0.6	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2	402.2
3	17	43.4	0.2	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3	400.3

In this study, we have imported the output CSV files into the HDFS, as shown in Figure 9. This will help manage to arrange, store, and clean RS data on a cluster of the data. Furthermore, integrating the preprocessed data in HDFS will not only stripe and mirror automatically but also run on commodity hardware, which does not require a very high-end server with large memory and processing processor. Hadoop can efficiently process a huge volume of data in just minutes, and petabytes in hours using

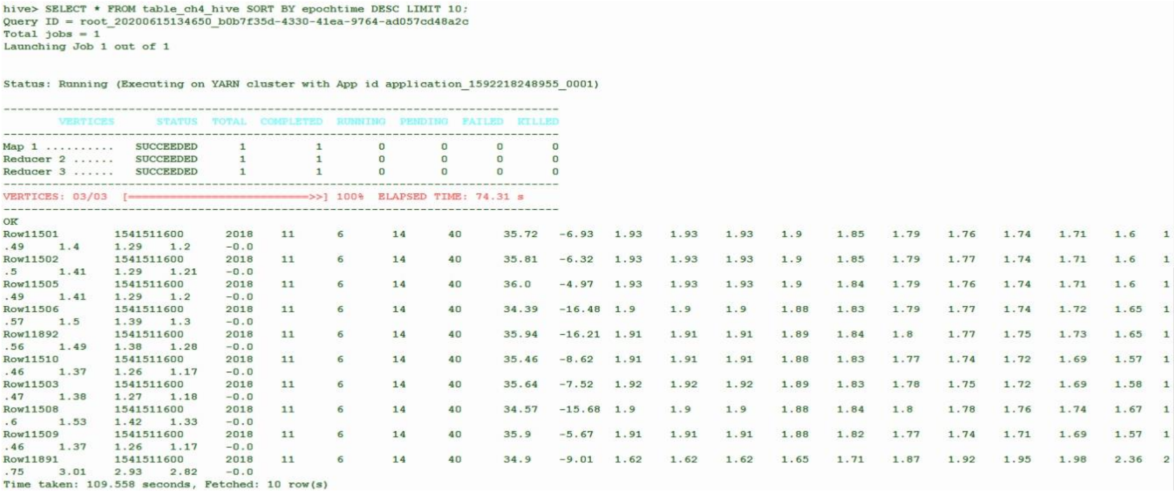
MapReduce. Importing the preprocessed RS data in HDFS is also replicated to other nodes in the cluster, which means that in the incident of failure, there is another copy existing for use. Importing many GB of data into the Hadoop cluster takes nearly a few minutes, and the visualization only a few seconds. The importation of gigabytes of ingested RS data from HDFS to HBase needs only a few minutes. However, HBase does not support SQL requests and consumes a large memory and high CPU performance to process huge inputs and outputs of data. In a shared cluster environment, the system includes fewer task slots per node to assign for HBase CPU requirements.

In this paper, we have stored the preprocessed RS data also in Hive external tables, as shown in Figure 10. Hive helps to simplify handling billions of rows, using the HiveQL, which is much similar to SQL than Pig. Hive also allows us to analyses the huge RS data with low skills in java programming for writing MapReduce programs for saving data from the Hadoop system. Importing some GB of data inside the Hive external table takes around a few minutes, and the visualization only a few seconds.



```
[root@controller ~]# hdfs dfs -cat /user/root/SATDATA/CH4.csv
Id;EpochTime;Y;M;D;H;M.1;Latitude;Longitude;LevelGround;Level0;Level1;Level2;Level3;Level4;Level5;Level6;Level7;Level8;Level9;Level10;Level11;Level12;Total_column
Row1;1541470800;2018;11;6;3;20;35.81;-1.4;1.85;1.88;1.91;1.93;1.93;1.92;1.92;1.92;1.92;1.91;1.9;1.89;1.87;1.86;-0.0
Row2;1541470800;2018;11;6;3;20;35.63;-10.03;1.91;1.91;1.91;1.88;1.83;1.77;1.74;1.71;1.69;1.57;1.46;1.37;1.25;1.17;-0.0
Row3;1541470800;2018;11;6;3;20;35.69;-10.62;1.9;1.9;1.9;1.87;1.81;1.76;1.73;1.7;1.67;1.55;1.43;1.34;1.22;1.14;-0.0
Row4;1541470800;2018;11;6;3;20;35.76;-11.24;1.91;1.91;1.91;1.88;1.82;1.76;1.74;1.71;1.68;1.56;1.44;1.36;1.24;1.15;-0.0
Row5;1541470800;2018;11;6;3;20;35.82;-11.9;1.9;1.9;1.9;1.88;1.82;1.76;1.73;1.7;1.68;1.56;1.44;1.35;1.23;1.15;-0.0
Row6;1541470800;2018;11;6;3;20;35.89;-12.6;1.91;1.91;1.91;1.89;1.83;1.77;1.75;1.72;1.69;1.58;1.46;1.37;1.26;1.17;-0.0
```

Figure 9. A snapshot of the HDFS file containing the CH_4 datasets



```
hive> SELECT * FROM table_ch4_hive SORT BY epochtime DESC LIMIT 10;
Query ID = root_20200615134650_b0b7f35d-4330-41ea-9764-ad057cd48a2c
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1592218248955_0001)

----------------------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 74.31 s
----------------------------------------------------------------------------------------------
OK
Row11501        1541511600      2018    11      6       14      40      35.72   -6.93   1.93    1.93    1.93    1.9     1.85    1.79    1.76    1.74    1.71    1.6     1
.49     1.4     1.29    1.2     -0.0
Row11502        1541511600      2018    11      6       14      40      35.81   -6.32   1.93    1.93    1.93    1.9     1.85    1.79    1.77    1.74    1.71    1.6     1
.5      1.41    1.29    1.21    -0.0
Row11505        1541511600      2018    11      6       14      40      36.0    -4.97   1.93    1.93    1.93    1.9     1.84    1.79    1.76    1.74    1.71    1.6     1
.49     1.41    1.29    1.2     -0.0
Row11506        1541511600      2018    11      6       14      40      34.39   -16.48  1.9     1.9     1.9     1.88    1.83    1.79    1.77    1.74    1.72    1.65    1
.57     1.5     1.39    1.3     -0.0
Row11892        1541511600      2018    11      6       14      40      35.94   -16.21  1.91    1.91    1.91    1.89    1.84    1.8     1.77    1.75    1.73    1.65    1
.56     1.49    1.38    1.28    -0.0
Row11510        1541511600      2018    11      6       14      40      35.46   -8.62   1.91    1.91    1.91    1.88    1.83    1.77    1.74    1.72    1.69    1.57    1
.46     1.37    1.26    1.17    -0.0
Row11503        1541511600      2018    11      6       14      40      35.64   -7.52   1.92    1.92    1.92    1.89    1.83    1.78    1.75    1.72    1.69    1.58    1
.47     1.38    1.27    1.18    -0.0
Row11508        1541511600      2018    11      6       14      40      34.57   -15.68  1.9     1.9     1.9     1.88    1.84    1.8     1.78    1.76    1.74    1.67    1
.6      1.53    1.42    1.33    -0.0
Row11509        1541511600      2018    11      6       14      40      35.9    -5.67   1.91    1.91    1.91    1.88    1.82    1.77    1.74    1.71    1.69    1.57    1
.46     1.37    1.26    1.17    -0.0
Row11891        1541511600      2018    11      6       14      40      34.9    -9.01   1.62    1.62    1.62    1.65    1.71    1.87    1.92    1.95    1.98    2.36    2
.75     3.01    2.93    2.82    -0.0
Time taken: 109.558 seconds, Fetched: 10 row(s)
```

Figure 10. A snapshot of the hive table file containing the CH_4 datasets

6. CONCLUSION

Nowadays, the world is witnessing many environmental issues, notably AP and climate change. Thus, RS techniques play an essential role in monitoring the AQ and supervise the climate changes. However, data provided by satellite sensors are pervasive, complex, and have a considerable size and high velocity. As a result, we have confirmed that satellite data are BD based on the eight salient of BD. Accordingly, such data processing is very challenging and goes beyond the capacity of current systems and architectures. For this intent, we adopted a Hadoop BD architecture that should tackle these issues.

In this manuscript, we focused only on the ingestion layer part enabling an efficient preprocessing of satellite data. The developed layer allowed a daily storage gain of eighty-six percent and improved the satellite dataset's accuracy of up to twenty percent. However, the processing took in important execution time, reaching ten hours because we used only a single standard computer. This architecture should be improved by integrating cloud computing technology and the high-performance computing (HPC) methods. It would be an interesting work to be conducted in the future. As a perspective, we are looking forward to developing an artificial intelligent (AI) algorithm based on MapReduce. That helps to interpolate RS data to fill satellite data gaps, validate RS data with MGS data to enhance their quality, and apply some meteorological models for data prediction helping in decision-makers.

REFERENCES

[1] D. J. Nowak, S. Hirabayashi, M. Doyle, M. McGovern, and J. Pasher, "Air pollution removal by urban forests in Canada and its effect on air quality and human health," *Urban Forestry & Urban Greening*, vol. 29, pp. 40-48, 2018.

[2] B. Boudriki Semlali, C. El Amrani, and S. Denys, "Development of a Java-based application for environmental remote sensing data processing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 1978-1986, 2019.

[3] Y. Ma et al., "Remote sensing big data computing: Challenges and opportunities," *Future Generation Computer Systems*, vol. 51, pp. 47-60, 2015.

[4] Andy, "How many satellites are orbiting the Earth in 2018?," *Pixalytics*, p. 3, 2018.

[5] L. Zhu, J. Suomalainen, J. Liu, J. Hyyppä, H. Kaartinen, and H. Haggren, "A Review: Remote Sensing Sensors," *Multi-purposeful Application of Geospatial Data*, 2018.

[6] B.-E. Boudriki Semlali and C. El Amrani, "Towards Remote Sensing Datasets Collection and Processing," *International Journal of Embedded and Real-Time Communication Systems*, vol. 10, no. 3, pp. 49-67, 2019.

[7] V. Karhila, "Bufr: A Meteorological Code for the 21st Century," *2010 EUMETSAT Meteorological Satellite Conference,* Cordoba, Spain, 2010, pp. 1-5.

[8] R. Rew and G. Davis, "NetCDF: an interface for scientific data access," *IEEE Computer Graphics and Applications*, vol. 10, no. 4, pp. 76-82, 1990.

[9] L. Gosink, J. Shalf, K. Stockinger, Kesheng Wu and W. Bethel, "HDF5-FastQuery: Accelerating Complex Queries on HDF Datasets using Fast Bitmap Indices," *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, Vienna, 2006, pp. 149-158.

[10] Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," *IEEE Access*, vol. 2, pp. 652-687, 2014.

[11] B.-E. Boudriki Semlali, C. El Amrani, and G. Ortiz, "Hadoop Paradigm for Satellite Environmental Big Data Processing," *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, vol. 11, no. 1, pp. 24-47, 2020.

[12] B.-E. Boudriki Semlali and C. El Amrani, "Towards Remote Sensing Datasets Collection and Processing," *in Transactions on Large-Scale Data- and Knowledge-Centered Systems XLI*, vol. 11390, A. Hameurlain, R. Wagner, and T. K. Dang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 286-294.

[13] W. J. Emery and A. Camps, "Introduction to satellite remote sensing: atmosphere, ocean, land and cryosphere applications," *Amsterdam, Netherlands; Cambridge, MA: Elsevier*, 2017.

[14] M. Akram, M. Amrani, and C. El, "Air2Day: An Air Quality Monitoring Adviser in Morocco," *International Journal of Computer Applications (IJCA)*, vol. 181, no. 17, pp. 1-6, 2018.

[15] S. Quesada-Ruiz et al., "Benefit of ozone observations from Sentinel-5P and future Sentinel-4 missions on tropospheric composition," *Atmospheric Measurement Techniques*, vol. 13, no. 1, pp. 131-152, 2020.

[16] G. Manogaran and D. Lopez, "Spatial cumulative sum algorithm with big data analytics for climate change detection," *Computers & Electrical Engineering*, vol. 65, pp. 207-221, 2018.

[17] T. Lukić et al., "Forest fire analysis and classification based on a Serbian case study," *Acta geographica Slovenica*, vol. 57, no. 1, pp. 51-63, 2017.

[18] El Amrani Chaker, "Remote Sensing for Real-time Early Warning of Environmental Disasters and WRF Modelling," 2015. [Online]. Available: http://asrenorg.net/eage2015/sites/default/files/files/7%20Chaker%20El%20Amrani.pdf.

[19] R. J. Boain', "A-B-Cs of Sun-Synchronous Orbit Mission Design," *AAS Publications Office*, pp. 1-20, 2004.

[20] J. Schmetz et al., "An Introduction to Meteosat Second Generation (MSG)," B*ulletin of the American Meteorological Societ*y, vol. 83, no. 7, pp. 977-992, 2002.

[21] M. Buchhorn, M. Lesiv, N.-E. Tsendbazar, M. Herold, L. Bertels, and B. Smets, “Copernicus Global Land Cover Layers-Collection 2,” Remote Sensing, vol. 12, no. 6, p. 1044, 2020.

[22] C. Wang, F. Hu, X. Hu, S. Zhao, W. Wen, and C. Yang, "A Hadoop-Based Distributed Framework For Efficient Managing And Processing Big Remote Sensing Images," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-4/W2, pp. 63-66, 2015.

[23] JIN Hailiang, LU Xiaoping, and LIU Huijie, "Large-Scale Terrain Realistic Rendering Based on Programmable GPU Hardware," *Geomatics and Information Science of Wuhan Univers*, 2010.

[24] R. Ranjan, "Streaming Big Data Processing in Datacenter Clouds," *IEEE Cloud Computing*, vol. 1, no. 1, pp. 78-83, 2014.

[25] B.-E. Boudriki Semlali, C. El Amrani, and G. Ortiz, "Adopting the Hadoop Architecture to Process Satellite Pollution Big Data," *International Journal of technology and engineering studies*, vol. 5, no. 2, pp. 30-39, 2019.

[26] A. Erraissi, A. Belangour, and A. Tragha, "Meta-Modeling of Data Sources and Ingestion Big Data Layers," *SSRN Electronic Journal*, pp. 1-5, 2018.

[27] B.-E. Boudriki Semlali, C. El Amrani, and G. Ortiz, "SAT-ETL-Integrator: An Extract-Transform-Load Software for Satellite Big Data Ingestion," *Journal of Applied Remote Sensing (JARS)*, vol. 14, no. 1, p. 28, 2020.

[28] C. El Amrani, G. L. Rochon, T. El-Ghazawi, G. Altay, and T. Rachidi, "Development of a real-time urban remote sensing initiative in the Mediterranean region for early warning and mitigation of disasters," *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, 2012, pp. 2782-2785.

[29] "EOSDIS," 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=EOSDIS&oldid=808578344.

[30] "NOAA NESDIS website," 2018. [Online]. Available: https://www.nesdis.noaa.gov/.

[31] Esa, "ESA website," 2019. [Online]. Available: https://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview3.

[32] Francis Breame, "BUFRextract_User_Guide_v30.pdf." 2018. [Online]. Available: http://www.elnath.org.uk/BUFRextract_User_Guide_v30.pdf.

[33] S. Siemen, S. Lamy-Thepaut, F. Li, and I. Russell, "The next generation of ECMWF's meteorological graphics library-Magics++," *Newsletter Feature Article*, no. 110. pp. 36-41, 2007.

[34] A. Thusoo et al., "Hive-a petabyte scale data warehouse using Hadoop," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, Long Beach, CA, USA, pp. 996-1005, 2010.

[35] J. Wei, D. Liu, and L. Wang, "A general metric and parallel framework for adaptive image fusion in clusters: A General Metric and Parallel Framework for Adaptive Image Fusion," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 7, pp. 2191-2502, 2014.

BIOGRAPHIES OF AUTHORS

Badr-Eddine Boudriki Semlali received his master's degree in the computer system and network engineering from the Faculty Sciences and Techniques of Tangier (FSTT) in 2017. Currently, he is a Ph.D. student at the UAE of Morocco and a researcher in the RSLab, TSC department, UPC, Barcelona, Spain. Specialized in BD analytical of remotely sensing Earth observatory and cloud computing. Badr-Eddine is also a reviewer in the IJEMA journal. He has authored some peer-reviewed papers and contributed to many international conferences. He has benefited from several international scholarships, particularly the ERASMUS+, VLIRUOS, and CMN of Murcia.

Chaker El Amrani is a Doctor in Mathematical Modelling and Numerical Simulation from the University of Liège, Belgium (2001). He joined Abdelmalek Essaâdi University, Morocco, in 2003. He is currently Chair of the Computer Engineering Department at the Faculty of Science and Technology, Tangier. He is the NATO Partner Country Project Director of a real-time remote sensing initiative for early warning and mitigation of disasters and epidemics in Morocco. He lectures distributed systems and is promoting High-Performance Computing education in the University. He joined in 2001 Thales Information Systems Company based in Brussels and worked as Air Traffic Control Software Engineer.