A comparison of extreme gradient boosting, SARIMA, exponential smoothing, and neural network models for forecasting rainfall data

R Agata* and I G N M Jaya
Department of Statistics, Universitas Padjadjaran, Indonesia

*Corresponding author e-mail: ruthagataprb@gmail.com

Abstract. Extreme gradient boosting, is a combination of gradient descent and boosting that can be used to build an optimal model for time series data. This method was used to forecast the rainfall data in city of Bandung for period 2018-2019 and compared to Seasonal Autoregressive Integrated Moving Average (SARIMA) exponential smoothing, and artificial neural network which were used as benchmarks. Data used in this study were monthly rainfall from 2000 through 2017. The extreme gradient boosting had the lowest mean absolute deviance, root mean squared error deviance, and mean absolute percentage error. This indicates the extreme gradient boosting model performed better than the SARIMA, exponential smoothing, and neural network. Based on the extreme gradient boosting model, it is concluded that the highest rainfall will occur between September 2018 and May as a rainy season in Bandung, and the lowest rainfall will occur between June and August as a dry season.

1. Introduction
Rainfall is the most important climate element affecting human daily life with very complex phenomenon and varies by time and space. However, rainfall is predictable [1], [2]. Developing the accurate forecasting methods of rainfall becomes an interesting topic [3] and becomes the biggest difficulty from different areas such as weather and data mining [4], environmental [5], operational hydrology [6], and statistical forecasting [7]. The objective is to obtain the accurate future prediction when the series data have a complex pattern. Small error with high power is important part in evaluating forecasting models. Several methods have been developed for univariate time series data such as Box-Jenkin ARIMA and exponential smoothing models. ARIMA estimates the model based on the autocorrelation function while exponential smoothing models are based on decompose trend and seasonality in the data ([8], [9], [10]). Both methods are developed by using linear models where the prediction is a weighted linear sum of recent past observation or lag [8]. If the deterministic process can be identified easily, the accurate predicted can be obtained.

However, the linear models will be successfully applied to data if data generated from deterministic processes are not a random process [11]. Rainfall data are random process and have non-linear pattern. A non-linear model may be better than ARIMA and exponential smoothing models for forecasting purpose.

A hybrid multi-model method such as artificial neural network (ANN) and gradient boosting were developed to overcome the non-linearity problem and random process in time data [3]. The main advantage of neural network lies in its ability to capture any arbitrary function such as nonlinear function ([12], [13]). However, the “black box” nature in neural network becomes the problem because there is
no information provided how and why the neural network came up with a specific output. In some applications the estimation process becomes the most important step. The bad or good result can be identified from the estimation process. Extreme gradient boosting (XGBoost) was developed to captures the interaction and nonlinearity in data structured where the estimation process can be known [14]. Using XGBoost, the most important structure that explains the generating series data can be defined.

The aim of this study is to examine whether extreme gradient boosting can outperform the classical ARIMA and exponential smoothing and the other machine learning artificial neural network for rainfall forecasting. This research provides more evidences as the usefulness of extreme gradient boosting for rainfall forecasting by performing training and testing data method. The comparison method is not done by simulation procedure. We use the real data and compare the model quality based on several criterion including mean absolute deviation (MAD), root mean squared error deviance (RMSED), and mean absolute percentage error (MAPE).

Here we focus on several alternative methods where the software was available in R-library. We believe this research is needed by practitioners due to the limitations in understanding statistics and being better able to operate computer programs.

The paper is structured as follows: in Section 2, we define data and methodology, move on with result and discussion in Section 3. Finally, Section 4 concludes the paper with final remarks.

2. Material and Method

2.1. Material

We obtained historical data of average monthly rainfall for Bandung city over the period of 2000-2015 for the purpose of model identification and those of 2016-2017 for forecast validation of the chosen model from the National Metrological Center, Bandung-Indonesia. The data were presented in Figure 1.

![Rainfall data in Bandung city, 2000-2017](image)

**Figure 1.** Rainfall data in Bandung city, 2000-2017

Figure 1 presents the temporal pattern of monthly rainfall data, 2000-2017. We observed that there is no temporal trend however, it clearly presents seasonal pattern.

2.2. Method

2.2.1. ARIMA

Box-Jenkins ARIMA is the most popular ordinary time series forecasting technique for stationary and non-stationarity data. The aim is to estimate an autoregressive integrated moving average (ARIMA)
model which is used to explains the temporal pattern of rainfall data. The basic Box-Jenkins expressed
by the following form [11]:

\[ y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} - \sum_{j=1}^{q} \beta_j e_{t-j}, i = 1, ..., p \text{ and } j = 0,1, ..., q \]  

(1)

where \( y_t \) denotes a rainfall data, \( \alpha_0 \) is an intercept term, and \( e_t \) is error term. The error term usually
assume to be withe noise. Then, the \( i \) and \( j \) terms are autoregressive and moving average parts of the
model. Eq.1 classified as an ARMA\((p, q)\) model, where \( p \) denotes the number of lag, and \( q \) is the number
of lagged forecast errors. Integrated with number differencing \( d \) may be applied to satisfy the stationarity
assumption. Eq.1 can be extended for seasonal model becomes ARIMA\((p, d, q)(P, D, Q)\)\([5]\), where
\( P, D, Q \) are similar to \( p, d, q \) but here as a part of seasonal components with \( s \) denotes the seasonal order.
The t-test is used to evaluate the significane SARIMA parameters.

2.2.2. Exponential Smoothing (ETS)
Exponential smoothing (ETS) is the simplest deterministic method is usually used in forecasting with
no clear trend or seasonal pattern were identified. The forecast is calculated using a linear weighted
average that reduces exponentially as observations are associated with the earliest observation in the
past-smallest weights.

\[ y_{t+1|t} = \delta y_t + \delta(1-\delta)y_{t-1} + \delta(1-\delta)^2y_{t-2} .... \]  

(2)

where \( 0 \leq \delta \leq 1 \) denote the smoothing parameter. The forecast for time \( t+1 \) is a weighted average of
all series, \( y_1, ..., y_T \). Parameter \( \delta \) controls the rate at which the weights decrease \([8]\).

2.2.3. Artificial Neural Network (ANN)
The most popular machine learning method for forecasting is artificial neural network (ANN). The ANN
uses the technique of error back propagation to train the network configuration. The algorithm of the
ANN consists hidden layers, output layer and number hidden layers, and a number of neurons in the
input layer. The ANN forecasting model is defined as \([3]\):

\[ y_{t+T+(m-1)r}^F = f(y_t, \omega, \theta, m, h) = \theta_0 + \sum_{j=1}^{m} \omega_j^{out} \phi \left( \sum_{i=1}^{m} \omega_{ji} y_{t+(i-1)r} + \theta_j \right) \]  

(3)

where \( \phi \) is the activation function, \( \omega_{ji} \) denotes the weights defining the link between the \( i \)-th input layer
and \( j \)-th the hidden layer; \( \theta_j \) are the bias for the \( j \)-th node of the hidden layer; \( \omega_j^{out} \) for the \( j \)-th node of
the hidden layer and the node of the output layer; and \( \theta_0 \) is the bias at the output node.

2.2.4. Extreme Gradient Boosting (XGB)
Extreme gradient boosting is one of the most loved gradient boosting algorithm. It can be used for
regression, classification, and is constructed based on gradient boosting framework. Suppose we have a
data set \( D = \{x_t; y_t\} \) containing \( T \) observation, where \( x \) denotes a training variable and \( y \) is a target
variable. The XGB approach assumes that there are \( G \) additive function of boosting as follows \([14]\):

\[ \hat{y}_t = \sum_{g=1}^{G} f_g(x_t) \]  

(5)

where \( \hat{y}_t \) denotes the prediction for time \( t \) at \( g \)-th boost, where \( f_g \) is a tree structured \( L \) with leaf \( j \) having
a weight score \( \omega_j \). The estimation process of the XGB approach is based on minimizing the loss function
\( L_g \) as:

\[ L_g = \sum_{t=1}^{T} l(y_t, \hat{y}_t) + \gamma T + \frac{1}{2}\lambda \| \omega \| \]  

(6)
where \( l \) is a differentiable convex loss function that measures the different between prediction \( \hat{y}_t \) and target value \( y_t \). \( T \) denotes the number of leaves with coefficient \( y \) and \( \lambda \) denotes the smoothing parameters. The following steps are involved in extreme gradient boosting:

(a) Initialize the boosting algorithm

\[
F_0(y) = \text{argmin}_{(y, \lambda)} \left( \sum_{t=1}^{T} l(y_t, \hat{y}_t) + yT + \frac{1}{2} \lambda \|\omega\| \right)
\]  
(7)

(b) Compute the gradient of the loss function iteratively

\[
r_{m} = -\alpha \left[ \frac{\partial L(y_t, F(y_t))}{\partial F(y_t)} \right]_{F(y) = F_{m-1}(y)}
\]  
(8)

where \( \alpha \) denotes the learning rate

Fit each \( h_m(y) = y - F_m(y) \) on the extreme gradient obtained

(c) Derive the multiplicative factor \((y, \lambda)m\) for each terminal node

Define the boosted model \( F_m(y) \) as follows

\[
F_m(y) = F_{m-1}(y) + (y, \lambda)m h_m(y)
\]  
(9)

2.3. Forecast evaluation methods

In order to compare the performance of the extreme gradient boosting, artificial neural network, ARIMA, and exponential smoothing time series model in forecasting rainfall data, several criteria will be used. The first is mean absolute deviance (MAD), as given below [9]:

\[
MAD = \frac{1}{H} \sum_{h=1}^{H} |y_{t+h} - \hat{y}_{t+h}|
\]  
(10)

where \( y_{t+h} \) denotes data testing at \( t + h \) period and \( \hat{y}_{t+h} \) is the forecast value for \( t + h \) period with \( H \) is length of forecast. The second and third criteria are root means squared deviance (RMSED) and mean absolute percentage deviance (MAPE) given below:

\[
RMSED = \sqrt{\frac{1}{H-1} \sum_{h=1}^{H} (y_{t+h} - \hat{y}_{t+h})^2}
\]  
(11)

\[
MAPE = \frac{1}{H} \sum_{h=1}^{H} \left| \frac{y_{t+h} - \hat{y}_{t+h}}{y_{t+h}} \right| 100\%
\]  
(12)

The smallest values of MAD, RMSED, and MAPE indicates the best model.

All computation process will be done by R-software using forecast and xgboost packages.

3. Results and discussion

Four different models were compared to determine the best for forecasting the rainfall data in city of Bandung, 2018-2019.

3.1. Data exploration

Diagnostic checking is the first step in order to define the temporal pattern in the data and to test the stationarity. We present the ACF plot for evaluating the temporal pattern and the Dicky Fuller test for testing stationarity assumption.
Figure 2. Autocorrelation function plot for training data (2000-2015)

Figure 2 shows the autocorrelation function plot with a clear seasonal pattern at time lag 12. It indicates that the rainfall data in the city of Bandung have seasonal pattern recurring every year (s=12 months).

Figure 3. Time series decomposition (2000-2015)

Time series decomposition (Figure 3) supports the idea that the rainfall data have strongly seasonal pattern. There is no clear pattern of linear trend. The Dicky Fuller stationarity test rejects the null hypothesis (H0: data series is not stationary) with a p-value of 0.01. It concludes that the rainfall data were stationary. Based on this diagnostic, the seasonal component have to be included in the models. Using forecast package, we obtained the best model for rainfall data is ARIMA(1,0,0)(2,0,0)[12]. Parameter estimates are presented in Table 1.
Table 1. SARIMA estimates

| Parameter | Estimate | Standard error | t-value | p-value |
|-----------|----------|----------------|---------|---------|
| AR(1)     | 0.1998   | 0.0755         | 2.6464  | 0.0090  |
| SAR(1)    | 0.2414   | 0.0709         | 3.4048  | 0.0008  |
| SAR(2)    | 0.1951   | 0.0753         | 2.5910  | 0.0105  |

Figure 4. Residual analysis

Figure 4 presents the residual analysis where the ACF and PACF plots show that the residuals satisfy the white noise assumption. This means that the ARIMA (1,0,0)(2,0,0)[12] is the best model for the rainfall training data. The best model of ETS was obtained based on an additive error, without temporal trend and with additive seasonal with smoothing parameter of 1e-04, while the forecast with ANN was obtained based on 20 networks, each of which is a 13-7-1 network with 106 weights and seasonal lag is s=12. The model of extreme boosting gradient is constructed using 28 features. The most important feature with the highest gain is lag 12 and the second one is lag 11. It indicates that there is a seasonal component for order 12 and 11.

We present the model comparison based on the mean absolute deviance (MAD), root mean squared error deviance (RMSED) and mean absolute percentage error (MAPE) of testing data in Table 2.

Table 2. Model comparison

| Model   | MAD    | RMSED  | MAPE   |
|---------|--------|--------|--------|
| SARIMA  | 116.9017 | 162.6560 | 65.7756 |
| ETS     | 117.1898 | 158.5143 | 64.8473 |
| ANN     | 122.8627 | 180.1054 | 69.6140 |
| XGB     | 112.5225 | 152.8657 | 63.7136 |

Based on the Table 2, the best model for rainfall data is the extreme boosting gradients as it has the smallest MAD, RMSED, and MAPE values.
Figure 5 shows the observed versus fitted rainfall data by means of SARIMA, EST, ANN and XGB gradient model. Based on this result, the best fitting model is BOOSTING Gradient model where the observed and fitted values are relatively similar.

**Forecast by means Extreme Boosting Gradient**

**Table 3.** Forecasted value of rainfall (2018-2019)

| Year | Jan  | Feb  | Mar  | Apr  | May  | Jun  | Jul  | Aug  | Sep  | Oct  | Nov  | Dec  |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 2018 | 144.73 | 190.59 | 279.26 | 299.08 | 201.52 | 83.25 | 100.52 | 74.72 | 147.23 | 198.06 | 317.76 | 178.32 |
| 2019 | 155.02 | 268.64 | 278.22 | 292.42 | 214.30 | 67.88 | 55.63 | 45.70 | 28.27 | 259.71 | 361.25 | 246.36 |

Table 3 shows the forecast values for 2018-2019. The low values of forecast were found around June – September which can be seen clearly in Figure 6.
Figure 6 shows the predicted (2000-2017) and forecasted value (2018-2019) of rainfall data in city of Bandung. The forecast values of rainfall seem have similar pattern with rainfall in the previous years. The high rainfall will occur between September and May as rainy season in Bandung and the lowest rainfall will occur between June and August as dry season. The temporal effect has a significant effect on the rainfall. According to Jaya et al. [15], a temporal pattern must be considered in the rainfall modeling.

5. Conclusion
This study compares SARIMA, exponential smoothing (ETS), neural network (NN) and extreme gradient boosting models to forecast monthly rainfall data in Bandung. The result shows that the extreme gradient boosting forecast was considerably more accurate with smallest MAD, RMSED, and MAPE than those of the traditional SARIMA, exponential smoothing, and neural network, which were used as benchmarks. The mean absolute deviance, root mean squared error deviance and mean absolute percentage error were all lower on average for the extreme gradient boosting than for the SARIMA, exponential smoothing, and neural network. We took monthly data from 2000 to 2015 as training data and 2016-2017 as testing data set. The extreme gradient boosting model performed better than the SARIMA, exponential smoothing, and neural network probably because the extreme gradient boosting considers more parameters including lag time and seasonal parameters. This result must be interpreted carefully because the large number of parameters may lead to overfitting problem. The SARIMA forecast is worse than expected. It is might be due to the ignoring model validation. The neural network performs worst due to small number observations.

Reference
[1] Hastenrath S, Greischar L and J Heerden 1995 American Meteorological Society 8 1511
[2] Luk K C, Ball J E, and Sharma A 2001 Mathematical and Computer Modelling 33 883
[3] Sumi M S, Zaman F M and Hirose H 2012 Int. J. Appl. Math. Comput. Sci. 22 841
[4] Yang Y, Lin H, Gou Z and Jiang J 2007 Computers Geosciences 33 20
[5] Hong W 2008 Applied Mathematics and Computation 200 41
[6] Li P and Lai E 2004 Development 288 189
[7] Pucheta J, Patino D and Kuchen B 2009 Computer and Computing Technologies in Agriculture II vol 2 IFIP Advances in Information and Communication Technology (Boston: Springer) pp 787–798
[8] Hyndman R J and Athanasopoulos G 2018 Forecasting: Principles and Practice (Melbourne: Texts)
[9] Box G E P, Jenkins G M, Reinsel G C and Ljung G M 2016 Time Series Analysis Forecasting and Control ed 5 (Hoboken: John Wiley & Sons, Inc.)
[10] Hyndman, R J, Koehler A B, Ord J K and Snyder R D 2008 Forecasting with Exponential Smoothing The State Space Approach (Berlin: Springer)
[11] Kohzadi N, Boyd M S, Kermanshahi B and Kaastra J 1996 Neurocomputing 10 169
[12] Cybenko C 1989 Math. Control Signals Systems 2 303
[13] Hornik K 1989 Neural Network 2 359
[14] Chen Z Y, Zhang T H, Zhang R, Zhu Z M, Yang J, Chen P Y, Ou C Q and Guo Y 2019 Atmospheric Environment 202 80
[15] Jaya I G N M, Ruchjana B N, Abdulah A S and Andriyana A 2018 International Journal of Geoinformatics 14 27