



Research article

Days-ahead water level forecasting using artificial neural networks for watersheds

Lemuel Clark Velasco^{1,2,*}, John Frail Bongat², Ched Castillon², Jezreil Laurente² and Emily Tabanao²

¹ Premier Research Institute of Science and Mathematics, Mindanao State University-Iligan Institute of Technology, Iligan City, 9200, The Philippines

² College of Computer Studies, Mindanao State University-Iligan Institute of Technology, Iligan City, 9200, The Philippines

* **Correspondence:** Email: lemuelclark.velasco@g.msuiit.edu.ph.

Abstract: Watersheds of tropical countries having only dry and wet seasons exhibit contrasting water level behaviour compared to countries having four seasons. With the changing climate, the ability to forecast the water level in watersheds enables decision-makers to come up with sound resource management interventions. This study presents a strategy for days-ahead water level forecasting models using an Artificial Neural Network (ANN) for watersheds by conducting data preparation of water level data captured from a Water Level Monitoring Station (WLMS) and two Automatic Rain Gauge (ARG) sensors divided into the two major seasons in the Philippines being implemented into multiple ANN models with different combinations of training algorithms, activation functions, and a number of hidden neurons. The implemented ANN model for the rainy season which is RPROP-Leaky ReLU produced a MAPE and RMSE of 6.731 and 0.00918, respectively, while the implemented ANN model for the dry season which is SCG-Leaky ReLU produced a MAPE and RMSE of 7.871 and 0.01045, respectively. By conducting appropriate water level data correction, data transformation, and ANN model implementation, the results of error computation and assessment shows the promising performance of ANN in days-ahead water level forecasting of watersheds among tropical countries.

Keywords: artificial neural network; days-ahead water level forecasting; watersheds; water level forecasting; multilayer perceptron neural network

1. Introduction

Due to climate change, erratic and undetermined water levels among bodies of water affect both lives and properties. This has led to the importance of predicting the occurrence of flash floods in watersheds as a critical technique for ensuring the safety and well-being of nearby residents. The phases of water level are designed to raise awareness among local authorities on the level of risks posed by the rising water level with the goal that an emergency arrangement can be initiated for the welfare of the local community residing near these bodies of water [1–4]. Forecasting water levels in lakes, rivers, watersheds, and groundwater levels have different factors to consider such as the experimental dataset for training the model, different forms of methods to be used for the training function, and the different input and output parameters [2,3]. There have been numerous existing models and technologies used in water level forecasting with the most common machine learning framework to be Artificial Neural Network (ANN) along with traditional statistical techniques such as Auto Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA) [2,5–9]. Several studies have developed a hybrid model by combining two different models and have been proven to provide better accuracy over individual models alone [7,8,10]. However, these models are expected to slow down the training process which would in turn affect its utilization. Like non-hybrid models, if there is not enough training data available, then accurate estimation and prediction become difficult. ANN is a mathematical model based on the structure and function of biological neural networks. When sufficiently trained with cleaned data, this model can solely identify non-linear patterns of meteorological behavior [8,11]. Thus, ANN should be ideally suited for the modelling of hydrological data which are known to be non-linear and complex [3,11–13]. To accurately predict the water level, the dataset must be prepared and the significant input variables must be properly selected along with choosing the suitable structure, parameters, and other variables. Consequently, if the chosen settings do not suit the data, the outcome will result in poor performance of the model's accuracy.

Tropical countries like the Philippines only have two seasons of the dry season from December to May and rainy the season from June to November. During the rainy and typhoon season, the occurrence of floods around the affected areas of the Mandulog River Watershed in Southern Philippines has been causing major problems to local administrators, government officials, and people's lives and property. With an area of about 78,228 hectares, the watershed collects from smaller rivers connected by a network of waterways more than 50 kilometres long as a network of channels serving as one of the main drains of rainfall water from its surrounding mountains [14]. The current technology used in the watershed is the Water Level Monitoring Station (WLMS) which determines the water level at a point in time every 5–10 minutes interval. The challenge faced by decision-makers in monitoring the watershed goes beyond capturing the past and present water levels of the watershed but is focused in the prediction of the water level. With currently no existing system that can predict the water level of the watershed, this study attempts to develop a days-ahead water level forecasting system by conducting data preparation of the captured water level dataset and implementing artificial neural networks as a machine learning framework. By evaluating its forecasting performance, close to accurate prediction of the watershed's water level can serve as inputs for local administrators in the early identification of flooding resulting to timely protection of property and evacuation of affected individuals.

2. Methodology

2.1. Water level data preparation

In order to implement a well performing system, water level data from the Mandulog River Watershed needs to undergo data selection, data correction, and data transformation for it to be fed into the ANN. Initially, choosing the right dataset was conducted by identifying reliable data source along with selecting the range of time to be considered. Studies considered parameters that can be used for the ANN inputs such as rain, rain rate, air temperature, humidity, air pressure, and solar radiation in water level forecasting [2–4,9,13]. In this study, the researchers used the rainfall and water level data from 2013 to 2018 collected by the Philippine Department of Science and Technology-Advanced Science and Technology Institute (DOST-ASTI). The collected water level data from a WLMS has a 10-minute interval with a unit of measurement in meters, while the rainfall data from one Automatic Rain Gauge (ARG) has a 15-minute interval and the rainfall data from two ARGs has a 30-minute interval, with a unit of measurement in millimeters. As shown in Table 1, the water level data from the WLMS were in a.csv format with the columns specific for the date, time, and water level delivered. These raw data from the .csv file format were then later imported and stored in a PostgreSQL database.

Table 1. Sample water level raw data.

DATE	TIME	WATER LEVEL DELIVERED
XX	XX	XX
XX	XX	XX

Data correction was then conducted for the detection, correction, or removal of corrupt and inaccurate records from the water level dataset which may refer to incomplete, incorrect, or irrelevant parts of the data. Researchers conducted a manual visual inspection to determine the missing values in the spreadsheet of water level and rainfall data. Data correction or cleaning was then conducted to handle missing values in the WLMS raw dataset and was replaced using applicable imputation methods [4,15]. The percentage of missing data in the dataset was then computed in order to determine how much data is erroneous in term of missing values, skipping time and time inconsistency. As suggested by authors, the researchers used two imputation methods for filling the missing data values with the regression analysis method applied for the missing water level data and the linear interpolation method applied for the missing rainfall data [1,10,16]. In formulating the regression model, as shown Equation 1, the predicted value y is the water level and x is the rainfall. The regression model with the dependent variable y and independent variable x_1, x_2, \dots, x_n is defined as:

$$y_i = b_0 + b_j x_{ij} + \dots + b_i x_{ip} + e_i \quad (1)$$

where y_i is the amount of i th dependent variable p is the number of predictors, b_j is the amount of i th coefficient, $j = 0, \dots, p$ x_{ij} is the value of i th of j th predictor and e_i is the observed error of the value for i th [16].

The linear interpolation method has been used by researchers for filling the missing values in time series and found that it has good results for rainfall data [17,18]. In formulating the linear interpolation model, the predicted value x is the rainfall data. Equation 2 shows the Linear Interpolation Equation where x is the independent variable, x_0 is the known value of the independent variable and $f_1(x)$ is the value of the dependent variable for a value x of the independent variable [17].

$$f_1(x) = b_0 + b_1(x - x_0) \quad (2)$$

$$\text{where } b_0 = f(x_0) ; b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

In dealing with the issue of time inconsistency, this study considered that each data row with time inconsistency will be associated to the nearest original time interval by replacing each row to the nearest time interval. The researchers used Python's `resample(how)` function as the process of replacing each row to the nearest original time interval and filling in the gaps of skipping time, where `how` is the preferred frequency or time interval of the data, which in this study is 10-minute for water level data and 15-minute for rainfall data.

Data transformation was then conducted by normalizing the dataset input with a certain finite range followed by the process of partitioning the data will be done into three subsets such as training, testing, and validation sets. The process of transforming or normalizing is important because without it the training of the network will be slow [9,10,19]. All input data to the ANN was normalized using the min-max normalization method shown in Equation 3.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3)$$

where z is the new water level normalized data, x is the water level data value to be normalized, $\max(x)$ is the maximum water level data value and $\min(x)$ is the minimum water level data point value in a dataset. The min-max transformation method is a linear transformation of data to a smaller range, typically in the 0 to 1 range without outliers.

2.2. ANN Model implementation

After data preparation, the dataset was then ready to be fed into the multilayer perceptron neural network. As shown in Figure 1, the architecture of the ANN involved a single input layer with seven input neurons identified as Year, Month, Day, Time, Mandulog Water Level, Digkilaan Rainfall, and Rogongon rainfall. One output layer was utilized having 144 output neurons which represents 3 days with 30-minute intervals. The three-day predictive projection was the request of the local Disaster Risk Reduction and Management Office as the ideal and optimal time period for decision making and information dissemination. This study used 1 hidden layer containing 149 hidden neurons along with identified parameters of 0.001 to 0.1 learning rate, 0.1 momentum and an epoch of 17,000. Two separate ANN models were implemented with an ANN model for the rainy season utilizing Resilient Propagation (RPROP) as the training algorithm and Leaky ReLU as the activation function while an ANN model for the dry season utilizing Scale Conjugate Gradient (SCG) as the training algorithm and Leaky ReLU as the activation function.

The researchers used Keras – an open-source neural network library written in Python, which runs on top of the machine learning platform TensorFlow. This library includes easy handling of data sets, supports several different activation functions and training algorithms, and it includes a framework for easy handling of training data sets. Keras is designed to enable fast experimentation with neural networks and focuses on being user-friendly, modular, and extensible. Keras library was imported into the systems project library to be able to implement the ANN model and used the built-in functions provided in the Keras Library. Figure 2 shows the web-based application that integrated the ANN model using Keras and a micro web framework written in Python language called Flask. Creating the

model, training the model, and testing the model, and the parameters were done using the Keras functions such as `add()` and `compile()`, `fit()`, `evaluate()`, and `predict()`.

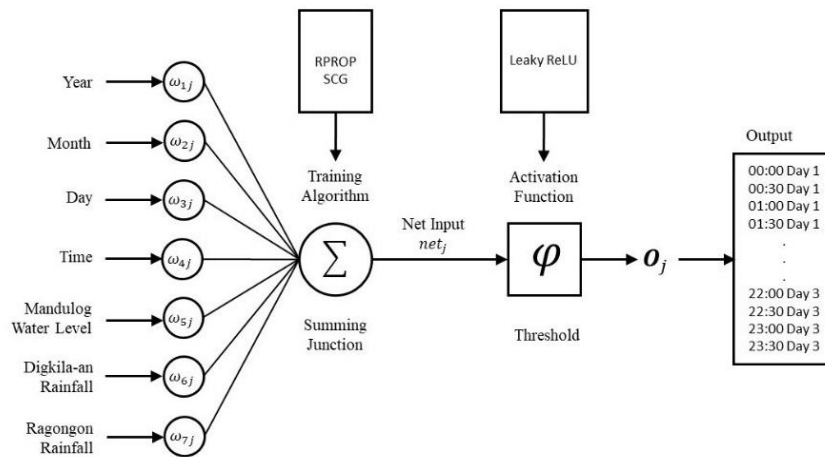


Figure 1. Block diagram of the ANN.

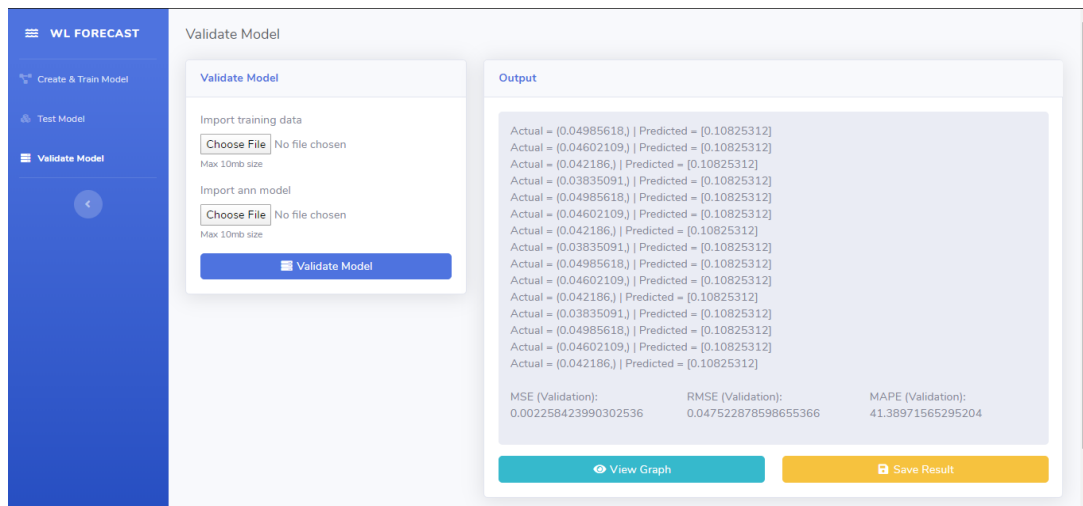


Figure 2. Screenshot of the web-based application.

The predictive ability of the ANN models for the two seasons were then tested and validated by comparing the results of the forecasted values to the actual water level values of the Mandulog River Watershed. The implemented ANN models were tested using the validation set from the water level data of March 2018 by producing 3-days ahead forecasted values. To evaluate the accuracy of the ANN models, two statistical formulas were selected, namely the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The RMSE describes the average magnitude of error of the observed and forecasted values with values near to 0 indicating the best fit to the data while the MAPE measures the size of the error in percentage terms. Smaller RMSE and MAPE values indicate consistency and accuracy of the developed ANN model. Equation 4 shows how RMSE was computed where h_i is the observed water level, \hat{h}_i is the forecasted water level from the model, and N is the number of data points. Equation 5 shows how MAPE was computed where h_i is the observed water

level, \hat{h}_i is the forecasted water level from the model, and N is the number of data points.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (h_i - \hat{h}_i)^2}{N}} \quad (4)$$

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{h_i - \hat{h}_i}{h_i} \right| \quad (5)$$

A tabular and graphical representation of the computed results was then generated to illustrate the comparison between the observed and forecasted values. As an additional validation, results were transformed and compared with the actual values to visualize if they were accurate to the real-world scenario. In this study, the validation data set from June 2018–October 2018 will be used to generate the validation results for rainy season and the validation data set from December 2017–May 2018 for dry season. These results were compared to actual water level values to determine if they were accurate by denormalizing the predicted water level and comparing them to the actual water level values by graphing them into a line graph. For the selected model, the denormalized values at each 3-day iteration will be graphed against the actual values of the same day. Once the graph was generated, they will be assessed for accuracy by way of visual inspection. The scheme that will be used for visual inspection is by observing it on a per day per 3-day basis. This means that for every day in the month of March, the next 3 days will be collected and graphed. Equation 6 shows the Min-Max denormalization formula where x is the original pre-normalized value of water level, z is the normalized equivalent of x , $max(x)$ is the maximum value and $min(x)$ is the minimum value.

$$x = z (max(x) - min(x)) + min(x) \quad (6)$$

2. Results and discussion

3.1. Water level data preparation results

The obtained water level data for the Mandulog River Watershed captured by the WLMS and the rainfall data for the neighboring vicinities of Digkilaan, Rogongon, and Pugaan captured by the ARGs were originally stored in a .csv file format. Figure 3 shows a map containing the locations of the four sensors with the WLMS at the Mandulog River Watershed and the ARGs located in Rogongon, Digkilaan, and Pugaan surrounding the Luinab catchment. The rainwater run-off from the Luinab catchment flows to the Luinab creek, which is a tributary of the Mandulog River [20]. Thus, rainfall data from Digkilaan and Rogongon were selected to be used in this study since these are the nearest ARGs installed near the Mandulog River Watershed WLMS. The rainfall data from Pugaan was not included since it is too far from the Mandulog River Watershed.

As shown in Table 2, the water level data from the Mandulog River Watershed WLMS contains 6 years of data from June 2012 to October 2018 with a 10-minute interval. The rainfall data from Digkilaan ARG contains 6 years of data from July 2012 to October 2018 with a 15-minute interval, and the rainfall data from Rogongon ARG contains 5 years of data from February 2013 to October 2018 with a 15-minute interval. In order to match the start and end date/time, this study used 5 years of data that ranges from February 2013 to October 2018. Several researchers used 2–4 years of historical data to forecast water level and proved enough to obtain satisfactory predictions [13,21]. Thus, in this paper, the 5-year historical data was considered be enough in developing the predictive model. Trimming the data was necessary for two reasons in order to the water level and rainfall data

to coincide so that they have the same starting date/time and end date/time and for the simplified conversion to 30-minute interval since the water level data has a 10-minute interval while rainfall data has a 15-minute interval. Table 3 shows the details of the trimmed data from the three sensors, a total of 573,189 rows of data were collected with each set having 3 variables of 264,246 records from Mandulog River Watershed water level, 156,279 records from Digkilaan rainfall, and 152,664 records from Rogongon rainfall. From this point onward of this study, the term dataset will refer to these trimmed data.



Figure 3. Map containing the location of the sensor sources.

Table 2. Details on the selected sensors.

Sensor	Date range	Number of rows	Time interval
Mandulog WLMS	Jun 2012–Oct 2018	298, 592	10-minute
Digkilaan ARG	Jul 2012–Oct 2018	176, 795	15-minute
Rogongon ARG	Feb 2013–Oct 2018	152, 709	15-minute

Table 3. Details on the trimmed selected sensors.

Sensor	Date range	Number of rows	Time interval
Mandulog WLMS	Feb 2013–Oct 2018	264, 246	10-minute
Digkilaan ARG	Feb 2013–Oct 2018	156, 279	15-minute
Rogongon ARG	Feb 2013–Oct 2018	152,664	15-minute

After conducting a manual inspection of the dataset, it was found out that time inconsistency, skipping time, and missing values were present due to the limitations of the device. Time inconsistency refers to a time that is not within the time interval e.g. 6:32, 8:17, and 22:46. Table 4 shows the sample raw data of a chosen sensor originally in a 15-minute interval with its actual time inconsistency compared to the ideal time consistency.

Table 4. Sample dataset with its time inconsistency.

Date	Digkilaan ARG	
	Actual time inconsistency	Actual time consistency
06-17-2016	23:16:00	23:15:00
06-18-2016	00:32:00	00:30:00

Table 5 shows the amount of time inconsistency per sensor and its percentage when compared to the calculated total amount of data. The total amount of raw data was defined as the number of rows of the data gathered directly from the sensor, while calculated total amount of data is the expected complete number of rows of data. Results showed that the sensor obtaining the highest amount of time inconsistency is the Digkilaan ARG with 4.76% while the Mandulog River Watershed WLMS has the least amount of time inconsistency with 0.023%. The presence of time inconsistency in the data may be due to sensor failure or faulty transmission of data, and for this reason it is beyond human capability to make perfect data.

Table 5. Amount of time inconsistency per sensor.

Sensor	Time inconsistency	Total amount of raw data	Calculated total amount of data	Percentage of time inconsistency
Mandulog WLMS	69	264,246	298,653	0.023%
Digkilaan ARG	9,438	156,279	199,102	4.74%
Rogongon ARG	838	152,664	199,102	0.42%

Skipping time refers to rows of time that do not conform to the specific time interval, e.g. from 5:30 directly skipping to 10:15 with the difference between skipping time and time inconsistency is that the time in skipping time jumps forward widely. As shown in the sample data of the Mandulog WLMS in Table 6, with the time 11:40, it jumped right to 12:10 when it should be 11:50 since water level data has a 10-minute interval.

Table 6. Sample dataset with its skipping time.

Mandulog WLMS		
Date	Time	Water level (m)
12-15-2013	11:40:00	1.6
12-15-2013	12:10:00	1.65

Table 7 shows the amount of skipping time per sensor and its percentage when compared to the actual total amount of data. Total amount of raw data is defined as the number of rows of the data gathered directly from the sensor, while calculated total amount of data is the expected complete number of rows of data. Results showed that the sensor having the highest amount of skipping time is the Rogongon ARG with 23.41% while the Mandulog River Watershed WLMS has the least amount of skipping time with 11.57%. The same as time inconsistency, the presence of skipping time in the data may be due to measurement equipment malfunction or other measurement errors.

Table 7. Amount of skipping time per sensor.

Sensor	Skipping time	Total amount of raw data	Calculated total amount of data	Percentage of skipping time
Mandulog WLMS	34,407	264,246	298,653	11.52%
Digkilaan ARG	42,632	156,279	199,102	21.50%
Rogongon ARG	46,440	152,664	199,102	23.32%

In dealing with the issue of time inconsistency, each data row with time inconsistency was associated to the nearest original time interval by replacing each row to the nearest time interval. For example, as shown in Table 8, with the Digkilaan ARG, rows of data have a time inconsistency of 23:16:00, 23:31:00, 0:01:00. By replacing each row to the nearest original time interval, the new actual time consistency became 23:15:00, 23:30:00, 0:00:00. For skipping time, as shown in Table 9, this was dealt by filling in the gaps with the difference between the succeeding time and the previous time. The process of replacing each row to the nearest original time interval and filling in the gaps of skipping time was performed using the Python function `resample(how)`, where `how` is the preferred frequency or time interval of the data, which in this case is 10-minute for water level data and 15-minute for rainfall data.

Table 8. Sample states of before and after time inconsistency intervention.

Digkilaan ARG			
Before		After	
Date	Time	Date	Time
06-17-2016	23:16:00	06-17-2016	23:15:00
06-17-2016	23:31:00	06-17-2016	23:30:00
06-18-2016	00:01:00	06-18-2016	00:00:00

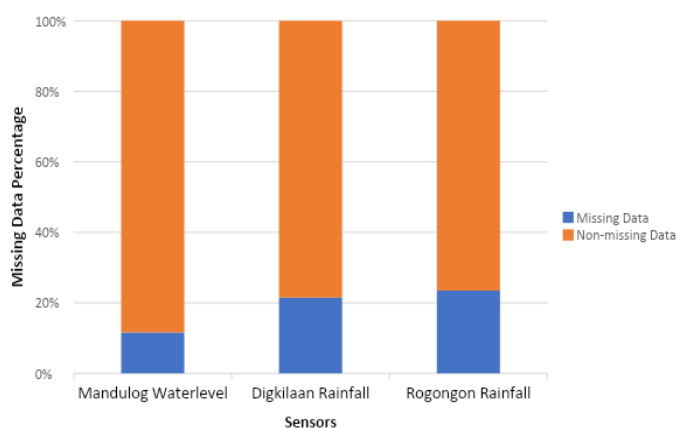
Table 9. Sample States of Before and Skipping Time Intervention.

Mandulog WLMS					
Before			After		
Date	Time	Water level (m)	Date	Time	Water level (m)
02-15-2013	11:40:00	1.6	02-15-2013	11:40:00	1.6
02-15-2013			02-15-2013	11:50:00	
02-15-2013			02-15-2013	12:00:00	
02-15-2013	12:10:00	1.65	02-15-2013	12:10:00	1.65

In the case of missing values, the presence of missing values causes a significant bias in the results and reduces the efficiency of the dataset. Ignoring the missing data is generally not valid for time-series prediction in which the currently predicted value of a system commonly depends on the historical time data of the system [1,17,22]. Table 10 shows the number of missing values per sensor and its percentage in terms of being sensor-based or being aggregated. Percentage of missing values in terms of sensor-based is defined as the percentage of the number of missing values from the raw data received directly from the sensor over the total amount of raw data, while percentage of missing values in terms of aggregated is defined as the percentage of the number of missing values from the calculated total amount of data. Figure 4 shows a graphical representation of the percentage of missing data per sensor. According to the results, the sensor that exhibits the least number of missing values is the Mandulog River Watershed WLMS with 11.61%, while the Rogongon ARG with 23.46% is the sensor that exhibits the greatest number of missing values. The large number of missing values is most likely to have been caused by sensor failure and recording process. In a research, ANNs do not suffer much with missing data up to about 30% [23]. So the missing values are still acceptable for processing.

Table 10. Missing values per sensor.

Sensor	Sensor-based			Aggregated		
	Number of missing values	Total amount of data	Percentage of missing values	Number of missing values	Calculated total amount of data	Percentage of missing values
Mandulog WLMS	145	264,246	0.05%	34,552	298,653	11.61%
Digkilaan ARG	12	156,279	0.008%	42,644	199,102	21.50%
Rogongon ARG	92	152,664	0.061%	46,532	199,102	23.46%

**Figure 4.** Graphical representation of missing data percentage per sensor.

After calculating the percentage and the amount of missing data for rainfall, an imputation method was used to handle the missing data. In this study, linear interpolation was used to fill in the missing values of rainfall data. Table 11 shows the state of the dataset pre-imputation and post-imputation of the missing values for a chosen rainfall sensor.

Table 11. Sample state of rainfall data pre and post imputation.

Rogongon ARG					
Pre-imputation			Post-imputation		
Date	Time	Rainfall (mm)	Date	Time	Rainfall (mm)
3-1-2018	11:15:00	0	3-1-2018	11:15:00	0
3-1-2018	11:30:00		3-1-2018	11:30:00	0
3-1-2018	11:45:00		3-1-2018	11:45:00	0

In imputing the water level data, regression analysis was used to fill in the gaps of the missing data for water level. Table 12 shows the state of the dataset pre-imputation and post-imputation of the missing values for the Mandulog River Watershed WLMS water level.

Table 12. Sample state of water level data pre and post imputation.

Mandulog MWLS					
Pre-imputation			Post-imputation		
Date	Time	Water level (m)	Date	Time	Water level (mm)
3-1-2018	11:00:00	1.28	3-1-2018	11:00:00	1.28
3-1-2018	11:10:00	1.29	3-1-2018	11:10:00	1.29
3-1-2018	11:20:00		3-1-2018	11:20:00	1.28333
3-1-2018	11:30:00		3-1-2018	11:30:00	1.27666

In this research, the Mandulog water level data which has a unit of measurement in meters was converted to millimeters for the water level and rainfall to have the same unit of measurement, since both Digkilaan rainfall data and Rogongon rainfall data were in millimeters. After converting the water level into millimeters, the dataset was converted into a 30-minute interval, since it is the least common multiple of the rainfalls' fifteen-minute interval and water levels' 10-minute interval. Since the water level data was in a ten-minute interval and the rainfall data was in a fifteen-minute interval, the maximum water level among the three per ten-minute recordings was chosen for the half-hour and the maximum rainfall among the two per fifteen-minute recordings was chosen for the half-hour. The process of choosing the maximum water level from the three ten-minute records and the rainfall from the two fifteen-minute records was performed using a python code created by the researchers. The data from the three sensors were then concatenated into a single dataset based on the date and time. The total number of rows of the dataset was then reduced to 99,552.

After converting the dataset to a 30-minute interval, the dataset was represented with numeric values. The time variable in the 30-minute interval was converted into numerical values since ANN models cannot be fed with variables represented with a colon symbol. Starting with 00:00, which is 12:00 AM, time was represented with a value of 1. For every increment of 30 minutes, the representation value was incremented by 1. The process iterates until 23:30 is converted to up to 48. The researchers also divided the Date variable to have the year, month, and day variable since Keras does not accept data with semicolons, commas, and other non-numeric symbols. As shown in Table 13, these attributes including the Time, were then represented by binary values.

Table 13. Attribute binary table.

Year	Binary value	Month	Binary value	Day	Binary value	Time	Binary value
2013	00000001	January	00000001	1	00000001	1	00000001
2014	00000010	February	00000010	2	00000010	2	00000010
2015	00000011	March	00000011	3	00000011	3	00000011
2016	00000100	April	00000100	4	00000100	4	00000100
2017	00000101
2018	00000110	December	00001100	31	00011111	48	00110000

After representing the attributes by binary values, the water level and rainfall was normalized using the Min-Max Normalization that scales the data into a 0 to 1 range. Table 14 shows the sample dataset after normalization.

Table 14. Sample dataset after data representation.

Year	Month	Day	Time	Mandulog water level	Digkilaan rainfall	Rogongon rainfall
00000001	00000010	00001101	00000001	0.120805369	0.0	0.0
00000001	00000010	00001101	00000010	0.121764142	0.0	0.0
00000001	00000010	00001101	00000011	0.120805369	0.0	0.0

After data representation, the dataset was partitioned into training set, testing set, and validation set. As shown in Table 15, the dataset was divided into two major seasons in the Philippines reflected as June to November 2013–2018 being used as the rainy season dataset while data records during the dry season from December to May 2013–2018 were used as the dry season dataset. Each season was then proportioned to 70% for training set, 15% for testing set, and 15% for validation set [2].

Table 15. Data partitioning.

Study set	Rainy season (50, 640)			Dry season (48, 912)		
	Dates	Months	Data rows	Dates	Months	Data rows
Training set	June to Nov 2013–2016	24 months	35,136	Dec 2013– 2015, Jan to May 2013–2016	24 months	31,440
Testing set	June to Nov 2017	6 months	8,784	Dec 2016, Jan to May 2017	6 months	8,736
Validation set	June to Oct 2018	5 months	6,720	Dec 2017, Jan to May 2018	5 months	8,736

2.2. ANN Model implementation results

Predictive performance of the ANN models for both the rainy and dry season were assessed using RMSE and MAPE. Table 16 shows the error computation results on the validation set where the ANN model for the rainy season had a MAPE of 6.731 and RMSE of 0.00918 while the ANN model for the dry season had a MAPE of 7.871 and RMSE of 0.01045.

Table 16. Error computation results.

Season	Training algorithm	Activation function	Mape	Rmse
Rainy Season	Resilient Propagation	Leaky ReLU	6.731	0.00918
Dry Season	Scaled Conjugate Gradient	Leaky ReLU	7.871	0.01045

Once the values of the models have been denormalized, the values were compared to the actual water level values by graphing them into a line graph. For the two ANN models, the denormalized values at each 3-day iteration were graphed against the actual values of the same day. Once the graphs have been generated, the models were assessed for accuracy for visual inspection. The scheme that was used for visual inspection is by observing it on a per 3-day basis. Figure 5 depicts predicted outputs compared to actual values for the rainy season. The graph showcased the 3-day iteration of the months

of June 2018 to October 2018 for the predictive performance of the Resilient Propagation–Leaky ReLU combination implemented for the rainy season. It can be observed from the graph that the prediction follows the actual water level of the Mandulog River Watershed exhibiting close to accurate predictive values across the months with the values from June 3, 2018 to June 9, 2018 and on July 3, 2018 being very close to the actual values.

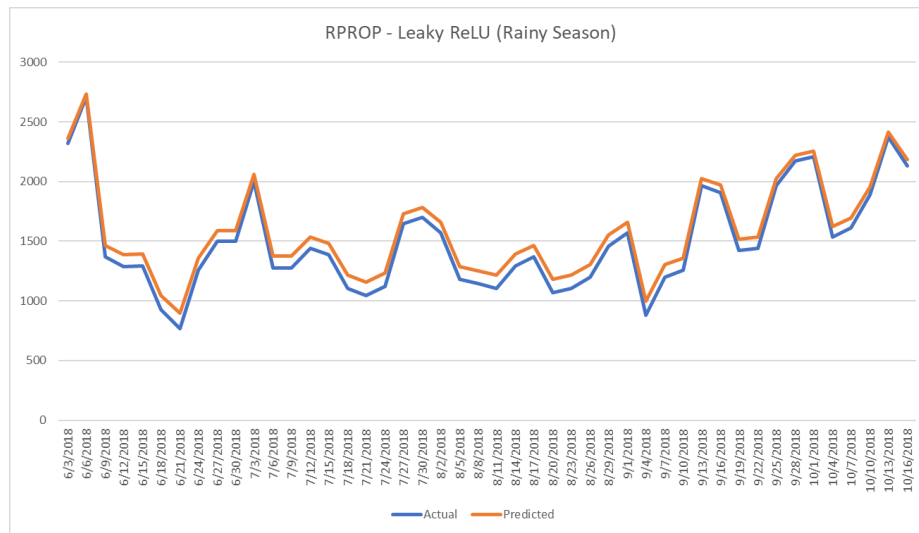


Figure 5. Actual vs. forecasted water level for the rainy season.

Figure 6 shows the performance for the SCG–Leaky ReLU combination for the dry season. It can be observed that though the actual and the predicted water levels have a margin, the rise and fall was evidently consistent to follow the pattern.

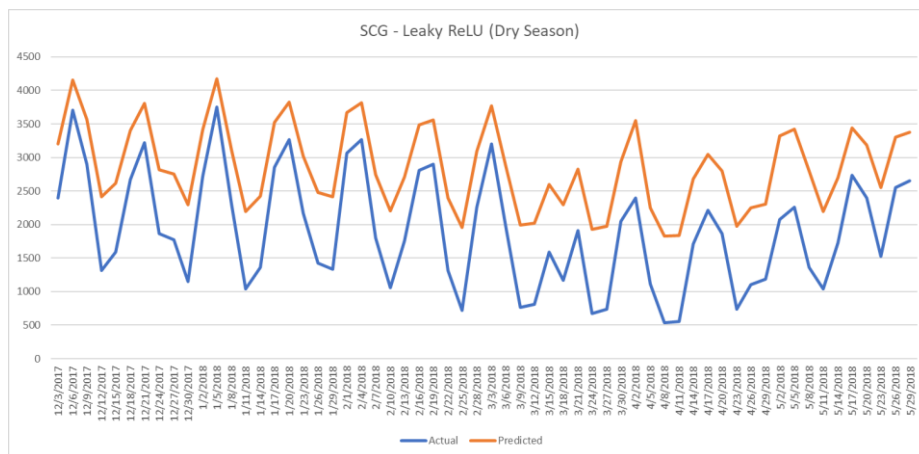


Figure 6. Actual vs. forecasted water level for the dry season.

As for the overall comparison of the two seasons, the forecasted outputs of rainy season’s RPROP–Leaky ReLU ANN model was closer to the actual values than that of the dry season’s SCG–Leaky ReLU ANN, but still produced good results in predicting the water level of the Mandulog River Watershed. The ANN models for the two tropical seasons exhibited an acceptable MAPE of below 15% by weather forecast standards [24]. While this study only used water level data to develop a water level

forecasting model for rainy and dry season, other researchers have used other climatic factors other than water level, such as rainfall, precipitation, temperature, and evaporation, which yielded good results [1,2,6–11,13,25,26]. A study used hourly rainfall and multiple water level data to predict the water level at the Anyangcheon stream in South Korea using ANN forecasting model showing a fairly good forecasting performance with an RMSE of 0.0936 which indicates that ANN models can simulate accurate water level forecasts [3]. However, there are also some studies that proved that water level alone can be used to develop ANN water level forecasting. A study implemented ANN, adaptive-neuro-fuzzy inference system (ANFIS), gene expression programming GEP, and ARMA to forecast daily water level, where in the four of them have almost the same accuracy with ANN having an RMSE of 0.114 for the 3-day ahead prediction [27]. Their result showed that ANN model was able to provide almost the same performance to the ANN models implemented in this study.

3. Conclusions and recommendations

In this study conducted in a tropical country, separate ANN models were implemented for each rainy and dry seasons in order to predict the water level of the Mandulog River Watershed. The general objective of this study was to develop days-ahead water level forecasting using ANN from the provided data of WLMS and ARGs conducted through a thorough data preparation and ANN model implementation process. In data preparation, the imputation process was a critical part in addressing the issues of time inconsistency, skipping time along with the missing and incorrect values in the data sets as it can significantly affect the result of the forecasting model. The linear interpolation method was able to fill in the missing values of both rainfall values on the dataset, while the regression method was able to fill in the 11.61% missing water level values. In implementing the ANN models into a web application, Keras library was successfully integrated with the application in setting up the environment for the development. In the validation of models, the researchers provided a calculation of MAPE and RMSE as well as a graphical visualization of the comparison between the actual and the forecasted water levels. It was shown that the ANN models exhibited good forecasting performance showcasing the Resilient Propagation–Leaky ReLU combination implemented for the rainy season exhibiting a MAPE of 6.731 with RMSE of 0.00918 and the Scaled Conjugate Gradient–Leaky ReLU combination implemented for the dry season exhibiting a MAPE of 7.871 with RMSE of 0.01045. It is also worth noticing that the rainy season ANN model has considerably better predicted outputs than the dry season ANN model.

Based on the findings of the study, the researchers would like to recommend further studies on the methods in data correction especially with the case if there are a lot of time inconsistencies, missing time, and more importantly the empty values in the water level dataset. As an improvement to the models, the researchers would also like to recommend further network training as well as using other ANN libraries aside from Keras. While Keras is an open-source neural network library, it has a limited training algorithm and activation function. The possibility of using other neural network libraries might gain a better result and can lead to a better forecasting system. Moreover, the researchers also suggest exploring different methods in selecting ANN parameters and other ways of performing training, testing, and validation as this might help in establishing a reliable ANN model for water level forecasting. Lastly, the researchers highly recommend installing other climatic monitoring systems in the Mandulog River Watershed such as rainfall and temperature that could be used as another input variable. Overall, the results of this study showed that ANN has the capability to be a promising days-

ahead water level forecasting model with proper data preparation and ANN model implementation. Water level forecasting among watersheds is a necessary tool to help identify the occurrence of flash floods for affected areas as it can help the local authorities in developing wise decisions for initiating emergency management or risk reduction management for the welfare of the local community.

Acknowledgements

The authors would like to thank the support of the Mindanao State University-Iligan Institute of Technology (MSU-IIT) Office of the Vice Chancellor for Research and Extension for their assistance in this study. This work is supported by MSU-IIT as internally funded research under the Premier Research Institute of Science and Mathematics (PRISM)- Applied Mathematics and Statistics (AMS) Research Group. The authors would also like to thank the Philippine Department of Science and Technology-Advance Science and Technology Institute (DOST-ASTI) for the data used in this study.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. M. H. Khalifelloo, M. Mohammad, M. Heydari, Multiple imputation for hydrological missing data by using a regression method (Klang River Basin), *IJRET Int. J. Res. Eng. Technol.*, (2015), 2321–7308. Available from: <http://www.ijret.org>
2. I. Sušanj, N. Ožanić, I. Marović, Methodology for developing hydrological models based on an artificial neural network to establish an early warning system in small catchments, *Adv. Meteorol.*, **2016** (2016), 9125219. <https://doi.org/10.1155/2016/9125219>
3. J. Y. Sung, J. Lee, I.-M. Chung, J.-H. Heo, Hourly water level forecasting at tributary affected by main river condition, *Water (Basel)*, **9** (2017), 644. <https://doi.org/10.3390/w9090644>
4. S. H. Arbain, A. Wibowo, Time series methods for water level forecasting of dungun river in Terengganu Malaysia, *Int. J. Eng. Sci. Technol.*, **4** (2014), 1802–1811. Available from: <http://www.ijest.info/docs/IJEST12-04-04-280.pdf>
5. G. Xu, Y. Cheng, F. Liu, P. Ping, J. Sun, A water level prediction model based on ARIMA-RNN, in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2019, pp. 221–226. <https://doi.org/10.1109/BigDataService.2019.00038>
6. A. N. Ahmed, A. Yafouz, A. H. Birima, O. Kisi, Y. F. Huang, M. Sherif, et al., Water level prediction using various machine learning algorithms: A case study of Durian Tunggal river, Malaysia, *Eng. Appl. Comput. Fluid Mechan.*, **16** (2022), 422–440. <https://doi.org/10.1080/19942060.2021.2019128>
7. D. O. Faruk, A hybrid neural network and ARIMA model for water quality time series prediction, *Eng. Appl. Artif. Intell.*, **23** (2010), 586–594. <https://doi.org/10.1016/j.engappai.2009.09.015>
8. A. S. Azad, R. Sokkalingam, H. Daud, S. K. Adhikary, H. Khurshid, S. N. A. Mazlan, et al., Water level prediction through Hybrid SARIMA and ANN models based on time series analysis: Red hills reservoir case study, *Sustainability*, **14** (2022), 1843. <https://doi.org/10.3390/su14031843>

9. J. Marcela, M. Castillo, J. Manuel, S. Cspedes, H. E. E. Cuchango, Water level prediction using artificial neural network model, *Int. J. Appl. Eng. Res.*, **13** (2018), 14378–14381. Available from: https://www.ripublication.com/ijaer18/ijaerv13n19_45.pdf
10. S. X. Liang, M. C. Li, Z. C. Sun, Prediction models for tidal level including strong meteorologic effects using a neural network, *Ocean Eng.*, **35** (2008), 666–675. <https://doi.org/10.1016/j.oceaneng.2007.12.006>
11. D.-H. Lee, D.-S. Kang, The application of the artificial neural network ensemble model for simulating streamflow, *Proced. Eng.*, **154** (2016), 1217–1224. <https://doi.org/10.1016/j.proeng.2016.07.434>
12. M. K. Akhtar, G. A. Corzo, S. J. van Andel, A. Jonoski, River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information: Case study of the Ganges River basin, *Hydrol. Earth Syst. Sci.*, **13** (2009), 1607–1618. <https://doi.org/10.5194/hess-13-1607-2009>
13. M. Campolo, A. Soldati, P. Andreussi, Artificial neural network approach to flood forecasting in the River Arno, *Hydrol. Sci. J.*, **48** (2003), 381–398. <https://doi.org/10.1623/hysj.48.3.381.45286>
14. J. C. Pagatpat, A. C. Arellano, O. J. Gerasta, GSM & web-based flood monitoring system, in *IOP Conference Series: Materials Science and Engineering*, **79** (2015). <https://doi.org/10.1088/1757-899X/79/1/012023>
15. S. Zhang, C. Zhang, Q. Yang, Data preparation for data mining, *Appl. Artif. Intell.*, **17** (2003), 375–381. <https://doi.org/10.1080/713827180>
16. H. Khalifeloo, M. Mohammad, M. Heydari, Application of different statistical methods to recover missing rainfall data in the Klang River catchment, *Int. J. Innov. Sci. Math.*, **3** (2015).
17. M. N. Noor, A. S. Yahaya, N. A. Ramli, A. M. M. AI Bakri, Mean imputation techniques for filling the missing observations in air pollution dataset, *Key Eng. Mater.*, **594–595** (2014), 902–908. <https://doi.org/10.4028/www.scientific.net/KEM.594-595.902>
18. J.-C. Baltazar, D. E. Claridge, Study of cubic splines and fourier series as interpolation techniques for filling in short periods of missing building energy use and weather data, *J. Solar Energy Eng.*, **128** (2005), 226–230. <https://doi.org/10.1115/1.2189872>
19. R. M. Pattanayak, H. S. Behera, Higher order neural network and its applications: A comprehensive survey, in *Progress in Computing, Analytics and Networking*, (2018), 695–709.
20. F. dela Rama-Liwanag, D. Mostrales, K. Sanchez, R. Tudio, V. Malales, M. T. Ignacio, GIS-based estimation of catchment basin parameters and maximum discharge calculation using rational method of Luinab catchment in Iligan City, 2018.
21. T. Ajayi, D. L. Lopez, A. E. Ayo-Bali, Using artificial neural network to model water discharge and chemistry in a river impacted by acid mine drainage, *Am. J. Water Resour.*, **9** (2021), 63–79. <https://doi.org/10.12691/ajwr-9-2-4>
22. S. Chiewchanwattana, C. Lursinsap, C.-H. H. Chu, Imputing incomplete time-series data based on varied-window similarity measure of data sequences, *Pattern Recogn. Letters*, **28** (2007), 1091–1103. <https://doi.org/10.1016/j.patrec.2007.01.008>
23. M. K. Gill, T. Asefa, Y. Kaheil, M. McKee, Effect of missing data on performance of learning algorithms for hydrologic predictions: Implications to an imputation technique, *Water Resour. Res.*, **43** (2007), W07416. <https://doi.org/10.1029/2006WR005298>
24. G. Mestre, A. Ruano, H. Duarte, S. Silva, H. Khosravani, S. Pesteh, et al., An intelligent weather station, *Sensors*, **15** (2015), 31005–31022. <https://doi.org/10.3390/s151229841>

25. H. R. Maier, G. C. Dandy, Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications, *Environ. Model. Software*, **15** (2000), 101–124. [https://doi.org/10.1016/S1364-8152\(99\)00007-9](https://doi.org/10.1016/S1364-8152(99)00007-9)
26. W. J. Wee, N. B. Zaini, A. N. Ahmed, A. El-Shafie, A review of models for water level forecasting based on machine learning, *Earth Sci. Inform.*, **14** (2021), 1707–1728. <https://doi.org/10.1007/s12145-021-00664-9>
27. O. Kisi, S. Karimi, J. Shiri, O. Makarynskyy, H. Yoon, Forecasting sea water levels at Mukho Station, South Korea using soft computing techniques, *Int. J. Ocean Climate Syst.*, **5** (2014), 175–188. <https://doi.org/10.1260/1759-3131.5.4.175>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)