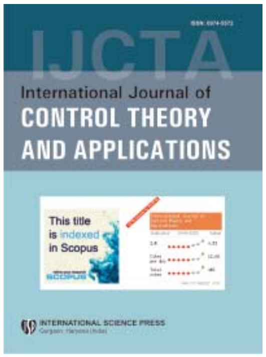


International Journal of Control Theory and Applications

ISSN : 0974–5572

© International Science Press

Volume 9 • Number 40 • 2016

Comparative Study of XGBoost4j and Gradient Boosting for Linear Regression

S. Ramraj[a] Aeshita Saini[a] and Gurleen Kaur[a]

[a]*Software Engineering, SRM University, Chennai Tamil Nadu, India*

E-mail: ramrajitsrm333@gmail.com, Asaini122@gmail.com, Gurleenkaur.95.gk@gmail.com

Abstract: Enterprises and companies now-a-days rely on large scale data analysis as a part of their core services for automating many manual activities. Machine learning [1] provide computers with the ability to learn without being programmed explicitly. Many machine learning methodologies can be implemented to analyze and predict data, using such algorithms which can iteratively learn from past data, making the job easier for companies to make better decisions for their products. Prediction is the one of the application of machine learning, linear regression is a supervised machine learning approach which is prominent in prediction of numerical data. It is closely related to computational statistics, which focuses on prediction-making through the use of computers. In this paper we are going to validate the performance, efficiency of XGBoost 4j package in a distributed programming environment called Apache Spark for predicting the speed of wind. Taking the algorithm as XGBoost which is short for Extreme Gradient Boosting [2] is a tool motivated by supervised learning, where we use the training data *xi* to predict a target variable *yi*. Added that we will compare the results obtained from our framework with Gradient Boosting, which will hence evaluate the efficiency, accuracy etc. for the prediction of the dataset. Therefore, XGBoost is developed with both deep considerations of system optimization and principles of machine learning. The goal of this library is to push extreme computation limits of the machine to provide a scalable, portable and accurate library. Therefore, to improvise the predictions further *i.e*. the chances of finding the errors in the data can be done by using Apache Spark [3]. Spark is ideal for iterative processing and interactive processing. It also executes faster batch processing jobs. Also in this framework we will compare the results of XGBoost4j in a distributed programming environment versus the Gradient Boosting in normal programming environment.

Keywords: *XGBoost, Gradient Boosting, Apache Spark, Regression Model.*

1. INTRODUCTION

Today's world still relies on fossil fuels as their primary energy resources. Because of some drawbacks human beings work on alternative energies like wind, hydro and solar etc. for primary energy resources as these energies don't get deplete. Wind and solar energies are quite similar as they both depend on atmospheric conditions as they are affected due to rain, cloudy days etc. Wind is renewable, non-polluting, free source of energy which is the reason for much research efforts in this field [4]. Also, the wind turbines don't produce atmospheric emissions and are available in various sizes which means vast range of people from village to town can make good use of them.

The major advantage of utilizing wind energy as a resource is that it acts like a clean fuel, also it is free, can be captured easily also is a renewable source of energy. The disadvantage includes that the strength of wind is not constant that means that wind turbines are not able to produce same amount of energy always, they are noisy and create pollution when are being manufactured.

There are different methods for wind forecasting which depend on time period. Starting from ultra-short forecasting to long forecasting varying from a range of few seconds to a week ahead. In this paper we concentrate on short term forecasting method which includes the prediction range from 0-4 hours. There has been a lot of work done upon short term forecasting, such as the physical approach and the statistical approach. Different methods for wind forecasting are 1) Hybrid method includes the combination of regression, physical method, and other techniques to produce forecast that can overcome the limitation of individual methods [5]. 2) Numerical weather prediction utilizes mathematical model of atmosphere and oceans to predict the weather bases on the persistence weather conditions. 3) Persistence method provides an assumption that the wind forecast made at a certain time $y + z$ will be same for y. This method is helpful for ultra-short forecasting.4) Artificial Neural Network method, 5) Physical method is based on the Numerical Wind Prediction data which includes surface temperature, surface pressure etc. as atmosphere parameters which provides precise prediction.

The dataset is taken from National Institute of Wind Energy on which the entire experimental analysis have been performed. The dataset includes a yearly recorded data of wind speed, direction, surface temperature, surface pressure, etc. for a particular location in Chennai.

This paper presents a detailed review on comparative analysis of XGBoost and Gradient Booting for wind power predictions based on time period and identifies possible developments in the future. This paper is mainly divided into three parts 1) Appropriate machine learning method used for analyzing the data given. 2) Tool to be used for predicting the error rate efficiently.3) Comparisons and conclusions with the existing technologies.

2. METHODOLOGIES

2.1. Supervised Learning Algorithm

We have taken supervised learning as in it each sample is a pair consisting of an input object (typically a vector) and an output object also called the supervised signal [6]. In simple language its like questions and solutions are given to a child and asked for a better solution. Working of supervised algorithm can be explain in simple terms like we have input data as '*a*' and output data corresponds tom data '*b*' and this algorithm help to map a function from input to output

$$b = f(a),$$

which is useful when we have new input data '*a*' that can predict new output variable for data '*b*'.

2.2. Technology Used - Apache Spark

It is an open source clustering-computing framework [7]. Our framework is built upon Apache Spark which enables distributed learning using many computing cores on a cluster where the data is continuously accessed and is cached to running memory, thus accelerating the learning of deep models. The requirement of Apache Spark includes distributed storage system. It supports multiple languages like Java, Scala and Python also it supports various set of libraries. Apache spark also helps in classification and predictions using the machine learning libraries incorporated in MLib features. The structure of apache spark is shown in the (figure1). It consists of a spark core at its bottom and the top layer contains Spark SQL, Mlib, Spark streaming and Spark GraphX which are used for data processing [8]. Apache Spark provides in memory computing which aims at reducing the latency and imparts better performance. Spark has Mesos which is its own cluster manager. Therefore, in our project we have worked on Mlib which is termed as the machine learning library for spark. It is easy to use as it is usable in Java, Scala, Python, and R. In terms of performance, it computes the algorithm 100 times faster than MapReduce. Spark promotes iterative computations which yields to faster manipulation of Mlib as it contains high quality algorithms that reduces the iteration which in turn increases the performance.

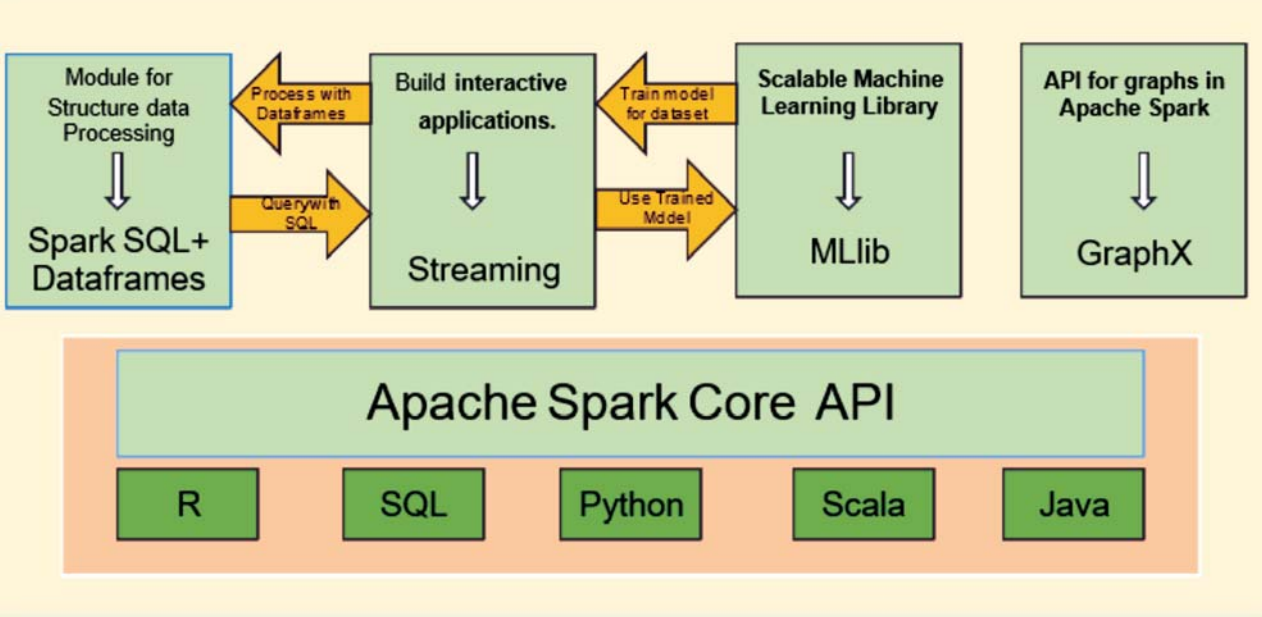


Figure 1: Architecture of Apache Spark

In order to prove the successful working of our proposed framework, we implement a regression model using XGBoost I.e Extreme Gradient Boosting [2] as the algorithm working on a computing cluster and train the model using the linear regression parameters to train the dataset which is collected by National Institute of Wind Energy. The main concept of XGBoost is to divide data under different trees in order to decrease lookup times. This makes performance fast as time taken to train models is decreased. The results obtained gave us a significant improvement of using XGBoost over other conventional machine learning algorithms improving the error rate from a significant percentage. Thus, XGboost has, been seen to be superior due to the application of multithreading. Thus concludes the learning time of machine models is decreased due to paralleled working of cores in Spark as compared to computation in a single machine. The main objective of Spark platform is to tackle the volume, velocity, and volatility aspects of the data provided to it [8]. The objective of spark platform is to tackle the volume of the provided data. As, the volume aspect is handle by dividing the learning task into small divisions so that no single machine is loaded with massive volume as one chunk, thus speeding up the process.

2.3. Programming Language Used- Scala

The programming language used is Scala I.e. ‘scalable language’ as programs written in it are concise and elegant. Nowadays, Scala is used by many companies like Facebook, Pinterest etc. Scala being compiler based and functional language is faster and suits best with Apache spark compare to other programming languages [9]. The main reason we used Scala as our programming language is due to the fact that Apache Spark is implemented on Scala, using Scala allows us to explore the latest features as most of the features are first available in Scala then only can be mapped to other languages.

2.4. Linear Regression Equation

It is commonly used predictive analysis. It is a best way to analyze data. It can be explained as the relationship between a dependent variable an independent variable b, in other words single independent variable help to predict the value of dependent variable, for more than on independent variable we can use multiple linear regression [10].

The linear regression equation is given as follows:

$$P = m + nQ,$$

Where Q is independent variable and P is dependent variable, the slope of the line is n and intercept is *m*.

Linear regression consists of a straight line which passes through maximum number of solutions, this line is known as

Regression line. For example,

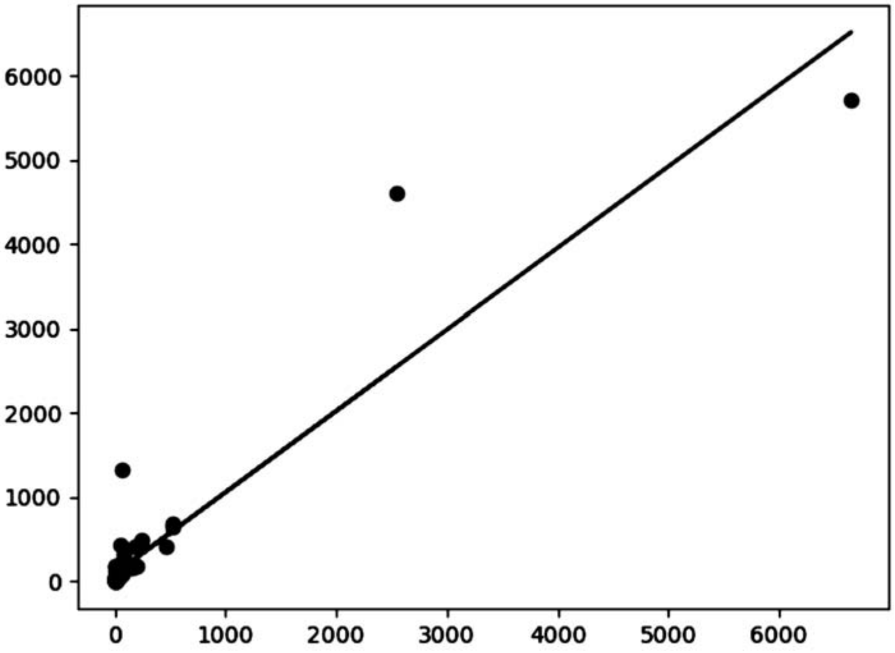


Figure 2: Linear Regression Equation Graph

Linear regression basically includes the line that best fits the maximum number of points. Therefore, the straight blue line passing through maximum number of points is known as the regression line in (Figure2). The above graph is based on the brain and body dataset. The regression line consists of the predicted reaction of brain with respect to each value of body.

In the above graph the straight line consists of prediction and the points represent the actual data. The coagulated points in the graph represent the errors of prediction. The farther the point equates to increment in the error rate.

3. ALGORITHM

3.1. Spark Session

1. Learning and training data for big data analytics is quite a tedious and demanding task. This is because of having billions of parameters which are involved in learning models. The purpose of the framework is accelerating the decision making process of the algorithm by parallelizing the computations of the machine models for a high performance computing cluster.

Therefore, our Spark framework basically consists of two parts:

1. Spark Master
2. One or more than one Spark workers

The Spark Master initializes the Spark Driver that manages the execution of many fragments of machine models in a group of one or more than one workers. At each iteration of the deep learning algorithm, each worker node learns a partial deep model.

Since we know that Spark is an in-memory computation of data. In our project we have created data frames. Spark SQL consists of data abstraction tabular analysis which are known as data frames. Therefor a data frame is an abstraction of structured or unstructured data which basically illustrates data sets with schema. Two data frames are made in which the wind data is stored and manipulated. The data basically consists of each month's hourly manipulated wind factors viz surface temperature, surface pressure, direction, wind speed etc. for a particular location in Chennai. The Spark parameters consists of one Spark Master and 4 worker nodes each consisting of 2GB of execution memory with 4 cores. The Spark Driver memory consist of 1 Gb storage. These configurations instantiate the Spark Session. So the master is switched on which triggers the Ethernet switch which connects the worker nodes for initiating the spark session.Two data frames are made, one for the training set and the other for the test set. In conclusion of predicting the wind speed the algorithm is iterated to some n number of iterations till an ideal is summed up.

3.2 Gradient Boosting

Gradient boosting algorithm [11] was developed for very high predictive capability. Still its adoption was very limited because the algorithm requires one decision tree to be created at a time in order to minimize the errors of all previous trees in the model. So it took a large amount of time to train even those models that were small in size. The concept of gradient boosting involves basically three steps. First, a proper differentiable loss function should be identified that is suitable for the given problem. Second, a weak learner is created to make the predictions. In gradient boosting a decision tree is chosen as a weak learner. Therefore the regression tress are used to get the real output value from the splits and the outputs from these divisions are added together to predict the entire model.This approach enables the improvement of the residuals in the predictions leading to more precise predictions. The trees are created in a greedy manner and often certain constraints are imposed in order to ensure that the weak learners continue to be weak learners and still the trees can be created using a greedy approach. Third, creation of an additive model to add up the predictions of the weak learners so as to reduce the loss function. This process of adding the trees happens one at a time. The output produced in the new tree is then added to the output of the pre-existing sequence of trees in order to improve the final output of the model. This process stops once the proper optimized value for the loss function is reached.

3.3. XGBoost

XGBoost stands for 'Extreme Gradient Boosting', which works on numeric vector and helps in solving tabular data. It is similar to gradient boosting but in terms of performance it is more efficient, approximately 10 times faster compare to old gradient boosting implementations. It supports various roles including regression, classification, user defined objects etc. Its ease of use make it a better platform for implementation, the platforms include windows, linux etc for execution purposes. In our proposed framework we have configured the parameters on the basis of the requirements of the customer.

Adopting this as an apt algorithms was mainly due to following:

1. It promotes parallel processing which increases the efficiency as compared to gradient boosting.
2. It supports user define object which makes it more flexible compare to other algorithms.
3. Since there were no missing data values for our project so there was no problem in managing the data. But XGBoost manages the missing data very efficiently.
4. The parameters basically included the regular liner regression parameters which were programmatically configured according to the requirements of our project [12].
5. The algorithm was executed for as many iterations till an ideal number was found.

4. RESULT

The dataset provided by National Institute of Wind Energy included factors such as surface pressure, surface temperature, date and time, direction, speed etc. By using this dataset various functions were performed viz regression, classification etc. We were able to predict the wind speed by dividing the dataset into two parts i.e. training dataset and testing dataset. This division was only possible since we didn't have any missing values. Different number of iterations were performed in order to find the ideal number of rounds. The result includes the average, maximum, minimum operations on the dataset. The calculations are done by manipulating the result from the algorithm. The metrics evaluation concludes the analysis of efficiency, performance and accuracy for both Gradient Boosting and Xgboost.

Firstly the analysis of XGBoost on a distributed environment i.e. Spark was accomplished which gave the following result:

The graph (figure3) plotted shows the average wind speed (actual) versus the predicted wind speed for number of iterations 100 and 250.

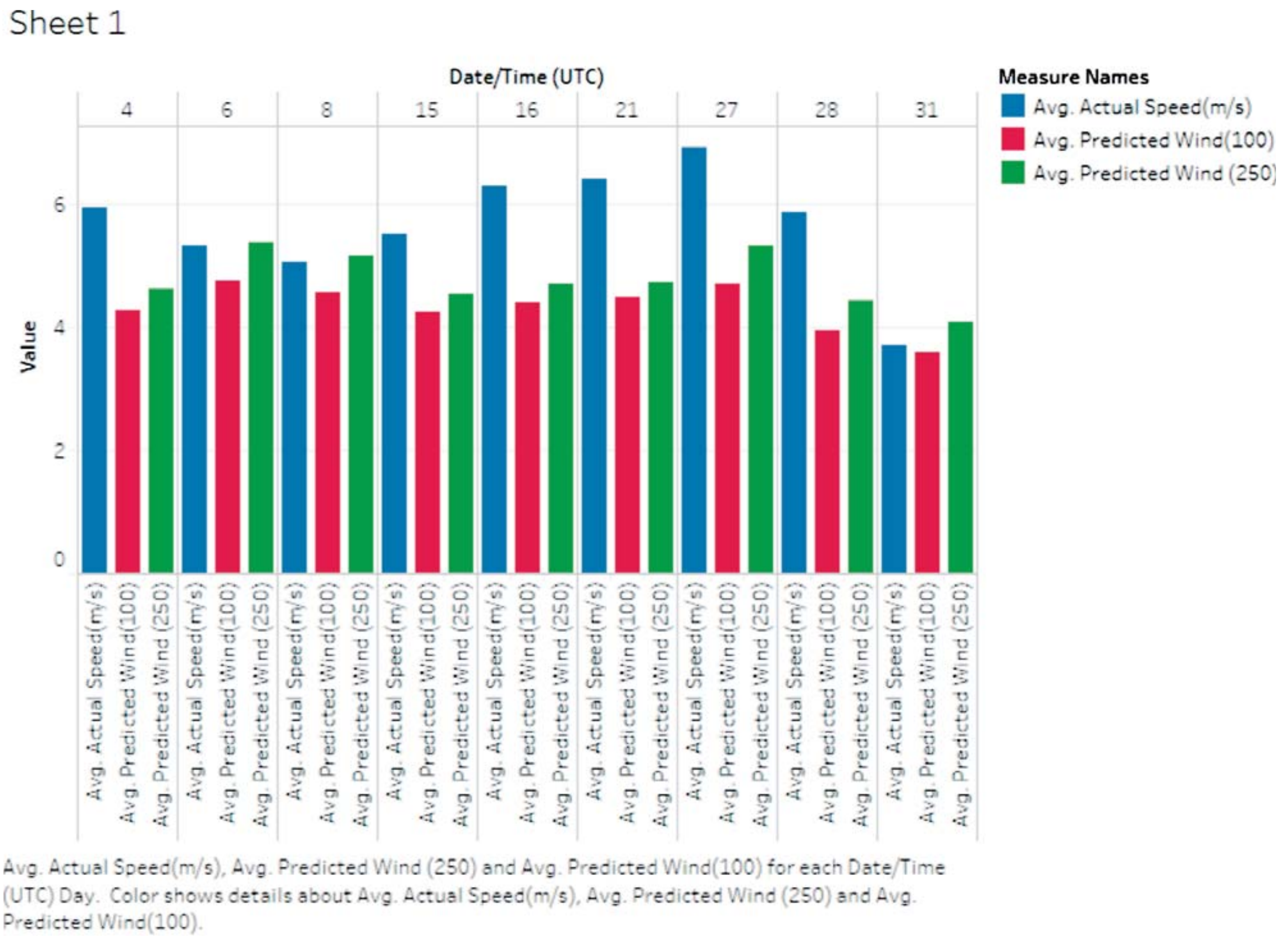
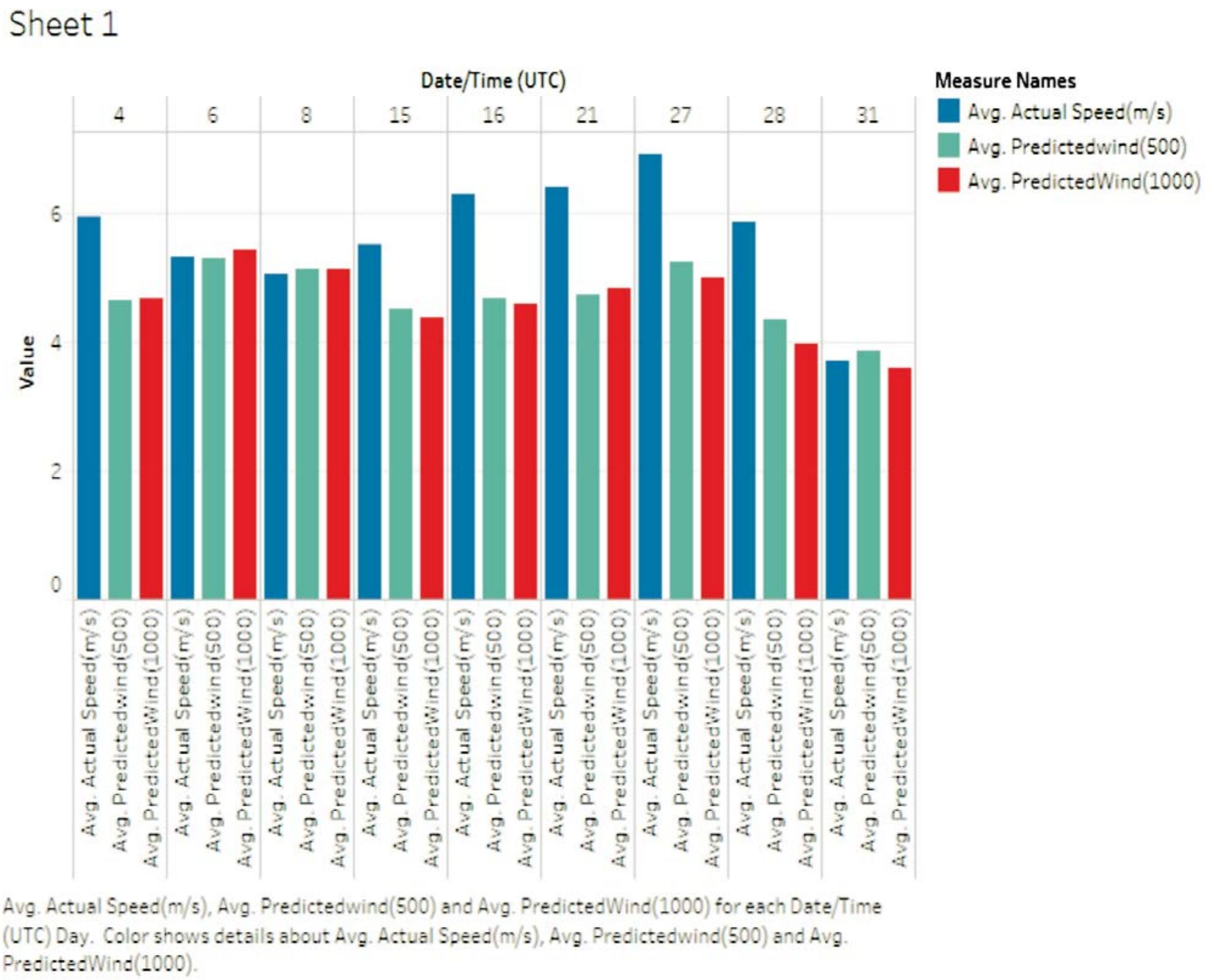


Figure 3: Comparison of Averages of Actual VS Predicted for (100-250) iterations

Thus in the above graph we can clearly see that the predicted speed for the dataset is varying to a greater level which concludes that the number of iterations for the algorithm is not suitable enough to calculate the wind speed based on the factors like surface temperature, surface pressure, direction etc. Thus the program was run with other values of iterations which lead to ideal number of iterations. The (figure: 4) shows the next set of iterations upon which we performed.

The graph shows the average wind speed (actual) versus the predicted wind speed for number of iterations 500 and 1000. The predicted wind speed for a set of 500-1000 iterations.

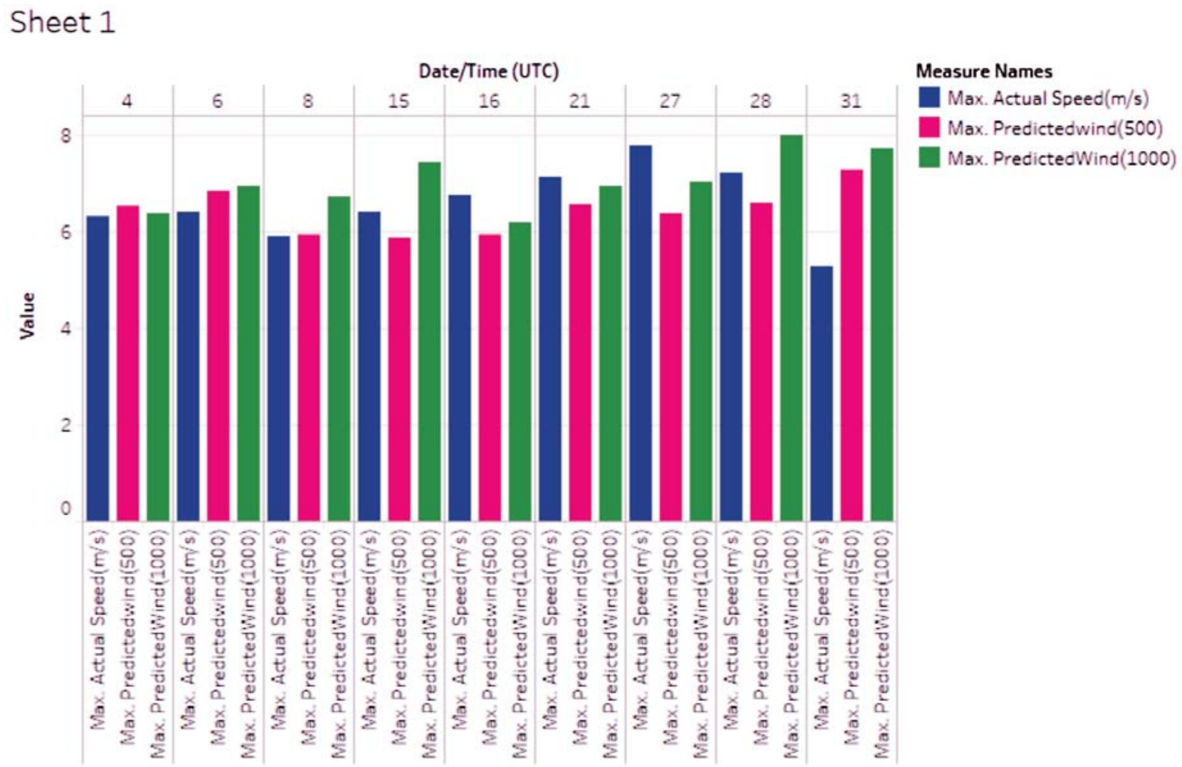


Avg. Actual Speed(m/s), Avg. Predictedwind(500) and Avg. PredictedWind(1000) for each Date/Time (UTC) Day. Color shows details about Avg. Actual Speed(m/s), Avg. Predictedwind(500) and Avg. PredictedWind(1000).

Figure 4: Average Actual Speed Versus Average Predicted Speed for (500-1000) Iterations

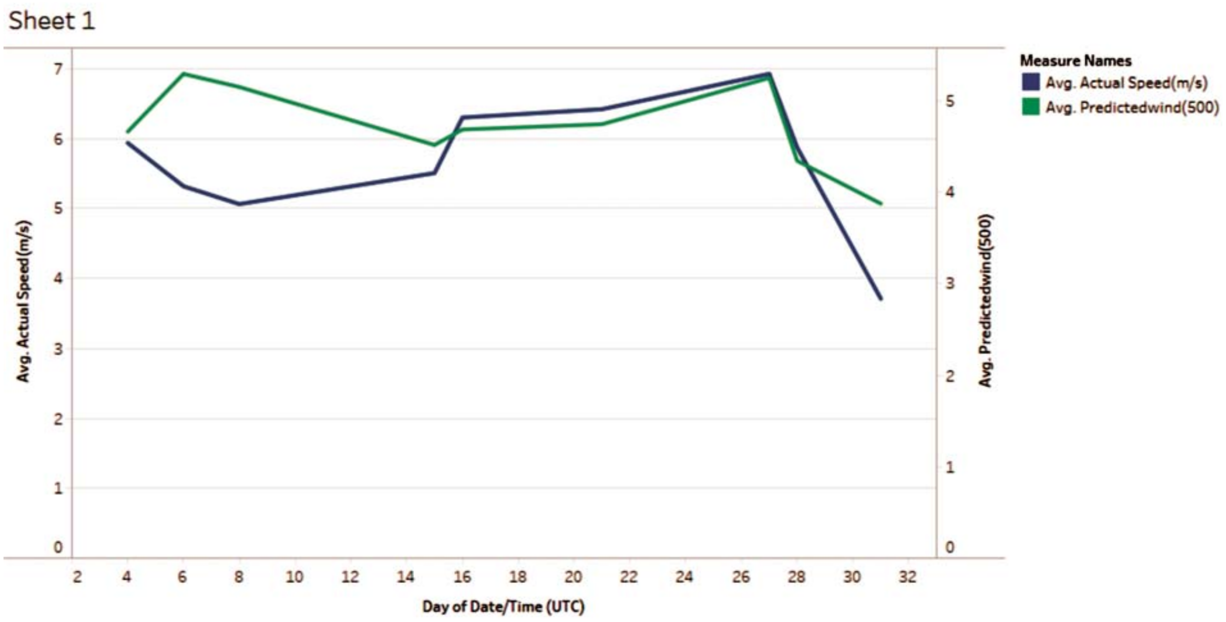
Thus the above graph (figure4) shows the average speed versus the predicted speed which is relatively closer compare to the previous graph plotted for 100 and 250 iterations. The deviation was quite high in the case of 250 iterations. The outcome was inferred to find out the ideal number of rounds for this data set which were concluded in this graph. Thus we concluded the ideal number of iterations for our project when our algorithm was configured for 500 number of rounds.

The below graph (figure5) is plotted for the maximum values of actual speed versus the maximum values of predicted speed. We can infer that 500 iterations can be taken as the ideal for the dataset which is provided to us by the National Institute of Wind Energy. As the results are quite close to the actual value, which shows how accurate the training and the test sets were, in terms of predicting the value.



Max. Actual Speed(m/s), Max. PredictedWind(1000) and Max. Predictedwind(500) for each Date/Time (UTC) Day. Color shows details about Max. Actual Speed(m/s), Max. PredictedWind(1000) and Max. Predictedwind(500).

Figure 5: Maximum Actual Speed VS Maximum Predicted Speed for (500-1000) iterations



The trends of Avg. Actual Speed(m/s) and Avg. Predictedwind(500) for Date/Time (UTC) Day. Color shows details about Avg. Actual Speed(m/s) and Avg. Predictedwind(500).

Figure 6: Average Actual Speed Versus Average Predicted Speed for Ideal Iterations

Therefore in the above graph (figure6) we can observe the ideal number of iterations for wind prediction. The graph is plotted for actual speed versus the predicted speed for 500 iterations. The results are pretty close signifying the apt working of the predictor algorithm. The graph is plotted for the testing data set in order to compare the results between training and testing sets. Thus we can infer that the training and test set gave significantly similar traced graph showing the accuracy of the algorithm in a distributed environment.

Now the comparative analysis for the same dataset was done using Gradient Boosting and Xgboost for calculating the efficiency and performance of both the algorithms in the stand alone mode. Therefore the same data set was configured as per the parameters of liner regression. The following graphs were made and its inference is given as follows:

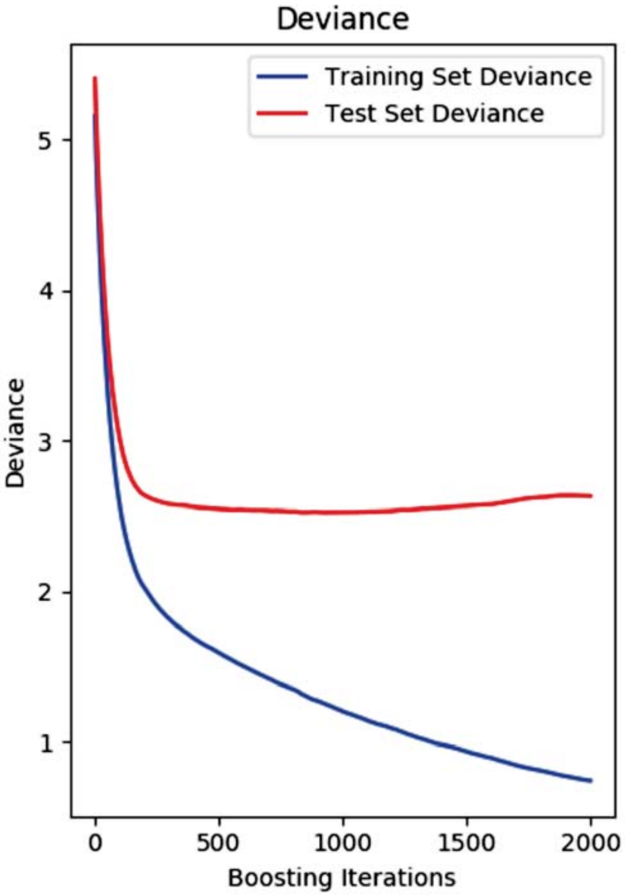


Figure 7: Gradient Boosting Training Set Deviance Versus Test Set Deviance

The above graph plotted between the training and test data sets showing the deviation when the algorithms are executed. This deviance is done for a set of 2000 iterations. Significantly we can observe from the (figure7) that gradient boosting done for a set of 2000 iterations shows high deviation. The situation of over-fitting occurs in this case which infers to less efficiency of gradient boosting for the given dataset. Added that the training curve starts at 5 which is less than XGBoost4j(figure8) in which it starts from 5.5 illustrating the fact that training is faster in the case of XGBoost4j making it better than Gradient Boosting. The decision making process in XGBoost4j is faster than Gradient Boosting which makes the training data set faster to be learned as compared to Gradient Boosting aiming its performance to be high. Therefore the deviation (actual and prediction) in XGBoost4j is less making it more efficient as the gap between the test curve and the train curve is less as compared to Gradient Boosting. Also when we change the number of iterations there is no siginficant change observerd in gradient boosting, but in XGBoost4j the gap between test and train curve is less when we are increasing the iterations. Thus these results inferred the reason of opting for XGBoost4j over Gradient Boosting as the algorithm for wind data analysis. Added that the outcome for more efficient error rate were observed in distributed environment *i.e*. Spark more than the stand alone mode, which made us clear about having more efficient analysis and a better performing algorithm to be taken as XGBoost's execution in Apache Spark.

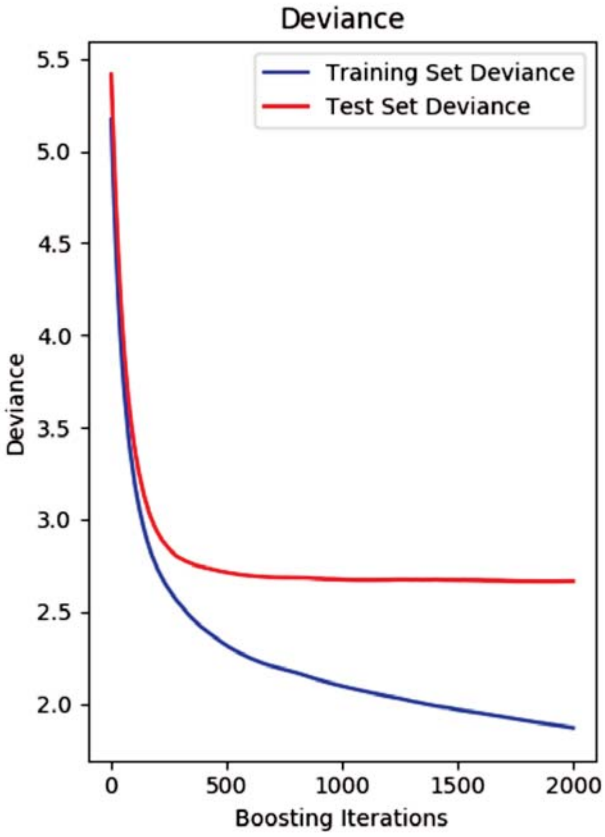


Figure 8: Xgboost4j Training Set Deviance Versus Test Set Deviance

The below graph(figure9) is plotted among the actual wind speed versus the predicted wind speed calculated using XGBoost4j package in a distributed environment which is Apache Spark versus the predicted wind speed calculated using the Gradient Boosting algorithm in a normal programming environment. Therefore we can infer that the green line in the graph depicts the XGBoost4j is seemingly close to the actual wind speed which is indicated by the orange line on most of the points. Thus concluding the efficiency of XGBoost4j as compared to Gradient Boosting in the above graph. In terms of determining the efficiency most of the times yellow line which depicts Gradient Boosting is found under the actual wind speed illustrating less precision of prediction for the given dataset. This clearly depicts the problem of under-fitting. but for the XGBoost4j we can't derive this similar pattern of under-fitting. Therefore Gradient Boosting needs additional tuning for better performance. But XGBoost4j does not need tuning as it gives a better result than Gradient Boosting implying to its performance level, efficiency and accuracy as higher than Gradient Boosting in a normal programming environment. Hence the accuracy of prediction is improvisied in XGBoost4j with respect to increase in the number of iterations.Thus implicating the reason for opting XGBoost4j than Gradient Boosting for the analysis of wind dataset evaluating better analysis of data in a distributed programming environment for fast learning and better execution in terms of results.

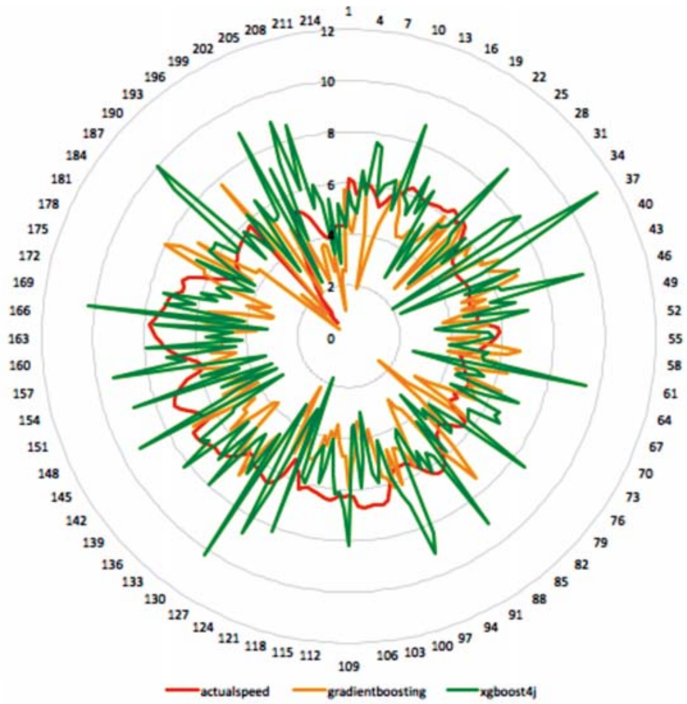


Figure 9: Xgboost4j in Apache Spark Versus Gradient Boosting in Normal Programming Environment

5. CONCLUSION AND FUTURE SCOPE

Thus, in this paper we propose a linear regression model which is performed using an open source clustering framework *i.e*. Apache Spark. Moreover, the experimental results provide us the information regarding the improving efficiency of the learning models programmed in a normal runtime environment versus the distributed computing environment. A detailed comparison of wind speed in both XGBoost and Gradient Boosting was performed based on the datasets which were provided by NIWE. It can be observed that the speed of execution of XGBoost is superior to Gradient Boosting, and also the accuracy of the algorithm is increased due to its working in the distributed computing environment. This is possible only in the consideration of a condition that there are no missing values in the program and the dataset so given had all the values for evaluation. Therefore, the future scope of the project will include robust models, which includes the implementation of Artificial Neural Network Technologies [13] which can predict the results more efficiently (long term prediction) and the processing rate for the dataset in these algorithms is also decreased leading to provide an efficient prediction (in terms of execution time). Also a suggestion of improving the XGBoost for complex datasets which will incorporate the training of real-world datasets where the occurrence of anomalies and noise are inevitable. Therefore, the introduction of ANN for implementation will provide a better dimension for the execution of various operations viz, regression and classification for better prediction of wind power forecasting.

REFERENCES

[1] Panos Louridas and Christof Ebert.'Machine Learning'. IEEE Software, 2016.

[2] XGBoost and XGBoost4j document,https://xgboost.readthedocs.io/en/latest/.

[3] Apache Spark https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm

[4] Elizabeta Lazarevska, Faculty of Electrical Engineering and Information Technologies – Skopje, University Ss. Cyril and Methodius - Skopje Wind speed prediction with Extreme Learning Machine, IEEE 8th International Conference on Intelligent Systems, 2016.

[5] Wen-Yeau Chang Department of Electrical Engineering, St. John's University, New Taipei City, Taiwan. A Literature Review of Wind Forecasting Methods. Journal of Power and Energy Engineering, 2014, 2, 161-168 Published Online April 2014

[6] https://en.wikipedia.org/wiki/Supervised_learning, supervised learning algorithm, Wikipedia.

[7] Book-Mastering Apache Spark, Gain expertise in processing and storing data by using advanced techniques with Apache Spark by Mike Frampton, 2015 Packt Publishing.

[8] Apache Spark, Lighting fast cluster computing, http://spark.apache.org/ .

[9] Spark Scala, Why I choose scala for Apache Spark project, Published on April 24, 2015, Featured in: **Software Engineering**.

[10] LinearRegression, https://en.wikipedia.org/wiki/Linear_regression .

[11] Greedy Function Approximation: A Gradient Boosting Machine, Jerome H. Friedman, IMS 1999 Reitz Lecture, February 24, 1999

[12] How to use XGBoost algorithm in R in easy steps, Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/01/xgboost-algorithm-easy-steps/

[13] K. Sreelakshmi, P. Ramakanthkumar.,"Neural Networks for Short Term Wind Speed Prediction", International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol: 2, No:6, 2008.