Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Ensemble learning for predicting mortality rates affected by air quality

To cite this article: Kartika C Dewi *et al* 2019 *J. Phys.: Conf. Ser.* **1192** 012021

View the article online for updates and enhancements.

You may also like

- Artificial neural network based particle swarm optimization in predictions mortality rate of broiler chicken
  D Hasman, D Gustian, R B Saputra et al.
- Mortality and cancer registration experience of the Sellafield workers known to have been involved in the 1957 Windscale accident: 50 year follow-up
  D McGeoghegan, S Whaley, K Binks et al.
- Estimating the mortality burden attributable to temperature and $PM_{2.5}$ from the perspective of atmospheric flow
  Ling Han, Zhaobin Sun, Juan He et al.

Recent citations

- XGBoost in handling missing values for life insurance risk prediction
  Deandra Aulia Rusdah and Hendri Murfi

The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada
Abstract submission deadline: **Dec 3, 2021**

Connect. Engage. Champion. Empower. Acclerate.
We move science forward

Submit your abstract

This content was downloaded from IP address 88.70.113.41 on 05/11/2021 at 13:14

Ensemble learning for predicting mortality rates affected by air quality

Kartika C Dewi[1], **Widya F Mustika**[2], **H Murfi**[3]
Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia

E-mail: kartika.chandra71@sci.ui.ac.id[1], widya.fm@sci.ui.ac.id[2] and hendri@ui.ac.id[3]

Abstract. Methods in machine learning are very helpful to solve various problems, especially those related to large data. The mortality rate is one of the problems related to large data which is fluctuating depending on the factors that influence it. One of the factors that affect the mortality rates is air quality. Methods that can be used to predict the mortality rate of a population are Random Forest and Extreme Gradient Boosting (XGBoost), which is an ensemble method with decision trees as the basic model. The missing values in the data used to cause the low level of accuracy. In this paper, we discuss how to handle missing values and comparing the accuracy level of ensemble methods that we used to predict the mortality rate. By the simulation results, it shown that handle the missing values in the data is best overcome by removing the missing values (Drop NaN). Mean Square Error (MSE) value generated by the Random Forest and XGBoost methods are 0.007239 ± (1.699 x 10^{-7}) and 0.04019. Based on the MSE values of both methods, Random Forest gives better accuracy than XGBoost to predict mortality rate affected by air quality.

1. Introduction

Technology cannot be separated from everyday life for society in general in this modern era. Technology is developing very rapidly regarding both software and hardware. Technological developments that increasingly enhance the function of a device aim to help human work. For example, solve population problems that exist in a country. Population structure in a country influenced by demographic processes, namely birth (fertility), death (mortality), and population mobilization. In planning the development of a country, the process is already affected by a large demographic. But three components of the demographic process will continue to change over time. One of the important component that influences the process is mortality.

Mortality rate defined as a measure of the death of a population in a particular place, time, and condition [1]. Mortality rate influenced by various factors including age, sex, disease, weather, air quality, and many other factors. Many factors that influence mortality rates cause difficulties in predicting mortality rates quickly and manually. So, it needs to apply a method that can predict the mortality rate of a population by considering the factors that influence it.

One of the common methods that can be used to a predict mortality rate is machine learning [2]. Machine learning is a model that can learn from data to obtain knowledge in the data. One of the learning methods in machine learning is supervised learning. The purpose of supervised learning is to build a model that can produce the most appropriate output for all training data. One of the supervised learning method is the ensemble learning. Ensemble learning is one of the popular learning methods in the field of Data Mining and Machine Learning. This method combines several models into one and produces a

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

more accurate output [3]. Ensemble method is a method that can improve the accuracy of base models such as decision trees, artificial neural networks, and Naive Bayes [4]. Ensemble method consists of two main approaches, namely, bagging and boosting. Bagging, which is to build several base models independently and the final prediction is averaging or voting from these base models. Boosting, which builds final models from several base models in a sequence. A base model depends on the performance of the previous base model. One of the advantages of the ensemble method is in handling large data especially a large volume of data. This ability is very important because most of the real problems have a large volume. Also, some simulations also show that the ensemble method gives better accuracy than other methods for both regression and classification problems [5].

Random Forest and XGBoost are popular ensemble methods for bagging and boosting, respectively. Both methods use decision trees as their base models. In this paper, we compare the accuracy of both models for the problem of mortality rate prediction which is a regression problem. But, missing values in the data used to cause the low accuracy of both methods. Therefore, we simulate the preprocessing data stage to handle missing values. After the preprocessing data stage, we get accuracy level that seen from the MSE values generated by both methods. From the level of accuracy obtained, it is known which method is more accurate in predicting mortality rates affected by air quality. Based on the MSE values of both methods, Random Forest gives better accuracy than XGBoost to predict mortality rate affected by air quality.

The rest of the paper is organized as follows: In section 2, the reviews of related works are presented. Section 3 describes the methodology. Section 4 describes the simulation. In section 5, we discuss the results of the simulations. Finally, we give the conclusion in Section 6.

2. Related works

Research on the prediction of mortality rates conducted by Mitchell in 2013 with a mortality index modeled as Normal Inverse Gaussian for modeling and predicting mortality rates [6]. Li, Yuenan et al. in 2017 predicts nonaccidental mortality rates associated with PM2.5 concentrations exceed national annual standards in China. Also, the study also shows the risk of death spatially increasing in Beijing. The model used is the weighted regression (GWR) [7]. In the year 2018 Wu, R. et al. apply the Gaussian process method by using mixed covariance functions to improve accuracy in predicting mortality rates [8].

Wang, G, et al. in 2015 predict mortality rates after radical cystectomy for bladder cancer with seven machine learning techniques, namely Back-Propagation Neural Network (BPN), Radial Base Function (RBFn), Extreme Learning Machine (ELM), Regularized ELM (RELM), Support Vector Machine (SVM), naive Bayes (NB) Classifier and K-Nearest Neighbor (KNN). The results showed that the RELM resulted in the highest average predictive accuracy value of 0.8 with a fast learning speed [9]. In 2018, Parreco et al. using machine learning to predict central line accurately, CLABSI placement and death in patients admitted to the ICU. This study applies different machine learning algorithms, namely Logistic Regression, Gradient Boosted Trees, and Deep Learning. The Gradient Boosted Tree model can be used to identify the most important variables and can be used as a guide for efforts to prevent infection. The use of prediction models in the study has implications for quality and cost reduction [10].

Over the past few decades, the ensemble method has been very popular in the fields of computational intelligence and machine learning. This is because the ensemble method has proven to be very effective and versatile in a variety of problems and applications in the real world. Initially, the ensemble method was developed to reduce variance to improve the accuracy of the automatic decision-making system. Since then, the ensemble method has been successfully used to overcome various machine learning problems, such as feature selection, trust estimation, missing features, incremental learning, error correction, class imbalance data, learning concept deviation from non-stationary distributions, and others [11].

The missing value in the data used must be handled with several strategies. Aljuaid et al (2016) in their paper comparing of imputation techniques such as mean, mode, KNN, Hot-Deck, Expectation Maximization and C5.0 for Missing Values (MVs). The study found that mean imputation disturbs normality assumptions, as well as reducing association with other variables. It can be used if the missing data less than 5% [12].

Therefore, application of the ensemble method developed in this study, namely Random Forest and XGBoost to predict mortality rate affected by air quality in the population in a region. Furthermore, error value from the predictions of two methods compared so that the best method is used to predict mortality rates in a particular area.

3. Methods

In general, the learning process in machine learning divided into supervised learning and unsupervised learning. In supervised learning, training data is accompanied by targets, while in unsupervised learning training data is not accompanied by targets. Two main problems of supervised learning are classification (target in the form of class) and regression (continuous/real value target). One approach to improving the performance of a model is to combine several models with a particular mechanism known as an ensemble method [2]. Ensemble method is a machine learning paradigm where training datasets are trained to solve the same problem. Machine learning approach learns a hypothesis from training data. In the ensemble method, the hypothesis is used to construct a set of hypotheses then combined [10].

One of the basic models used in the ensemble method is a decision tree. The decision tree is a model like a flow chart consisting of nodes and branches. The node represents the testing of a particular feature. The branch represents the results of the test. Each leaf node states a label or decision after calculating all features. The path from the root to leaf states a rule. The advantages of the decision tree are simple models to be understood, interpreted, and can be visualized. The decision tree is a white box model, where rules can be explained easily using Boolean logic. Meanwhile, most other machine learning models are a black box where rules are difficult to interpret [2].

Ensemble method consists of two main approaches that represent state-of-the-art learning approaches, namely Bagging and Boosting. Bagging method that is building several models independently is used to increase predictive accuracy sequentially at a certain time, where the final prediction is average of the prediction for regression problems and voting for classification problems [13]. One popular bagging method is the Random Forest. Boosting method, which builds several models in a sequence where the error function used to build a model depends on the performance of the previous model. Boosting method process depends on the accuracy of the basic model of the smallest weak hypothesis error [14]. One method of boosting is XGBoost.

3.1 Random Forest

Random Forest method is a machine learning method introduced by Leo Breiman in the 2000s [15]. Random Forest has advantages such as classification, feature selection, feature weighting, and outlier detection. Random Forest method is used to improve the accuracy of a prediction. This method has been developed to solve both classification and regression problems [16].

Random Forest model is an ensemble method, which is a machine learning technique that aims to combine predictions from some basic estimators to improve the accuracy of some of the basic estimators. The basic estimator commonly used is a decision tree model. Input datasets will be divided into several bootstraps, then a decision tree created from each bootstrap. For a regression problem, the predicted value is the result of the average prediction of all decision trees that have been made.

The algorithm below is the Random Forest algorithm used to solve regression problems. The Random Forest method itself is one of the bagging approaches in the ensemble method. The basic model of the Random Forest method is a randomized decision tree. In scikit-learn, each randomized decision tree is built based on the samples taken randomly with returns from learning data. In the process of selecting a feature as a node from a randomized decision tree, the feature chosen is no longer the best feature of all features, but the best feature of bootstrap is randomly formed. Furthermore, the final prediction is the average value of the predictions of each randomized decision tree model for the case of regression and voting for the classification case [2].

Table 1. Random Forest algorithm.

```
Algorithm 1: Random Forest for regression [17]
• Input: Given a dataset $[(\boldsymbol{x_1}, \boldsymbol{y_1}), \dots, (\boldsymbol{x_n}, \boldsymbol{y_n})] = \boldsymbol{N}$
  Process :
   For $\boldsymbol{b} = \mathbf{1}$ to $\boldsymbol{B}$:
    a) Draw a bootstrap sample $\boldsymbol{Z}^*$ of size $\boldsymbol{N}$ from the training data.
    b) Grow a random-forest tree $\boldsymbol{T_b}$ to the bootstrapped data, by recursively
       repeating the following steps for each terminal node of the tree, until the
       minimum node size $n_{min}$ is reached.
         i.   Select $m$ variables (features) at random from $p$ variables (features).
         ii.  Pick the best variable/split-point among the $m$.
         iii. Split the node into two daughter nodes.
• Output: The ensemble of trees $\{\mathrm{T_b}\}_1^{\mathrm{B}}$
 To predict a new point $x$ : $\boldsymbol{\hat{f}_{rf}^{B}(x)} = \boldsymbol{\frac{1}{B}\sum_{b=1}^{B} T_b(x)}$
```

*3.2 Extreme Gradient Boosting (*XGBoost*)*

Research on gradient boosting has done from a variety of sciences since the 1990s, Leo Breiman 1997 first introduce that boosting can be interpreted as an appropriate optimization algorithm [18]. Freund and Schapire (1997), Jerome H. Friedman (1999) [19], and Mason's 1999 perspective on general function gradient boosting [20]. The boosting process is very simple, which is to minimize the Root Mean Square Error (RMSE) sequentially, the gradient boosting concept is located in its development, which is an additional expansion of the criterion fittings [21].

Extreme Gradient Boosting (XGBoost) method is the development of gradient boosting models that have superior results and process speeds and can process data with missing values without preprocessing data stage, automatically parallel calculations on a single machine, both used for large data with results the good one. XGBoost method can solve regression and classification problems properly, for regression problems used by XGBoost Regression. The algorithm below shown the process of XGBoost.

Table 2. XGBoost algorithm.

```
Algorithm 2: XGBoost algorithm [22]
• Input: Given a dataset $\{\boldsymbol{D}, \boldsymbol{y}\}$ and $\boldsymbol{p}$ CARTs $\boldsymbol{f(x)}$ as weak learners.
  Process :
    The ensemble technique sequentially adds weak learners from previous
    ensemble residual values. If $\boldsymbol{k} > \mathbf{0}, \boldsymbol{k} \in \boldsymbol{N}$ is boosting as much as k, then
    ensemble $\boldsymbol{F_k(x)}$ is as much as k that is,
                    $\boldsymbol{F_k(x)} = \boldsymbol{\sum_{i=1}^{k} f_i(x)}$
   Minimize $\boldsymbol{\mathcal{L}^{(k)}}$ :
                    $\underset{\boldsymbol{f_k}}{\boldsymbol{min}}\, \boldsymbol{\mathcal{L}^{(k)}} = \underset{\boldsymbol{f_k}}{\boldsymbol{min}} \sum_{i=1}^{n} \boldsymbol{l}(\boldsymbol{y_i}, \boldsymbol{F_{k-1}(x_i)} + \boldsymbol{f_k(x_i)}) + \boldsymbol{\Omega(f_k)}$
                    $\boldsymbol{\Omega(f_k)} = \boldsymbol{\gamma T} + \frac{1}{2}\boldsymbol{\lambda}||\boldsymbol{w}||^2$
    with
    $\boldsymbol{\gamma}$ dan $\boldsymbol{\lambda}$: regularization and the minimum error value of hyperparameters
    $\boldsymbol{T}$: number of tree nodes
    $\boldsymbol{w}$: node weight
    Output: minimize $\boldsymbol{\mathcal{L}^{(k)}}$
```

Learning process using the XGBoost method introduced by Xu et al. in 2017 predicts the complex condition production process uses the EMD-XGBOST-RLSE model [23]. There are three reasons why XGBoost has better performance than other boosting methods, including the introduction of a regular loss function, the weight of each new tree can be reduced by the constant given with the aim of reducing the effect of a single tree on the final score, taking samples of the same column like the random forest method [24].

XGBoost algorithm builds models accurately on conventional data or also called structured data. XGBoost algorithm stage begins by dividing the dataset into two parts, namely training and testing data. Then boosting is made to create a model from the parameters of the XGBoost method. Predetermined training data used to create the first model. This model is then evaluated using testing data. If the error is still found, the model will be rebuilt by minimizing the error by using gradient descent, and the data will be updated again according to the learning rate. This continues until the minimum error value obtained.

4. Simulation

The data used in this study taken from Kaggle. This data contains Id (id for a region), region (identifier of a region), date (date information in one year for each region), O_3 mean (average ozone level calculated per day in region), PM_{10} (average particle diameter ≤ 10 micrometers per day), $PM_{2.5}$ (mean particle ≤ 2.5 micrometers per day), NO_2 mean (nitrogen dioxide, average per day), T_2M (Temperature at 2 m, average per day), mortality rate (number of deaths per 100,000 people). The following table is data statistics used.

Table 3. Initial data statistics on Kaggle[1].

	Mortality rate	O_3	PM_{10}	$PM_{2.5}$	NO_2	T_2M
Count	18403	18394	18394	15127	11833	18403
Mean	1.301737	45.325857	13.712272	7.498714	12.045813	283.002235
Std	0.304161	16.22133	7.421616	5.758357	8.296675	5.182186
Min	0.439	0.988	2.02	0.904	1.104	265.562
25%	1.102	35.07425	8.65625	3.624	6.056	279.3215
50%	1.281	45.836	11.7045	5.636	9.769	283.27
75%	1.474	55.881	16.589	9.3265	15.858	287.2405
Max	2.841	105.693	60.627	45.846	76.765	297.209

[1] https://www.kaggle.com/c/predict-impact-of-air-quality-on-death-rates/

This dataset contains five features (O_3, PM_{10}, $PM_{2.5}$, NO_2, and T_2M) and consists of 18,403 lines that indicating the number of the dataset. Based on table 3 (count), it is known that the amount of data for each feature is not the same. The features O_3 and PM_{10} have 18,394 number of data, $PM_{2.5}$ has 15,127 number of data, NO_2 has 11,833 number of data, and T_2M has 18,403 number of data. This is due to the different missing values for each feature. The existence of missing values in data results in a low accuracy of the model. Therefore, preprocessing of data is carried out to overcome the problem of missing values so that the accuracy of the model can be improved. The dataset divided into two parts, 80% as training data and 20% as testing data. These following are the steps taken in the simulation process.

4.1 Pre-processing data

This preprocessing data stage is needed because there is a missing value in the data used. The method used to overcome this missing value consists of:

- *FillNaN = 0.* Change the missing value with a value of 0 (zero).
- *FillNaN = Mean.* Change the missing value with the average value of the data in the same column/row.
- *FillNaN = Median.* Change the missing value with the median of the data in the same column/row.
- *FillNaN = Most frequent.* Change the missing value with the value that most often appears from the data in the same column/row.
- *FillNaN = Drop NaN.* Remove the missing value in the data in the same column/row.

In this paper, we delete data based on rows for Drop NaN strategy because the data used only has five features, so the features will be very small if the data is deleted based on the column. After the preprocessing data stage, data obtained is a data that does not contain a missing value in accordance with the preprocessing method used. Then, data is divided into training data and testing data with a composition of 80% for training data and 20% for testing data.

4.2 Learning process

Learning process aims to determine the best parameter value of the method in training data provided. The learning process is carried out using the Random Forest and XGBoost methods, then random forest and XGBoost simulation methods are performed using Spyder with Python Script 3.6.

4.3 Evaluation model

Evaluation model process aims to determine the level of accuracy model. The accuracy of the model obtained by looking at the value of Mean Square Error (MSE). Mean Square Error is an error function measurement unit that used as a basic statistical method to measure model performance, for example, is air quality on regression problems. The range of MSE values is between 0 and 1. If the MSE value is closer to 0 then the model used is better. MSE calculations for the dataset are as follows [25]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2$$

with

n = number of samples
e_i = error model, where $e_i = t_i - x_i$

5. Result and analysis

The level accuracy of the method used in this paper obtained by looking at error value from the prediction of mortality rate generated by Random Forest and XGBoost methods. Error values for both methods calculated with the mean square error. In the Random Forest model, there is a randomness factor every time a simulation performed. This results in different MSE values generated each time a simulation is performed. To overcome this, MSE variance value calculated from the Random Forest model to obtain the level of accuracy. Variance values describe how quantitative data are distributed. Low variance values indicate that the data points are very close to the average value or expectation value and between each other. Random Forest method simulation process is carried out five times to get the average MSE value and variance sought. Whereas XGBoost method is stable so that the same MSE values and accuracy levels obtained for each simulation. Therefore, the model variance value in XGBoost method is equal to 0 (zero). Following is the table of simulation results.

Table 4. MSE value.

Pre-processing Data	MSE	
	Random Forest	XGBoost
Drop NaN	0.007239 ± (1.699 x 10^{-7})	0.040192 ± 0
Mean	0.008156 ± (8.974 x 10^{-8})	0.045209 ± 0
Median	0.008094 ± (5.670 x 10^{-8})	0.044953 ± 0
Most frequent	0.008072 ± (1.528 x 10^{-6})	0.044732 ± 0
Zero	0.008196 ± (3.165 x 10^{-8})	0.044650 ± 0

According to table 4 it can be seen that the variance value of the MSE average in the Random Forest method has a very small value, which is 1,699 x 10^{-7} for Drop NaN strategy, 8,974 x 10^{-8} for Mean strategy, 5,670 x 10^{-8} for Median strategy, 1,528 x 10^{-6} for Most Frequent strategy and 3,165 x 10^{-8} for Fill NaN = 0 strategy. This very low variance value indicates that the MSE value point is skewed with its mean value. In other words, the MSE value generated from five simulations for each preprocessing data strategy is very little difference. While the variance for each preprocessing data strategy in the XGBoost method is 0 (zero) because this method produces the same MSE value in each simulation.

From table 4, the highest average MSE value for the Random Forest method obtained from the prediction of mortality rates obtained by the Fill NaN strategy = 0 by 0.008196 ± (3.165 x 10^{-8}). While the lowest average MSE value obtained with Drop NaN strategy of 0.007239 ± (1,699 x 10^{-7}). Meanwhile, the highest MSE value for the XGBoost method from the predicted mortality rate obtained by the Mean strategy of 0.045209. While the lowest MSE value obtained with a Drop NaN strategy of 0.040192.

The simulation results show that the smallest MSE values for both methods obtained by preprocessing data using Drop NaN. The smallest average MSE value with Drop NaN in Random Forest is 0.007239 ± (1.699 x 10^{-7}), while for XGBoost is 0.040192. But, the use of Drop NaN strategy can reduce the amount of data (if data is deleted by rows) or features (if data is deleted by column). This is certainly not recommended for very large data with many missing values, because the amount of data deleted can affect the learning process and the level of accuracy.

For other Drop NaN preprocessing strategy, the minimum MSE value for Random Forest obtained by Most Frequent that is 0.008072 ± (1.528 x 10^{-6}) and Fill NaN = 0 obtains minimum MSE value for XGBoost that is 0.044650. It can be seen that the minimum MSE value of Random Forest is smaller than XGBoost.

6. Conclusion

Based on the simulation results for the Random Forest and XGBoost methods, it is known that the best data preprocessing strategy in the case of predicting mortality rates using Drop NaN. From MSE value obtained, it concluded that the Random Forest method has a higher level of accuracy than XGBoost method to predict mortality rates that are influenced by air quality.

Acknowledgments

This work was supported by Universitas Indonesia under PITTA 2018 grant. Any opinions, findings, and conclusions or recommendations are the authors' and do not necessarily reflect those of the sponsor.

References

[1] Porta M 2001 *A Dictionary of Epidemiology* (Oxford: Oxford University Press)

[2] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Berkeley: Springer)

[3] Seni G and Elder J F 2010 Ensemble methods in data mining: improving accuracy through combining predictions *Synthesis Lectures on Data Mining and Knowledge Discovery* 2 pp 1–126

[4] Kim Y 2009 Boosting and measuring the performance of ensembles for a successful database marketing *Expert Systems with Applications* 36 pp 2161–76

[5] Piao Y, Park H W, Jin C H and Ryu K H 2014 Ensemble method for classification of high-dimensional data 2014 *International Conference on Big Data and Smart Computing* (*BIGCOMP*) pp 245–249

[6] Mitchell D, Brockett P, Ariga R M and Muthuraman K 2013 Modeling and forecasting mortality rates *Insurance: Mathematics and Economics* 52 pp 275–285.

[7] Li Y, Chen Z and Li J 2017 How many people died due to PM2.5 and where the mortality risks increased? A case study in Beijing *IEEE International Geoscience and Remote Sensing Symposium* (*IGARSS*) pp 485–488

[8] Wu R and Wang B 2018 Gaussian process regression method for forecasting of mortality rates. *Neurocomputing*

[9] Wang G, Lam K M, Deng Z and Choi K S 2015 Prediction of mortality after radical cystectomy

for bladder cancer by machine learning techniques *Computers in Biology and Medicine* 63 pp 124–132

[10] Parreco J P, Hidalgo A E, Badilla A D, Ilyas O and Rattan R 2018 Predicting central line-associated bloodstream infections and mortality using supervised machine learning *Journal of Critical Care* 45 pp 156–162

[11] Zhang C and Ma Y 2012 *Ensemble Machine Learning Methods and Applications* (London: Springer)

[12] Aljuaid T and Sasi S 2016 Proper imputation techniques for missing values in data sets *International Conference on Data Science and Engineering (ICDSE)*

[13] Oliveira E M and Oliveira F L C 2017 Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods *Energy* 144 pp 776–788

[14] Freund Y and Schapire R E 1997 A decision-theoretic generalization of on-line learning and an application to boosting *Journal of Computer and System Sciences* 55 pp 119–139

[15] Biau G 2012 Analysis of a Random Forests model *Journal of Machine Learning Research* 13

[16] Goel E, Abhilasha E 2017 Random Forest: a review *IJARCSSE* 7

[17] Hastie T, Tibshirani R and Friedman J 2008 *The Elements of Statistical Learning* (Stanford: Springer)

[18] Breiman L 1997 Arcing The Edge

[19] Friedman J H 1999 Greedy function approximation: a gradient boosting machine

[20] Mason L, Baxter J, Bartlett P L and Frean M 1999 Boosting algorithm as gradient descent *12th Internal conference on Neural Information Processing Systems* pp 512 - 518

[21] Friedman J H 2001 Greedy function approximation: a gradient boosting machine *Ann. Stat* 29 p 5

[22] Chen T & Guestin C 2016 XGBoost: a scalable tree boosting system *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp 785–794

[23] Xu Z, Yan L and Wang M 2017 Complex production process prediction model based on EMD-XGBOOST-RLSE *The 9th International Conference on Modelling, Identification and Control (ICMIC)* pp 940–947

[24] Pan B 2018 Application of XGBoost algorithm in hourly pm2.5 concentration prediction *IOP Conf. Series: Earth and Environmental Science* 113 pp 1 – 7

[25] Chai T and Draxler R R 2014 Root Mean Square Error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature *Geosci Model Development* 7 pp 1247–50