An artificial neural network model for forecasting air pollution

W D W A Manan and N R Abdullah

1Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), 21080 Kuala Terengganu, Terengganu, Malaysia

*Corresponding author: wand@uitm.edu.my

Abstract. Air pollution has caused a lot of problems to people in terms of health and economy, as well as affecting various floras and faunas. Thus, monitoring air quality levels and forecasting the occurrence of air pollution is important so that preventive measures could be taken. In this study, Artificial Neural Network (ANN) was used to forecast the air pollution index (API) in Kuala Terengganu. This study focused on the prediction of API based on 5 years of data of main pollutants’ daily concentration taken at the air quality monitoring station in Kuala Terengganu. The aim was to develop an Artificial Neural Network model that can predict the API. A Multilayer Perceptron Neural Network (MLP) engine was implemented in the system prototype and developed by using Keras, a deep learning library in Python. The model’s performance was evaluated using the Mean Squared Error (MSE) statistical method and functionality tests were done to ensure the prototype was working correctly. In order to get a good performance model, a hyperparameter tuning process was carried out and the best hyperparameters values were selected. The performance of the model in making predictions was good as the MSE value was 0.0195.

1. Introduction

Clean air is significant for the survival of humans and other living beings. It impacts the well-being and economic development of a country. Air pollution is a grave problem that leads to some crucial concerns in many parts of the world. These worries are mainly about the effects of air pollution on human well-being and health [1], [2], [3]. Air pollution not only endangers people but can also affect crops, forests and animals [1], [4]. It is common knowledge that air pollution also impacts the environment where it causes changes in the climate, creating holes in the ozone layer and increasing the pollution of particulate matter in the air [1], [2]. Therefore, some actions could be taken to reduce its effects if the occurrence of air pollution could be predicted.

This fact motivated this project, which tried to predict air pollution incidence in the city of Kuala Terengganu using an artificial neural network model. The project had three objectives; first, to study algorithms that can be used to predict the occurrence of air pollution. Second, to develop an artificial neural network (ANN) model that can predict air pollution in Kuala Terengganu. Third, to evaluate whether the system prototype built from the ANN model is functioning correctly. At the end of the project, a prototype was built so that users can predict the occurrence of air pollution and find out the status of air quality in Kuala Terengganu.

The user of the prototype is targeted to be someone who wants to predict air pollution in Kuala Terengganu such as workers of the Department of Environment (DOE) and the Meteorological
Department. The prototype could also be used by the general public and other government departments as long as they have the required input data. As the capital of the state of Terengganu, Kuala Terengganu was a suitable focus area for this study. Furthermore, it also has a fairly significant industrial area which is one of the major contributors to air pollution.

All required data for the project were retrieved from the DOE. Five years data of air pollution index (API) and daily concentration of main pollutants such as particulate matters under 10µm (PM10), carbon monoxide (CO), sulphur dioxide (SO2), ozone (O3) and nitrogen dioxide (NO2) in Kuala Terengganu were collected. Figure 1 below shows the conceptual framework of this project.

![Conceptual Framework](image)

**Figure 1.** Air Pollution Forecasting in Kuala Terengganu using Artificial Neural Network Project Conceptual Framework.

### 2. Related Work

#### 2.1. Air Pollution Forecasting

Forecasting of air pollution started in the 1960s, due to rising awareness of the trends and influence of air pollution [4]. The purpose of air pollution forecasting is to predict the composition of air pollution in the atmosphere at a particular time and area. Lately, numerous researchers have conducted various studies to provide accurate air pollution forecasting using ANN. The forecast may result in the prediction of the pollutants’ concentration or air quality status. [5] stated that to prevent the rising of air pollution, it is important to forecast the content of pollutants in the air.

Similar to this project, many other kinds of research concentrated on building models to predict the air pollution index. For instance [5] constructed a Feedforward Neural Network (FFNN) model, while [6] used Long Short-Term Memory (LSTM), [7] implemented MLP and [8] used various machine learning techniques in building their prediction models.

#### 2.2. Machine Learning

Most research mentioned above, used machine learning techniques in constructing their prediction models. The Artificial Neural Network technique used in this research is one of the most popular machine learning techniques. [9] defined machine learning as “an artificial intelligence branch that uses intelligent software to enable machines to perform their jobs skilfully”. Machine learning gives
computers the ability to learn without being explicitly programmed to do so. They stated that methods of statistical learning are the backbone of intelligent software used to develop this intelligence. The learning process starts with observations, direct experience data or feedback which is used to search for patterns of data and make better decisions in the future from the obtained past examples. The main goal of machine learning is to automatically enable computers to learn without human intervention and to take appropriate actions accordingly.

2.3. Applications of Machine Learning and Artificial Neural Network

Machine learning has been applied in areas as diverse as the medical field, the stock market and agriculture. In medical diagnosis, machine learning algorithms were used in diagnosing and detecting several diseases such as heart disease and malaria disease. [10] for example, used Naive Bayes to predict the occurrence of heart disease using machine learning algorithms based on attributes that are related to heart disease.

In time series forecasting, two machine learning algorithms, namely Support Vector Machine (SVM) and K-nearest Neighbour (KNN) were used by [11] in predicting rainfall in the coming year. In the financial market analysis area, research to predict closing stock prices using ANN was conducted by a group of researchers [12]. Machine learning was also used in the agriculture area which can be seen in the research done by [13] that used SVM, ANN and Naive Bayes techniques in the prediction of crop yield.

Artificial Neural Network is a type of machine learning algorithm which is modelled on the human brain. ANN models are designed to mimic the way humans learn. The model is made up of a set of artificial neurons that are interconnected with other neurons that depend on the neural network’s weights. The weights are the connections between neurons that will determine the effect of one neuron on another [14]. ANN is a great tool to identify patterns that are too complex for programmers to extract and teach the machine to recognize them automatically.

In many research, ANN has been used in various application areas such as medical diagnosis as well as business and time series forecasting. We can conclude that machine learning algorithms including ANN can be used in many areas due to their ability to improve over time. These algorithms can also provide immediate adaptation without the presence of a human. Thus, these will add intelligence value to the technology or program that uses the machine learning algorithm.

2.4. Similar Application

In predicting the occurrence of air pollution, one of the popular algorithms used is ANN. For example, [5] used two types of artificial neural network models. One of the models was for a short-term prediction that used Feedforward Neural Network (FFNN) and the other one was to model a spatial prediction that used Elman Recurrent Neural Network. In another research on predicting micro particulate matters (PM10), [7] applied MLP algorithm in developing their prediction model. [8] used MLP as one of the machine learning algorithms to develop an air quality prediction model for New Delhi, India. Each research focused on one specific area and several parameters such as main air pollutants, speed of wind and temperature were used.

The researchers also stated that the ANN model can give satisfactory prediction results and better accuracy than other algorithms. For example, the results obtained by [10] indicated that the Feedforward Neural Network model gave more than 70% accuracy rate. Furthermore, the MLP model has proven its great ability in solving complex and nonlinear data. This can be seen from the results of the research done by [7] where the designed MLP model was ready for operational use. Thus, this shows that ANN can be used to forecast air pollution and will be able to produce a good prediction result.
3. Research Methodology

3.1. Project Phases
This research was conducted in six phases namely preliminary study, data collection, system design, implementation, evaluation and documentation. A brief description of each phase is shown in the figure 2 below.

3.2. Analysis Phase
3.2.1. Data Collection. All required data were retrieved and collected from the DOE. The data were on the daily concentration of main air pollutants namely particulate matters under 10µm (PM10), carbon monoxide (CO), sulphur dioxide (SO2), ozone (O₃), nitrogen dioxide (NO₂) and API. Five years of data from 2014 until 2018 were collected from the department for this study.

Since the project was focused on the area of Kuala Terengganu, all data including the API were retrieved from the air quality monitoring station in Kuala Terengganu, located at Sekolah Kebangsaan Chabang Tiga, Kuala Terengganu. Table 1 shows an example of data collected from the DOE.
Table 1. Data from Department of Environment.

| Site Id | Site Location                  | API | SO2  | NO2  | O3   | CO   | PM10 |
|---------|--------------------------------|-----|------|------|------|------|------|
| CA0034  | Sek. Keb. Chabang Tiga, Kuala Terengganu | 54  | 0.003| 0.007| 0.042| 0.63 | 58.0 |
| CA0034  | Sek. Keb. Chabang Tiga, Kuala Terengganu | 44  | 0.004| 0.010| 0.044| 0.63 | 23.0 |
| CA0034  | Sek. Keb. Chabang Tiga, Kuala Terengganu | 49  | 0.004| 0.019| 0.049| 0.81 | 47.0 |
| CA0034  | Sek. Keb. Chabang Tiga, Kuala Terengganu | 53  | 0.007| 0.009| 0.042| 1.08 | 56.0 |

3.2.2. Data Pre-processing. Pre-processing of data was done to improve the representation of the collected data and to get a clean dataset. A clean dataset is important as it can affect the accuracy of the machine learning model. Furthermore, a dataset can affect the ability of the model to learn in the training process. Thus, to develop an ANN engine, correct pre-processing data techniques are needed in order to get quality and clean data for the development of the ANN model. Three pre-processing techniques were used in this project, namely data selection, data inspection and data normalization. All these were carried out by using a pre-processing tool provided in the Waikato Environment for Knowledge Analysis (WEKA).

In the data selection process, required data attributes were selected and unnecessary data attributes such as site id and site name were removed from the dataset. Next, further pre-processing was done by inspecting the dataset for missing data. All missing data were replaced by the mean of each data attributes by using the ‘replace missing values’ filter in WEKA. Then the data were normalized using the ‘normalize’ filter in WEKA. Normalization was done to make sure the range for all predictors is in the common range of 0 to 1 [15], [16]. After data pre-processing is completed, a clean dataset was produced as pictured in table 2. 80% of this dataset was used to train the MLP engine, while the other 20% was set aside for testing and evaluating the model.

Table 2. Clean Dataset.

| SO2       | NO2       | O3       | CO       | PM10   | API  |
|-----------|-----------|----------|----------|--------|------|
| 0.358974  | 0.145773  | 0.395349 | 0.206897 | 0.053586 | 54   |
| 0.487179  | 0.233236  | 0.418605 | 0.206897 | 0.014513 | 44   |
| 0.487179  | 0.495627  | 0.476744 | 0.275862 | 0.041306 | 49   |
| 0.871795  | 0.204082  | 0.395349 | 0.37931  | 0.051353 | 53   |
| 1         | 0.145773  | 0.290698 | 0.206897 | 0.03684  | 43   |
| 0.871795  | 0.204082  | 0.453488 | 0.172414 | 0.023444 | 47   |
| 0.615385  | 0.437318  | 0.465116 | 0.206897 | 0.034607 | 48   |
| 0.487179  | 0.233236  | 0.27907  | 0.275862 | 0.032375 | 39   |
| 0.149009  | 0.204082  | 0.209302 | 0.206897 | 0.020095 | 28   |
| 0.149009  | 0.244186  | 0.103448 | 0.011164 | 29     |

3.3. Design Phase
In the design phase, determining the input and output of the system and designing the user interface was done as part of building the prototype. Next, a multilayer perceptron neural network engine with ten hidden layers was designed for data training and prediction (see Figure 3). From the available data, the
inputs were determined to be the daily concentration of the main pollutants and the output data will be
the predicted air pollution index and the air quality status such as “good”, “moderate”, “unhealthy”,
“very unhealthy” and “hazardous”. The air quality status is set from the standard range of API status as
classified by the DOE, shown in table 3. A user interface as shown in figures 7 and figure 8 was also
designed in this phase.

Table 3. Standard Air Quality Status.

| API   | Status     |
|-------|------------|
| 0-50  | Good       |
| 51-100| Moderate   |
| 101-200| Unhealthy |
| 201-300| Very Unhealthy |
| 301-400| Hazardous |

3.4. Multilayer Perceptron Neural Network Algorithm
A multilayer perceptron (MLP) is a perceptron that is combined with other perceptrons, lined and
arranged in multiple layers which can solve complex and larger problems. An MLP usually consists of
a few layers of nodes, namely the input layer as the first layer, the output layer as the last layer and
hidden layers [17]. The first layer is the layer that receives information, while the last layer is the output
layer where the solution to the problem is produced. The hidden layers meanwhile act as intermediate
layers that divide the input and output layers. In this project, the MLP consists of one input layer, ten
hidden layers and one output layer. Figure 3 shows the architecture of the MLP that has fully-connected
layers.

![Figure 3. Multilayer Perceptron Architecture Used in This Project.](image)

3.5. Implementation Phase
In this phase, several activities were conducted to develop a prototype containing the MLP training
engine for predicting air pollution. The prototype was developed using Python in the Eclipse IDE. All
the input and output data were stored in an excel file for users’ reference. Testing and debugging were
done upon completion of the prototype. In developing the prototype, a personal laptop with 64 bit and
4 GB DDR4 (Double Data Rate 4) memory was used as well as three software, namely Eclipse IDE and Pydev as the code editor and Microsoft Excel for storing data.

3.5.1 Multilayer Perceptron Training Engine. The multilayer perceptron neural network (MLP) engine was developed by using the Keras deep learning library in Python. The Keras library is a high-level API that is written in Python and used for running high-level neural networks [18]. A few steps were needed to build a neural network model using the Keras library [18], [19].

The first step was to define the neural network. In this step, a sequential class which is the container for the layers was defined and created. A Sequential API and the densely-connected layer type were used for creating the structure of the model. The model of this project has twelve densely-connected layers of neurons which consist of one input layer, one output layer and ten hidden layers. This allows the neural network to think widely first before converging to the final prediction. Densely connected layers were used because a multilayer perceptron model must use fully connected layers of neurons. The number of neurons in the input layer and hidden layers was 203 and the dropout rate for each layer was set to 0.0318. The number of inputs or features was specified in the first layer by the `input_dim` attribute which was 5. In the first layer and hidden layers, the Rectified linear (RELU) based activation function was used. These number of hidden layers, neurons per layer, dropout rate and activation function were selected based on an experimental setup described in Section 4.1 as they produced the best score for the mean squared error (MSE). The RELU-based activation function is one of the standard activation functions used today and is defined as \( y = \max(0, x) \) [20]. For the output layer, no activation function was used since this model is a regression model. As we wanted to predict a real-valued or numeric prediction, no activation function was necessary.

The network was compiled after it was defined. The loss function used to evaluate the network and the optimizer used to train the network needed to be specified when compiling the training model. Mean squared error (MSE) loss function and the adaptive moment estimation (Adam) optimization algorithm were used for this model. The Adam optimizer is an efficient stochastic gradient descent algorithm and a widely used optimizer as it can automatically tune itself and produce good results [18]. The learning rate for the Adam optimizer was set to 0.0053. The selection of these two parameters was chosen from an experiment described in the next section.

The third step in developing the MLP engine was to fit the network. Fitting the network means adapting the weights on the training dataset. This is done to ensure that the memory will not be loaded with too many input patterns at the same time. The batch size defines the number of patterns shown by the network before its weight is updated in the epoch, while the epoch defines the number of times the model is exposed to the training dataset. At the start, the number of epochs was set to 50 and the batch size was set to 128 in the `fit()` function. Once fit, the network is ready to be trained.

The next step was to evaluate the network. The performance of the network was evaluated using the test dataset. The MSE value was used to measure the performance of this model. The lower the MSE value, the better the training model. The final step was to make predictions. Once the performance of the model is satisfied, the model can be used to make predictions on new data by using the `predict()` function.

3.5.2 Experiment Setup. An experiment called hyperparameter tuning was conducted to find the best hyperparameters for this model. This is to ensure the best MLP regression model was developed in terms of speed, performance and quality. [18] defined a hyperparameter as a parameter whose value is set before the process of learning starts in training an ANN model. The hyperparameters involved in these experiments were the number of hidden layers, the number of neurons per layer, learning rate, dropout rate, activation function, optimizer and batch size as shown in figure 4.

For this experiment, a total of 25 different trials were conducted. For each trial, the possible combinations of hyperparameter values were evaluated and all output scores of the trials were recorded. The output score was the value of MSE since mean squared error was used as the loss function. The combination of hyperparameter values that produced the best output score among the 25 trials was then
selected and used for this model. The best values of the hyperparameters and the score are shown in figure 5.

![Diagram](image)

**Figure 4.** Details of the Experiment Setup.

![Table](image)

**Figure 5.** The Best Values of Hyperparameters and the Output Score.

3.6. Evaluation Phase
The evaluation was done by conducting functionality tests on the prototype. For measuring the performance of the MLP model, a statistical method, namely the mean squared error method (MSE) was used. The MSE method was used to measure the performance of the model in making predictions since it was widely used by many researchers for regression problems. MSE is calculated as the average of squared differences between the predicted API value produced by the prototype and the actual values of the API in the test dataset. The lower the MSE, the better the performance of the model in making predictions as there would be a good match between the actual and predicted dataset [21].

4. Result

4.1 Training Result
The training process was visualized using a loss graph as shown in figure 6. The figure shows that the training and validation values decreased as the number of epochs increased. This means that the loss value, which is the MSE value has reached the minimum value which is zero as shown in figure 6. The
learning curve of the loss graph produced from the training process is a good fit or an optimal fit. A good fit learning curve is the goal of the learning algorithm and also shows that the performance of the model on both training and validation dataset were good [22]. The plot of both training and validation loss also decreased to a point of stability. Thus, this means the training model could be implemented in the system prototype and could be used to predict air pollution.

Figure 6. Loss Graph of Training Process.

4.2 System Interface
The system interface is developed to make sure the user can easily understand how to use the system. The system prototype consists of two user interfaces. The first interface contains several input fields, namely date, ozone, sulphur dioxide, carbon monoxide, nitrogen dioxide and particulate matter under 10. The user is required to fill in all the input fields and then clicking on the ‘predict button’ to make the prediction. The user can clear the input fields using the ‘clear button’ and close the system prototype using the ‘exit button’. Figure 7 shows the first interface of this prototype.

After a user has clicked the predict button, the results of the prediction are shown in the second interface. The second interface consists of two entries which are prediction result and air quality status. The prediction result is in the form of numerical data as it is the value of the predicted API and the air quality status is based on the category of the predicted API. The date of the prediction is displayed at the top of the interface. The user can return to the first interface by clicking the ‘back button’ and exit the prototype by clicking the ‘exit button’. Figure 8 shows the second interface of this prototype.
4.3 Testing and Evaluation

The MLP model was evaluated using test data collected from the DOE. The MSE value during the model evaluation was 0.0195. The lower the MSE, the better the performance of the model in making predictions. Several different research has indicated that the MSE value that is closer to zero is better [21], [23]. Thus, this proved that the performance of the model developed in this project is good and it can be used to predict the air pollution index.

Besides using MSE, predictive and actual results were also compared. The prediction results stored in the excel file were compared with the actual result to determine the accuracy and effectiveness of the system prototype in making the prediction. For this, 140 sets of data were used to evaluate the system prototype. The correct predictions were summed and multiplied by 100 to calculate the percentage of accuracy. The formula used to calculate this is shown in equation (1) below:

\[
\frac{\text{(Correct predictions)}}{\text{(Total predictions)}} \times 100 = 79\%
\]

From this calculation, the accuracy of the system prototype was determined to be 79% where 110 out of 140 predictions were the same as the actual air quality status. This thus indicated that this system prototype was able to successfully predict the air pollution in Kuala Terengganu.

Functionality tests were also done to make sure the system prototype could run properly and was able to make a prediction on air pollution from the data inserted into the system. Several users are selected to test the system and evaluate the system based on their preferences. Several criteria have been selected as the aspects of evaluation based on several observations on a good existing system. The results of the functionality test are summarized in table 4.
Table 4. Evaluation Aspects in Functionality Test.

| Aspect | Dissatisfied | Neutral | Satisfied |
|--------|--------------|---------|-----------|
| Users can input the data. | ✓ | | |
| Users can click the buttons. | ✓ | ✓ | |
| Users can understand the function of each button. | ✓ | ✓ | |
| The prototype gives the desired output. | ✓ | ✓ | |
| The input and output data can be saved into an excel file. | ✓ | ✓ | |
| User-friendliness. | ✓ | ✓ | |
| Prototype efficiency. | ✓ | ✓ | |

5. Conclusion

In conclusion, all three objectives have been achieved after all activities were completed. All the stated methodologies have been implemented in this project. This project has proven that the selected algorithm which is ANN, can be used to solve the problems and is able to give a reasonable prediction result when predicting the air pollution in Kuala Terengganu. This shows that the first objective which is to study algorithms that can be used to forecast air pollution in Kuala Terengganu has been achieved. A system prototype that implemented the ANN algorithm as the engine has also been successfully developed. Hence, the second objective which is to develop an artificial neural network system model that can predict air pollution in Kuala Terengganu was also fulfilled. Evaluation and testing have been done to this system prototype to prove its functionality in predicting the air pollution index. From the evaluation and testing, this system prototype is capable of giving the desired prediction result. Thus, the third objective which is to evaluate the functionality of the system prototype has been achieved. As an overall conclusion, this prototype can be used to forecast air pollution in the city of Kuala Terengganu and is able to give a satisfactory result.

Acknowledgment

Thank you to the Department of Environment Malaysia (DOE) for providing the data for this project.

References

[1] Manisalidis I, Stavropoulou E, Stavropoulos A and Bezirtzoglou E 2020 Environmental and Health Impacts of Air Pollution: A Review Front. public Heal. 8 p 14.
[2] Usmani R S A, Saeed A, Abdullahi A M, Pillai T R, Jhanjhi N Z and Hashem I A T 2020 Air pollution and its health impacts in Malaysia: a review Air Qual. Atmos. Heal. 13(9) p 1093-1118.
[3] Mohd Pu’Ad M F, Gunawan T S, Kartiwi M and Janin Z 2019 Development of Air Quality Measurement System using Raspberry Pi 2018 IEEE 5th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA (Thailand) p 28-30.
[4] Bai L, Wang J, Ma X and Lu H, 2018 Air pollution forecasts: An overview Int. J. Environ. Res. Public Health 15(4) p 1-44.
[5] Rahman P A, Panchenko A A and Safarov A M 2017 Using neural networks for prediction of air pollution index in industrial city IOP Conf. Ser. Earth Environ. Sci. 87 4.
[6] Bui T-C, Le V-D and Cha S K 2018 A Deep Learning Approach for Forecasting Air Pollution in South Korea Using LSTM.
[7] Abdullah S, Ismail M and Ahmed A, 2019 Multi-Layer Perceptron Model for Air Quality Prediction Malaysian J. Math. Sci. 8(1) p 15-34.
[8] Srivastava C, Singh S and Singh A P 2019 Estimation of air pollution in Delhi using machine learning techniques 2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018 (India)
[9] Mohammed M, Khan M B and Bashier E B M 2016 *Algorithms and Applications* Boca Raton: Taylor & Francis Group LLC.

[10] N R, Maneesha T, Hafeez S and Krishna H 2018 Prediction of heart disease risk using machine learning *Int. J. Mech. Prod. Eng. Res. Dev.* 8(Special Issue 2) p 605-614.

[11] Garg A and Pandey H, 2019 Rainfall prediction using machine learning techniques *Int. J. Innov. Technol. Explor. Eng.* 8(7) p 957-963.

[12] Vijh M, Chandola D, Tikkiwal V A and Kumar A 2020 Stock Closing Price Prediction using Machine Learning Techniques *Procedia Comput. Sci.* 167 p 599-606.

[13] Palanivel K and Surianarayanan C 2019 An Approach for Prediction of Crop Yield Using Machine Learning and Big Data Techniques *Int. J. Comput. Eng. Technol.* 10(3) p 110-118.

[14] Han S-H, Kim K W, Kim S and Youn Y C 2018 Artificial Neural Network: Understanding the Basic Concepts without Mathematics *Dement. Neurocognitive Disord.* 17(3) p 83.

[15] Mclean S, Kaiser J and Richard B 2019 Environmental Modelling & Software A review of artificial neural network models for ambient air pollution prediction *Environ. Model. Softw.* 119 p 285–304.

[16] Brownlee J, 2020, How to use Data Scaling Improve Deep Learning, *Machine Learning Mastery*. [Online]. Available: https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/. [Accessed: 22-Feb-2021].

[17] Little Z, 2020 Multi-Layer Layer Perceptron (MLP) *Medium*. [Online]. Available: https://xzz201920.medium.com/multi-layer-perceptron-mlp-4e5c020fd28a. [Accessed: 22-Feb-2021].

[18] Brownlee J 2020 Your First Deep Learning Project in Python with Keras Step-By-Step, *Machine Learning Mastery*. [Online]. Available: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/. [Accessed: 18-May-2020].

[19] Brownlee J 2020 5 Step Life-Cycle for Neural Network Models in Keras - Machine Learning Mastery, *Machine Learning Mastery* [Online]. Available: https://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/. [Accessed: 18-May-2020].

[20] Liu D, 2017, A Practical Guide to ReLU, *Medium*. [Online]. Available: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f17. [Accessed: 11-Jul-2020].

[21] Kazumali E Kr. Biswas S and W Kelsey T 2016 What is the Acceptable MSE value and Coefficient of determination(R2)? *Research Gate*. [Online]. Available: https://www.researchgate.net/post/What_is_the_Acceptable_MSE_value_and_Coefficient_of_determinationR2. [Accessed: 11-Jul-2020].

[22] Brownlee J 2019 How to use Learning Curves to Diagnose Machine Learning Model Performance, *Machine Learning Mastery*. [Online]. Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/. [Accessed: 22-Feb-2021].

[23] Taghavifar H and Mardani A 2014 Application of artificial neural networks for the prediction of traction performance parameters *J. Saudi Soc. Agric. Sci.* 13(1) p 35-43.