INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Wind speed prediction using tree ensemble

Pooja Varshini R
poojavarshinir@gmail.com
Valliammai Engineering College, Kanchipuram, Tamil Nadu

Farzana Begum S
safrinafarzu@gmail.com
Valliammai Engineering College, Kanchipuram, Tamil Nadu

Saranya M
saranyamani311996@gmail.com
Valliammai Engineering College, Kanchipuram, Tamil Nadu

Dr. B. Vanathi
mbvanathi@gmail.com
Valliammai Engineering College, Kanchipuram, Tamil Nadu

ABSTRACT

The wind energy has emerged as one of the safest growing renewable energy to address the crisis witnessed in power generation. Wind speed is an important factor in wind power production and integration. However, the complex nature of wind speed limits the dependability and induces high fluctuation in power generation. The accurate prediction of wind speed energy with minimum accepted errors will increase to harness the energy content in a wind efficiently. In recent years, machine learning algorithms are used to analyze and predict data to make better decisions. Ensemble model is one of the supervised machine learning approaches to predict numerical data. In this project, the speed of wind is predicted through XGBoost 4j package in a distributed programming environment of Apache Spark. XGBoost is a short form of Extreme Gradient Boosting tool of supervised machine learning, where the training data set xi is used to predict a target variable yi. The results are compared with different iterations in order to minimize the uncertainties and to evaluate the efficiency and accuracy. The main findings were that Ensemble model was the most accurate method.

Keywords: *Tree Ensemble, XGBoost, RMSE, Apache Spark.*

1. INTRODUCTION

Wind energy is becoming a viable solution to global warming and fossil fuel problem. But the wind power industries face numerous technical issues one such issue is the prediction of wind speed. Errors in wind speed forecasts lead to errors in forecasting for both demands of electricity and economic benefits of power supply from wind farms. Therefore, small improvements in wind-speed forecasts constitute much larger improvements in wind power generation forecasts. There are two model classes for prediction tasks such as physical numerical weather simulation and machine learning algorithms. In the recent years, the machine learning algorithms are used widely. Since these models derive functional dependencies directly from the observations. They have also known as data-driven models. The project uses the ensemble model which is an accurate method used in machine learning wind prediction.

2. LITERATURE STUDY

(Pan Zhao et al, 2010), in this paper, the wind speed forecasting was carried out using the algorithm of support vector regression (SVR) techniques. The outcome is listed by comparing its performance with a back propagation neural network model through simulation. Both algorithms are applicable for prediction the wind speed time series in future; the prediction effect of support vector regression outperforms the back propagation neural network model as indicated by the mean square errors and mean absolute errors. Three different stages of the wind speed curve are analyzed, the results show that the proposed algorithm fit the original wind speed curve well at the whole process, but the back propagation neural network failed during the rise stage when the ascent rate suddenly become flatness of the original wind speed curve.

(Xinxin Bai et al, 2012) studied that the inherent randomicity and intermittency of wind resource brings challenges to operators of power systems and wind farms. Therefore, preliminary forecasting of the wind power is necessary. The study proposed a statistical method based on linear regression model and will be a time-consuming process.

(Wawan et al, 2014) The study analyses research works carried out by various authors in wind speed prediction using different machine learning algorithms such as Fuzzy inference systems, Artificial neural network, Particle swarm optimization, Linear Method and Support vector machines at different prediction periods. The choice of model is based on the availability of weather parameters.

(Senthil Kumar et al, 2016) The study present Relief feature selection method for wind speed forecast with important features such as wind direction, temperature, and humidity. Further, the bagged neural network is implemented to forecast wind speed. However, the neural network model has limitations in pattern matching and computational time is more.

(Xiaoyan Shao et al, 2016) study records that the renewable energy forecasting becomes increasingly important as the contribution of solar or wind power production to the electrical power grid constantly increases. Significant improvement in forecasting accuracy has been demonstrated by developing more sophisticated solar irradiance forecasting models using statistics and/or numerical weather predictions. The development of a machine learning based multi-model blending approach for statistically combing multiple meteorological models is used to improve the accuracy of solar power forecasting. The system leverages upon multiple existing physical models for prediction including numerous atmospheric and cloud prediction models based on satellite imagery as well as numerical weather prediction (NWP) products.

(Peter Fischer et. al, 2015) Gradient boosted regression (GBT) is a non-parametric regression technique used in various applications, suitable to make predictions without having an in-depth knowledge about functional dependencies between the predictors and response variables. The paper uses digital elevation model. Boosting is an extension of the initial regression tree algorithm. Instead of creating a single tree, boosting creates iteratively ensemble trees which aim to minimize the residuals of the initial model. The final model consists of gradient descent strategy, where different loss functions are possible. GBT is capable to model the complex behavior of wind streams and results are reliable within a given accuracy.

(Elizabeta Lazarevska, 2016) This paper presents an alternative approach to forecasting the wind speed based on extreme learning machine. The wind speed is modeled by means of available meteorological data such as total solar radiation, ambient temperature, humidity, barometer pressure, etc. The applied modeling algorithm belongs to the class of extreme learning machine methods, which are gaining an increasing interest in the scientific and research community. The conducted research has clearly shown that the Extreme Learning Machine does not depend notably on the selection of the activation function for the hidden neurons. However, the random selection of the hidden layer parameters significantly influences its approximation capacity and generalization property. The presented Extreme Learning Machine model offers an alternative approach to modeling the wind speed on a short-term basis and the simulation results show clearly its advantages and very good performance indices. The extreme learning machine method has the attributes of simplicity, good approximation performance, and fast computation.

(Wenbin Chen et al, 2017) study focus on efficient classification model for the complex radar signal. This paper records that performance of XGBoost is better compared with classic models such as support vector machine, relevant vector machine, and deep belief network.

(Mohammad Abu Alsheikh, et al, 2016) studied that the proliferation of mobile devices, such as smartphones and Internet of Things gadgets, has resulted in the recent mobile big data era. Collecting mobile big data is unprofitable unless suitable analytics and learning methods are utilized to extract meaningful information and hidden patterns from data. Specifically, distributed deep learning is executed as an iterative Map Reduce computing on many Spark workers. Each Spark worker learns a partial deep model on a partition of the overall mobile, and a master deep model is then built by averaging the parameters of all partial models. This Spark-based framework speeds up the learning of deep models consisting of many hidden layers and millions of parameters. It uses a context-aware activity recognition application with a real-world dataset containing millions of samples to validate our framework and assess its speedup effectiveness.

3. PROCESS FLOW

The steps involved in Short-term wind speed prediction is shown in Figure 1. Wind data set is read, pre-processed if any missing values. The XGBoost algorithm has been executed using Scala programming. Further data set is split into the training and testing set and used for validation and testing.

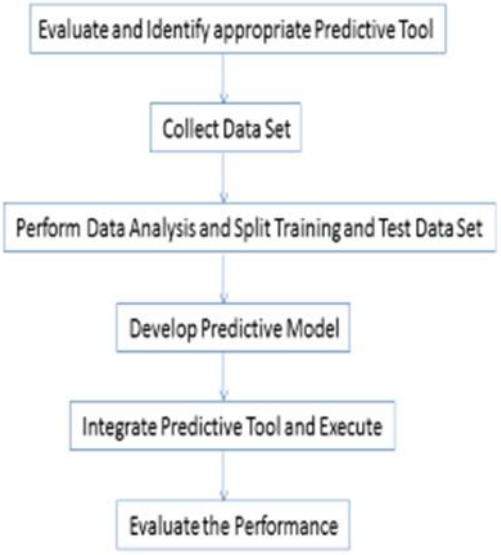


Figure 1. The process flow of Short-term wind speed prediction

4. METHODOLOGY USED

Tree ensemble algorithms empower predictive models with high accuracy, stability and ease of interpretation. They are adaptable to solving any kind of problem. The decision trees are typically drawn upside down such that leaves are the bottom and roots are the tops. The tree formation follows both regression and classification types.

Like every other model, the tree ensemble also suffers from the bias means and variance of the split. Some of the commonly used ensemble methods are bagging and boosting. Bagging is a technique used to reduce the variance of predictions by combining the results of multiple classifiers modeled on different sub-samples of the same dataset.

(Leslie Valiant) Boosting refers to a family of an algorithm which converts "weak learner" to be a "Strong learner" by using some kind of modifications. From a statistics point of view, this process was similar to creating a "good hypothesis" from a relatively "poor hypothesis". A poor learner or a "weak hypothesis" is a model whose performance is slightly better than random chance. (Jerome 1999) Gradient boosting algorithm was developed for such high productive and capability. It builds an ensemble of trees one-by-one, then the predictions of individual trees are summed as given in equation 1 (with an assumption as ensemble has 3 trees).

$$D(\mathbf{x}) = d_{\text{tree }1}(\mathbf{x}) + d_{\text{tree }2}(\mathbf{x}) + d_{\text{tree }3}(\mathbf{x}) \quad (1)$$

The next tree (tree 4) in the ensemble should complement well the existing trees and minimize the training error of the ensemble as given in equation 2.

$$D(\mathbf{x}) + d_{\text{tree }4}(\mathbf{x}) = f(\mathbf{x}). \quad (2)$$

To get a closer destination, the tree is trained to reconstruct the difference between the target function and the current prediction of an ensemble, which is called the residual /errors as given in equation 3.

$$R(\mathbf{x}) = f(\mathbf{x}) - D(\mathbf{x}). \quad (3)$$

The losses currently supported by GBTs in MLlib are Log Loss (classification), Squared Error (Regression) and Absolute Error (Regression and outlier model. The steps to be followed in the Loss calculation

(i) Consider a whole set as the root node, Calculate Mean and Sum of Squared Error (SSE)
(SSE $= \sum_{i=1}^{n}(x_i - Mean)^2$
(ii) Split the root node into leaf nodes, Calculate Mean and SSE for the leaf nodes
(iii) Find SSE Drop = SSE of Root Node – (Sum of SSE of Leaf nodes)
(iv) Select the Tree which has a high value of SSE Drop

Consider the following set of Wind Speed (m/s)is observed in various Wind power densities: 4.4, 5.1, 5.6, 6.0, 6.4, 6,8, 7.0, 9.4. The Tree can be formed in many ways. Consider the two trees represented in Figure 2 and 3.

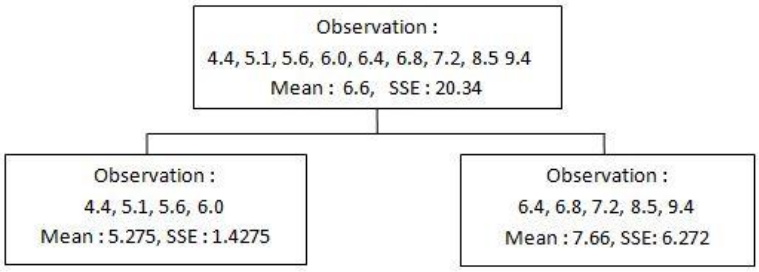


Figure 2. Tree 1

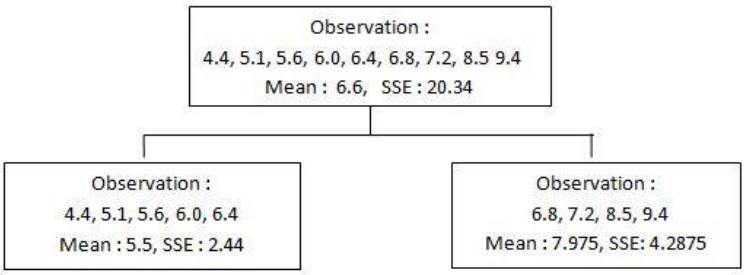


Figure 3. Tree 2

- SSE Drop in Tree 1 : 20.34 – 1.4275 – 6.272 = 18.915
- SSE Drop in Tree 2 : 20.34 – 2.44 – 4.2875 = 13.6125
- Since SSE Drop is high in Tree 1, it is chosen as a fit model in the first level iteration.

Further to solve over-fitting issues, the residual tree (error) will be formed and analyzed for prediction.

Developed by (Tianqui chen et al, 2016) XGBoost is an advanced implementation of Gradient boost algorithm. Some of the high performance capabilities of XGBoost are listed as

- known as 'regularized boosting' technique
- implements parallel processing and support faster computation
- allows users to define custom optimization objectives and evaluation criteria
- has an in-built routine to handle missing values
- Tree splitting up to the max-depth and remove splits beyond which there is no positive gain
- Allows the user to run a built-in cross-validation

The most important factor behind the success of the XGBoost is its scalability in all scenarios. The scalability of XGBoost is due to several important systems and algorithmic optimizations. These innovations include: a new tree learning algorithm for dealing with sparse data; theoretically reasonable weighted quintile sketch program can handle instance weights in approximate tree learning. Parallel and distributed computing make learning faster, enabling faster model exploration.

A precise short-term wind speed prediction is important for a safe and sustainable balancing of the electricity grid. This work focus on short-range forecasting model using XGBoost regression in a distributed environment.

System Design:

The Wind Speed Prediction using machine learning algorithm depicts simple block diagram as shown in Figure 4. The Wind Speed Historical Data set is obtained, pre-processing is limited, since the data set is in the form of numerical information and there is no missing value. The data set is also divided into training and testing set.

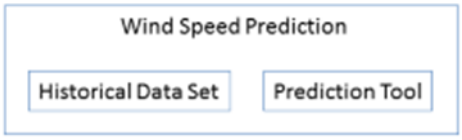


Figure 4. Wind Speed Prediction Block Diagram

The prediction tool XGBoost4J performs the data processing and serve as complete data analytic pipeline on top of deployment frameworks such as Apache Spark. The combination of technologies used in the product environment is as follows:

- Apache Spark for the processing engine,
- Scala for the Programming Language
- XGBoost for the Supervisied Learning Algorithm

To create a solution that can make accurate predictions, the predicted model to be designed as shown in Figure 5. The training set is a sample of data used to fit the model. The training data set is the actual data set, used to train the model. The validation data set is used to provide an unbiased evaluation of a model fit on the training data set and fine tune the model. The test data set is the sample of data used to provide an unbiased evaluation of a final model fit on the training data set. The validation set is released initially along with training set and the actual test set is released only when the model fit is derived. The split mainly depends on two things such as the total number of sample in data and actual model of training. Initially, the data set is split into training and test data set. The random choice of X % of train data set makes remaining (100- x) % as a validation test. However, when the model has more hyper parameters, the validation test is large, otherwise, it is small or nil. This is known as cross-validation and avoids over fitting during iterations.

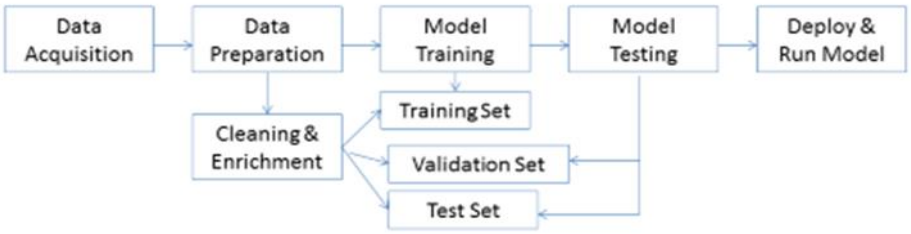


Figure 5. Iterative Flow of Data Processing in Predictive Model

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. Scala is an acronym for "Scalable Language". It is a general-purpose programming language designed for the programmers who want to write programs in a concise, elegant, and type-safe way. Scala can be regarded as a model for distributed computation of large amounts of data sets which are Java Virtual Machine (JVM) based and supports high scalability over other languages such as Java, Python, and R.

XGBoost is used for supervised learning problems, where one use the training data (with multiple features) x_i to predict a target variable y_i. Like decision trees, GBTs handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearity and feature interactions. MLlib supports GBTs for binary classification and for regression, using both continuous and categorical features. MLlib implements GBTs using the existing decision tree implementation.

The algorithm of XGBoost:

Gradient boosting iteratively trains a sequence of decision trees. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labeled to put more emphasis on training instances with poor predictions. Thus, in the next iteration, the decision tree will help correct for previous mistakes. The specific mechanism for re-labeling instances is defined by a loss function (discussed below). With each iteration, GBTs further reduce this loss function on the training data.

GBTS train one tree at a time, which increases the likelihood of over-fitting and scalability. There are few parameters to be discussed before defining the algorithm:

- numIterations: This sets the number of trees in the ensemble. Each iteration produces one tree. Increasing this number makes the model more expressive, improving training data accuracy. However, test-time accuracy may suffer if this is too large.
- learningRate: This parameter should not need to be tuned. If the algorithm behavior seems unstable, decreasing this value may improve stability.
- loss: Different losses can give significantly different results, depending on the dataset.
- algo: The algorithm or task (classification vs. regression) is set using the tree [Strategy] parameter

However, it requires careful tuning, slow to train but fast to predict. Also, it cannot be extrapolated. The steps involved in the algorithm is listed as follows:

1. Input Dataset
2. Pre-process, do dataset cleaning and submit for training
    - The classifier looks for feature extraction and suggests best possible split
    - The optimal splitting is found through a greedy approach

- Upon splitting, Percentile value is assigned for each split using an approximate algorithm

3. Based on the percentile value, scores are assigned in each split
4. Based on the scores, ranks are allotted to these splits.
5. During prediction, the scores are added to find the best fit and strengthen the weak learner.
6. The issues in real-time data such as missing values and empty space are addressed with the help of scores calculated in previous events with a separate path, while the normal decision tree faces misleading with such events.
7. Prediction accuracy is ensured through loss calculation and minimization

5. SYSTEM ARCHITECTURE

XGBoost is a Machine learning library designed and optimized for tree boosting. The distributed XGBoost system runs magnitude faster than existing alternative of distributed machine learning algorithms and uses a few resources. Despite the benefits, there is a gap between the implementation of data processing frameworks and machine learning libraries, which prohibits the smooth connection. The common workflow utilizes the Spark/Flink to preprocess or cleand data and creates additional overhead. To resolve the situation, a new-brewed XGBoost4J, for JVM platform is introduced as shown in Figure 6. It aims to provide the clean Java/Scala APIs and the integration with the most popular data processing system.

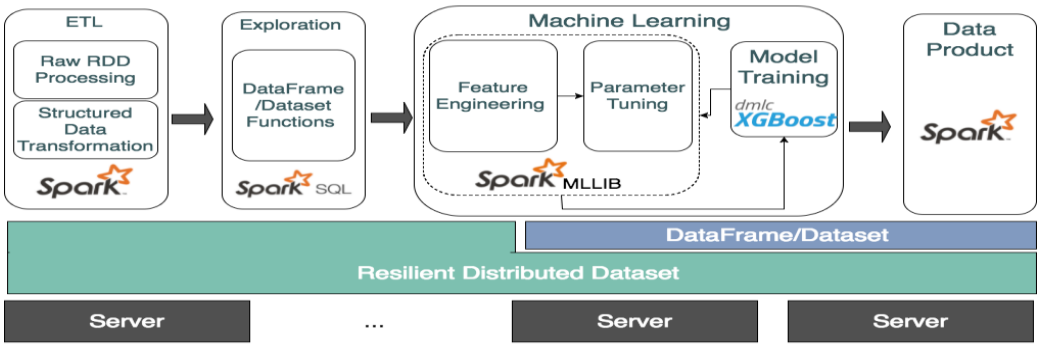


Figure 6. A new pipeline Architecture with XGBoost4J and Spark

Implementation

Upon completion successful completion of Apache Spark in a distributed environment, the wind speed prediction is executed using the proposed machine learning XGBoost algorithm through open source clustering framework. The study assumes that there are no missing values in the dataset for simplicity. The detailed implementation and experimental results are presented herewith.

The dataset fed into the programming is obtained through SRM Automated Weather Station (AWS). The sample dataset format is given in Figure 7. The dataset has other values such as wind direction, air temperature, etc., which helps in calculating wind power energy. They can also be used in an algorithm with multivariate analysis. XGBoost supports CSV and libsvm formats for training and inference.

Date	Time	Wind Speed ((kmph))	Wind Direction ((Degrees))	Air Temperature (Degree C)	Solar Radiation ((Wpm2))	Barometric Pressure ((KPa))	Humidity ((%))	Min Air Temperature (Degree C)	Max Wind Speed ((kmph))	Max Air Temperature (Degree C)
2/3/2018	14:00:14	5.6649	292.768	34.4128	818.768	999.873	28.2507	34.21	15.2082	34.79
2/3/2018	13:45:11	5.7491	255.122	34.3635	842.754	999.981	28.2602	34.09	15.2082	34.79
2/3/2018	13:30:10	6.712	262.263	34.3063	868.07	1000.34	26.837	34.06	10.863	34.51
2/3/2018	13:15:10	7.0073	283.3	34.5723	874.491	1000.89	27.8665	34.27	13.7598	34.71
2/3/2018	13:00:14	5.4838	287.868	34.3957	838.375	1001.09	30.0169	34.21	10.863	34.58
2/3/2018	12:45:14	5.4275	319.203	34.3292	832.518	1001.29	32.0937	34.16	13.7598	34.44
2/3/2018	12:30:10	5.4254	317.274	34.19	815.568	1001.44	37.1682	34.03	13.0356	34.33
2/3/2018	12:15:13	4.3211	291.227	33.8512	810.079	1001.56	39.3478	33.56	8.6904	34.21
2/3/2018	12:00:14	3.9147	282.906	33.5182	794.449	1001.84	44.5336	33.38	9.4146	33.56
2/3/2018	11:45:10	2.9292	289.68	33.0607	756.807	1002.03	47.4542	32.64	6.5178	33.38

Figure 7. Sample Dataset obtained from SRMAWS

XGBoost performs remarkably well in machine learning operations because it robustly handles a variety of data types, relationships and distributions, and a large number of parameters that can be tweaked and tuned for improved fits. The tree boosting parameters are set to provide optimum value. The list of parameters is eta, subsample colsample_bytree, colsample_bylevel(to prevent overfitting), max_depth of a tree and min_child_weight. The eval_metric sets as Root Mean Square Error (RMSE) for regression. The RMSE represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are called residuals/loss when the calculations are performed over the data sample that was used for estimation and are called prediction errors when computed out-of-sample. The objective parameter specifies the learning task and the corresponding learning objective and set as reg: linear. The resource allocation can be plan ahead when the load requirement is known ahead of time and the process completion can be made within a given time frame using numworkers and numRound. There are no hard and fast rules which can guarantee best results in parameter tuning. One must understand, what the hyperparameters control and theoretical understanding of XGBoost algorithm. The most important parameters include depth of the trees, shrinkage (eta) and subsample. The setting of parameters in XGBoost algorithm is given in Figure 8.

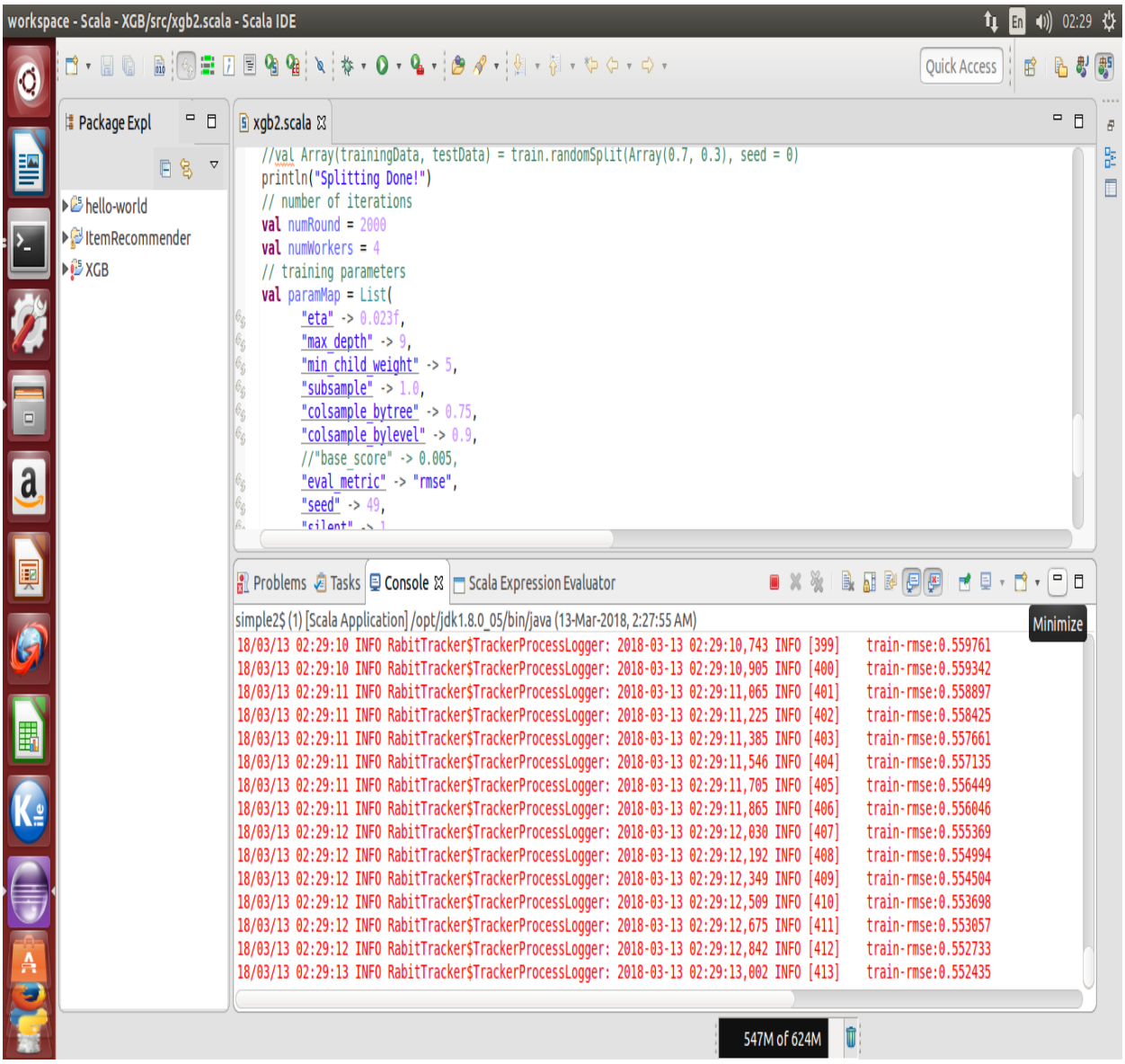
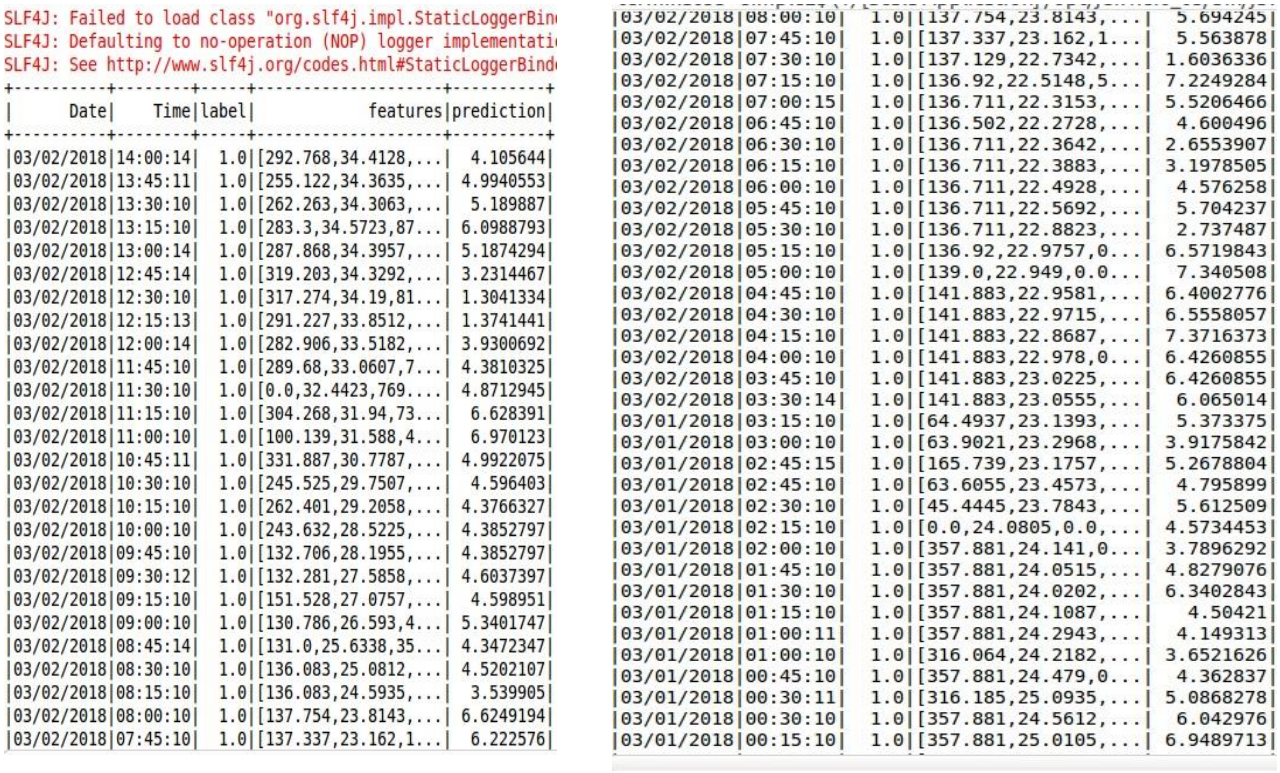


Figure 8. Parameters setting along with RMSE calculation at Run time

The different terminal ids used in distributed environment for task scheduler is tracked and consolidated at the end when all the nodes complete their jobs. The warnings are also listed in case of non-issuance of allocated resources by nodes. This helps in consolidating available free resources for further processing. Figure 9 depicts the feature extraction along with prediction at various time intervals. The metrics evaluation concludes the performance efficiency of XGBoost.



```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBin
SLF4J: Defaulting to no-operation (NOP) logger implementati
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBind
+----------+--------+-----+------------------+----------+
|      Date|    Time|label|          features|prediction|
+----------+--------+-----+------------------+----------+
|03/02/2018|14:00:14|  1.0|[292.768,34.4128,...|  4.105644|
|03/02/2018|13:45:11|  1.0|[255.122,34.3635,...| 4.9940553|
|03/02/2018|13:30:10|  1.0|[262.263,34.3063,...|  5.189887|
|03/02/2018|13:15:10|  1.0|[283.3,34.5723,87...| 6.0988793|
|03/02/2018|13:00:14|  1.0|[287.868,34.3957,...| 5.1874294|
|03/02/2018|12:45:14|  1.0|[319.203,34.3292,...| 3.2314467|
|03/02/2018|12:30:10|  1.0|[317.274,34.19,81...| 1.3041334|
|03/02/2018|12:15:13|  1.0|[291.227,33.8512,...| 1.3741441|
|03/02/2018|12:00:14|  1.0|[282.906,33.5182,...| 3.9300692|
|03/02/2018|11:45:10|  1.0|[289.68,33.0607,7...| 4.3810325|
|03/02/2018|11:30:10|  1.0|[0.0,32.4423,769....| 4.8712945|
|03/02/2018|11:15:10|  1.0|[304.268,31.94,73...|  6.628391|
|03/02/2018|11:00:10|  1.0|[100.139,31.588,4...|  6.970123|
|03/02/2018|10:45:11|  1.0|[331.887,30.7787,...| 4.9922075|
|03/02/2018|10:30:10|  1.0|[245.525,29.7507,...|  4.596403|
|03/02/2018|10:15:10|  1.0|[262.401,29.2058,...| 4.3766327|
|03/02/2018|10:00:10|  1.0|[243.632,28.5225,...| 4.3852797|
|03/02/2018|09:45:10|  1.0|[132.706,28.1955,...| 4.3852797|
|03/02/2018|09:30:12|  1.0|[132.281,27.5858,...| 4.6037397|
|03/02/2018|09:15:10|  1.0|[151.528,27.0757,...|  4.598951|
|03/02/2018|09:00:10|  1.0|[130.786,26.593,4...| 5.3401747|
|03/02/2018|08:45:14|  1.0|[131.0,25.6338,35...| 4.3472347|
|03/02/2018|08:30:10|  1.0|[136.083,25.0812,...| 4.5202107|
|03/02/2018|08:15:10|  1.0|[136.083,24.5935,...|  3.539905|
|03/02/2018|08:00:10|  1.0|[137.754,23.8143,...| 6.6249194|
|03/02/2018|07:45:10|  1.0|[137.337,23.162,1...|  6.222576|
|03/02/2018|08:00:10|  1.0|[137.754,23.8143,...|  5.694245|
|03/02/2018|07:45:10|  1.0|[137.337,23.162,1...|  5.563878|
|03/02/2018|07:30:10|  1.0|[137.129,22.7342,...| 1.6036336|
|03/02/2018|07:15:10|  1.0|[136.92,22.5148,5...| 7.2249284|
|03/02/2018|07:00:15|  1.0|[136.711,22.3153,...| 5.5206466|
|03/02/2018|06:45:10|  1.0|[136.502,22.2728,...|  4.600496|
|03/02/2018|06:30:10|  1.0|[136.711,22.3642,...| 2.6553907|
|03/02/2018|06:15:10|  1.0|[136.711,22.3883,...| 3.1978505|
|03/02/2018|06:00:10|  1.0|[136.711,22.4928,...|  4.576258|
|03/02/2018|05:45:10|  1.0|[136.711,22.5692,...|  5.704237|
|03/02/2018|05:30:10|  1.0|[136.711,22.8823,...|  2.737487|
|03/02/2018|05:15:10|  1.0|[136.92,22.9757,0...| 6.5719843|
|03/02/2018|05:00:10|  1.0|[139.0,22.949,0.0...|  7.340508|
|03/02/2018|04:45:10|  1.0|[141.883,22.9581,...| 6.4002776|
|03/02/2018|04:30:10|  1.0|[141.883,22.9715,...| 6.5558057|
|03/02/2018|04:15:10|  1.0|[141.883,22.8687,...| 7.3716373|
|03/02/2018|04:00:10|  1.0|[141.883,22.978,0...| 6.4260855|
|03/02/2018|03:45:10|  1.0|[141.883,23.0225,...| 6.4260855|
|03/02/2018|03:30:14|  1.0|[141.883,23.0555,...|  6.065014|
|03/01/2018|03:15:10|  1.0|[64.4937,23.1393,...|  5.373375|
|03/01/2018|03:00:10|  1.0|[63.9021,23.2968,...| 3.9175842|
|03/01/2018|02:45:15|  1.0|[165.739,23.1757,...| 5.2678804|
|03/01/2018|02:45:10|  1.0|[63.6055,23.4573,...|  4.795899|
|03/01/2018|02:30:10|  1.0|[45.4445,23.7843,...|  5.612509|
|03/01/2018|02:15:10|  1.0|[0.0,24.0805,0.0,...| 4.5734453|
|03/01/2018|02:00:10|  1.0|[357.881,24.141,0...| 3.7896292|
|03/01/2018|01:45:10|  1.0|[357.881,24.0515,...| 4.8279076|
|03/01/2018|01:30:10|  1.0|[357.881,24.0202,...| 6.3402843|
|03/01/2018|01:15:10|  1.0|[357.881,24.1087,...|   4.50421|
|03/01/2018|01:00:11|  1.0|[357.881,24.2943,...|  4.149313|
|03/01/2018|01:00:10|  1.0|[316.064,24.2182,...| 3.6521626|
|03/01/2018|00:45:10|  1.0|[357.881,24.479,0...|  4.362837|
|03/01/2018|00:30:11|  1.0|[316.185,25.0935,...| 5.0868278|
|03/01/2018|00:30:10|  1.0|[357.881,24.5612,...|  6.042976|
|03/01/2018|00:15:10|  1.0|[357.881,25.0105,...| 6.9489713|
```

Figure 9. Feature Extraction along with Prediction

6. CONCLUSION

The literature available for wind speed prediction modeling depicts that most of the models are used in wind power generation demand forecasting. There are several models ranging from linear regression, support vector model, artificial neural network and tree ensemble model etc., the efficiency of new tree ensemble model followed in XGBoost algorithm is to optimize the value of the objective function. The benefits of XGBoost constructs the boosted trees to intelligently obtain the feature scores, thereby indicating the importance of each feature to train the model.

The present study has taken the input from single machine, univariate model in automated weather station of SRM. However, the power output of a single system is not high enough to cause relevant effects on the electric grid, so the output of entire wind park to be studied and analyzed. The need of multi-machine, the multivariate model suggests including robust techniques in data pre-processing to deal missing values and inconsistency, data conversion and normalization, selection of proper input parameter, a combination of supervised machine learning algorithms with appropriate loss calculation and implementation. Our study focused on short-term wind prediction with XGBoost and with the objective function as regression: linear. The long-term prediction insists on make use of objective function as an artificial neural network for improving the results in wind forecasting. Thus the appropriateness of machine learning model in wind speed prediction is purely based on time series of wind prediction, variables used in the dataset, type of prediction model.

7. REFERENCES

[1] S.M. Lawan, W.A.W.Z Abidin, W.Y.Chai, A. Baharun and T.Masri (2014), Different Models of Wind Speed Prediction, A comprehensive Review, International Journal of Scientific & Engineering Research, Volume 5, Issue 1, January 2014

[2] K. Sreelakshmi, P.Ramkanth Kumar, (2008), Performance evaluation of short-term wind speed prediction techniques, International Journal of Computer Science and Network Security, Vol. 8, No. 8

[3] Peter Fischer, Christophe Etienne, Jiaojiao Tian, Thomas Krua, (2015), Prediction of wind speeds based on digital elevation models using boosted regression trees, International Archives of the Photogrammetry, Remote sensing, and special information sciences, Vol XL-1/W5

[4] Senthil Kumar P, Daphne Lopez (2016), Forecasting Wind speed using feature selection and neural networks, International Journal of Renewable energy Research, vol 6, No. 3

[5] Jerome H. Friedman, IMS 1999 Greedy Function Approximation: A Gradient Boosting MachineReitz Lecture,
[6] Tianqi Chen, Carlos Guestrin, (2016) XGBoost : A scalable tree boosting system, March 9, 2016, arXiv: 1603.02754
[7] Leslie Valiant, Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World, ISBN-13: 978-0465060726
[8] Mohammad Abu Alsheik, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan and Zhu Han. Mobile big data analytics using deep learning and apache spark. Aieee network (Volume: 30, Issue: 3), 2016.
[9] Xinxin Bai, Ting Dut, Haifeng Wang, Xiaoguang Rui, Wenjun Yin, Chen Wang, Weijun He. Efficiency improvement of short-term forecast for wind power. Service Operations and Logistics, and Informatics (SOLI), IEEE, 2012.
[10] Xiaoyan Shao,Siyuan Lu,Hendrik F. Hamann. SOLAR RADIATION FORECAST WITH MACHINE LEARNING Active-Matrix Flatpanel Displays and Devices (AMFPD), The 23rd International orkshop on Active-Matrix Flatpanel Displays and Devices, IEEE, 2016.
[11] Pan Zhao,Junrong Xia,Yiping Dai. Wind speed prediction using support vector regression. Industrial Electronics and Applications (ICIEA), IEEE, 2010.
[12] Wenbin Chen, Kun Fu, Jiawei Zuo, Xinwei Zheng, Tinglei Huang, Wenjuan Ren, 2017, Radar emitter classification for large data set based on weignthed xgboost, IET Radar, Sonar & Navigation
[13] Elizabeta Lazrevska 2016, Wind speed prediction with extreme machine learning, intelligent systems, IEEE 8th International conference

