Environmental Technology & Innovation 27 (2022) 102776

Contents lists available at ScienceDirect

Environmental Technology & Innovation

journal homepage: www.elsevier.com/locate/eti

Spatiotemporal data partitioning for distributed random forest algorithm: Air quality prediction using imbalanced big spatiotemporal data on spark distributed framework

Marjan Asgari [a,*], Wanhong Yang [a], Mahdi Farnaghi [b]

[a] Department of Geography, Environment & Geomatics, University of Guelph, Canada
[b] Faculty of Geo-Information Science and Earth Observation, University of Twente, The Netherlands

ARTICLE INFO

Article history:
Received 29 April 2021
Received in revised form 10 March 2022
Accepted 19 June 2022
Available online 27 June 2022

Keywords:
Big spatiotemporal data
Distributed systems
Air quality prediction
Distributed random forest algorithm
Imbalanced data

ABSTRACT

Spatiotemporal air quality datasets are typically collected hourly in monitoring stations deployed non-uniformly across a metropolitan city. These datasets are not only big, which poses challenges on the storage and processing capacity of centralized computing systems but also imbalanced and spatially heterogeneous, which may result in biased air quality prediction. To address these challenges, we designed and developed a parallel air quality prediction system equipped with a spatiotemporal data partitioning method, a distributed machine learning algorithm, Hadoop's distributed data storage platform and its resource scheduler/manager, and Spark's efficient and in-memory execution environment, which is suitable for running iterative algorithms, e.g., machine learning. Our proposed spatiotemporal partitioning method accounted for imbalance and spatial heterogeneity features of big air quality data in predictive models, which comply with the load-balancing requirement of distributed computing systems. Distributed Random Forest algorithm in the H2O library of the Spark framework was selected as the distributed machine learning algorithm to develop the air quality predictive model. This algorithm is an ensemble forest with algorithm-level adjustments to perform as efficiently as possible for big imbalanced datasets. An application of the parallel quality prediction system for Tehran, Iran showed that the parallel prediction system had considerable speedup gain and improved both the overall accuracy and class precision of air quality prediction when working with imbalanced big spatiotemporal air quality datasets. A future research direction is to add data streaming and visualization functions to the system to provide rapid and reliable air quality prediction for supporting environmental health management.

1. Introduction

The United States Environmental Protection Agency (US EPA) characterizes Air Quality Index (AQI) by specific gases that affect human health the most, such as ground-level ozone (O3), Carbon Monoxide (CO), and Particles having Aerodynamic Diameters of 10 µm or under and 2.5 µm or under (PM10 and PM2.5) (Kadri et al., 2012). By investigating the relationship between non-accidental daily mortality and the AQI in Tehran, Iran, Amini et al. (2019) showed that the AQI had an immediate effect on daily mortality that increased steadily over weeks with higher air pollution levels. Kan et al. (2008)

[*] Corresponding author.
E-mail address: masgari@uoguelph.ca (M. Asgari).

https://doi.org/10.1016/j.eti.2022.102776

also identified a remarkable correlation between cardiovascular mortality and the air pollution level on the death day and a day before. Thus, air quality prediction systems are essential to estimate air quality levels and provide rapid, accurate and reliable information for environmental protection and health care agencies to minimize adverse effects of air pollution on human health and the environment.

The decreasing costs of in-situ and remote sensors have significantly expanded the extent of air quality monitoring networks, resulting in an explosion in the volumes of spatial and temporal air quality datasets (Chen et al., 2014; Du et al., 2017; Song et al., 2019). Statistical models, which based on regression analysis investigate the causal effects of meteorological factors on the change of pollutant concentration(s) (Zhu et al., 2018), have been applied widely for air quality prediction. However, big datasets are linked to highly nonlinear processes and unknown dynamics in air pollution formation, and Liao et al.'s (2021) extensive review of statistical models for air quality forecasting reveals that due to the neglection of these complexities by statistical models, these models show limited performance in air pollution prediction using data with nonlinear relationships or non-normal distributions. In recent years, developing air quality predictive models based on machine learning algorithms has emerged as one of the research directions to address the complexities of big spatiotemporal air quality datasets by learning the hidden relationship within historical data (Chen et al., 2014). However, the development of machine learning-based air quality predictive models faces three significant challenges.

The first challenge is the capacity constraints of centralized computing systems in processing big spatiotemporal air quality datasets as machine learning algorithms need to load their required datasets entirely into the system memory when operating (Qiu et al., 2016). To overcome this challenge, distributed air quality prediction systems using Hadoop and Spark parallel computing frameworks have been developed. Ghaemi et al. (2015) showed that Hadoop MapReduce speeded up the air quality prediction process, and Ayyalasomayajula et al. (2016) showed that the Spark execution engine even surpassed in speeding from Hadoop MapReduce by 20%–25%. Asgari et al. (2017) extended the previous results by showing that Spark-based distributed forecasting system not only achieved high execution speed but also benefited from Hadoop distributed file system in storing large datasets on a cluster of computers. Li et al. (2019) developed Logistic Regression and Random Forest (RF) machine learning models using Spark MLlib library on both a local computer and a distributed system, Amazon Elastic MapReduce (EMR) cloud. The results demonstrated that the RF algorithm took 23.7 s to train a model on a local computer while the distributed algorithm on a four-processor Spark parallel system used 61 ms. These studies revealed that distributed computing significantly improved the processing speed of air quality predictive models with big spatiotemporal data (Peteiro-Barral and Guijarro-Berdiñas, 2013).

The second challenge is to address the imbalance feature of big spatiotemporal air quality datasets. The tendency towards deploying more air quality monitoring stations in highly polluted zones in urban areas, different starting date/month/year of monitoring stations, and potential equipment failures lead to potentially imbalanced big air quality datasets with greatly-differed sizes of data collected for different air quality classes and/or monitoring stations (Del Río et al., 2014). The significant variations between the size of collected data samples may cause biased learning and poor performance of air quality predictive models (Weiss and Provost, 2003). Rebalancing the distribution of classes using under-sampling, over-sampling, or hybrid approaches (Wu et al., 2020) and combining multiple diverse classifiers (Salunkhe and Mali, 2016) were developed to characterize imbalanced datasets for machine learning algorithms. A survey by Leevy et al. (2018) showed that Ensemble Methods such as RF machine learning algorithms had the advantage of handling imbalanced datasets. Li et al. (2019) developed parallel computing for air quality prediction and demonstrated that the RF machine learning algorithm outperformed the Logistic Regression algorithm in model accuracy. Zhang and Yuan (2015) implemented the Distributed Random Forest (DRF) algorithm on a Spark-based distributed system for fine particulate matter (PM2.5) prediction and achieved a relatively high overall model accuracy value of above 70%. However, these studies did not examine the RF's capability in improving the per-class accuracy for the minority classes. Triguero et al. (2016) proposed a two-layer data parallelization method to address the imbalanced big dataset issue. Their method partitioned imbalanced data randomly into data blocks in the first layer, and in the second layer, an evolutionary under-sampling method was used to balance the data partitions. Even though this approach was not applied for partitioning big air quality datasets, their results showed that such approaches could improve the classification accuracy for extremely imbalanced datasets.

The third challenge is to address the spatial heterogeneity characteristic of big air quality datasets, which refers to the uneven distribution of air pollutant concentrations across geographical areas (Sun et al., 2021). Most machine learning-based air quality predictive models assume that input data is uniformly distributed. Without accounting for the spatial structure of the input data, these algorithms ignored the underlying spatial relationships affecting the air quality prediction (Yang et al., 2018). To address this challenge, Georganos et al. (2021) suggested working on local predictive models instead of one global model for the entire study area. Training local machine learning models using only nearby observations for each data location instead of training a global model with collected data for the entire extent of the study area helped characterize the spatial relationships between model variables and attributes to improve the model prediction accuracy. To address both imbalanced datasets and spatial heterogeneity, Rastogi et al. (2018) used the k-Nearest Neighbours algorithm with the Locality Sensitive Hashing partitioning method on Spark framework to partition data based on their closeness. Then, they used Synthetic generation of minority class data to balance the data partitions. However, these approaches had not been applied to data partitioning for air quality predictive models.

Although the development of machine learning-based air quality predictive models tackled the aforementioned challenges to some certain extent, there still exists a knowledge gap to address all three challenges together. Thus, this study aimed to address the knowledge gap to develop and apply a parallel air quality predictive model based on a machine learning algorithm with the characterization of imbalance and spatial heterogeneity features of big spatiotemporal air quality datasets. In the next section, we introduced a parallel computing system for air quality prediction using distributed partitioning of big spatiotemporal data and the DRF algorithm. In Section 3, the application of the parallel computing system for air quality prediction in the Tehran city of Iran was presented. We discussed the performance of the parallel air quality prediction system in Section 4. The paper ended with the conclusion section to highlight the key points of the paper and discuss future research directions.

2. Parallel air quality prediction system: Design and implementation

In this study, we designed and developed a parallel air quality prediction system based on (1) Hadoop and Spark powerful distributed computing frameworks, (2) a double-layer partitioning method for big spatiotemporal data, and (3) a distributed machine learning algorithm. A brief technical context on Hadoop and Spark frameworks along with the design and implementation of the system are elaborated in the following subsections.

2.1. *Hadoop and Spark parallel computing frameworks*

Hadoop is a platform to store and process big datasets across clusters of either virtually- or physically-distributed computers (Mavridis and Karatza, 2017), and Spark is a parallel computing system designed for *fast* big data processing (Zaharia et al., 2016). Detailed information about Hadoop and Spark can be found in Anuradha (2015) and Salloum et al. (2016), respectively. In our study, Hadoop is adopted to provide a distributed big data storage platform (HDFS) and a resource manager and scheduler (YARN), and Spark is utilized to provide a fast big data processing environment (Fig. 1). It should be mentioned that, both Hadoop and Spark work on a Master-Slave architecture, which is a parallelization scheme on independent parallel processing units. Slave or data processor nodes receive input data blocks from the master node, which is only a parallel job/data controller/tracker node and does not take part in parallel data processing (Czarnul, 2021), to process them and return the outputs back to the master node. The master node provides the final output from all slave nodes' results.

Hadoop Distributed File System (HDFS) is designed for storing, querying, and retrieving big datasets on/from distributed computers; it provides scalable and fault-tolerant data storage (Anuradha, 2015). HDFS implementation on the master node is called name node, which partitions the entire dataset into HDFS data partitions using random/nonrandom strategies, distributes them to data nodes, monitors data nodes and stores all required information for querying/retrieving data. Data nodes, HDFS implementation on the slave nodes, are responsible for reading/writing HDFS data partitions per name node's requests (Shetty and Manjaiah, 2016). Yet Another Resource Negotiator (YARN), with two nodes called resource manager and node manager, is designed to handle the resource allocation process for data processing engines compatible with Hadoop, such as Spark. The resource manager and node manager adjudicates available resources, memory, CPU, etc. of all computers, and decides which tasks, where and when should be run. The resource manager on the master node receives the application, manages available resources and, accordingly, schedules the execution of distributed applications, and communicates with node managers on slave nodes to send them allocated parallel jobs/tasks to be executed (Perwej et al., 2017).

Spark Core, the underlying execution engine, is designed for conducting distributed tasks through effective input/output operations and in-memory computing (Jonnalagadda et al., 2016). To ensure fast data processing, Spark Core only works with Resilient Distributed Dataset (RDD), which is Spark abstraction of a partitioned database in the memory of distributed computers throughout data processing. Using Spark build-in data partitioning methods, such as Hash or Range partitioner (more information in Mahmud et al., 2020), Spark Context creates logical RDD data partitions by partitioning/repartitioning datasets stored in external storage systems, e.g., HDFS partitions (Azeroual and Nikiforova, 2022). Despite supporting a large domain of data types, Spark RDD does not support spatial data types and, consequently, spatial data partitioning methods. The GeoSpark library, an in-memory distributed library, is developed to introduce Spatial RDD (SRDD), which stores spatial data, characterizes their spatial features, and supports developing spatial data partitioning techniques (Yu et al., 2016).

2.2. *Double-layer partitioning method for big spatiotemporal data*

Big data partitioning in parallel computing systems is applied to divide data into small data blocks and distribute them across a cluster of computers. The main consideration for data partitioning is to generate same-size data blocks to maintain data load balance across the cluster. However, when it comes to spatial data, the data partitioning methods should also preserve spatial structure of data objects to help characterize spatial heterogeneity (Yao et al., 2017).

Due to the tendency towards having concentrated monitoring stations in the most polluted areas and sparse stations in the least polluted ones in a metropolitan city (Fig. 1), the amount of data collected for all AQI classes is not equal. There exist the minority/majority classes, classes with the least/most collected amount of data, in these imbalanced air quality datasets. In air quality modelling, ignoring the data imbalance feature causes less chance of sampling for the minority classes, which may lead to inaccurate air quality prediction. As the first step, data partitioning based on the location

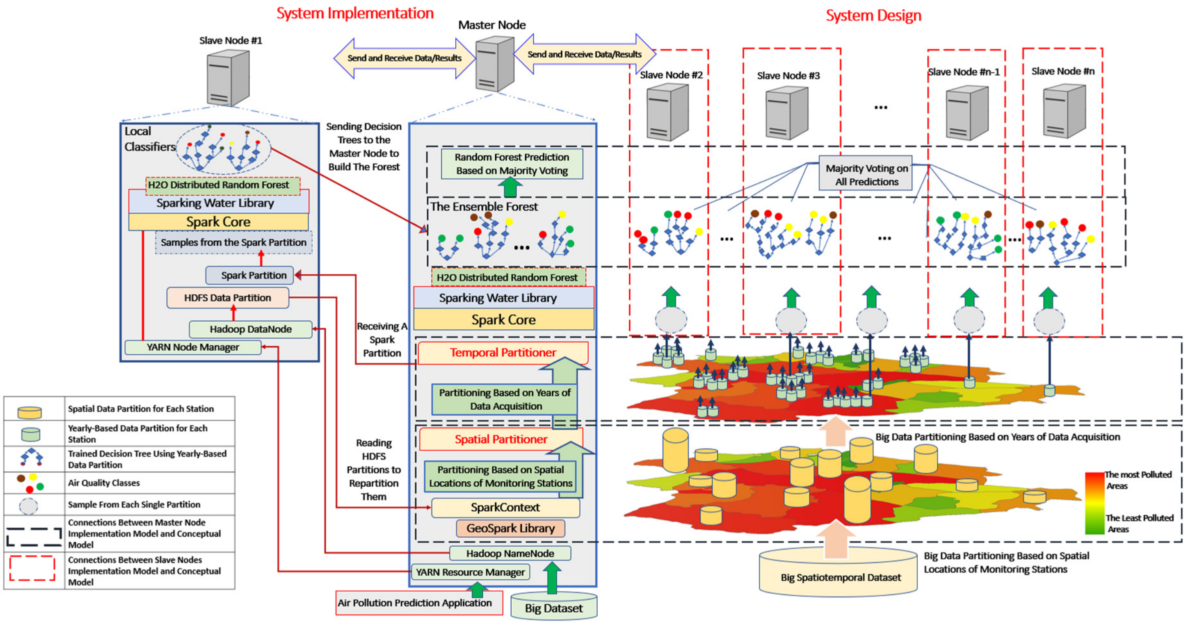

System Implementation
Master Node
Slave Node #1
Send and Receive Data/Results
Send and Receive Data/Results
Local Classifiers
H2O Distributed Random Forest
Sparking Water Library
Spark Core
Samples from the Spark Partition
Spark Partition
HDFS Data Partition
Hadoop DataNode
YARN Node Manager
Sending Decision Trees to the Master Node to Build The Forest
Random Forest Prediction Based on Majority Voting
The Ensemble Forest
H2O Distributed Random Forest
Sparking Water Library
Spark Core
Receiving A Spark Partition
Temporal Partitioner
Partitioning Based on Years of Data Acquisition
Spatial Partitioner
Partitioning Based on Spatial Locations of Monitoring Stations
Reading HDFS Partitions to Repartition Them
SparkContext
GeoSpark Library
Hadoop NameNode
YARN Resource Manager
Air Pollution Prediction Application
Big Dataset
Spatial Data Partition for Each Station
Yearly-Based Data Partition for Each Station
Trained Decision Tree Using Yearly-Based Data Partition
Air Quality Classes
Sample From Each Single Partition
Connections Between Master Node Implementation Model and Conceptual Model
Connections Between Slave Nodes Implementation Model and Conceptual Model
System Design
Slave Node #2
Slave Node #3
...
Slave Node #n-1
Slave Node #n
Majority Voting on All Predictions
Big Data Partitioning Based on Years of Data Acquisition
The most Polluted Areas
The Least Polluted Areas
Big Data Partitioning Based on Spatial Locations of Monitoring Stations
Big Spatiotemporal Dataset

Fig. 1. The design and implementation of the air quality prediction system.

of monitoring stations and developing air quality classifiers for each partition increases the sampling rate of minority classes, which helps address the spatial heterogeneity feature of the dataset, as well (Bai et al., 2018; Xie et al., 2017). However, some air pollution monitoring stations have more or less collected data compared to the others due to different deployment dates and potential technical failure at various locations and times, as shown in Fig. 1. Generating data partitions *only* based on the spatial locations ends up having frequently thin or oversized data partitions. This is against the load-balance requirement of distributed systems, and it increases not only the idle time of the processors working on thin partitions but also the volume of communications of processing units with the master node. Thus, the second step in data partitioning aims to reduce the number of thin and oversized data partitions. Due to the seasonal fluctuations of air pollutant levels (Song et al., 2019), a yearly basis for temporal partitioning helps meet the load-balancing requirement by coming up with equally-sized partitions (Fig. 1). With these two steps, we designed an innovative double-layer distributed partitioning method for big spatiotemporal air quality data based on the location of air pollution monitoring stations and data acquisition years.

Using the GeoSpark library, we designed the first layer of our double-layer partitioning method, as shown in Fig. 1. The first layer creates SRDD partitions by repartitioning HDFS data based on the spatial location (Latitude, Longitude) of the monitoring stations. This process has two sub-steps, which are (1) converting HDFS data objects to spatial data objects based on latitudes and longitudes of monitoring stations and (2) partitioning spatial data objects with the same number of monitoring stations. In the second layer, using a temporal partitioner, the spatial partitions are divided further into smaller partitions based on the year of collected data objects. The number of data partitions for each station depends on the number of years covered by that station. In the end, each partition only contains data objects for one monitoring station and one year of the data coverage for that station. At this stage, partitions are ready to be assigned to slave nodes and sampled by the DRF algorithm for building parallel classifiers.

2.3. Air quality predictive model development: Distributed random forest algorithm

RF algorithm is a decision tree ensemble that builds and trains multiple decision trees (classifiers) using the bagging sampling method and a random subset (partition) of the entire training dataset (Breiman, 1996). This algorithm is well-known for its capability of dealing with big imbalanced datasets, easy extraction of knowledge, not requiring normally-distributed training datasets, and providing accurate results (Nagarajan and Ld, 2019). In addition to these advantages, the independent construction of trees in the RF algorithm paved the way for developing the DRF algorithm on clusters of computers to speed up the model development process. In fact, the DRF algorithm constructs decision trees simultaneously on parallel computers by sampling from distributed partitions of data. Using the DRF algorithm along with our proposed spatiotemporal partitioning method, as shown in Fig. 1, we can develop local classifiers for all spark partitions individually. This process consists of two stages, which are (1) building local decision trees (classifiers) using locally available data partitions and (2) building the ensemble forest from all local decision trees. The first stage is carried out simultaneously on all slave nodes, while the second stage starts at the end of the first stage on the master node (Fig. 1).

The DRF algorithm's classifiers on each computer are fed and trained using samples from only one spatiotemporal data partition. Thus, based on a user-defined number of decision trees (two trees as illustrated in Fig. 1) on each distributed computer, a local classifier for each data partition is developed independently from other classifiers. Accordingly, the chance of sampling from minority classes increases, which addresses the challenge of imbalanced datasets. In addition, the spatial factors impacting air quality indices are considered implicitly, which addresses the spatial heterogeneity of air quality data in the modelling process. In the second stage, all developed classifiers across the cluster of computers are submitted to the master node, which builds the ensemble forest for air pollution prediction in our proposed parallel computing system. Then, each tree (classifier) in this forest does the classification individually and votes for a class. The class with maximum votes will be chosen as the DRF's predicted class.

In the implementation, we used the DRF algorithm in the H2O library, which provides robust, fast, and scalable machine learning algorithms for parallel processing engines/frameworks (Aiello et al., 2016). The reason for choosing DRF is because it has adopted several algorithm-level approaches for efficient and accurate performance for big imbalanced datasets. This algorithm provides the chance of adjusting parameters, such as Class_Sampling_Factor parameter, which defines the under- or over-sampling ratio for each class, Balance_Classes, which enables the algorithm to oversample only the minority classes in a dataset, and Sample_Rate_Per_Calss, which defines the sampling rate for each class (Cook, 2016). Besides, DRF supports the Grid Search method for tuning hyperparameters that are parameters that cannot be optimized by learning from the training process. This method defines a range of values for each hyperparameter, develops DRF models with all possible combinations of values in those ranges, and finds the best combination based on the models' accuracy assessment (George and Sumathi, 2020). DRF algorithm in the H2O library works with "H2O Dataframe", which is an in-memory 2D array of data objects. Thus, in the Spark framework, the process of conversion into/from SRDD data objects from/into H2O data frame at the beginning and the end of running DRF is required. Sparkling Water library is developed to integrate the Spark parallel computing framework with the H2O library. This library takes care of all data types conversions and runs the H2O DRF algorithm on a Spark-based distributed system (Malohlava et al., 2016).

3. Tehran's air pollution prediction system

We applied the parallel computing framework for air quality prediction in the Tehran city of Iran. This section presents the spatiotemporal air quality datasets in 3.1 and then the air quality predictive model based on the DRF algorithm and input data in 3.2.

3.1. Tehran's spatiotemporal air quality big datasets

Tehran locates in the north of Iran (Fig. 2) with a static and dynamic population of about 8.5 and 12.5 million, respectively. The poor air quality in Tehran is closely related to its rapid population growth, industrial development, urbanization, and high fuel consumption, along with its high altitude and being surrounded by the Alborz Mountain Range from its north (Henger and Sarraf, 2018). Across Tehran city, 21 monitoring stations are deployed to collect air quality data for calculating and reporting AQI on an hourly basis using Carbon Monoxide (CO), Nitrogen Dioxide (NO2), Ozone (O3), Particulate Matters with a diameter of 10 μm or less (PM10), and Particulate Matters with a diameter of 2.5 μm or less (PM2.5). Fig. 2 shows the geographic location of Tehran city in Tehran Province and the "uneven" distribution of 21 air pollution monitoring stations. The southwestern part of the city has a concentration of monitoring stations, with some zones having more than one station while some central zones have no monitoring stations.

This study used an hourly air quality dataset collected by these monitoring stations from 2009 to 2013. We plotted the maximum AQI recorded in this city for all five years of study (Fig. 3). The pattern showed a yearly fluctuation of AQIs (temporal variation) along with a high frequency of AQIs of nearly 500. Further, the number of days with AQIs above 200 (very unhealthy level in class 5 Mirabelli et al., 2020) was quite high in this city. A related study by Pishgar et al. (2020) examined the spatiotemporal pattern of the mortality rates due to respiratory tract diseases caused by air pollution in Tehran from 2008 to 2013, which supported the existence of yearly variations of air pollution and high AQI levels in Tehran. The authors also showed that the peaks for air pollution-related mortalities occurred in winter and fall seasons, and the mortality rates for 2010, 2012 and 2013 were similar but higher than those for 2009 and 2011. These results were in line with the temporal variations of AQI levels from 2009 to 2013, as shown in Fig. 3.

We plotted the proportions of AQIs recorded in each station during the five years of study (Fig. 4). The pattern showed a considerable spatial variability of AQI, and the majority of recorded AQIs belong to three AQI classes, which are classes 7, 6 and 5. This pattern confirmed that the collected air quality data were imbalanced, and AQI class 4, 3, 2 and 1 formed the minority classes of this study's big dataset. This figure also showed that not all the stations were operating throughout all years of study, e.g., stations 4 and 15 were operating all years, stations 4, 17, and 16 were operating for four years, and stations 21, 6, and 20 were operating for only two years.

Based on Figs. 2, 3, and 4, our big air quality dataset, which has a total of 4,100,000 data records, exhibits (1) imbalance feature, with around 40% of data allocated to the most major and 1% to the most minor classes, (2) spatial heterogeneity reflected by concentrated and sparse monitoring stations across the city, and (3) AQI temporal variations throughout the study years.

In this study, meteorological variables, e.g., wind speed, wind direction, cloud cover, temperature, relative humidity, and air pressure data, along with geospatial variables of air pollution monitoring stations, e.g., absolute height, and the

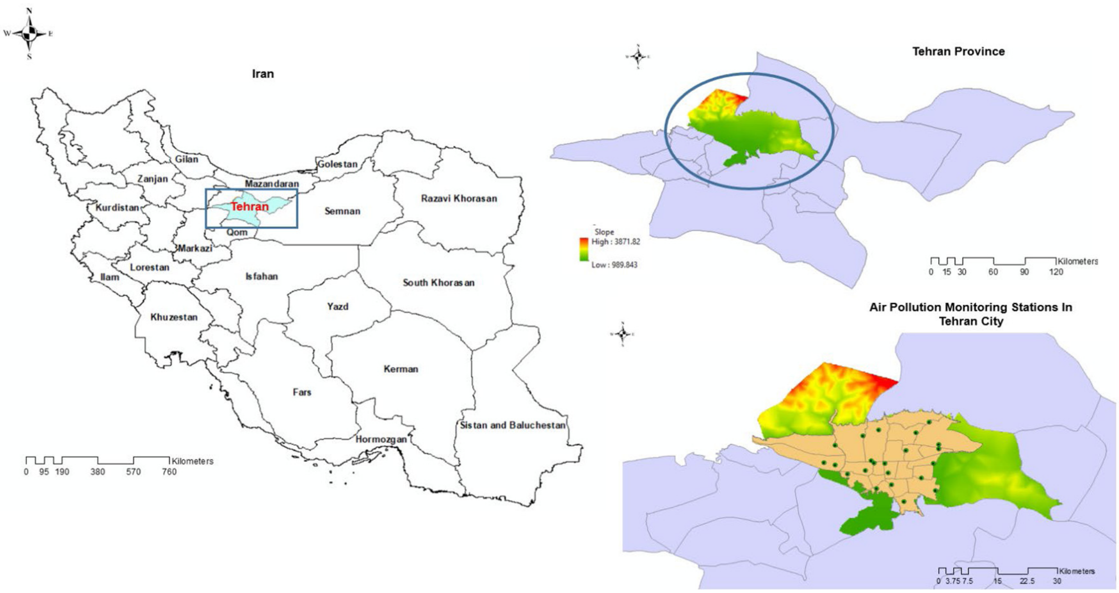
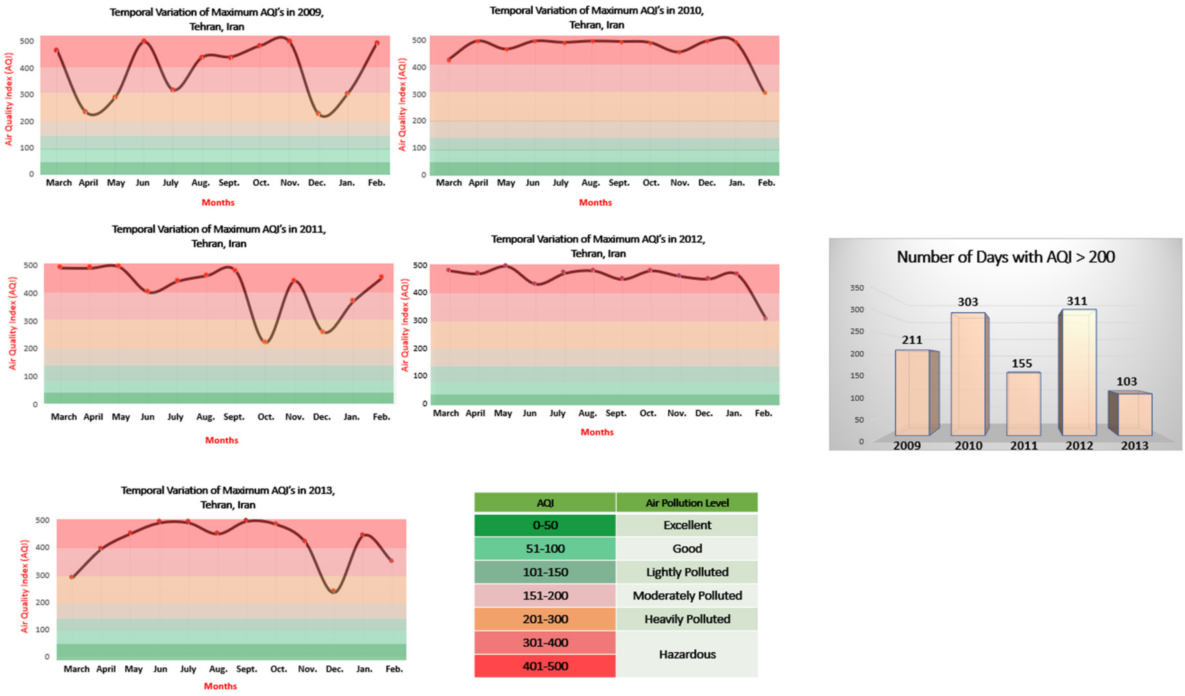


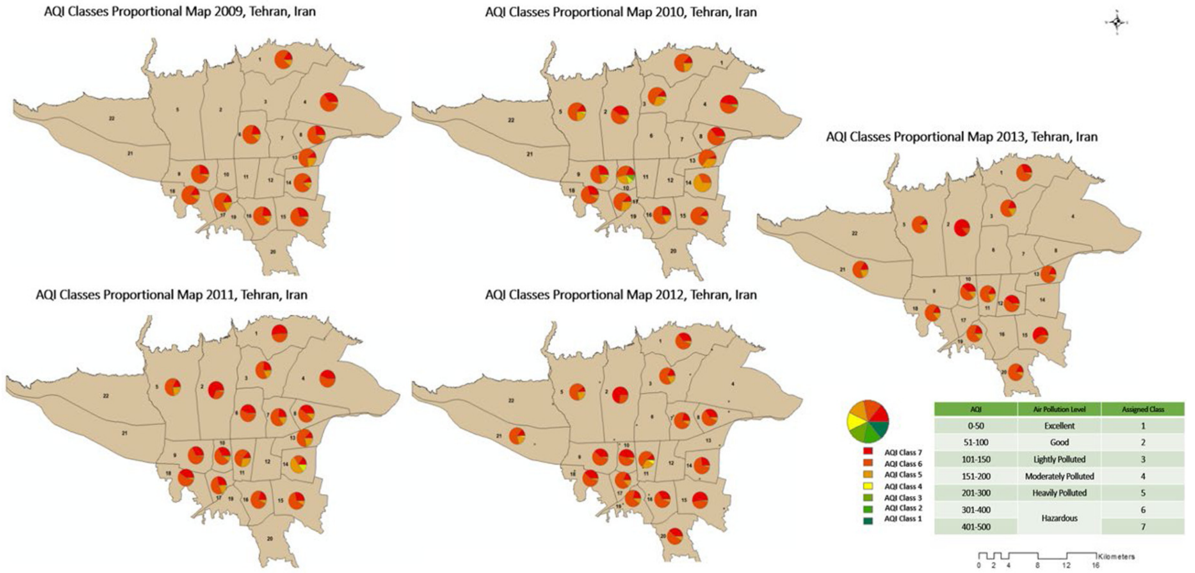
Fig. 2. Geographic locations of air pollution monitoring stations in Tehran, Iran (Data source for Iran/Tehran Shapefiles is www.map.tehran.ir and data source for air pollution monitoring stations is www.air.tehran.ir).



AQI	Air Pollution Level
0-50	Excellent
51-100	Good
101-150	Lightly Polluted
151-200	Moderately Polluted
201-300	Heavily Polluted
301-400	Hazardous
401-500	

Fig. 3. AQI temporal variation in 2009–2013, Tehran, Iran along with associated air quality class lookup table adapted from Taghizadeh et al. (2019).

distance from primary and secondary roads (Bignal et al., 2007), were used in developing the air quality predictive models. Wind speed plays a role in horizontal displacement, re-suspension, and dispersion of air pollutants. Wind direction has a major role in the movement and re-suspension of air pollutants (Sayegh et al., 2014). Cloud cover and air temperature are related to the formation of some pollutants such as PM10. The higher the air temperature, the slower the air movement and the more stable the atmospheric conditions. Due to the dissolution of air pollutants in water molecules, relative humidity is an important variable. High air pressure in cold seasons brings the air pollutants to the proximity of ground causing the amount of oxygen to decrease and the number of hazardous gases to increase (Hooyberghs et al., 2005).


AQI Classes Proportional Map 2009, Tehran, Iran
AQI Classes Proportional Map 2010, Tehran, Iran
AQI Classes Proportional Map 2013, Tehran, Iran
AQI Classes Proportional Map 2011, Tehran, Iran
AQI Classes Proportional Map 2012, Tehran, Iran
AQI Class 7
AQI Class 6
AQI Class 5
AQI Class 4
AQI Class 3
AQI Class 2
AQI Class 1
Kilometers
0
2
4
8
12
16

AQI	Air Pollution Level	Assigned Class
0-50	Excellent	1
51-100	Good	2
101-150	Lightly Polluted	3
151-200	Moderately Polluted	4
201-300	Heavily Polluted	5
301-400	Hazardous	6
401-500		7

Fig. 4. Air quality indices proportional maps for 2009–2013, Tehran, Iran, along with associated air quality classes lookup table adapted from Taghizadeh et al. (2019).

Table 1
The variables of the air quality predictive model based on distributed random forest algorithm.

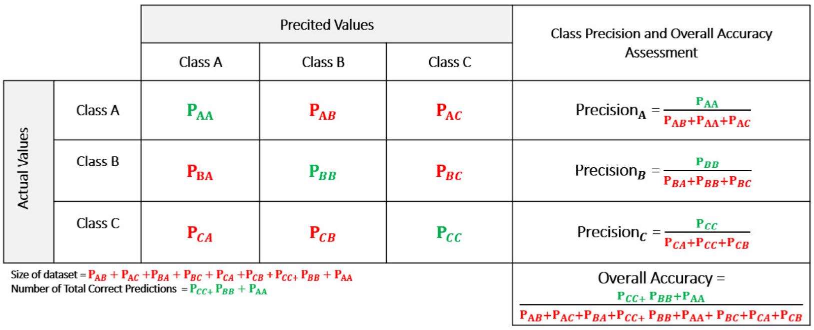
Variable(s) type	Data category	Model variable(s)	Definition
Models input variables	Air pollution data	$AQI^{-24\text{ h}}$, $AQI^{-25\text{ h}}$, …, $AQI^{-72\text{ h}}$	Hourly AQI starting from 24 h to 72 h before the prediction time
	Meteorological data	$Temp^{-24\text{ h}}$, $Temp^{-48\text{ h}}$, $Temp^{-72\text{ h}}$	Temperature (Temp) for 24, 48 and 72 h before the prediction time
		$P^{-24\text{ h}}$, $P^{-48\text{ h}}$, $P^{-72\text{ h}}$	Air Pressure (P) for 24, 48 and 72 h before the prediction time
		$RH^{-24\text{ h}}$, $RH^{-48\text{ h}}$, $RH^{-72\text{ h}}$	Relative humidity (RH) for 24, 48 and 72 h before the prediction time
		$CloudCov^{-24\text{ h}}$, $CloudCov^{-48\text{ h}}$, $CloudCov^{-72\text{ h}}$	Cloud Cover (CloudCov) for 24, 48 and 72 h before the prediction time
		$WS^{-24\text{ h}}$, $WS^{-48\text{ h}}$, $WS^{-72\text{ h}}$	Wind Speed (WS) for 24, 48 and 72 h before the prediction time
		$WD^{-24\text{ h}}$, $WD^{-48\text{ h}}$, $WD^{-72\text{ h}}$	Wind Direction (WD) for 24, 48 and 72 h before the prediction time
	Geospatial data	Elevation Dist_P_Road Dist_S_Road	Elevation of the monitoring station Distance of the monitoring station to primary roads Distance of the monitoring station to secondary roads
Dependent variable		AQI class	Air quality class for the prediction day

3.2. The air quality predictive model based on distributed random forest algorithm and input data

In this study, we developed an air quality predictive model by training the DRF algorithm with a training dataset including 3,500,000 data rows. During the training stage, the DRF algorithm was fed with data for independent variables, as shown in Table 1 (Ghaemi et al., 2015), to predict the dependent variable, which was the AQI class for the next day (+24 h) called the prediction day. The accuracy of the constructed model was evaluated using the test dataset with 600,000 data rows.

3.3. Evaluation methods

We evaluated the DRF algorithm overall accuracy, single class precision, and the parallel execution time and parallel speedup for Tehran's air pollution prediction system.



		Precited Values			Class Precision and Overall Accuracy Assessment
		Class A	Class B	Class C	
Actual Values	Class A	P_{AA}	P_{AB}	P_{AC}	$\text{Precision}_A = \frac{P_{AA}}{P_{AB}+P_{AA}+P_{AC}}$
	Class B	P_{BA}	P_{BB}	P_{BC}	$\text{Precision}_B = \frac{P_{BB}}{P_{BA}+P_{BB}+P_{BC}}$
	Class C	P_{CA}	P_{CB}	P_{CC}	$\text{Precision}_C = \frac{P_{CC}}{P_{CA}+P_{CC}+P_{CB}}$

Size of dataset = $P_{AB} + P_{AC} + P_{BA} + P_{BC} + P_{CA} + P_{CB} + P_{CC+} P_{BB} + P_{AA}$
Number of Total Correct Predictions = $P_{CC+} P_{BB} + P_{AA}$

Overall Accuracy = $\frac{P_{CC+} P_{BB}+P_{AA}}{P_{AB}+P_{AC}+P_{BA}+P_{CC+} P_{BB}+P_{AA}+ P_{BC}+P_{CA}+P_{CB}}$

Fig. 5. Confusion matrix for a three-class classification problem.

- **Overall Accuracy and Class Precision Evaluation of the DFR Algorithm:** For classification problems, the confusion matrix is a two-dimensional matrix that indexes one dimension by the actual values for classification classes and the other by their predicted values. The representation of this matrix for a three-class classification, named A, B and C, is shown in Fig. 5. Each row in this matrix belongs to one class, and it contains the number of correct predictions (e.g., PAA for class A) and the number of wrong predictions (e.g., PAB and PAC for class A) made for that class by the classifier. The class precision is an evaluation measure, for individual classes, and it is calculated as the ratio of the number of correct predictions to the sum of wrong predictions (Fig. 5) (Deng et al., 2016). To show the performance of the DRF algorithm in achieving higher precision measures, we compared it with the results of the Naïve Bays algorithm. Unlike the RF algorithm that uses decision trees as rules for splitting data in a way with the least variation and combines them into an ensemble of trees for prediction, Naïve Bayes represent classifiers based on Bayesian techniques. These techniques calculate the conditional probability or the likelihood of the occurrence of a class based on previous knowledge, and these conditional probabilities are used for the prediction. Naïve Bayes delivers competitive classification accuracy and computational efficiency; however, it is not developed to work with big imbalanced datasets (Murphy, 2006). Thus, this algorithm is a good base algorithm for evaluating improvements in classification accuracy that DRF provides for big imbalanced datasets. In this study, both DRF and Naïve Bayes algorithms were trained using the same training dataset. In addition to class precision values, we also evaluated the overall accuracy for both algorithms, which is calculated by dividing the sum of correct predictions for all classes by the size of the dataset (Sethi and Mittal, 2019).
- **Speedup Gain**: To evaluate the speedup gain, we deployed a cluster of computers with four computers, each having 4 CPU processing cores and 50 GB memory. Since Previous studies, e.g., by George and Sumathi (2020), showed that Spark-based parallel computing systems achieved speedup gain compared to non-parallel systems, we measured the speedup gain when scaling our distributed system horizontally in this study. Horizontal scaling means increasing the number of processing computers at each evaluation stage in the cluster of computers. Parallel Speedup gain, accordingly, is calculated using Eq. (1). In this equation, TN and TP are parallel execution times obtained using n and p CPU processing cores, respectively.

$$SP = \frac{\mathrm{TN}}{\mathrm{TP}} \tag{1}$$

4. Results

Machine learning algorithms for classification problems can achieve high overall accuracy but may have low class precisions for minority classes, which may lead to biased predictive models towards the majority classes. Therefore, improving only the overall accuracy is not sufficient, and a balance between the overall accuracy and the class precision should be considered. Thus, the performance of the proposed parallel computing framework for air quality prediction was defined as its ability to achieve high overall accuracy and class precisions for big imbalanced datasets. Fig. 6. A shows the overall accuracy, and Fig. 6.B shows the class precision for all AQI classes in Tehran's air quality predictive model based on DRF and Naïve Bayes algorithms. The overall accuracy of 0.6802 shows that the DRF algorithm is the better fit model "overall" for a big imbalanced spatiotemporal air quality dataset compared to the Naïve Bayes algorithm (overall accuracy of 0.4541). The green percentages above the charts in Fig. 6.B show the accuracy improvements in the DRF compared to the Naïve Bayes algorithm in predicting each class. The DRF and Naïve Baye algorithms achieved the accuracy of 0.771 and 0.708, respectively, in the prediction of class 6 (a majority class); however, for minority classes, the DRF algorithm showed considerably higher accuracy, while the Naïve Bays algorithm showed a steep accuracy decrease. The accuracy improvement of the DRF vs. Naïve Bayes algorithm increases as the classes move from majority to minority classes (around 82% accuracy improvement for class 2). These results showed that DRF had the advantage of improving both overall accuracy and class precision in air quality prediction using big imbalanced datasets.

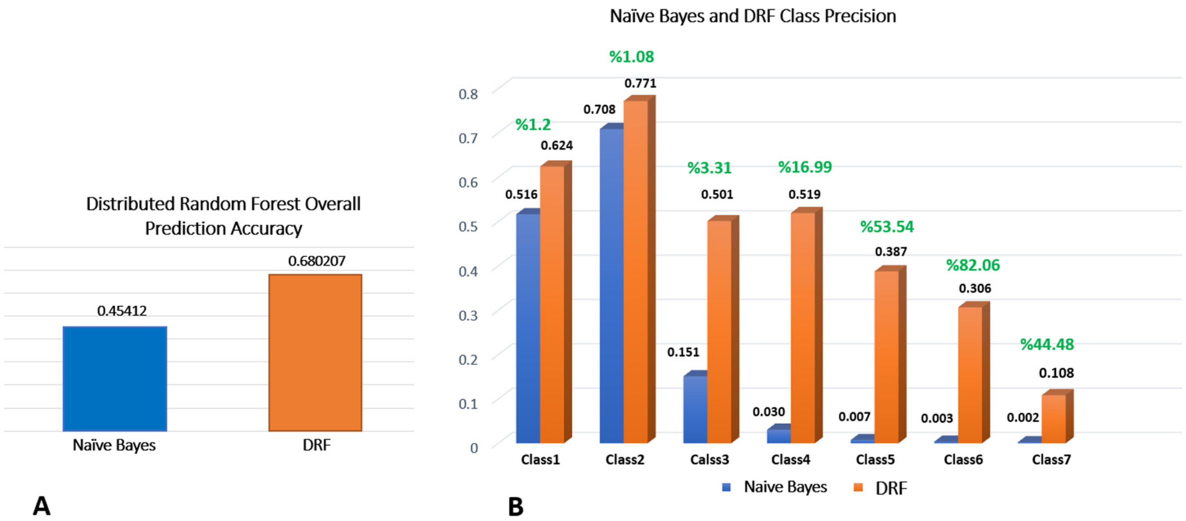


Fig. 6. (A) Overall prediction accuracy for DRF and Naïve Bayes algorithm, (B) Obtained class accuracies in and DRF and Naïve Bayes algorithms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

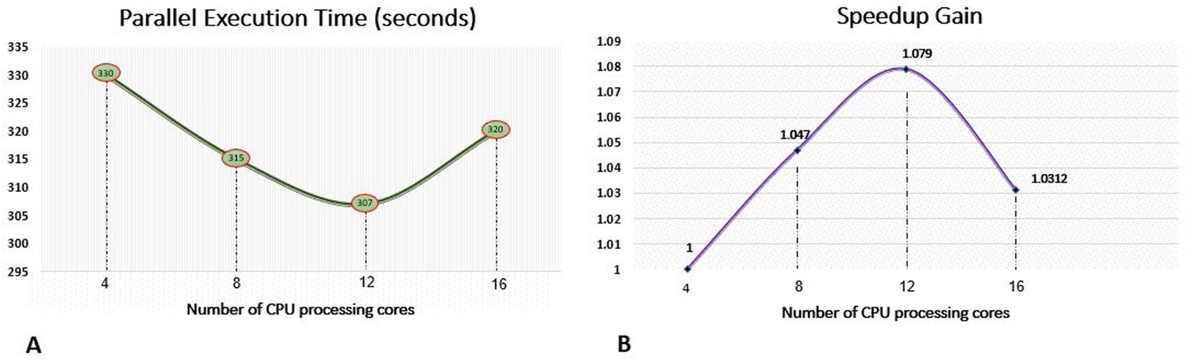


Fig. 7. (A) The parallel execution time and (B) Speedup gain of the parallel air quality predictive system.

Fig. 7.A shows the parallel execution time and Fig. 7.B shows the speedup gain of the parallel computing system for Tehran's air quality prediction on 4, 8, 12, and 16 processing units in four computers.

The execution time of a single run of DRF algorithm on a single computer with 4, 8 and 12 processing units were 330, 315 and 307 s. Till 12 processing units, a direct positive relationship appeared between the number of parallel processing units and the execution speedup; increasing the number of parallel processing units afterwards showed a decreased execution speed. This could be related to the increased communication overhead imposed on the system with increasing the number of "communicators" for assigning partitions to parallel processing units and getting the results back from them. Therefore, although the spatiotemporal partitioning guarantees that parallel processing units be assigned partitions of the same size to prevent having both over-loaded and idle processing units, it increases the number of parallel partitions and consequently the communications among processing units. There is a threshold number of parallel processing units for distributed systems, above which the system is "saturated" due to the communication overhead. This threshold, 12 processing units for our system, should be identified for each distributed computing system since the pattern of speedup gain and saturation is case-specific, which depends on dataset size, partitioning mechanism, computational architectures, etc. The optimal configuration of the distributed system can be identified using empirical analysis of the speedup gain and saturation pattern.

5. Conclusion

To provide rapid, accurate and reliable air quality information for environmental health management, the development of air quality prediction systems becomes critical. However, working with spatiotemporal air quality datasets, which are typically big in their volume and exhibit imbalance and spatial heterogeneity features, may cause data processing challenges and result in biased air quality prediction. To address these complexities in this study, we developed a parallel air quality prediction system based on Hadoop and Spark frameworks, a double-layer spatiotemporal partitioning method and the DRF algorithm implemented in the Sparkling Water machine learning library. The big spatiotemporal data partitioning method enabled us to increase the accuracy of air quality predictive models by addressing the imbalance feature of big datasets and characterizing their spatial heterogeneity. We applied the parallel computing system for air quality prediction in Tehran, Iran, based on an hourly dataset collected from 2009 to 2013. The results showed the air quality predictive model not only achieved high overall accuracy, approximately 68%, but also considerably higher class precisions for minority classes in the classification; around 82% and 44% precision improvements for two minority classes compared to Naïve Bay's obtained results. In addition, by increasing the number of parallel processing units, we observed an increasing trend in the execution speed of the rapid air quality prediction system, with the highest speedup gain of 1.079, until a saturation point, after which execution speed started to decrease. In future studies, data streaming components, such as the Spark Streaming module, can be incorporated to connect air quality monitoring stations to the air pollution prediction system and update predictive models immediately as data becomes available. In addition, the SparkR module can be included in the parallel air quality prediction system to provide data visualization through generating and displaying spatial and temporal distribution maps of air quality classes. Besides, for providing public access to the predicted air quality classes, spatial and temporal distribution maps of air quality information could be hosted on a WebGIS for the rapid air quality prediction based on the proposed parallel computing system.

CRediT authorship contribution statement

Marjan Asgari: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration, Funding acquisition. **Wanhong Yang:** Conceptualization, Writing – original draft, Writing – review & editing, Visualization. **Mahdi Farnaghi:** Conceptualization, Methodology, Resources, Writing – original draft, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aiello, S., Click, C., Roark, H., Rehak, L., Stetsenko, P., 2016. Machine Learning with Python and H2O. H2O. ai Inc, Edited by Lanford, J, Published by H20.

Amini, H., Nhung, N.T.T., Schindler, C., Yunesian, M., Hosseini, V., Shamsipour …, M., Künzli, N., 2019. Short-term associations between daily mortality and ambient particulate matter, nitrogen dioxide, and the air quality index in a Middle Eastern megacity. Environ. Pollut. 254, 113121.

Anuradha, J., 2015. A brief introduction on Big Data 5Vs characteristics and Hadoop technology. Procedia Comput. Sci. 48, 319–324.

Asgari, M., Farnaghi, M., Ghaemi, Z., 2017. Predictive mapping of urban air pollution using Apache Spark on a Hadoop cluster. Proceedings of the 2017 international conference on cloud and big data computing. pp. 89–93.

Ayyalasomayajula, H., Gabriel, E., Lindner, P., Price, D., 2016. Air quality simulations using big data programming models. IEEE, Proc. IEEE 2nd Int. Conf. Big Data Comput. Service Appl. (BigDataService). pp. 182–184.

Azeroual, O., Nikiforova, A., 2022. Apache spark and mllib-based intrusion detection system or how the big data technologies can secure the data. Information 13 (2), 58.

Bai, L., Wang, J., Ma, X., Lu, H., 2018. Air pollution forecasts: An overview. Int. J. Environ. Res. Public Health 15 (4), 780.

Bignal, K.L., Ashmore, M.R., Headley, A.D., Stewart, K., Weigert, K., 2007. Ecological impacts of air pollution from road transport on local vegetation. Appl. Geochem. 22 (6), 1265–1271.

Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

Chen, M., Mao, S., Liu, Y., 2014. Big data: A survey. Mob. Netw. Appl. 19 (2), 171–209.

Cook, D., 2016. Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI. O'Reilly Media, Inc.

Czarnul, P., 2021. Assessment of OpenMP master–slave implementations for selected irregular parallel applications. Electronics 10 (10), 1188.

Del Río, S., López, V., Benítez, J.M., Herrera, F., 2014. On the use of mapreduce for imbalanced big data using random forest. Inform. Sci. 285, 112–137.

Deng, X., Liu, Q., Deng, Y., Mahadevan, S., 2016. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inform. Sci. 340, 250–261.

Du, Y., Ma, C., Wu, C., Xu, X., Guo, Y., Zhou, Y., Li, J., 2017. A visual analytics approach for station-based air quality data. Sensors 17 (1), 30.

Georganos, S., Grippa, T., Gadiaga, A.Niang., Linard, C., Lennert, M., Vanhuysse …, S., Kalogirou, S., 2021. Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling. Geocarto Int. 36 (2), 121–136.

George, S., Sumathi, B., 2020. Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. Int. J. Adv. Comput. Sci. Appl. (IJACSA) 11 (9).

Ghaemi, Z., Farnaghi, M., Alimohammadi, A., 2015. Hadoop-based distributed system for online prediction of air pollution based on support vector machine. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XL-1/W 5, 215–219.

Henger, M., Sarraf, M., 2018. Air Pollution in Tehran: Health Costs, Sources, and Policies. World Bank.

Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F., Brasseur, O., 2005. A neural network forecast for daily average PM10 concentrations in Belgium. Atmos. Environ. 39 (18), 3279–3289.

Jonnalagadda, V.S., Srikanth, P., Thumati, K., Nallamala, S.H., Dist, K., 2016. A review study of apache spark in big data processing. Int. J. Comput. Sci. Trends Technol. (IJCST) 4 (3), 93–98.

Kadri, A., Shaban, K.B., Yaacoub, E., Abu-Dayya, A., 2012. Air quality monitoring and prediction system using machine-to-machine platform. In: International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, pp. 508–517.

Kan, H., London, S., Chen, G., Zhang, Y., Song, G., Jiang, L., Zhao, N., Chen, B., 2008. Season, gender, age, and education as modifiers of the effects of outdoor air pollution on daily mortality in Shanghai, China: the public health and air pollution in Asia (PAPA) study. Epidemiology 19 (6), S92.

Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A., Seliya, N., 2018. A survey on addressing high-class imbalance in big data. J. Big Data 5 (1), 1–30.

Li, L., Li, Z., Reichmann, L., Woodbridge, D., 2019. A Scalable and Reliable Model for Real-time Air Quality Prediction. IEEE, SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 51–57.

Liao, K., Huang, X., Dang, H., Ren, Y., Zuo, S., Duan, C., 2021. Statistical approaches for forecasting primary air pollutants: a review. Atmosphere 12 (6), 686.

Mahmud, M.S., Huang, J.Z., Salloum, S., Emara, T.Z., Sadatdiynov, K., 2020. A survey of data partitioning and sampling methods to support big data analysis. Big Data Min. Anal. 3 (2), 85–101.

Malohlava, M., Hava, J., Mehta, N., 2016. Machine Learning with Sparkling Water: H2o+ Spark. ai Inc, H2O.

Mavridis, I., Karatza, H., 2017. Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark. J. Syst. Softw. 125, 133–151.

Mirabelli, M.C., Ebelt, S., Damon, S.A., 2020. Air quality index and air quality awareness among adults in the United States. Environ. Res. 183, 109185.

Murphy, K.P., 2006. Naive bayes classifiers. Univ. Br. Columbia 18 (60), 1–8.

Nagarajan, G., Ld, D.B., 2019. Predictive analytics on big data-an overview. Informatica 43 (4).

Perwej, Y., Kerim, B., Sirelkhtem, M., Sheta, O.E., 2017. An empirical exploration of the yarn in big data. Int. J. Appl. Inf. Syst. (IJAIS) 12 (19).

Peteiro-Barral, D., Guijarro-Berdiñas, B., 2013. A survey of methods for distributed machine learning. Prog. Artif. Intell. 2 (1), 1–11.

Pishgar, E., Fanni, Z., Tavakkolinia, J., Mohammadi, A., Kiani, B., Bergquist, R., 2020. Mortality rates due to respiratory tract diseases in Tehran, Iran during 2008–2018: a spatiotemporal, cross-sectional study. BMC Public Health 20 (1), 1–12.

Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S., 2016. A survey of machine learning for big data processing. EURASIP J. Adv. Signal Process. 2016 (1), 1–16.

Rastogi, A.K., Narang, N., Siddiqui, Z.A., 2018. Imbalanced big data classification: a distributed implementation of smote. In: Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking, pp. 1–6.

Salloum, S., Dautov, R., Chen, X., Peng, P.X., Huang, J.Z., 2016. Big data analytics on Apache Spark. Int. J. Data Sci. Anal. 1 (3), 145–164.

Salunkhe, U.R., Mali, S.N., 2016. Classifier ensemble design for imbalanced data classification: a hybrid approach. Procedia Comput. Sci. 85, 725–732.

Sayegh, A.S., Munir, S., Habeebullah, T.M., 2014. Comparing the performance of statistical models for predicting PM10 concentrations. Aerosol Air Qual. Res. 14 (3), 653–665.

Sethi, J., Mittal, M., 2019. Ambient air quality estimation using supervised learning techniques. EAI Endorsed Trans. Scalable Inf. Syst. 6 (22).

Shetty, M.M., Manjaiah, D.H., 2016. Data security in Hadoop distributed file system. IEEE, International Conference on Emerging Technological Trends (ICETT). pp. 1–5.

Song, R., Yang, L., Liu, M., Li, C., Yang, Y., 2019. Spatiotemporal distribution of air pollution characteristics in Jiangsu Province, China. Adv. Meteorol. 2019.

Sun, X., Xu, W., Jiang, H., Wang, Q., 2021. A deep multitask learning approach for air quality prediction. Ann. Oper. Res. 303 (1), 51–79.

Taghizadeh, F., Jafari, A.J., Kermani, M., 2019. The trend of air quality index (AQI) in Tehran during (2011-2016). J. Air Pollut. Health 4 (3), 187–192.

Triguero, I., Galar, M., Merino, D., Maillo, J., Bustince, H., Herrera, F., 2016. Evolutionary undersampling for extremely imbalanced big data classification under apache spark, IEEE, Congress on Evolutionary Computation (CEC). pp. 640–647.

Weiss, G.M., Provost, F., 2003. Learning when training data are costly: The effect of class distribution on tree induction. J. Artificial Intelligence Res. 19, 315–354.

Wu, Q., She, Q., Jiang, P., 2020. Class Imbalance SS-ELM for Regional Air Pollution Prediction. IEEE, 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS). pp. 440–445.

Xie, X., Semanjski, I., Gautama, S., Tsiligianni, E., Deligiannis, N., Rajan …, R.T., Philips, W., 2017. A review of urban air pollution monitoring and exposure assessment methods. ISPRS Int. J. Geo-Inf. 6 (12), 389.

Yang, W., Deng, M., Xu, F., Wang, H., 2018. Prediction of hourly PM2. 5 using a space–time support vector regression model. Atmos. Environ. 181, 12–19.

Yao, X., Mokbel, M.F., Alarabi, L., Eldawy, A., Yang, J., Yun …, W., Zhu, D., 2017. Spatial coding-based approach for partitioning big spatial data in hadoop. Comput. Geosci. 106, 60–67.

Yu, J., Wu, J., Sarwat, M., 2016. A demonstration of GeoSpark: A cluster computing framework for processing big spatial data. IEEE, 32nd International Conference on Data Engineering (ICDE). pp. 1410–1413.

Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave …, A., Stoica, I., 2016. Apache spark: a unified engine for big data processing. Commun. ACM 59 (11), 56–65.

Zhang, C., Yuan, D., 2015. Fast fine-grained air quality index level prediction using random forest algorithm on cluster computing of spark. IEEE, 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom). pp. 929–934.

Zhu, D., Cai, C., Yang, T., Zhou, X., 2018. A machine learning approach for air quality prediction: Model regularization and optimization. Big Data Cogn. Comput. 2 (1), 5.