A Scalable and Reliable Model for Real-time Air Quality Prediction

Liying Li*, Zhi Li*, Lara G. Reichmann, Diane Myung-kyung Woodbridge
lli93,zli142,lreichmann,dwoodbridge@usfca.edu
Data Institute
University of San Francisco

***Abstract*—Air pollution has become a concern for people in California. Predictions of Air Quality Index are, on one hand, benefit by the high frequency, the large number of sampling stations, and the substantial size of relevant data. On the other hand, it is challenging to effectively store, manage, and process a vast amount of data in real-time. In this research, we explore a pipeline to store, process, and make predictions applying machine learning models to the air quality datasets. The pipeline was built using Apache Spark SQL and Spark machine learning libraries (MLlib) on AWS Elastic MapReduce (EMR). Using a 10-year air quality dataset of California, we developed Logistic Regression and Random Forest Classification machine learning models on a local machine as well as on a distributed system. We found that employing Amazon S3, MongoDB and Apache Spark, the distributed setting on a cluster achieved better computational performance than a non-distributed setting.**

***Index Terms*—Distributed computing, Distributed databases, Distributed information systems, Machine learning, Predictive models, Air pollution, Air quality**

I. Introduction

Due to the rapid global industrialization and urbanization process, environmental pollution issues such as air pollution have become more and more severe. Air pollutants are a mixture of solid particles and gases in the air. Major outdoor air pollutants include ozone (O_3), particle matter (PM), sulfur dioxide (SO_2), carbon monoxide (CO) and nitrogen oxides (NO_x) [1]. Some air pollutants can trigger adverse health problems, including respiratory disease, heart disease, lung cancer, brain damage, liver damage and kidney damage [2]. For instance, it has been reported that fine particulate ($PM_{2.5}$) and sulfur-oxide related pollution are associated with lung cancer and cardiopulmonary mortality [3].

The problem of air pollution is quite severe in California. According to the American lung association's annual "State of the Air" report, California has the eight of the USA's ten most-polluted cities in terms of ozone pollution and it has seven of the USA's ten most-polluted cities both in terms of by year-round particle pollution and by short-term particle pollution [4]. The air pollution issue in California is largely caused by the regular occurrences of wildfires [5]. According to the California Department of Forestry and Fire Protection, the average number of fires in California in five years exceeded 3,500, which led to more than 190 thousand acres being burned and more than 600 million dollars of economic loss on average [6]. Emissions from wildfires can increase the concentration level of PM, CO_2, NO as well as other air pollutants, resulting in poor air quality [7]. For example, Camp Fire, the most destructive fire on record, broke out in Butte County, Northern California on November of 2018, and caused the air pollution level of Northern California to be the worst in the world [8]. Due to the poor air quality, several public schools in Northern California had to be closed in response to the health warning. Poor air quality can also cause a significant increase in hospital emergency room visits for asthma, respiratory problems, eye irritation, and smoke inhalation [9].

Air quality forecasting is an effective way of protecting public health by providing an early warning against harmful air pollutants. As certain air pollutants can cause severe health problems, it is essential to predict air quality in real-time. Therefore, especially in California, it is of great importance to predict air quality to give advance warning and let the public sectors such as schools, emergency services, and health-care engage in early prevention and planning.

The Air Quality Index (AQI) is an index for reporting daily air quality and it can be used to warn the public when air pollution is hazardous. For instance, as the AQI increases, the public is likely to experience increasingly severe adverse health effects. The United States Environmental Protection Agency (US EPA) calculated the AQI for five major pollutants: particulate matter (PM), sulfur dioxide (SO_2), carbon monoxide (CO), nitrogen dioxide (NO_2), ground-level ozone (O_3). Air quality can be classified into six levels according to AQI. The AQI levels of 1 to 6 indicates excellent, good, mild pollution, moderate pollution, heavy pollution, and serious pollution, respectively. According to AQI technical specification HJ633-2010 (for trial implementation), Table I shows the AQIs and their corresponding air quality levels [10].

The calculation of AQI involves complex mathematical and statistical techniques to model air circulation that often require a large quantity of historical measurement data under various atmospheric conditions. In the absence of historical data, deterministic approaches can be used when there is an access to real-time emissions and the physical process in the atmosphere is known. Some deterministic forecasts couple meteorological-chemistry models that are composed of simplified or more comprehensive 3-D chemistry transport models. These approaches are computationally expensive and usually provide limited accuracy especially for predicting the

[*] These authors contributed equally.

TABLE I: AQIs and Air Quality Levels

Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0 to 50	Air quality is considered satisfactory, and air pollution posse little or no risk.
Moderate	51 to 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 to 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health alter : everyone may experience more serious health effects.
Hazardous	301 to 500	Health warnings of emergency conditions. The entire population is more likely to be affected.

extreme pollution points [11]. Now, with the development of big data, many studies have been able to achieve better accuracy in the prediction of air quality through machine learning or deep learning based models. For example, Zheng, et al. used both linear regression-based models and a neural network-based model to predict AQI over the next 48 hours in China [12]. Zhao, et al. applied a random forest approach for predicting air quality for the urban sensing system in Shenyang, China [13]. Yu, et al. used an improved Artificial Neural networks (ANN) model called GA-ANN to predict AQI in Tianjin, China, in which a genetic algorithm (GA) is used to select a subset of factors from the original set and the GA-selected factors are fed into ANN for modeling [14]. Li, et al. applied a novel spatiotemporal deep learning (STDL)-based model to predict air quality [15].

Unfortunately, many of the studies simply applied the machine learning models and did not take account of scalable approaches for storing, managing and processing big data. As the volume of data increases, it would be computationally expensive to train the aforementioned machine learning and deep learning models without using high-performance computing (HPC) or distributed computing systems. Especially, distributed systems can achieve high-quality aggregate performance by connecting many networked computers to compute and process tasks on one or multiple commodity machines. In addition, distributed computing provides high reliability by replicating data and saving copies into other machines or storing data dependencies [16], [17], [18]. Therefore, distributed systems enable us to process data in a fast, reliable and cost-efficient way.

In this study, we focus on using big data technologies including MongoDB and Apache Spark on Amazon Web Services (AWS) and applying machine learning algorithms to model and predict California daily AQI. This method will provide with better computation efficiency on the basis of historical meteorological data and air pollutant data. By doing this, this study seeks to provide advance warning to the public and the public sector so that they can engage in pre-event planning. To achieve these, our study evaluated Spark performance under various machine learning algorithms. We also compared the performance of similar algorithms running on various cluster settings using Spark and a single local machine using Scikit-learn, a popular Python machine learning library. This would help us to determine the benefits of implementing pipelines on distributed systems for machine learning algorithms to achieve better computational efficiency.

II. Background

A. NoSQL and MongoDB

There have been significant increases in the amount of data that various web and internet of things (IoT) applications collect. There are 2.5 quintillion bytes of data created every day and the schema of a relational database has to be changed frequently in order to store unstructured data, or data with frequent schema changes [19]. For storing data with explosive volume growth, the needs of an affordable but robust system arose. In the late 2000s, many research and open source projects including Google BigTable [20] and Amazon Dynamo [21] demonstrated great performance and scalability using newly developed non-relational databases, NoSQL (Not Only SQL). Many of the NoSQL databases support distributed data stores by dividing and storing data in different servers (shards) and improve availability by maintaining its replications in other servers. In addition, many NoSQL databases support storing schemaless data and are designed to store data which are closely related as an aggregate in the same server node.

MongoDB is an open-source document database, a type of NoSQL which stores data as a JSON document and allows to store data to have a various number of non-static fields and corresponding values [22]. Users can easily write queries for retrieving and updating databases using its query language. MongoDB also enables to store and query large volume data with high availability, automatic scaling and time cost efficiency. MongoDB allows to add additional machines as the data volume increases. Additionally, MongoDB automatically elects a primary node and secondary nodes among a replica set for storing original data and its copies in case of a system failure and for achieving high availability. When a large number of users try to access data, this distributed setting provides faster read and write accesses compared to a single node setting.

B. MapReduce and Apache Spark

Hadoop's MapReduce, introduced in 2004, was an innovative implementation of distributed computing in an attempt to speed up large scale data analysis [23]. MapReduce splits data into smaller chunks across different machines, and subsequently maps and processes a task, e.g., filtering and sorting, in parallel. The output of a mapped task becomes the input of a reduce operation, which returns a final answer to a driver node. This highly-effective model allows users to design programs with successive Map and Reduce operations, and is the most popular distributed processing model.

Apache Spark is an open-source distributed computing framework, which extends the MapReduce model with primitives for efficient data sharing by using resilient distributed datasets (RDDs). In general, Spark has several advantages over other distributed computing systems: fast speed, ease of use, generality and runs everywhere. To be more specific, Spark has an advanced directed acyclic graph (DAG) execution engine which enables it to run programs up to 100 times faster than Hadoop MapReduce in memory, or 10 times faster on disk. It is ease of use, as Spark can be used interactively with Scala, Java, Python and R. Spark provides Spark SQL and ML libraries for managing structured data and tackling machine learning problems. Spark can not only run on a standalone cluster mode which is an optimized resource manager for Spark, it can also run on other resource managers including Hadoop Yet Another Resource Negotiator (YARN) resource managers, originally built for Apache Hadoop but also working with Spark [24].

III. System Overview

A. System Workflow

For this study, the data pipeline was designed around scalability, cloud resources, and distributed computing methods. In this case, we developed our pipeline using Amazon Web Service (AWS), as AWS can provide high availability and scalability and makes its components including storage and processing engines to be compatible. Firstly, our preprocessed data was stored in AWS Simple Storage Service (S3) bucket [25], then the data was transferred and loaded into a MongoDB on AWS EC2. Later, our data was processed with machine learning algorithms using Apache Spark installed on AWS EMR (Figure 1).

1) Data Storage: Data was scraped from US Environmental Protection Agency (EPA) AQS Data Mart [26] using its APIs. After preprocessing data using Google BigQuery, data was stored in AWS S3 bucket. S3 was selected for its high scalability, reliability and efficient data I/O. AWS S3 also allows to store and retrieve any amount of data at any time, from anywhere on the web.

2) Data Management: Since MongoDB allows to store data of any structure and supports sharding (partitioning data) and replication (storing copied data), we chose MongoDB as our data management system. In our study, we deployed MongoDB on an AWS EC2 cluster with a total of 10 instances. Half of the instances were t2.large (2 CPUs with 8 GB RAM) while the other half were t2.medium (2 CPUs with 4 GB RAM) instances. AWS EC2 is a web service that provides secure, reliable and resizable capacity in the cloud. EC2 enables to build a MongoDB cluster by specifying configurations and deployment specifications.

3) Data Processing and Machine Learning: Data processing and machine learning was performed using Apache Spark SQL and Spark machine learning libraries (MLlib) on AWS Elastic MapReduce (EMR). AWS EMR can provide a managed Hadoop framework for processing vast amounts of data across AWS EC2 instances with an easy, fast and cost-effective setting. The EMR enables us to deploy Apache Spark applications and interact with data in other AWS data stores such as AWS S3 and MongoDB built on AWS EC2. For this research, we ran two different EMR cluster settings: one with three m4.large (2 CPUs, 8 GB RAM) instances another with three m4.xlarge (4 CPUs, 16 GB RAM) instances.

For each setting, one instance was set up as a master and the remaining two were launched as slaves. Once MongoDB was connected with AWS EMR, the data was processed using the following six steps: 1) Data transfer, 2) RDDs creation, 3) DataFrame creation, 4) DataFrame processing, 5) Machine learning model training, and 6) Validation. The performance of various cluster settings using AWS EMR was also compared with the performance of a MacBook Pro with 512 GB Flash Storage, 8 GB RAM, and 2.9 GHz Intel Core i5 (2 CPUs).

B. Algorithms

Spark MLlib supports various machine learning algorithms with enhanced performance using its distributed computing framework. In this study, we used the 47 features in Table II to predict the daily AQI category per city using the following supervised learning algorithms in MLlib.

1) Logistic Regression: Logistic regression is a predictive analysis used for binomial or multinomial classification problems [27]. Logistic Regression minimizes a linear combination of our input features through gradient descent. Unlike linear regression which returns continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes [28].

2) Random Forest Classification: Random Forest classification is an ensemble method that fits multiple decision tree classifiers on various bootstrap samples (which re-samples the data many times with replacement and re-estimates the model) [29]. The output class is the mode of the classes of the individual trees, which attains the majority of the votes. Due to the instability of a single regression tree, random forest classification tends to improve predictive accuracy as well as overcome an issue of overfitting to the training data set.

IV. Experiment Output

A. Data

The data that we use is from the AQS Data Mart, which contains the historical air quality information across the state.

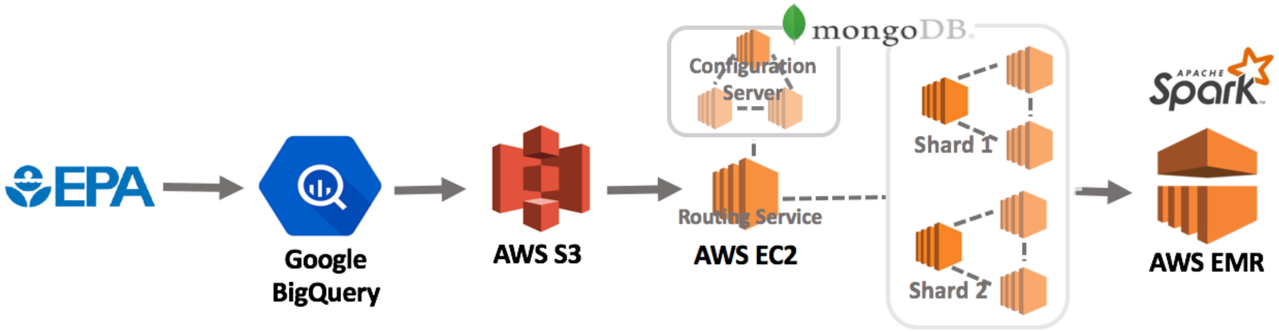


Fig. 1: System workflow

TABLE II: 47 features used for supervised learning

Feature	Value
month	month of the year
day	day of month
dow	day of week
county_code	The Federal Information Processing Standard (FIPS) code of the county in which the monitor resides
city_name	The name of the city where the monitoring site is located
longitude	City level averaged monitoring sites angular distance east of the prime meridian measured in decimal degrees
latitude	City level averaged monitoring sites angular distance north of the equator measured in decimal degrees
observation_count_co	The number of samples taken during the day for CO
observation_count_no2	The number of samples taken during the day for NO2
observation_count_o3	The number of samples taken during the day for O3
observation_count_pm10	The number of samples taken during the day for PM10 Mass
observation_count_pm25_frm	The number of samples taken during the day for PM2.5 FRM/FEM Mass
observation_count_pm25_nonfrm	The number of samples taken during the day for PM2.5 non FRM/FEM Mass
observation_count_pm25_speciation	The number of samples taken during the day for PM2.5 Speciation
aqi_co	AQI of CO for the day
aqi_no2	AQI of NO2 for the day
aqi_o3	AQI of O3 for the day
aqi_pm10	AQI of PM10 for the day
aqi_pm25_frm	AQI of PM2.5 FRM/FEM Mass for the day
aqi_pm25_nonfrm	AQI of PM2.5 non-FRM/FEM Mass for the day
aqi_so2	AQI of SO2 for the day
arithmetic_mean_co	The average value of CO for the day
arithmetic_mean_no2	The average value of NO2 for the day
arithmetic_mean_o3	The average value of O3 for the day
arithmetic_mean_pm10	The average value of PM10 for the day
arithmetic_mean_pm25_frm	The average value of PM2.5 FRM/FEM Mass for the day
arithmetic_mean_pm25_nonfrm	The average value of PM2.5 non-FRM/FEM Mass for the day
arithmetic_mean_pm25_speciation	The average value of PM2.5 Specification for the day
arithmetic_mean_pressure	The average value of barometric pressure for the day
arithmetic_mean_so2	The average value of SO2 for the day
arithmetic_mean_temp	The average value of temperature for the day
arithmetic_mean_wind	The average value of resultant wind for the day
max_value_co	The highest CO value for the day
max_value_no2	The highest NO2 value for the day
max_value_o3	The highest O3 value for the day
max_value_pm10	The highest PM10 value for the day
max_value_pm25_frm	The highest PM2.5 FRM/FEM Mass value for the day
max_value_pm25_nonfrm	The highest PM2.5 non-FRM/FEM Mass value for the day
max_value_pm25_speciation	The highest PM2.5 Speciation value for the day
max_value_pressure	The highest barometric pressure value for the day
max_value_so2	The highest SO2 value for the day
max_value_temp	The highest temperature value for the day
max_value_wind	The highest resultant wind value for the day
max_aqi	The Air Quality Index for the day
max_aqi_before_yesterday	The Air Quality Index for two days before the day
max_aqi_yesterday	The Air Quality Index for one day before the day
label	The Air Quality Index for the next day

It also includes associated aggregate values per 8-hour, day, year, etc. The factors include air pollutants that would affect AQI, such as CO, NO_2, O_3, PM_{10}, $PM_{2.5}$ firm, $PM_{2.5}$ Non-firm, $PM_{2.5}$ speciation and SO_2. It also includes other factors like temperature, wind, and pressure. As our objective was to benchmark performance rather than achieve marginal gains in prediction accuracy, we conducted feature engineering which was not too rigorous and performed analysis using daily aggregates. We used the concentration level of air pollutants with their associated AQI and other influential factors like temperature, wind to predict the AQI of the next day.

TABLE III: Cluster types

	Local Machine	Cluster 1	Cluster 2
Resource Manager	Standalone	YARN	YARN
Number of Nodes	1	3	3
Number of CPU	4	2	4
Memory (GB)	8	8	16

We used the BigQuery Python client library and BigQuery API to query the data using SQL. Based on the city name and date, the tables of different features were joined. Since our goal is to predict an AQI level on the city basis, the data was then grouped by the same city to get the mean value of each air pollutants, temperature, wind, pressure as well as the max AQI value for each city.

Our preprocessed data included 600,000 observations of California air quality information collected between the year of 2008 and the year of 2017 with a size of approximately 1.02 GB. Using this dataset allowed us to evaluate the computation efficiency of the system for storing, managing and processing data.

B. Results

Both Logistic Regression and Random Forest models were built on three different settings in Table III. To evaluate the performance of our models, we measured model fitting time as well as compared their accuracy and F1 score:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

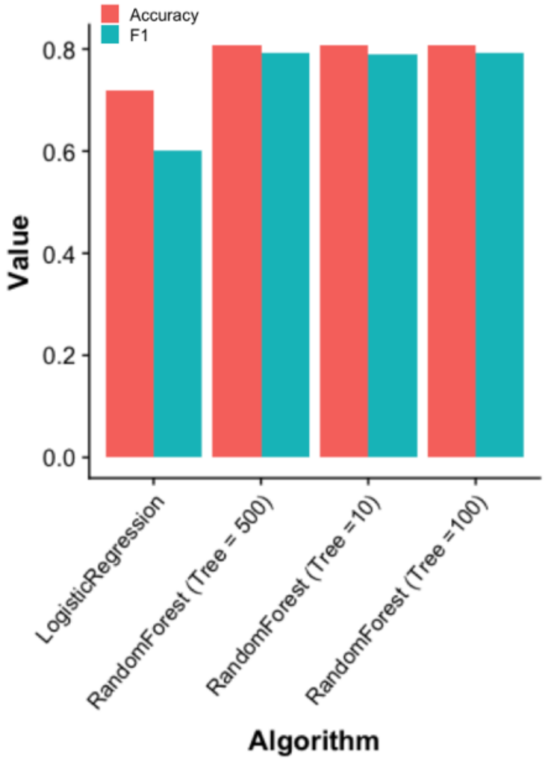


Fig. 2: Model evaluation on an EMR cluster with 8GB RAM

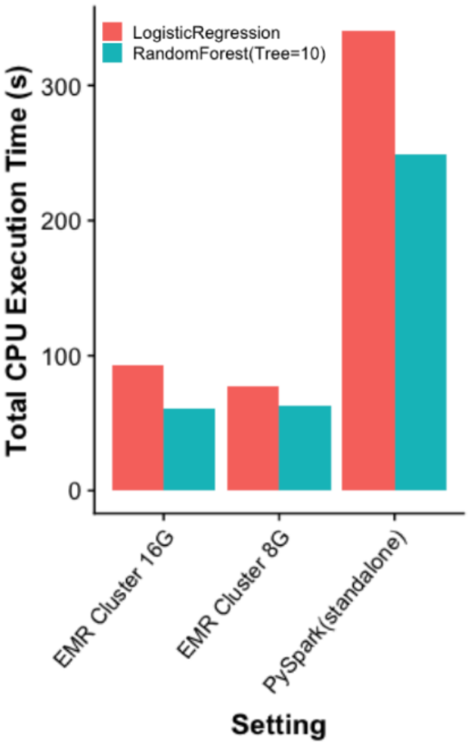


Fig. 3: Comparison of model training time

The experimental result in Figure 2 shows that Random Forest outperformed Logistic Regression in terms of both accuracy and F1 score. While Logistic Regression had 0.72 and 0.60 for accuracy and F1 respectively, Random Forest showed 0.81 and 0.79. In this case, the number of trees does not quite affect either accuracy or F1 score of Random Forest models. In addition, Random Forest required less training time than Logistic Regression using Spark (Figure 3). It took 341 and 249 seconds for Logistic Regression and Random Forest respectively on a single node setting, while it only took 77.6 and 62.7 seconds for each algorithm on the EMR Cluster 1. This proves that a distributed cluster setting can highly enhance execution time.

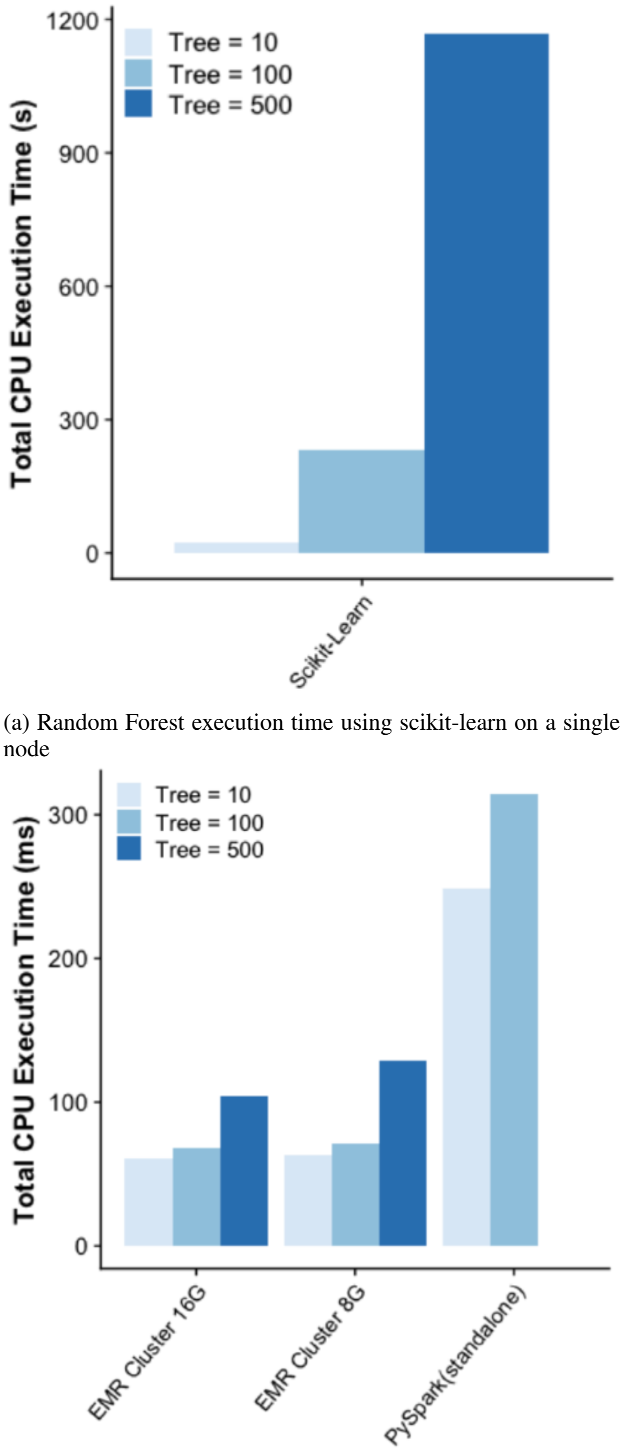
As the values of hyperparameters changes, supervised models could become more complex, thus taking more time for a machine to train a model. For instance, as the number of estimators changes, the Random Forest algorithm could become very complex. Basically, when the number of estimators grows, it would take longer to train the model. As the algorithm complexity increases, researchers and developers often face challenges including memory limits and execution time. In order to benchmark Spark MLlib performance, we developed Random Forest using Scikit-learn, the most popular machine learning library for Python on a local machine without a parallelizing process and Spark on three different cluster settings in Table III with three different tree sizes including 10, 100 and 500 trees. For the aforementioned algorithms, the Logistic Regression showed that it can deal with a large size of data relatively easily. Overall, compared to Random Forest with 10 trees, the Logistic Regression took longer to train. The differences between the execution time for these two algorithms were not significant when the models were trained on EMR clusters. However, there was a significant difference between the execution time of the two algorithms in the setting of PySpark standalone.

Figure 4 shows that Random Forest in Scikit-learn took 23.7 seconds to train a model with 10 trees, while the distributed computing-based algorithms took as short as 61 milliseconds which was 388.52 times faster. Comparing to a single node setting on the standalone mode, the two types of EMR clusters showed a significantly better time efficiency. When the tree count was 500, the local machine setting failed to execute due to memory shortage, while it took only 129 and 104 milliseconds for cluster 1 and 2 accordingly.

V. Conclusion

As air quality prediction is highly related to public health in major cities, it is critical to develop a system which can collect, store and manage large amount of data and apply machine learning algorithms in real-time for providing timely alerts. To achieve this goal, we developed a scalable and reliable data storage and processing pipeline, utilizing S3, a cloud data storage, MongoDB, a distributed database and Apache Spark, a distributed computing framework.

In our experiment, Random Forest showed 0.81 and 0.79 for accuracy and F1 respectively, showing better performance



(a) Random Forest execution time using scikit-learn on a single node

(b) Random Forest execution time using Spark on different cluster settings

Fig. 4: Random Forest execution time in different settings

than Logistic Regression. In addition, deploying the model on a cluster with various hyperparameter settings proved that a distributed setting on a cluster achieved better computational performance than a non-distributed setting. Compared to non-distributed settings, the study result is promising, proving that the designed pipeline can provide a scalable and efficient throughput of machine learning algorithms for air quality prediction.

Our research also shows that, when handling large datasets, a standalone machine has limited memory size to progress the data. In those cases, a distributed system which combines several standalone machines together to perform the computations, would succeed without exceeding memory. Theoretically, larger datasets could be handled by deploying more machines, with the caveat of incurring into higher costs. Therefore, we must carefully weigh the trade-off between the improvements in the power of data manipulation and the real cost for the particular application.

For future work, we will explore opportunities to combine air quality data collected from IoT devices to provide air quality prediction at a higher resolution.

REFERENCES

[1] L. Curtis, W. Rea, P. Smith-Willis, E. Fenyves, and Y. Pan, “Adverse health effects of outdoor air pollutants,” *Environment international*, vol. 32, no. 6, pp. 815–830, 2006.

[2] J. M. Samet, S. L. Zeger, F. Dominici, F. Curriero, I. Coursac, D. W. Dockery, J. Schwartz, and A. Zanobetti, “The national morbidity, mortality, and air pollution study,” *Part II: morbidity and mortality from air pollution in the United States Res Rep Health Eff Inst*, vol. 94, no. pt 2, pp. 5–79, 2000.

[3] C. A. Pope III, R. T. Burnett, M. J. Thun, E. E. Calle, D. Krewski, K. Ito, and G. D. Thurston, “Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution,” *Jama*, vol. 287, no. 9, pp. 1132–1141, 2002.

[4] A. L. Association. (2018) Most polluted cities. [Online]. Available: https://www.lung.org/our-initiatives/healthy-air/sota/city-rankings/most-polluted-cities.html

[5] Climate Central. (2017) Western wildfires undermining progress on air pollution. Climate Central. [Online]. Available: https://www.climatecentral.org/news/report-wildfires-undermining-air-pollution-progress-21753

[6] C. D. of Forestry and F. Protection. (2018) Cal fire jurisdiction fires, acres, dollar damage, and structures destroyed (1933-2016). [Online]. Available: http://cdfdata.fire.ca.gov/pub/cdf/images/incidentstatsevents_270.pdf

[7] N. R. Council *et al.*, *Air quality management in the United States*. National Academies Press, 2004.

[8] J. Dineen and G. Wu. (2018) Northern california air quality rated the worst in the world, conditions hazardous. San Francisco Chronicle. [Online]. Available: https://www.sfchronicle.com/california-wildfires/article/Smoke-still-plagues-Bay-Area-skies-a-week-after-13394932.php

[9] R. J. Delfino, S. Brummel, J. Wu, H. Stern, B. Ostro, M. Lipsett, A. Winer, D. H. Street, L. Zhang, T. Tjoa *et al.*, “The relationship of respiratory and cardiovascular hospital admissions to the southern california wildfires of 2003,” *Occupational and environmental medicine*, vol. 66, no. 3, pp. 189–197, 2009.

[10] US EPA. (2011) Air quality index(aqi) a guide to air quality and your health. [Online]. Available: https://www.airnow.gov/index.cfm?action=aqibasics.aqi

[11] V. M. Niharika and P. S. Rao, “A survey on air quality forecasting techniques,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 103–107, 2014.

[12] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, “Forecasting fine-grained air quality based on big data,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2267–2276.

[13] H. Zhao, J. Zhang, K. Wang, Z. Bai, and A. Liu, “A ga-ann model for air quality predicting,” in *Computer Symposium (ICS), 2010 International*. IEEE, 2010, pp. 693–699.

[14] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, “Raq–a random forest approach for predicting air quality in urban sensing systems,” *Sensors*, vol. 16, no. 1, p. 86, 2016.

[15] X. Li, L. Peng, Y. Hu, J. Shao, and T. Chi, “Deep learning architecture for air quality predictions,” *Environmental Science and Pollution Research*, vol. 23, no. 22, pp. 22 408–22 417, 2016.

[16] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed systems: concepts and design*. pearson education, 2005.

[17] A. Howard, T. Lee, S. Mahar, P. Intrevado, and D. Woodbridge, “Distributed data analytics framework for smart transportation,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2018, pp. 1374–1380.

[18] S. Suma, R. Mehmood, N. Albugami, I. Katib, and A. Albeshri, “Enabling next generation logistics and planning for smarter societies,” *Procedia Computer Science*, vol. 109, pp. 1122–1127, 2017.

[19] B. Marr. (2018) How much data do we create every day? Forbes. [Online]. Available: https://www.forbes.com/

[20] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable: A distributed storage system for structured data,” *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, p. 4, 2008.

[21] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: amazon’s highly available key-value store,” in *ACM SIGOPS operating systems review*, vol. 41, no. 6. ACM, 2007, pp. 205–220.

[22] MongoDB. (2018) Mongodb for giant ideas. [Online]. Available: https://www.mongodb.com/

[23] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[24] A. Spark. (2019) Apache spark: Lightning-fast cluster computing. [Online]. Available: http://spark.apache.org

[25] Amazon Web Service . (2019) Amazon s3. [Online]. Available: https://aws.amazon.com/s3/

[26] US Environmental Protection Agency. Air quality system data mart. United States Environmental Protection Agency. [Online]. Available: http://www.epa.gov/ttn/airs/aqsdatamart

[27] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*. Irwin Chicago, 1996, vol. 4.

[28] J. S. Cramer, “The origins and development of the logit model,” *Logit models from economics and other fields*, pp. 149–158, 2003.

[29] I. Barandiaran, “The random subspace method for constructing decision forests,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, 1998.