The comparison between extreme learning machine and artificial neural network-back propagation for predicting the dengue incidences number in DKI Jakarta

S Tiffany¹, D Sarwinda¹, B D Handari¹, G F Hertono¹, a)¹
¹Department of Mathematics, Faculty of Mathematics and Natural Sciences (FMIPA), University of Indonesia, Depok 16424, Indonesia

Corresponding author’s e-mail: a) gatot-f1@ui.ac.id

Abstract. The existence of COVID-19 in Indonesia is not the only disease which we must be aware of. The Health Ministry has said that Dengue Hemorrhagic Fever is as dangerous as COVID-19 and must also be treated with caution. Based on data, until July 2020, there are 71,633 dengue cases in Indonesia and DKI Jakarta has the sixth-highest dengue incidence number. One of the factors that affects the spread of dengue vector is weather. It is necessary to predict the number of dengue incidences so that the dengue handling and prevention efforts can be done optimally. In this study, the number of dengue incidences will be predicted by involving weather factors (rainfall, temperature, and humidity) using Extreme Learning Machine and Artificial Neural Network-Back Propagation and also comparing the both of their performance. The result shows that Extreme Learning Machine can give the dengue incidence prediction in DKI Jakarta with the best RMSE testing result of 0.04584, which is more accurate than the dengue incidence prediction that is given by using Artificial Neural Network-Back Propagation with 100 epochs. Moreover, Extreme Learning Machine can do the training process faster than Artificial Neural Network-Back Propagation.

1. Introduction
The problem about coronavirus outbreak is currently being discussed in Indonesia. However, The Health Ministry said that the health problem in Indonesia is not only about the coronavirus outbreak. In fact, there is a disease that is no less dangerous and is as deadly as the disease caused by coronavirus, called Dengue Hemorrhagic Fever (DHF). DHF is an endemic disease caused by dengue virus infection through its RNA which is included in the Flaviridae family and the genus Flavivirus. This disease is transmitted to humans through the bite of an infected mosquito, especially Aedes aegypti and Aedes albopictus, which is generally found in tropical and subtropical areas [1]. In Indonesia, there are 71,633 dengue cases that are reported until July 2020 and DKI Jakarta has the sixth-highest dengue incidence number [2].

High population densities and mobility in DKI Jakarta allow viruses to transmit easily so that DHF will spread faster and infect many people. Furthermore, another factor that is very influential in the spread of DHF is weather. Some of these factors contribute to the spread of parasites and dengue vectors: temperature, rainfall, humidity, water surface, and wind speed [3].

It is necessary to predict dengue incidences in the future so that the dengue prevention and handling effort can be done optimally. Prediction process is usually related by machine learning method.
Machine learning is a branch of artificial intelligence which focuses on developing a system based on data. Various machine learning methods that have been used to get the best prediction of dengue incidences include Neural Network and Nonlinear Regression [4], Least Squares Support Vector Machine [5], and Artificial Neural Network [6].

Artificial Neural Network (ANN) is one of the method that is most frequently used to make a prediction. ANN generally use the back propagation algorithm in training process. Although ANN-BP is mostly used in many application, it still has some limitations such as slow learning speed [7], the presence of local minima [8], and the inappropriate learning rate that can cause overtraining [8]. To overcome all of these limitations, Huang proposed the Extreme Learning Machine (ELM) method.

ELM can do the training process rapidly because this method does not require an iterative process for tuning parameters. Moreover, the generalization ability of ELM is considered to be superior to that of ANN-BP [8]. Adzkiya, Irawan, and Najar (2018) have done a prediction of DHF outbreak risk level using ELM by involving the influence of weather factors. They compared the performance between ELM using binary sigmoid activation function and ELM using bipolar sigmoid activation function. That study concluded that ELM using binary sigmoid activation function showed the best performance in predicting the risk level of DHF outbreak [9].

In this study, we will predict the dengue incidence number by involving the influence of weather factors (rainfall, temperature, and humidity) and the influence of previous cumulative dengue incidences number using Extreme Learning Machine and Artificial Neural Network-Back Propagation and also comparing their performances. This paper will be organized as follows. Section 2 provides the description of data and the method that is used in this study. Section 3 provides the result of prediction simulation and also the explanation about that result. The last section in this paper presents the conclusion from this study.

2. Material and methods

2.1. Dataset

In this study, we will use two types of data: the daily cumulative dengue incidences number data and daily weather data from five municipalities of DKI Jakarta, including Central Jakarta, East Jakarta, West Jakarta, South Jakarta, and North Jakarta (except the Thousand Islands), that are recorded from 6 January 2009 until 25 September 2017. Daily cumulative dengue incidences data is obtained from the Epidemiological Surveillance Section of DKI Jakarta Health Agency website, while daily weather data, that is consisted by three variables include the cumulative rainfall (mm), the average temperature (°C), and the average humidity (%), are obtained from Meteorology, Climatology, and Geophysics Agency. The predictor variables in this study are weather variables and the previous cumulative dengue incidences number variable is used as the input variable. The variable that was used as the target variable is the daily cumulative dengue incidences number.

2.2. Artificial neural network

ANN is a computational system that is designed to simulate how the human brain analyzes and processes information. It has a self-learning ability that allows them to get a better result when more data is available [10]. ANN is a supervised learning method which is very popular [11]. The structure of ANN generally consists of three parts: the input layer, the hidden layer, and the output layer. The number of hidden layers in ANN can be more than one. It depends on the number of data and also the complexity of data that is used. In this study, we will use one hidden layer. The structure of ANN with one hidden layer is shown in Figure 1.

In the training process of ANN, we will get the updated weight and bias values which can minimize error values. Training process of ANN is divided into two phases: the forward phase and the backward phase. The first phase is the forward phase. In Figure 1, input layer receives \( N \) different observation data, \( x^1_i, x^2_i, ..., x^s_i \), where \( i = 1, 2, ..., N \) and \( s = 1, 2, ..., n \), is the number of input variables. Furthermore, all of those input data are sent to the hidden layer.
In the hidden layer, input data will be used in the process of calculating the activation function value by using the following formula [11]:

$$a_i^j = g\left(\sum_{s=1}^{N} x_i^s w_{sj} + \beta_j\right), i = 1, 2, ..., N, j = 1, 2, ..., \bar{N},$$

where \(\bar{N}\) is the number of hidden nodes, \(a_i^j\) is the \(j\)th hidden node output, \(w_{sj}\) is the weight connecting the \(s\)th input node and the \(j\)th hidden node, \(\beta_j\) is the \(j\)th hidden node bias, and \(g\) is the activation function.

![Figure 1. The structure of artificial neural network [12]](image)

Next, the value of \(a_i^j\) is sent to the output layer, which will be used as input to calculate the activation function value in the output layer by using the following formula [11]:

$$v_r^i = g\left(\sum_{j=1}^{\bar{N}} a_i^j u_{jr} + y_r\right), i = 1, 2, ..., N, r = 1, 2, ..., m,$$

where \(m\) is the number of output nodes, \(v_r^i\) is the \(r\)th output node activation function value by using the \(i\)th observation data, \(u_{jr}\) is the weight connecting the \(j\)th hidden node and the \(r\)th output node, \(y_r\) is the \(r\)th output node bias, and \(g\) is the activation function. The process of calculating the activation function value in the hidden and output layers is called feed-forward which is carried out during the forward phase [11].

The method that is commonly used in learning process of ANN is back propagation which is carried out during the backward phase. In back-propagation method, weight and bias parameters will be updated gradually based on the error value. The error value can be calculated by using the following formula [11]:

$$Loss = \frac{1}{2N} \sum_{i=1}^{N} (y_r^i - v_r^i)^2,$$

where \(y_r^i\) is the target output value of the \(r\)th output node by using the \(i\)th data observation whose target value has been determined before based on the data used and \(v_r^i\) is the prediction output value of the \(r\)th output node by using the \(i\)th data observation. Parameters are updated in this process using the gradient descent algorithm. The gradient descent formula for updating the weight and bias can be written as follows [13]:

$$u_{jr}' = u_{jr} - \eta \frac{\partial Loss}{\partial u_{jr}}, \quad w_{sj}' = w_{sj} - \eta \frac{\partial Loss}{\partial w_{sj}},$$
\[ y'_{r} = y_{r} - \eta \frac{\partial \text{Loss}}{\partial y_{r}}, \quad \beta'_j = \beta_j - \eta \frac{\partial \text{Loss}}{\partial \beta_j}, \]  

(5)

where \( \eta \) is the learning rate, \( u'_{jr} \) is the weight connecting the \( j \)th hidden node and the \( r \)th output node that has been updated, \( y'_{r} \) is the \( r \)th output node bias that has been updated, \( w'_{sj} \) is the weight connecting the \( s \)th input node and the \( j \)th hidden node that has been updated, and \( \beta'_j \) is the \( j \)th hidden node bias that has been updated.

### 2.3. Extreme learning machine

ELM is a feedforward neural network method that only has one hidden layer (SLFN). The architecture of this method is shown in Figure 2. ELM is a supervised learning method proposed by Huang (2006) to overcome the limitations of back-propagation, such as the slow learning speed that has been a significant obstacle in its application over the last few decades [7], the presence of local minima which can affect back-propagation performance [8], and the inappropriate learning rate that can cause overtraining [8].

![Figure 2. The architecture of extreme learning machine](image)

For \( N \) arbitrary distinct samples \((x_i, y_i)\) where \( x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n \) and \( y_i = [y_{i1}, y_{i2}, ..., y_{im}]^T \in \mathbb{R}^m \), SLFNs with activation function \( g(x) \) and \( \tilde{N} \) hidden nodes generally are mathematically modeled as follows [8]:

\[
\sum_{j=1}^{\tilde{N}} \beta_j g_j(x_i) = \sum_{j=1}^{\tilde{N}} \beta_j g(w_j \cdot x_i + b_j) = o_i, \quad i = 1, 2, ..., N,
\]  

(6)

where \( w_j = [w_{j1}, w_{j2}, ..., w_{jn}]^T \) is the weight vector which connects the \( j \)th hidden node and input nodes, \( \beta_j = [\beta_{j1}, \beta_{j2}, ..., \beta_{jm}]^T \) is the weight vector which connects the \( j \)th hidden node and output nodes, \( b_j \) is the \( j \)th hidden node bias, and \( w_j \cdot x_i \) denotes the inner product of \( w_j \) and \( x_i \).

SLFNs with \( \tilde{N} \) hidden nodes and activation function \( g(x) \) can approximate \( N \) samples with zero error means that \( \sum_{i=1}^{N} ||o_i - y_i|| = 0 \) such that [8]

\[ H\beta = Y, \]  

(7)
where $H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_N \cdot x_1 + b_N) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_N \cdot x_N + b_N) \end{bmatrix}_{N \times N}$ is the hidden layer output matrix (the $j$th column of $H$ is the $j$th hidden node output), $\beta = [\beta_1, \beta_2, \ldots, \beta_j]^T$ is the weight matrix which connects hidden layer and output layer, and $Y = [y_1, y_2, \ldots, y_N]^T$ is the target matrix.

In ELM, the weight parameter which connects the input layer and the hidden layer ($w_j$) and the hidden layer bias parameter ($b_j$) is determined in the beginning of process randomly, while the weight parameter which connects the hidden and output layers ($\beta$) is calculated analytically [8]. In the training process of ELM, we will get the optimum value of $\beta$ by finding the least squares solution of linear equation $H \beta = Y$ using the following formula [8]:

$$\beta = H^T Y,$$

(8)

where $H^T$ is Moore-Penrose generalized inverse of matrix $H$ that has a formula as follows [8]:

$$H^T = (H^T H)^{-1} H^T.$$  

(9)

After we get the matrix $H^T$, output weight $\beta$ can be calculated by using the formula in the equation (8).

### 2.4. Performance criteria

Performance from each model is evaluated by calculating the difference between the predicted dengue incidences number and the actual number of dengue incidences. The performance of each model will be evaluated by using the Root Mean Square Error (RMSE) formula as follows [14]:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (f_i - y_i)^2}{n}},$$

(10)

where $f_i$ is the prediction result, $y_i$ is the actual value, and $n$ is the number of observation data.

### 2.5. Proposed method

In this study, the data used contains missing values and noise in the daily weather data. We will preprocess data by doing the data imputation using mean and cleaning the noise in the daily weather data. After that, we will convert the daily data into the weekly data.

The next step is determine time lag between the predictor variable and the weekly dengue incidence number. To find the time lag between the cumulative dengue incidence previous number variable and the weekly cumulative dengue incidences number, we will use autocorrelation, and to find the time lag between the weather variable in the previous week and the weekly cumulative dengue incidences number, we will use cross-correlation. The autocorrelation formula is written as follows [15]:

$$\text{Corr}(Y_t, Y_{t-k}) = \frac{\sum_{i=1}^{n} (Y_i - \bar{Y}) (Y_{i-k} - \bar{Y})}{\sum (Y_i - \bar{Y})^2},$$

(11)

and the cross-correlation formula is written as follows [15]:

$$\text{Corr}(X_{t-k}, Y_t) = \frac{\sum_{i=1}^{n} (X_{i-k} - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}},$$

(12)

where $k$ is lag, $Y_{t-k}$ is the number of dengue incidences in the previous week, $X_{t-k}$ is the weather variable in the previous week, $Y_t$ is the number of dengue incidences in the $t$th week dengue.
incidences number, and \( \bar{Y} \) is the average value of dengue incidences number, \( \bar{X} \) is the average value of weather variable. In this study, one week is set as the minimum time lag and eight weeks is set as the maximum time lag by considering the mosquito life cycle period and also the virus incubation period, both in the mosquito’s body (extrinsic incubation) and in the human body (intrinsic incubation).

Next, the weekly data will be normalized by using the Min-Max formula. In this study, the activation function that will be used is binary sigmoid so that the Min-Max formula can be written as follows [16]:

\[
x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}
\]

where \( x'_i \) is the normalized data, \( x_i \) is the data that will be normalized, \( \min(x_i) \) is the minimum value of \( x_i \), and \( \max(x_i) \) is the maximum value of \( x_i \).

After that, we will make the dataset which consists of predictor variables according to the previously determined time lag and weekly dengue incidence number. This dataset will be divided into two parts: training data and testing data with the composition of 90% training data and 10% testing data based on historical time series. Next, the ELM model and the ANN-BP model will be constructed which the predictor variables act as input variable and the dengue incidences number acts as target variable.

In this study, parameters \( \eta \) that will be used are \( \eta = 0.001 \), \( \eta = 0.1 \), and \( \eta = 0.5 \). The performance of each model will be evaluated by using the RMSE formula and we will compare the prediction result of each model. The research flow of proposed method is shown in Figure 3.

![Figure 3. The research flow of proposed method](image)

3. Result and discussion

In this section, we discussed the implementation of ELM and ANN-BP models, evaluated their performance, and compared the prediction results obtained when we used ELM and ANN-BP. In this study, the architectures of ELM and ANN-BP consist of one hidden layer in which the optimum number of hidden nodes is choosed by trial and error, start from 5, 10, 15, ..., 50 hidden nodes. We use four predictor variables, so that the input layer of ELM and ANN-BP consists of four nodes. There is one target of this study, namely the daily cumulative dengue incidences number, so that the output layer of ELM and ANN-BP consists of one node. Binary sigmoidal function is used as the activation function. The best learning rate is assign by trial and error which there are three trials, namely 0.001; 0.1; and 0.5. ELM and ANN-BP models will be used to predict the dengue incidence number in five municipalities of DKI Jakarta, including Central Jakarta, East Jakarta, West Jakarta, South Jakarta, and North Jakarta (except Kepulauan Seribu).

By determining the biggest absolute value of cross-correlation, we get the time lag of predictor variables. The time lag of each predictor variable used to construct the ELM and ANN-BP models for predicting DHF incidence number in five municipalities of DKI Jakarta, including Central Jakarta,
East Jakarta, West Jakarta, South Jakarta, and North Jakarta (except Kepulauan Seribu), are shown in Table 1.

**Table 1.** The time lag of each predictor variable used for constructing the model to predict DHF incidence number in DKI Jakarta

| Region          | Previous Dengue Incidences Number | Cumulative Rainfall | Average Humidity | Average Temperature |
|-----------------|-----------------------------------|---------------------|------------------|---------------------|
| North Jakarta   | 2                                 | 6                   | 8                | 1                   |
| South Jakarta   | 1                                 | 7                   | 5                | 5                   |
| East Jakarta    | 1                                 | 7                   | 8                | 8                   |
| West Jakarta    | 1                                 | 7                   | 6                | 8                   |
| Central Jakarta | 1                                 | 8                   | 7                | 8                   |

After we get the optimum time lag of each predictor variable, we implement ELM and ANN-BP. By using the optimum time lag in Table 1 and the data proportion of 90% training data and 10% testing data, the number of training and testing data that will be used is 402 and 45 datasets, respectively, except in South Jakarta, which has 403 and 45 training and testing datasets, respectively. We determine the ELM model and the ANN-BP model in each region that can yield the smallest RMSE testing value, which is shown in Table 2.

**Table 2.** The best ELM model and the best ANN-BP model for predicting the dengue incidence number in DKI Jakarta based on RMSE testing value

| Region          | ELM                  | ANN-BP               |
|-----------------|----------------------|----------------------|
|                 | Hidden Nodes Number  | Training Time (second) | RMSE Testing | Learning Rate (η) | Hidden Nodes Number | Training Time (second) | RMSE Testing |
| North Jakarta   | 10                   | 0.02                 | **0.0975**     | 0.1              | 20                  | 193.09                 | 0.0985      |
| South Jakarta   | 5                    | 0.01                 | **0.0458**     | 0.1              | 5                   | 188.72                 | 0.0473      |
| East Jakarta    | 5                    | 0.01                 | **0.0758**     | 0.1              | 10                  | 116.66                 | 0.0764      |
| West Jakarta    | 10                   | 0.02                 | **0.0766**     | 0.1              | 10                  | 184.44                 | 0.0789      |
| Central Jakarta | 5                    | 0.1                  | **0.0702**     | 0.1              | 10                  | 106.65                 | 0.0760      |

In Table 2, it is known that ELM with 5 hidden nodes can yield the smallest RMSE testing value for South Jakarta, East Jakarta, and Central Jakarta. For North Jakarta and West Jakarta, the smallest RMSE testing value is obtained when ELM consists of 10 hidden nodes. From Table 2, we can also see that RMSE testing value of ANN-BP with learning rate η = 0.1 is smaller than the RMSE testing value of ANN-BP with learning rate η = 0.001 and learning rate η = 0.5. When the hidden layer consists of 10 nodes, ANN-BP with η = 0.1 can obtain the smallest RMSE testing value for East Jakarta, West Jakarta, and Central Jakarta. When predicting the DHF incidence number in South Jakarta, the smallest RMSE testing value is obtained by ANN-BP (η = 0.1) with 5 hidden nodes. For South Jakarta, 20 hidden nodes were required for ANN-BP (η = 0.1) to yield the smallest RMSE testing value. The plot of the prediction result using the best ELM and ANN-BP models in each region can be seen in Figure 4.
In Table 2, there are numbers in bold that indicate the smallest RMSE testing value between ELM and ANN-BP model. We can see that the best RMSE testing value is obtained when using ELM is smaller than the RMSE testing value which resulted from using ANN-BP for five municipalities of DKI Jakarta. It means that by using the ELM model, we can get the dengue incidence number prediction which is closer to the actual value than using ANN-BP. Moreover, the training process of the ELM model runs faster than that of the ANN-BP model.
Figure 4. The prediction result of dengue incidences number using the best ELM model and The best ANN-BP model (a) in North Jakarta, (b) in South Jakarta, (c) in East Jakarta, (d) in West Jakarta, and (e) in Central Jakarta.
4. Conclusion
In this study, we applied ELM and ANN-BP to predict the dengue incidence numbers in five municipalities of DKI Jakarta, including North Jakarta, South Jakarta, East Jakarta, West Jakarta, and Central Jakarta by using a proportion of 90% training data and 10% testing data. Based on the dataset that is used in this study, ANN-BP with $\eta = 0.1$ using 100 epochs has the better performance than the performance of ANN-BP with $\eta = 0.001$ and with $\eta = 0.5$. However, ELM can yield the better dengue incidence number prediction than that which was obtained by ANN-BP with $\eta = 0.1$. The training process of ELM runs faster than that of ANN-BP.

Acknowledgement
This research is supported by PUTI 2020 Research Grant No: NKB-1011/UN2.RST/HKP.05.00/2020 from Universitas Indonesia.

References
[1] World Health Organization 2011 Comprehensive Guideline for Prevention and Control of Dengue and Dengue Haemorrhagic Fever (India: WHO)
[2] Rokom 2020 Hingga Juli, Kasus DBD di Indonesia Capai 71 Ribu (Jakarta: Kemkes) accessed on 9 July 2020 see http://sehatnegeriku.kemkes.go.id/baca/rismedia/20200709/3134413/hingga-juli-kasus-dbd-indonesia-capai-71-ribu/
[3] Ariati J and Musadad D A 2012 Kejadian Demam Beradarah Dengue (DBD) Dan Faktor Iklim Di Kota Batam, Provinsi Kepulauan Riau Jurnal Kesehatan 11 279-286
[4] Husin N A and Salim N 2008 A Comparative Study for Back Propagation Neural Network and Nonlinear Regression Models for Predicting Dengue Outbreak Jurnal Teknologi Maklumat 20 97-112
[5] Mustaffa Z and Yusof Y 2011 Dengue Outbreak Prediction: A Least Squares Support Vector Machines Approach International Journal of Computer Theory and Engineering 3 489-493
[6] Balasaravanan K and Deb P 2018 Machine Learning : The New ‘Big Thing’ for Competitive Advantage Int. J. Knowledge Engineering and Data Mining 7 13-15
[7] Suhareri, Nawi N M, and Fathurrahman M 2018 Early Detection of Dengue Disease Using Extreme Learning Machine International Journal on Advanced Science Engineering Information Technology 8 2219-24
[8] Huang G B, Siew C K, and Zhu Q Y 2006 Extreme Learning Machine : Theory and Applications Neurocomputing 70 489-501
[9] Adzkiya D, Irawan M I, and Najar A M 2018 Extreme Learning Machine Method for Dengue Hemorrhagic Fever Outbreak Risk Level Prediction International Conference on Smart Computing and Electronic Enterprise (ICSCEE) pp 1–5
[10] Barhoom A M, Bassem S, Khalil A J, Muskeh M, and Samy S 2019 Predicting Titanic Survivors using Artificial Neural Network International Journal of Academic Engineering Research 3 8-12
[11] Putra J W G 2019 Pengenalan Konsep Pembelajaran Mesin dan Deep Learning (Tokyo)
[12] Rajesh R, Sateeshkumar J, and Sreekanth M S 2015 Extreme Learning Machine for The Classification of Rainfall and Thunderstorm Journal of Applied Sciences 15 153-156
[13] Gandhiadi G K, Harini L P I, and Putra I M D U 2016 Implementasi Backpropagation Neural Network dalam Prakiraan Cuaca di Daerah Bali Selatan E-Jurnal Matematika 5 126-132
[14] Budiman H 2016 Analisis dan Perbandingan Akurasi Model Prediksi Rentet Waktu Support Vector Machines dengan Support Vector Machines Particle Swarm Optimization untuk Arus Lalu Lintas Jangka Pendek Systemic 2 19-24
[15] Chan K and Cryer J D (2008) Time Series Analysis With Applications in R (USA: Springer)
[16] Birmohan S and Dalwinder S 2019 Investigating The Impact of Data Normalization on Classification Performance Applied Soft Computing Journal 1-23