

# Performance Analysis of Multilayer Perceptron Neural Network Models in Week-Ahead Rainfall Forecasting

Lemuel Clark P. Velasco<sup>1</sup>, Ruth P. Serquiña<sup>2</sup>, Mohammad Shahin A. Abdul Zamad<sup>3</sup>, Bryan F. Juanico<sup>4</sup>,  
Junneil C. Lomocso<sup>5</sup>

Mindanao State University-Iligan Institute of Technology  
Premier Research Institute of Science and Mathematics  
Iligan City, The Philippines

**Abstract**—Multilayer perceptron neural network (MLPNN) is considered as one of the most efficient forecasting techniques which can be implemented for the prediction of weather occurrence. As with any machine learning implementation, the challenge on the utilization of MLPNN in rainfall forecasting lies in the development and evaluation of MLPNN models which delivers optimal forecasting performance. This research conducted performance analysis of MLPNN models through data preparation, model designing, and model evaluation in order to determine which parameters are the best-fit configurations for MLPNN model implementation in rainfall forecasting. During rainfall data preparation, imputation process and spatial correlation evaluation of weather variables from various weather stations showed that the geographical location of the chosen weather stations did not have a direct correlation between stations with respect to rainfall behavior leading to the decision of utilizing the weather station having the most complete weather data to be fed in the MLPNN. By conducting performance analysis of MLPNN models with different combinations of training algorithms, activation functions, learning rate, and momentum, it was found out that MLPNN model having 100 hidden neurons with Scaled Conjugate Gradient training algorithm and Sigmoid activation function delivered the lowest RMSE of 0.031537 while another MLPNN model having the same number of hidden neurons, the same activation function but Resilient Propagation as training algorithm had the lowest MAE of 0.0209. The results of this research showed that performance analysis of MLPNN models is a crucial process in model implementation of MLPNN for week-ahead rainfall forecasting.

**Keywords**—Multilayer perceptron neural network; performance analysis; rainfall forecasting

## I. INTRODUCTION

Multilayer perceptron neural network (MLPNN) is considered as a widely used artificial neural networks architecture in predictive analytics functions. The architecture of an artificial neural network, that is, its structure and type of network is one of the most important choices concerning the implementation of neural networks as forecasting tools. The design of MLPNN is motivated by the structure of a biological neuron system capable of parallel processing like a human brain, but the processing elements of this machine learning tool has gone far from their biological inspiration [1, 2, 3]. For this

reason, MLPNN have been successfully used by most of the researchers in the field of forecasting, science and engineering to predict the behavior of both linear and nonlinear systems without the need to make assumptions that are implicit in most traditional statistical approaches [2, 4, 5, 6]. With all its promising results, the biggest challenge with MLPNN is the selection of an appropriate model since there are different MLPNN model structures, training algorithms, activation functions, learning rate, momentum and number of epochs to choose from [1, 7]. This makes it hard to find the proper model for a particular problem [4]. Modelers and researchers who use MLPNN in forecasting still rely on performance analysis of MLPNN models in order to implement domain-specific applications that generate close to accurate predictions.

The field of rainfall forecasting is one of the domains that utilize MLPNN in generating predictions of various granularities [1, 3, 5]. Rainfall is the metric used to measure the amount of rain that accumulates at any given point in the earth's surface. This measurement is usually reported in millimeters and is most often associated with its more violent counterpart which is flooding. Out of the historical data collected from various rain gauges, MLPNN models show great potential in discovering patterns from preprocessed data which in turn forecast rainfall used for life-saving applications such as flood management and airport administration. The nature of the combination of meteorological parameters such as relative humidity, air pressure, wet bulb temperature, cloudiness, and rainfall at the point of measurement as well as from surrounding stations poses challenges in data preparation as well as in the input and hidden layers of MLPNN models [7, 8]. Furthermore, in the output layers of MLPNN applications for rainfall forecasting, modelers usually generate week-ahead forecast to give ample time for decision makers in the dissemination of disaster preparedness measures to the affected stakeholders [3]. With the consideration of the continuous data gathered from rainy and non-rainy periods, data representation, data cleaning, correlation evaluation and data transformation are also modeling challenges that need to be considered before using any MLPNN model as a supervised learning framework in the forecast of life-saving predictions [1, 5, 8]. With this, performance analysis of MLPNN models that takes into consideration appropriate data preparation which optimizes

various neural networks parameters is an important function in week-ahead rainfall predictions.

The choice of a dataset and the quality of its data is a defining factor in the accuracy of the MLPNN model's output. Data quality for the dataset has to be maintained else the prediction process of the MPNN and its testing will potentially suffer from anomalies and inconsistencies [9, 10]. In addition, the selection of the study area, time span of the data, and important variables in the dataset must be conducted in order to produce the best possible case scenario for the research problem. Furthermore, without the proper representation of MLPNN model results, discussions, and error assessment, rainfall forecasting will fail to capture the validity of its output and leave these implementation efforts vulnerable to misinterpretation [2, 8]. Thus, the choice of the MLPNN model, multiple runs of data pre-processing, model construction, and the analysis and presentation of MLPNN model performance are all required to present a working solution to the prediction of rainfall and other weather phenomena. This research aims to focus on evaluating the performance of MLPNN models in choosing a suitable candidate for implementation in week-ahead rainfall forecasting. Specifically, this study exhibits foundational methodologies in MLPNN model design creation which involves data preparation procedures and decisions on the parameter values to be implemented. The results of this study can provide methods on testing the validity and accuracy of MLPNN models as well as comparing and measuring the performance of its various forecasting parameters. This study hopes to contribute to the recent technology of rainfall forecasting by evaluating MLPNN models which can be used to optimally implement close to accurate predictions that provide accurate rainfall forecast to specific localities.

## II. METHODOLOGY

### A. Rainfall Data Preparation

Data preparation involves the exploration, analysis, and other general pre-processing methods and techniques that must be performed before data is fed to the MLPNN model. Initially, data selection and data representation which are the processes of choosing the appropriate dataset and the representation of

key variables to be considered as well as the transformation of non-numeric variables into numerical representations need to be followed by testing these variables for correlation with rainfall and spatial autocorrelation along with other geographic locations [9, 11, 12]. Weather data gathered from Tutiempo Network S.L. of seven weather stations in Mindanao, the Philippines was considered due to the geographical surface area and proximity within the path of a number of storms and typhoons. Moreover, the weather dataset shown in Table I was segregated into multiple years for each of the seven stations segregated by their month with each month constituting of daily recorded observations. The 12-year weather data from 2006-2017 from the seven weather stations totaling to 398,853 units of data underwent data preparation. Additionally, ISO 8601 standard for dates, yyyy/mm/dd was also used to represent the dates corresponding the weather data.

Missing data is a type of data anomaly in weather and climate data that occurs when measuring instruments fail, leaving behind gaps in the dataset. The percentage of missing data in the dataset was computed in order to determine how much data was missing. It is important to determine the percentage of missing data in a dataset because it can cause significant prediction error when data is not uniform [11]. It is an important calculation to make because without understanding the scale of missing data, it would be difficult to gauge how much the imputation process will affect overall accuracy. The larger the amount of missing data, the larger amount of values that have to be filled in by the imputation process, thus lesser missing data implies better overall accuracy. Aside from determining the total of missing data per set, the missing data per climate variable is also an important metric to determine. It needs to be accounted for due to later steps involving individual variables being used for correlation measurements. A variable that has a large number of missing data will also affect the computation of the Pearson's correlation coefficient to be conducted in the study. Random Forests Imputation method was then used to fill in the identified missing data. The Random Forests Imputation method is an ensemble learning method for classification and regression which uses multiple decision trees and outputs either the mode for classification problems or the mean prediction for regression problems of the individual trees.

TABLE I. WEATHER DATA VARIABLES

SYMBOL	CLIMATE FEATURE	UNIT	SYMBOL	CLIMATE FEATURE	UNIT
T	Average Temperature	°C	V	Average Wind Speed	km/h
TM	Maximum Temperature	°C	VM	Maximum Sustained Wind Speed	km/h
Tm	Minimum Temperature	°C	VG	Maximum speed of wind	km/h
SLP	Atmospheric Pressure at Sea Level	hPa	RA	Indicate whether there was rain or drizzle	0 or null
H	Average Relative Humidity	%	SN	Indicate if it snowed	0 or null
PP	Total Rainfall/ Snowmelt	mm	TS	Indicate whether there was a storm	0 or null
VV	Average Visibility	km	FG	Indicate whether there was fog	0 or null

Variable correlation evaluation was then conducted to determine which variables in the weather dataset are correlated against the target variable which is rainfall. As suggested by researches, to confirm which of the variables in the dataset fit the criteria, the Pearson's Correlation Formula was used to determine the correlation strength of each climatological variable with regards to rainfall [13]. Results range between  $[-1, +1]$ , indicating weak to strong correlation with values close to 0 indicating no correlation. For this study, a 95% confidence interval was used with a considered  $p$ -value of less than or equal to 0.005. As shown in Equation 1, the Pearson's Correlation Formula was run multiple times for each rainfall combination per weather station. The Pearson's Correlation Coefficient  $r$  provided values of two different variables  $x_i$  and  $y_i$  of equal cardinality  $n$  where  $\bar{x}$  and  $\bar{y}$  are the means of the two variables respectively.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

As applied in this study,  $x_i$  was used to denote rainfall values and  $y_i$  was used to denote values for one other climate variable aside from rainfall like average temperature and humidity.

In order to increase the predictive power and include more data for training the MLPNN, the identification of clusters was conducted in the entire geographic area using Spatial Autocorrelation. This will result in the identification of a base weather station along with other stations in the initial study area that has rainfall values spatially correlated with one another. The data present in the base station and the identified stations that exhibit correlation was included as inputs for the MLPNN to predict the rainfall values of the base station. To test for particular locations that exhibit local spatial autocorrelation of their values, Local Moran's  $I$ , an extension of the Pearson's Correlation formula with the addition of a spatial weights matrix which represents the weight given to the distance between points in space was used. A 95% confidence interval was used, meaning that a  $p$ -value of less than or equal to 0.005 was considered. The Local Moran's  $I$  for location  $i$  provided two locations in space  $i$  and  $j$  as shown in Equation 2, where  $z_i$  and  $z_j$  are the deviations from the mean at both locations,  $S_i^2$  being the standard deviation at location  $i$  and  $\omega_{ij}$  being a spatial weights matrix.

$$I_i = \frac{z_i}{S_i^2} \sum_j w_{i,j} z_j \quad (2)$$

As applied in this study,  $z_i$  and  $z_j$  are deviations from the mean at weather stations  $i$  and  $j$ . To acquire a spatial weights matrix, a list of neighbors was needed. The  $k$ -nearest neighbor algorithm was used to generate a list of neighbors. The algorithm returns a list of neighbors that correspond to the number  $k$  attached to each weather station. The local Moran formula was applied in different stages, at each stage increasing the number of neighbors for each weather station. After the list of neighbors was acquired, a row standardized weight matrix was calculated from it. Since the list of neighbors differs at every iteration  $k$ , a different weight matrix was generated at every step. Once a spatial weight matrix has been generated, the local Moran formula was then calculated.

Since the spatial weights matrix differs at every iteration of  $k$ , a different set of Moran indices were calculated. After transforming the dataset using Min-Max Normalization, the dataset was then partitioned into different sets namely the Training Set and the Testing Set. The dataset was partitioned according to the number of years instead of percentages as suggested by researches on rainfall forecasting conducted in tropical counties [11, 12].

### B. MLPNN Model Evaluation

The MLPNN architecture and model define the structure of the neural network which includes the number of layers, the direction of data flow in each layer, number of neurons per layer, and how these neurons are arranged. The neurons comprising the input layer is completely and uniquely determined once the specifications of the training data have been identified with the number of neurons comprising the input layer to be equal to the number of features in the data set. According to researches, the three most significant data inputs in rainfall prediction aside from the actual daily rainfall or precipitation values are relative humidity, air pressure and average temperature as these core elements constitutes the formation of rain or storm [4, 14]. Temperature affects the evaporation process causing increase in humidity while pressure affects the flow of air carrying these two. In order to predict rainfall with a high level of accuracy, these three parameters should be used. But since this research also considers the correlation between stations and its variables, other input data such as wind speed and visibility will also be tested with the aim to find out if the results yields an acceptable correlation evaluation, then the variable will be included as an input. The next matter to be resolved following the identification of the input layer is the number of hidden layers to be used along with its hidden neuron. According to the studies, a single hidden layer of a MLPNN is sufficient enough to approximate any complex nonlinear function with any desired accuracy [2, 4, 10]. As for the number of neurons in the hidden layer, the formula shown in Equation 3 as suggested by a study would give an upper bound limit of values that will not result in over fitting [15]. Stathakis' formula uses an arbitrary scaling factor from 2-10 that is multiplied by the sum of the total input plus total output in order to gradually decrease the value of the number of neurons as the arbitrary factor reaches to 10.

$$Nh = \frac{Ns}{(\alpha * (Ni + No))} \quad (3)$$

In this formula,  $Nh$  is the total number of hidden neurons to be calculated. This is done by dividing the total number of samples in the training data set  $Ns$  with the product of the arbitrary scaling factor  $\alpha$  multiplied to the sum of the total input  $Ni$  with the addition of the total number of output  $No$ . The result will be tested individually as the number of input neurons in the hidden layer decreases. Along with other MLPNN parameters, models that reach the local minima with the lowest MAE and RMSE will be selected as the optimal number of hidden neurons. Since this study aims to predict rainfall data on a weekly basis, the number of output neurons will correspond to the requirement which is to produce 7 prediction outputs corresponding to 7 days as represented by the 7 output neurons.

This work is supported by the Mindanao State University-Iligan Institute of Technology (MSU-IIT) as an internally funded research under the Premier Research Institute of Science and Mathematics (PRISM) Applied Mathematics and Statistics Group.

Training algorithms, activation functions, learning rates and the momentum are important MLPNN model parameters that should be identified. The MLPNN training algorithm is the parameter that tunes the network so that its outputs are close to the desired values [16]. In choosing the training algorithm, several factors have to be considered including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal and whether the network is being used for pattern recognition or function approximation [5, 16]. Since this study is dealing with rainfall forecasting which relies heavily on statistical calculations given historical data in order to get future values, the researchers focused on the function approximation algorithms. Function approximation algorithm shown in Equation 4 allowed researchers to find ways of separating objects into different classification given an input vector  $x$ , a weight vector  $w$ , and a threshold value  $T$ , an output of 1 indicating membership of a classification, consequently an output of 0 indicating exclusion from the class [17]. With this, a select function approximation algorithm will be used for the training algorithms.

$$\sum_i w_i (x_i) > T \quad (4)$$

The activation function indicates the output of a neuron in terms of its input. Activation functions are important in order for the MLPNN to learn and make sense of complicated and non-linear complex functional mappings between inputs and response variable [6, 12]. Its main purpose is to convert the input signal of a node to an output signal which will be used as input in the next layer. There are a number of activation functions that can be used such as Sigmoid, Threshold and Linear activation functions. Among MLPNN implementations, the activation functions often chosen for rainfall forecasting are the logistic sigmoid and hyperbolic tangent [2, 3, 8, 16]. These functions are used because they are mathematically convenient and are close to linear near origin while saturating rather quickly when getting away from the origin allowing MLPNN to model strongly and mildly nonlinear mappings.

As for the learning rate  $\eta$ , which determines how fast weights changes in order to reach local minimum, the goal is to find a value low enough that the network converges to an acceptable result but high enough that the network do not have to spend years just training. Some studies in rainfall prediction use 0.8 as the default value for learning rate [6, 18, 19]. There can be a situation in the MLPNN model where the algorithm converges to a local minimum or saddle point and may think it reached the global minima leading to a sub-optimal result. Momentum is used to avoid this situation though a value between 0 and 1 that increases the size of the steps taken towards the minimum by trying to jump from local minima. If the momentum is large, then the learning rate should be kept small. A large value of momentum also means that the convergence will happen fast. But if both the momentum and learning rate are kept at large values, it might skip the minimum with a huge step, or else momentum cannot reliably avoid local minima and slows down training of the system.

Momentum also helps in smoothing out the variations, if the gradient keeps changing direction. A right value of momentum can be either learned by trial and error within 0.1 and 0.9 as suggested in a research or through cross-validation [5]. Thus, this study simulated different combinations of training algorithms and activation functions along with a range of values for momentum and learning rate in formulating the MLPNN models.

After identifying the model architecture and formulating different models, the researchers conducted a supervised training process of each model by feeding the training data set into the MLPNN. Training is an essential step in order for the MLPNN models to do forecasting [10]. It is during this process that the MLPNN adapts itself to a stimulus and eventually produces a desired response. In conducting the supervised training, the training data set already underwent data preparation in which it was imputed to fill the missing values, correlation evaluated to remove variables that have no significant influence in rainfall, normalization to normalize dataset into (0, 1). When feeding the training data set into the MLPNN model, an ideal or desired output was introduced along with the input stimulus. Then the response is compared with the desired output and if response differs from the desired value, the network generates an error signal, which was used to calculate the adjustment that should be made to the network's synaptic weights so that the actual output matches the target output possibly getting an error close to zero. In order to test the accuracy of the trained models, testing was conducted. Testing results was used to compute the MAE and RMSE for the error measurement in order to identify the optimal model. To properly compute the MAE and RMSE, the researchers group the data by week from Day 1-7 i.e. January 1 to January 7 as first week then increment the starting day of the next week by 1 each time. So that the second week starts at Day 2-8 i.e. January 2 to January 8, so on and so forth. This process continues until the whole result has been grouped by week. MAE and RMSE were then calculated per week. Once it was done, the average of all values obtained was calculated and recorded. These steps were repeated for all formulated models. The model that produced the smallest MAE and RMSE error will be chosen as the optimal MLPNN model of the performance analysis.

### III. RESULTS AND DISCUSSION

#### A. Rainfall Data Preparation Results

There were variables in the dataset that were found to be variables that have no bearing in the prediction of rainfall, as they merely indicate the occurrence of different weather phenomena. These variables were RA, SN, TG, and FG; these variables along with the columns they represent were removed from the dataset. The percentage of missing data in the dataset was then calculated in order to better understand the amount of information lost during the recording of the data. Table II shows the amount of missing data present per weather station and its percentage when compared to the total amount of data units.

TABLE II. MISSING DATA PER WEATHER STATION

Station	Missing data	Total data	%of missing data	Station	Missing data	Total data	%of missing data
Butuan	442	39789	1.11%	Malaybalay	2741	39789	6.89%
Davao	216	39789	0.54%	Surigao	974	39789	2.45%
Dipolog	359	39789	0.90%	Zamboanga	252	39789	0.63%
Hinatuan	601	39789	1.51%				

The three stations that exhibit the least amount of missing data are Davao Airport, Zamboanga, and Dipolog with 0.54%, 0.63%, and 0.90% respectively. These stations are the prime candidates to use as the base station due to the missing values being brought down to the minimum, ensuring that the accuracy of the dataset is true to the real world and not artificially filled in through imputation. Stations with the most missing data are Malaybalay with 6.89%, Surigao with 2.45%, and Hinatuan with 1.51%. Tables III and IV shows the state of the dataset during pre-imputation and post-imputation for a chosen weather station for the first 3 days of January 2006, respectively. This research requires the usage of an imputation technique due to succeeding methodologies requiring a complete set of data. Correlation formulas need as many existing data as possible in order to determine an accurate correlation measure. Removing the missing data while possible results in information loss; in some stations the information loss will be severe like Malaybalay. Furthermore, by not imputing the missing data the research loses out on predictive power when developing the MLPNN. This is an important factor to consider since without much data, the forecasting accuracy will be severely affected.

For each dataset, the rainfall variable and another variable in the same set were tested using Pearson’s Product-Moment Correlation Test. After the Pearson’s Correlation test was performed on the dataset, the results were given in pairs of two, the first element being the value of the coefficient, and the second being the *p*-value of the coefficient. This is an important step because more data was needed to include in the MLPNN and other climate variables are the best indicator for correlation with rainfall. Furthermore, since rainfall is the target climate variable to be forecasted, other climate variables are bound to influence the amount of, frequency, and severity of rainfall. Thus, correlation between variables and rainfall was calculated. It is important to recall that the study will be using a 95% confidence interval, so *p*-values less than or equal to 0.005 will be considered. Among the seven sets of data, Zamboanga station has the most variables correlated with rainfall being 8 and the least amount of variables being correlated is Hinatuan with 5. Dipolog and Malaybalay stations have 6 correlated variables, while Davao Airport and Surigao has 7 correlated variables. Fig. 1 and 2 graphically shows the Pearson’s *r* and their *p*-values, respectively.

Once the variables correlated with rainfall were determined, spatial autocorrelation was measured between stations in close proximity with each other using Local Moran’s *I*. The list of neighbors was acquired by using the *k*-

nearest neighbor algorithm. Each *k* indicates the number of neighbors attached to a weather station, so for example *k*=2 means that there are two weather stations attached to every station in the study area and *k*=5 means that there are five weather stations attached to every station. This step was conducted to determine potential clusters in the study area for initial consideration. Without determining potential clusters, the autocorrelation measurement can no longer be called a Local Indicator of Spatial Autocorrelation which cannot be used for the scope of this study. The results of the process are shown in Table V where each column marked by *k* represents the number of neighbors attached to a particular station. Davao’s closest neighbor would be Malaybalay at *k*=1, at *k*=2 there will be two stations attached as neighbors: Hinatuan and Malaybalay. This process repeats for all seven weather stations.

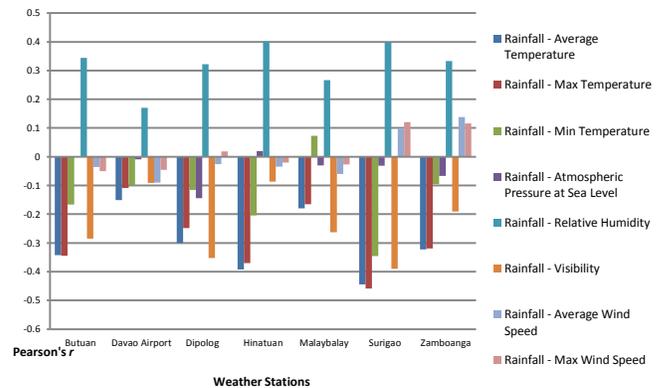


Fig. 1. Graph of All Collected Pearson’s *r*.

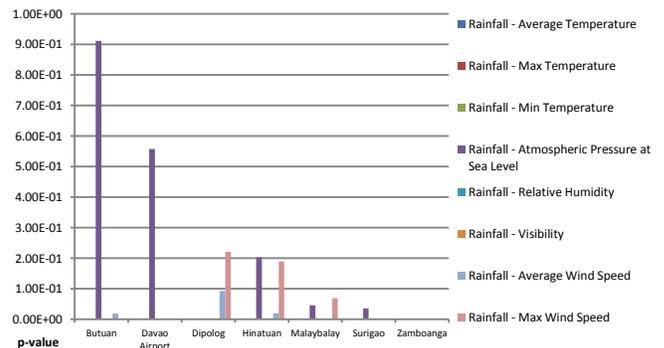


Fig. 2. Graph of All Collected *p*-Values.

After the list of neighbors was acquired, a row standardized weight matrix was calculated from it. A row standardized weight matrix is a matrix whose values represent the numerical weight the algorithm gives to emphasize the importance of the distance between two neighboring points in space. The higher the number, the more weight is given to the distance between locations. This weight matrix was needed to calculate the local Moran index for every identified potential cluster under consideration. Table VI shows the results of that process at  $k=1$ , where  $k$  is the number of neighbors attached to a weather station.

After the weights matrix was calculated, the required parameters of the local Moran formula were now met. The

process was iterated for every value of  $k$ , increasing the number of neighbors. A different set of Moran indices and  $p$ -values were calculated at every iteration. It was observed that due to extreme values of rainfall, there were cases when the Moran index returns a Not a Number (NaN). These situations were encountered when there exists a day where the rainfall values across all considered weather stations were 0. In this situation, the value was converted to a 0. Tables VII and VIII details a sample table of the local Moran indexes per day at  $k=1$  as well as the  $p$ -values associated with them, respectively. These results are important because they determine whether or not the Moran indices throughout time are uniform and consistent, as well as determine if these Moran indices are significant at the accepted 95% confidence interval.

TABLE III. PRE-IMPUTATION DATASET OF THE WEATHER STATION

Y	M	D	T	TM	Tm	SLP	H	PP	VV	V	VM
2006	1	1	28.1	32.3	23.5	1008.8	87	0	15.9	3.5	10.7
2006	1	2	27.6	28.8	25.7	1009.2	86	9.91	12.6	4.4	7.2
2006	1	3									

TABLE IV. POST-IMPUTATION DATASET OF THE WEATHER STATION

Y	M	D	T	TM	Tm	SLP	H	PP	VV	V	VM
2006	1	1	28	32	24	1009	87	0	16	3.5	11
2006	1	2	28	29	26	1009	86	9.9	13	4.4	7.2
2006	1	3	27	30	24	1010	87	11	14	3	9.9

TABLE V. RESULT OF THE K-NEAREST NEIGHBOR ALGORITHM

Station	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
Butuan	Surigao	Malaybalay	Hinatuan	Davao	Dipolog	Zamboanga
Davao	Malaybalay	Hinatuan	Butuan	Surigao	Dipolog	Zamboanga
Dipolog	Malaybalay	Zamboanga	Butuan	Surigao	Davao	Hinatuan
Hinatuan	Butuan	Malaybalay	Davao	Surigao	Dipolog	Zamboanga
Malaybalay	Butuan	Davao	Hinatuan	Surigao	Dipolog	Zamboanga
Surigao	Butuan	Hinatuan	Malaybalay	Dipolog	Davao	Zamboanga
Zamboanga	Dipolog	Malaybalay	Davao	Butuan	Surigao	Hinatuan

TABLE VI. RESULT OF THE SPATIAL WEIGHTS MATRIX AT  $K=1$

	Butuan	Davao	Dipolog	Hinatuan	Malaybalay	Surigao	Zamboanga
Butuan	0.000000	0.000000	0.000000	0.3333333	0.3333333	0.3333333	0.000000
Davao	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
Dipolog	0.000000	0.000000	0.000000	0.000000	0.500000	0.000000	0.500000
Hinatuan	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Malaybalay	0.3333333	0.3333333	0.3333333	0.000000	0.000000	0.000000	0.000000
Surigao	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Zamboanga	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000

TABLE VII. SAMPLE TABLE OF COLLECTED LOCAL MORAN INDEXES AT  $K=1$  OF A CHOSEN WEATHER STATION

Station	Day1	Day2	Day3	Day4	Day5	Day6	Day7
Butuan	-0.23741	0.11511024	-0.5840033	-0.4812947	0.34175831	-0.10306089	-0.0260774
Davao	0.1613777	0.41525523	0.381054	0.5636944	-0.03057254	-0.41368012	0.59088377
Dipolog	0.1359339	0.06561247	0.3668329	0.3555742	0.38138875	0.42227707	0.51195061
Hinatuan	-1.1727723	0.78774914	-0.9067904	-1.0368087	1.13056531	0.08142846	-0.04060727
Malaybalay	0.1839798	0.08734511	0.3770831	0.4908609	-0.01572634	0.09587238	0.37478806
Surigao	0.2302712	-0.22120921	-1.2262734	-0.8615195	-0.12905408	-0.63906866	-0.06406317
Zamboanga	0.1115774	0.06323564	0.3645246	0.2567042	0.80314769	0.39171443	0.51685907

TABLE VIII. SAMPLE TABLE OF COLLECTED P VALUES AT K=1 OF A CHOSEN WEATHER STATION

Station	Day1	Day2	Day3	Day4	Day5	Day6	Day7
Malaybalay	0.1815233	0.26302077	0.1015762	0.06259734	0.35697344	0.2583876	0.1023129
Surigao	0.2284913	0.53194755	0.8828206	0.77941166	0.48061866	0.7447528	0.4540085
Dipolog	0.2394622	0.31630871	0.1784729	0.18571222	0.14882041	0.1202729	0.1200828
Davao	0.2693683	0.19619077	0.2693726	0.20909464	0.43020454	0.634613	0.1968113
Zamboanga	0.3010429	0.3677157	0.275534	0.31943323	0.10507711	0.2183319	0.2207365
Butuan	0.5728002	0.24091351	0.8356553	0.76843441	0.10846802	0.4375946	0.3709408
Hinatuan	0.9703086	0.08033877	0.7969135	0.83260397	0.04684826	0.3648206	0.4435583

Each of the rows in the table correspond to each weather station, each of the columns represent the days in the time frame, and each data cell the Moran index associated with the day and station. As can be observed, some values do not conform to the typical range for the Local Moran's Index formula, which is  $-1$  to  $+1$ . A researcher has already established that the Local Moran Index formula does not actually have a set range of  $(-1, +1)$  [20]. Moreover, the exact range of indices actually conforms to the smallest and largest eigenvalue of among  $n-1$  eigenvalues of the weights matrix  $W$ . So this means that depending on the spatial weights matrix generated, the values for the indices will differ and might not conform to the usual standard range. The spatial weights matrix is more effective when the locations in question are close to each other, and thus have more weight established between them. Furthermore, the results differ at every day with values indicating a positive relationship other a negative relationship indicating a sporadic pattern.

As shown, the collected  $p$ -values for each Local Moran index of each weather station across all the days of the time frame, 2006-2017. Each of the rows in the table correspond to each weather station, each of the columns represent the days in the time frame and each data cell the  $p$ -value of the Moran index that correspond to the day and weather station. The  $p$ -values are all above 0.005, the maximum requirement for a value to be considered significant at a 95% confidence interval. This indicates that the calculated Moran indices are not considered to be significant for study which creates a problem. From the generated values, a line graph was drawn up to show the variation of each Moran Index value per day. The same process can also be generated for the  $p$ -values per day. The line graphs are for the Moran Indexes and  $p$ -values taken at  $k=1$  for a chosen weather station across the entire time span ranging from 2006 – 2017, respectively. Fig. 3 shows that for all the days in the duration 2006 – 2017, the calculated Moran indices for each day come out to an interval between 1 and  $-1.5$ . Although it does not conform to the range, it still indicates whether or not particular locations have correspondence. However, Fig. 4 further shows that the Moran indices do not exhibit uniform and consistent values through time. This means that these Moran indices are highly variable and differ at points in time, making these values sporadic and difficult to predict. As shown, none of the Moran indices calculated are within a 95% confidence interval. This means that none of these indices are significant and cannot be used as indicators for spatial autocorrelation.

Upon further calculation of Local Moran indexes and their  $p$ -values with increasing  $k$ , the number of neighbors attached, it was found out that any and all settings result to the same pattern. The pattern being indices not conforming to the standard range, and the  $p$ -values being greater than the 95% confidence interval will allow. This means that according to the formula, none of the locations in the geographic space have correlation with regards to their rainfall values, regardless of the number of neighbors attached. This may be due to factors that cannot be controlled, such as the topography between each station or the distance between locations. The shorter the distance the greater would be the correspondence, however as shown, it will seem that the distance between stations are too much to determine an accurate measure of relationship. Furthermore, the topography of Mindanao, the Philippines consists of flat plains and mountain ranges, and other types of topography which directly influences the behavior of rain clouds or storms as they approach each station.

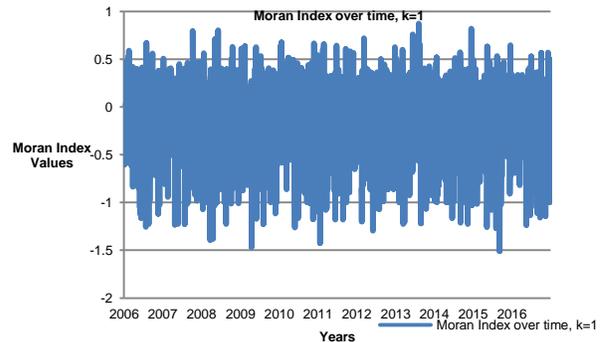


Fig. 3. Local Moran Indexes at  $k=1$  for a Chosen Weather Station.

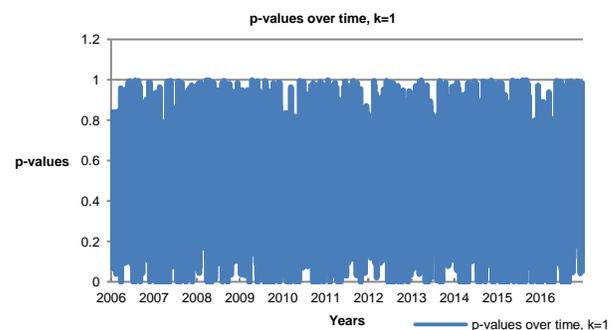


Fig. 4. Collected P-Values at  $K=1$  for a Chosen Weather Stations

With this, since none of the stations exhibit rainfall correlation, an alternate course of action was taken. Instead of using correlation and neighboring stations' rainfall values as an addition for feeding data into the MLPNN, a single station's data was used to feed data to the MLPNN. The selection of this single station depended primarily on the percentage of missing data on its dataset reducing the need to impute the remaining missing data and the number of local variables correlated with the location's rainfall attribute. Ultimately, data set from the Davao station with its 0.54% missing data percentage and having 7 climate variables correlated with rainfall was used in the alternate action. In total, the Davao dataset will bring with it 11 variables to be used as data for the MLPNN: 8 climate variables including the rainfall variable and 3 numerical variables, corresponding to the day of the month, month, and year, respectively. As shown in Table IX, the Training Set starts at the beginning of the time series which is the 1<sup>st</sup> of January 2006 and ends on the 31<sup>st</sup> of December 2016. The Testing Set starts right after the end of the Training Set which is the 1<sup>st</sup> of January 2017 and ends in the 31<sup>st</sup> of December 2017.

**B. MLPNN Model Evaluation Results**

The architecture defines the structure of the MLPNN which includes the number of inputs in the input layer, number of neurons in the hidden layer and the number of outputs in the output layer. Shown in Fig. 5 are the input layer, hidden layer and output layer of the MLPNN. With respect to the number of input neurons, results of the data preparation process led the researchers in identifying the final eleven variables to be used as inputs in the input layer namely (1) average temperature, (2) minimum temperature, (3) maximum temperature, (4) average wind speed, (5) maximum wind speed, (6) relative humidity, (7) total rainfall, (8) visibility, (9) day, (10) month, and (11) year. These parameters resulted in a high *p*-value indicating its correlation with respect to rainfall. This implies that these parameters influence the formation of rain at some point, thus its inclusion as inputs. A study found out that the three most significant data inputs in rainfall prediction aside from daily rainfall or precipitation are relative humidity, air pressure, and average temperature [14]. Air pressure on the other hand was not included as the final input after getting a *p*-value greater than 0.005 or not within a 95% confidence interval on the correlation evaluation of each variable with respect to rainfall as shown in Table X. This implies that for the Davao dataset, air pressure does not hold weight in rainfall forecasting. The logical explanation would be due to Davao's topography and geography that has been captured by the MLPNN during the training phase using years of data about Davao's air pressure readings.

TABLE IX. DATA PARTITIONING

Study Set	Month/s or Year/s Used	Number of months/years used	Amount of data records
Training Set	2006-2016	12 years	4019 records
Testing Set	2017	1 year	365 records

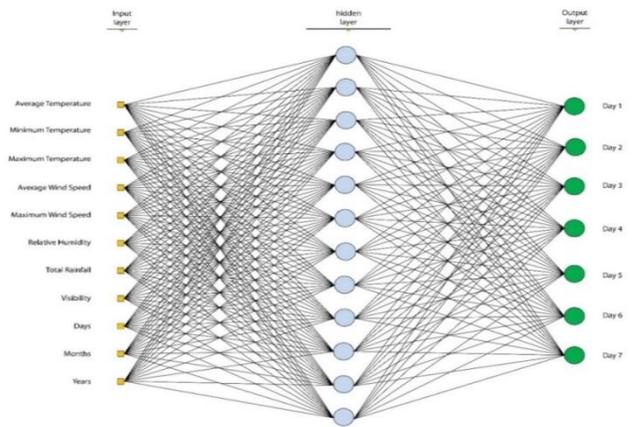


Fig. 5. The Architecture of the Multi-Layer Perceptron Neural Network.

In a study which used present hourly rainfall data and meteorological parameters of relative humidity, air pressure, temperature, visibility, and rainfall from surrounding rain gauge stations as input variables, the MLPNN was able to promisingly predict rainfall 1 to 6 hours ahead at 75 rain gauge stations as forecast point [4]. For the number of neurons in the hidden layer, the result of the Stathakis formula are shown in Table XI and was used as the values to be tested as the number of neurons in the hidden layer [15]. The number of hidden neurons is the maximum number of neurons that can be used with respect to the arbitrary scaling factor. Thus, if the scaling factor is 10, the number of hidden neurons the researchers can use are between 1-22 starting with the maximum value gradually decreasing as the researcher tests each value.

Another important observation the researchers had was the behavior of the models with respect to the number of hidden neurons. It was observed that Hyperbolic Tangent activation function does not converge to the maximum error unless the number of hidden neuron is less than 50. Anything above that number simply does not converge. However, Sigmoid exhibits the opposite since it converges to maximum error when the number of hidden neurons is greater than 50. Considering all these observations and running the MLPNN model multiple times with those range of values, 50 neurons for the Hyperbolic Tangent activation function and 100 neurons for the Sigmoid function were identified as the optimum number of hidden neurons to use for the respective activation functions. According to a study on hidden neurons in MLPNN, as the number of hidden nodes increases, the local minima point's also increases [21]. Increasing the number of hidden neurons enables the MLPNN to reach deeper local minima but also increases the possibility of getting stuck as the increase in the number of local minima is directly proportional with the increase in hidden nodes. Another study also found out that the Sigmoid's function values lies in the range from 0 to 1 which means that at some point in the graph, the gradient is approaching to zero and the network tends to stop learning on that point [22]. This can be addressed by increasing the number of neurons in the hidden layer in order to scale the Sigmoid Activation function. Thus, those behaviors observed might be due to these restrictions and limitations. As for the number of neurons in the output layer, the main objective of the study is to forecast week-ahead rainfall. This means that the MLPNN

model will be running on machine mode giving the neural network multiple output nodes, 7 output nodes to be exact, which represents the 7 days of the week from Monday to Sunday in no particular sequence to be predicted. The training algorithm is the parameter that tunes the network so that its outputs are close to the desired values [4, 10, 19]. Among the function approximation algorithms, most of the studies in rainfall forecasting use Backpropagation, Resilient Propagation and Quick Propagation as their training algorithms [1, 4, 7, 8, 18]. After evaluating more researches and looking at related studies, the researchers were able to identify two additional function approximation algorithms aside from the three mentioned that were suited for rainfall forecasting namely: Scaled Conjugate Gradient and Levenberg-Marquardt [7, 12, 19]. A total of five training algorithms, namely, Backpropagation, Resilient Propagation, Quick propagation, Scaled Conjugate Gradient, and Levenberg-Marquardt were used.

As for the learning rate, the researcher used the default standard value of 0.001-0.8 suggested by studies [5, 6, 18, 19].

With the 0.8 values, researchers were able to reach acceptable percent errors on their respective models which gives reasonable bases for using the same standard learning rate value. Moreover, higher learning rates speed the convergence process, but can result in overshooting or non-convergence. Consequently, lower learning rates product more reliable results at the expense of increased training time. For the momentum parameter, it is important to take note not to set the parameter too high as it can create a risk of overshooting the minimum values that can cause the system to become unstable but not too low as well as it cannot reliably avoid local minima and slow the training of the system. The optimal value of momentum can be achieved through trial and error between 0.1 and 0.9 as these values had been tested to work best with Backpropagation, Resilient Propagation, and Quick propagation approximation functions [1, 8, 18]. After these parameters had been identified, different models were formulated and used the same learning rate of 0.001 and momentum of 0.8 with training and testing results shown in Table XII.

TABLE X. EVALUATION RESULT FOR DAVAO VARIABLES

Davao Variables	Correlation Evaluation	p-Value	Davao Variables	Correlation Evaluation	p-Value
Ave Temp	-0.1512712	2.20E-16	Rel. Humidity	0.1703603	2.20E-16
Max Temp	-0.1082898	5.22E-13	Visibility	-0.09161644	1.04E-09
Min Temp	-0.09997687	2.70E-11	Ave. Wind Speed	-0.08984917	2.17E-09
Air Pressure	-0.00881795	0.5578	Max. Wind Speed	-0.0460988	0.00217

TABLE XI. RESULT OF STATHAKIS'S FORMULA

Arbitrary Scaling Factor	Hidden Neurons	Arbitrary Scaling Factor	Hidden Neurons	Arbitrary Scaling Factor	Hidden Neurons
2	112	5	45	8	28
3	74	6	37	9	25
4	56	7	32	10	22

TABLE XII. MLPNN MODEL EVALUATION TRAINING RESULTS

Models	Training Algorithm	Activation Function	Max Error Reached
Model 1	Back Propagation	Sigmoid	0.0011
Model 2	Resilient Propagation		0.00129
Model 3	Quick Propagation		0.00143
Model 4	Scaled Conjugate Gradient		0.00157
Model 5	Levenberg-Marquardt		Did not reach max error
Model 6	Back Propagation	Hyperbolic Tangent	0.00145
Model 7	Resilient Propagation		0.00136
Model 8	Quick Propagation		0.00141
Model 9	Scaled Conjugate Gradient		0.00138
Model 10	Levenberg-Marquardt		Did not reach max error
Model 11	Back Propagation	Gaussian	Did not reach max error
Model 12	Resilient Propagation		Did not reach max error
Model 13	Quick Propagation		Did not reach max error
Model 14	Scaled Conjugate Gradient		Did not reach max error
Model 15	Levenberg-Marquardt		Did not reach max error
Model 16	Back Propagation	Sin	Did not reach max error
Model 17	Resilient Propagation		Did not reach max error
Model 18	Quick Propagation		Did not reach max error
Model 19	Scaled Conjugate Gradient		Did not reach max error
Model 20	Levenberg-Marquardt		Did not reach max error

All MLPNN models were run using the same identified parameters. With respect to the number of neurons in the hidden layer, since the researcher found an important observation about the behavior of some activation function with respect to the number of neurons, models were categorized into two: (1) Models running with 100 hidden neurons (2) Models running with 50 hidden neurons. That means each model was run twice for 50 and 100 hidden neurons then identified which models do converge and reach max error. Results showed that most of Gaussian and Sin models did not reach the maximum error so these models will not be included in the testing phase while most of Sigmoid and Hyperbolic Tangent models except Models 5 & 10 with Levenberg-Marquardt training algorithm reached a maximum error. A study which had almost similar setup trained an MLPNN with Sigmoid activation function using 50 hidden neurons and found out that although 50 hidden neurons was faster to learn, the model produces a smooth curve with more error, thus increasing the number of hidden neurons to 300 solved that problem [7]. During the testing phase, models that reached maximum error were used. These trained MLPNN models were loaded back and the testing dataset were fed. As shown in Table XIII, MAE and RMSE were then calculated in order to assess the performance of the MLPNN models.

For models running in 100 hidden neurons, Model 2 with Sigmoid activation function and Resilient Propagation training algorithm got the lowest MAE while Model 4 with Sigmoid activation function and Scaled Conjugate Gradient training algorithm got the lowest RMSE. For those running in 50 hidden neurons, Model 9 with Hyperbolic Tangent activation function and SCG training algorithm has the lowest MAE and RMSE. A graphical representation of MAE and RMSE is shown in the Fig. 6.

In order to determine the optimum performing model for the 100 hidden neurons, the researchers decided to use RMSE as the deciding factor in determining the optimal MLPNN model since there is only a 0.000664 difference between model 2 and 4. Thus, the best optimal MLPNN model for the 100 hidden neurons was Model 4 and for the 50 hidden neurons was Model 9. It can be noticed that both of these models used SCG as their training algorithm.

TABLE XIII. MAE AND RMSE DURING TESTING PHASE

Models	MAE	RMSE	Remarks
Model 1	0.025051015	0.041653867	100 hidden neurons
Model 2	0.020899512	0.034208070	
Model 3	0.023180201	0.037797543	
Model 4	0.021564216	0.031537630	
Model 6	0.022685495	0.031717446	50 hidden neurons
Model 7	0.022258201	0.031266462	
Model 8	0.028053257	0.039392566	
Model 9	0.021483081	0.030660975	

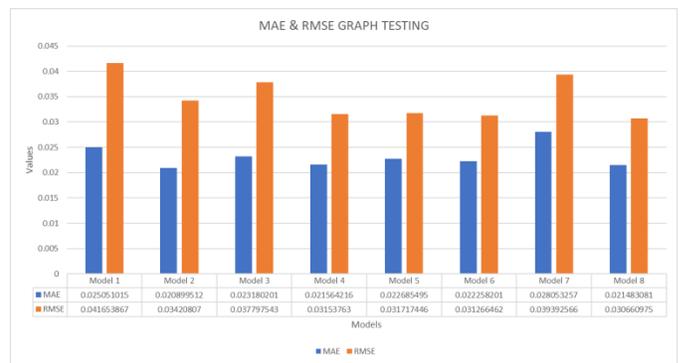


Fig. 6. MAE and RMSE During the Testing Phase.

#### IV. CONCLUSION AND RECOMMENDATIONS

Performance analysis of MLPNN models was conducted in this study among the weather station datasets in order to identify which MLPNN models can be optimally implemented in week-ahead rainfall forecasting. Techniques on weather data preparation, MLPNN model design along with its training and testing was conducted in this study. During rainfall data preparation, imputation process was a crucial part in addressing incorrect and inaccurate values in the datasets as it can greatly affect the outcome of the data being predicted. Random Forest Imputation technique was able to fill in the missing 5% rainfall values on the dataset. Pearson’s Correlation was also able to correlate 95% of the total inputs identified except for air pressure. However, the Moran’s Spatial Autocorrelation showed that geographical location of the stations did not have a direct correlation between stations with respect to rainfall prediction. During MLPNN model design creation, it was found out that the number of neurons for the hidden layer plays an important role in the prediction outcome as some models behaved differently with respect to the number of neurons. Other parameters such as activation function, training algorithm, learning rate and momentum was substantial to minimal effects on the outcome of the prediction. With this, an MLPNN model with Sigmoid activation function used 100 neurons in the hidden layer while an MLPNN model with Hyperbolic Tangent activation function used 50 hidden neurons. The MLPNN models that had the lowest MAE and RMSE were the ones who used Sigmoid and Hyperbolic Tangent as the activation function and Scaled Conjugate Gradient as the training algorithm with MAE of 0.021564 and 0.021483, RMSE of 0.031537 and 0.030660, respectively.

The researchers would like to recommend further studies on the aspect of hidden neuron selection and the behaviors of activation functions and training algorithm with respect to these hidden neurons. The need to explore different methods in selecting MLPNN parameters is also highly recommended as this will help establish a reliable MLPNN model performance analysis on rainfall forecasting. The researchers also suggest that further studies would be conducted on proper ways of performing training and testing that are suited and optimized for MLPNN architecture in weather forecasting as this will also help in improving the accuracy of the models which are subject to performance analysis. Overall, the results of this study showed that MLPNN models have the potential to be a viable week-ahead rainfall forecasting technique given that proper

data preparation, model architecture selection, model formulation and model validation are performed.

#### ACKNOWLEDGMENT

The authors would like to thank the support of the MSU-IIT Office of the Vice Chancellor for Research and Extension through PRISM-Premiere Research Institute in Science and Mathematics for their assistance in this study.

#### REFERENCES

- [1] Khalili, N., Khodashenas, S.R., Davary, K., Karimaldini, F. (2011). Daily Rainfall Forecasting for Mashhad Synoptic Station using Artificial Neural Networks. IPCBEE vol. 19, IACSIT Press, Singapore, 2011
- [2] L. C. Velasco, D. L. Polestico, G. P. Macasieb, M. B. Reyes and F. Vasquez, "Load Forecasting using Autoregressive Integrated Moving Average and Artificial Neural Network" International Journal of Advanced Computer Science and Applications(IJACSA), 9(7), 2018. <http://dx.doi.org/10.14569/IJACSA.2018.090704>
- [3] S. K. Nanda, D. P. Tripathy, S. K. Nayak, S. Mohapatra, "Prediction of Rainfall in India using Artificial Neural Network Models" IJISA International Journal of Intelligent Systems and Applications, Vol. 5(12), pp 1-22, 2013, doi:10.5815/ijisa.2013.12.01
- [4] N. Q. Hung, S. Babel, S. Weesakul, N. K. Tripathi, "An Artificial Neural Network Model for Rainfall Forecasting in Bangkok Thailand" Hydrology and Earth System Sciences, 13(1), pp 1413-1425, doi: 10.5194/hess-13-1413-2009, 2009
- [5] K. K. Kuok, S. Haron, C.P. Chan, "Hourly Runoff Forecast at Different Leadtime for a Small Watershed using Artificial Neural Networks" International Journal of Advances in Soft Computing and its Applications, 3(1), 2011.
- [6] R. Bustami, C. Hong, D. Lim, "River Bedup Catchment Water Level Prediction Using Pre-developed ANN Model of Siniawan Catchment" UNIMAS e-Journal of Civil Engineering, Volume 2(1), 2011
- [7] S. Mahajan, H. Mazumdar, H., "Rainfall Prediction using Neural Net based Frequency Analysis Approach," International Journal of Computer Applications, Volume 84 (9), pp 0975 – 8887, 2013
- [8] K. Abhishek, A. Kumar, R. Ranjan, S. Kumar, "A Rainfall Prediction Model using Artificial Neural Network" IEEE Control and System Graduate Research Colloquium, 2013, doi: 10.1109/ICSGRC.2012.6287140
- [9] J. V. Broeck, S. A. Cunningham, R. Eeckels, K. Herbst, "Data cleaning: Detecting, diagnosing, and editing data abnormalities" PLoS Medicine, Volume 2(10), 2005, doi:10.1371/journal.pmed.0020267
- [10] L. C. Velasco, C. Villezas, P. N. Palahang, and J. A. Dagaang, "Next day electric load forecasting using Artificial Neural Networks". In Proceedings of the IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu, 2015
- [11] V. K. Dabhi, S. Chaudhary, "Hybrid Wavelet-Postfix-GP Model for Rainfall Prediction of Anand Region of India," Advances in Artificial Intelligence, 2014(717803), pp 1-11, 2014, doi:10.1155/2014/717803
- [12] P. Singh, B. Borah B. "Indian summer monsoon rainfall prediction using artificial neural network," Stochastic Environmental Research and Risk Assessment, Volume 27(7), pp. 1585-1599, 2013, doi:10.1007/s00477-013-0695-0
- [13] C. J. Matyas, J. A. Silva, "Extreme Weather and Economic Well-being in Rural Mozambique. Natural Hazards, 66(1), 31-49. doi:10.1007/s11069-011-0064-6, 2013
- [14] J. B. Omotosho, A. A. Balogun, K. Ogunjobi, "Predicting Monthly and Seasonal Rainfall, Onset and Cessation of The Rainy Season in West Africa Using Only Surface Data". International Journal of Climatology, 20(1), pp 865–880. doi : 10.1002/1097-0088(20000630)20:8 2000
- [15] D. Stathakis, "How many hidden layers and nodes?", International Journal of Remote Sensing, 30(8), pp 2133-2147, 2009
- [16] N. K. Goyal, "Software development efforts prediction using artificial neural network", The Institution of Engineering and Technology, ISSN 1751-8806 doi:10.1049/iet-sen.2015.0061, 2015
- [17] W. H. Wilson, Function approximation, Artificial Intelligence Group, School of Computer Science and Engineering, University of NSW, 2019
- [18] V. K. Somvanshi, O. P. Pandey, P. K. Agrawal, N. V. Kalanker, M. R. Prakash, R. Chand, "Modelling and prediction of rainfall using artificial neural network and ARIMA techniques", The Journal of Indian Geophysical Union, 10(2), pp 141-151. doi=10.1.1.575.614, 2006
- [19] A. Mislan, S. Hardwinarto, D. Sumaryono, M. Aipassa, "Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggara Station, East Kalimantan – Indonesia," Procedia Computer Science, Volume 59(1), pp 142-151, 2015, doi: 10.1016/j.procs.2015.07.528
- [20] Y. Maruyama, "An alternative to Morans I for spatial autocorrelation" Statistics Theory, Cornell University, arXiv:1501.06260 (Math, Stat), 2015.
- [21] K. Vora, S. Yagnik, "A New Technique to solve local Minima problem with large number of hidden nodes on Feed Forward Neural Network", International Journal of Engineering Development and Research, 2(2), pp 1978-1981, ISSN: 2321-9939, 2014
- [22] D. Gupta, "Fundamentals of Deep Learning – Activation Functions and When to Use Them?", Deep Learning, Analytics Vidhya, 2017