ATENEO DE MANILA UNIVERSITY

PREDICTIVE MODELING OF MARIKINA RIVER WATER LEVELS ALONG THE STO. NIÑO AND MONTALBAN GAUGING STATIONS USING ARTIFICIAL NEURAL NETWORKS

A CAPSTONE PROJECT SUBMITTED TO
THE GRADUATE FACULTY OF
THE SCHOOL OF SCIENCE AND ENGINEERING
IN CANDIDACY FOR THE DEGREE OF
MASTER IN
DATA SCIENCE

DEPARTMENT OF INFORMATION SYSTEMS
AND COMPUTER SCIENCE

BY
BRYNX JUNIL T. ALEGARBES
NEIL BRYANT A. QUE
JEREMY MARCUS S. TAN

QUEZON CITY, PHILIPPINES
APRIL 2025

The CAPSTONE PROJECT entitled:

PREDICTIVE MODELING OF MARIKINA RIVER WATER LEVELS ALONG THE STO. NIÑO AND MONTALBAN GAUGING STATIONS USING ARTIFICIAL NEURAL NETWORKS

submitted by **Brynx Junil T. Alegarbes, Neil Bryant A. Que, Jeremy Marcus S. Tan** has been examined and is recommended for **Oral Defense**.

PATRICIA ANGELA R. ABU, Ph.D.
Chair

CLARK KENDRICK C. GO, Ph.D.
Adviser

VICTOR ANDREW A. ANTONIO, Ph.D.
Adviser

RAPHAEL A. GUERRERO, Ph.D.
Dean
School of Science and Engineering

The Faculty of the Department of Information Systems and Computer Science, School of Science and Engineering, Ateneo de Manila University
ACCEPTS THE CAPSTONE PROJECT entitled:

PREDICTIVE MODELING OF MARIKINA RIVER WATER LEVELS ALONG THE STO. NIÑO AND MONTALBAN GAUGING STATIONS USING ARTIFICIAL NEURAL NETWORKS

submitted by **Brynx Junil T. Alegarbes, Neil Bryant A. Que, Jeremy Marcus S. Tan** in partial fulfillment of the requirements for the degree of Master in Data Science.

JOHN PAUL C. VERGARA, Ph.D.
Member

ANDREI D. CORONEL, Ph.D.
Member

RAPHAEL B. ALAMPAY, Ph.D.
Member

CLARK KENDRICK C. GO, Ph.D.
Adviser

VICTOR ANDREW A. ANTONIO, Ph.D.
Adviser

RAPHAEL A. GUERRERO, Ph.D.
Dean
School of Science and Engineering

Grade:
Date: April 1, 2025

ABSTRACT

Water level forecasting has become increasingly important given the increased frequency of typhoons and extreme rainfall events worldwide. Accurate water level forecasting is vital in mitigating the effects of flood-related hazards. Using data obtained from the Metro Manila Development Authority and the Philippine Atmospheric, Geophysical, and Astronomical Services Administration, this study aims to create an accurate forecasting model for the water level at the Sto. Niño station along the Marikina River Basin. While deep learning models have demonstrated strong predictive capabilities, their effectiveness depends on the availability of relevant hydrological data and appropriate temporal modeling techniques. In this study, three artificial neural networks were trained and evaluated. The Long Short-Term Memory (LSTM) outperformed the basic time series model and the other artificial neural networks in terms of MSE and NSE in univariate and multivariate cases. The impact of data augmentation and lagged input representation on LSTM networks for water level prediction is then investigated. By incorporating additional data from an upstream station and experimenting with different lag structures, we assess how varying input configurations influence predictive performance. Results indicate that shorter lag values generally yield better performance. Furthermore, Shapley Additive Explanations (SHAP) analysis reveals that previous water levels play the most significant role in predictions, with select rainfall stations also contributing valuable information. These findings emphasize the potential of LSTMs for flood forecasting and demonstrate the benefits of targeted data augmentation and input selection in hydrological modeling.

ACKNOWLEDGMENTS

First of all, we would like to thank our thesis advisers, Dr. Clark Kendrick C. Go and Dr. Victor Andrew A. Antonio for giving us the opportunity to work on this research and for guiding us along the way.

Next, we would like to thank the MMDA, PAG-ASA, DOST, and DPWH for accommodating our requests and providing us with datasets that were an essential component of this research.

Furthermore, we would also like to thank our families and friends for their unwavering support as we went through multiple sleepless nights to complete the research.

Most important of all, we thank God for giving us the wisdom needed to understand what was once an unfamiliar topic to us and the strength to push through in spite of hardships.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Background of the Study

With rapid urbanization and climate change, rainfall events have become increasingly likely to cause flooding [1]. As cities expand and landscapes transform, the natural capacity of the environment to absorb and manage rainfall diminishes, leading to a heightened risk of flooding in urban low-lying areas. In 2022, Typhoon Karding brought rainfalls that highly increased the water level of the Marikina River, exceeding the critical point of 18 m, and caused severe flooding in parts of Marikina that rendered some roads inaccessible [52]. Then recently, in 2024, Typhoon Carina brought heavy rainfall causing the water level to reach 18.3 m and forcing residents to evacuate [7]. It is alarming to know that the difference between the second alarm (water level reaching 16 meters) and the third alarm (water level reaching 18 meters) was only around 4 hours. Because of climate change and continuous improper land use, severe flooding in the Marikina River Basin (MRB) has become more frequent [34]. This phenomenon not only threatens human lives but also poses significant risks to infrastructure, livelihoods, and ecosystems.

Recognizing the urgency and magnitude of these challenges, scientists, urban planners, and policymakers have devised effective strategies to anticipate and mitigate the effects of flood-related hazards. Collaborative efforts have facilitated flood management plans encompassing flood control infrastructure, early warning systems, emergency response protocols, and community engagement initiatives. One of the pivotal advancements in flood management and risk reduction is the development and implementation of flood forecasting and early warning systems [50]. These systems are particularly crucial for urban areas because of their susceptibility to rapid and severe flooding. By combining machine learning techniques, meteorological data, hydrological insights, and computational models, these systems aim to provide timely and accurate predictions of potential flood events.

Currently, Marikina City employs an alarm level system [41], where the water level of the Marikina River under the Sto. Niño bridge is tracked. This does not in any way predict or forecast future water levels. Instead, this is used to monitor flooding by raising an alarm if the water level reaches certain threshold values to propose necessary actions, such as preparation and evacuation, to be undertaken by the residents living in proximity to the river. While the current alarm level system that relies solely on the current water level in Sto. Niño has been of great use to Marikina City, improvements can still be made in the form of more robust water level forecasting methods that can predict future water levels in real-time without the need for extensive up-to-date data on the Marikina River. This would help the government and citizens in making decisions on when to evacuate from high-magnitude floods long before they happen.

Furthermore, predictions of water levels could also help engineers design and construct bridges, dams, levees, and other infrastructure projects to withstand potential flooding events. There is a need to integrate flood risk management into infrastructure development by considering the increased frequency and intensity of floods due to climate change, the limitations of current designs in handling extreme water loads, and how these predictions can inform choices about materials and design features [36]. In fact, the Federal Emergency Management Agency (FEMA) in the United States encourages "freeboard" as a safety factor usually expressed in feet above a flood level for floodplain management. It is an additional height or safety buffer incorporated into the design of structures like levees and dams. Its purpose is to account for various unforeseen factors that could potentially cause flood levels to exceed the calculated heights based on a specific flood size and floodway conditions. These factors may include wave action, obstructions from bridge openings, and the hydrological impacts of urbanization within the watershed area [13]. By including this extra margin, the freeboard helps compensate for uncertainties and unknowns that could contribute to higher-than-anticipated flood heights, thereby enhancing the overall flood protection capabilities of the infrastructure.

In addition, Olsen [36] explained that more accurate forecasting models can provide insights into the potential maximum water levels and flow rates that structures may face during their lifetime, enabling engineers to determine the necessary specifications, such as the height and strength of levees, bridge clearances, dam spillway capacities, and overall structural integrity to withstand expected hydraulic forces. By incorporating these predictions into the design process, infrastructure investments can be protected from the devastating impacts of severe flooding events, resulting in safeguarding lives, property, and economic assets. Moreover, this integration can lead to the development of innovative engineering solutions and adaptive strategies, ensuring the resilience of critical infrastructure in the face of a changing climate and evolving flood patterns.

Given the urgency and the frequency of flooding in areas surrounded by the Marikina River, this study aims to predict the water level at the Sto. Niño and Montalban gauging stations along the MRB using machine learning, specifically, different artificial neural networks.

1.2 Objectives of the Study

The main objective of this study is to predict the water levels at the Sto. Niño and Montalban gauging stations along the Marikina River Basin with the use of artificial neural networks (ANNs) using data obtained from the Metropolitan Manila Development Authority (MMDA). Specifically, the study aims to:

1. Establish ANNs as fundamental components of water level forecasting models in the Marikina River Basin.
2. Determine if ANNs are better forecasting models of the water levels in the Marikina River Basin than the basic time series models.
3. Determine if adding more input features and changing lag values will improve the forecasting models.
4. Provide recommendations and insights based on the study's findings, aiming to guide stakeholders, policymakers, and relevant authorities in enhancing flood preparedness, response strategies, and resilience measures within the study area and similar contexts.

1.3 Significance of the Study

This study addresses critical challenges brought by flooding along the Marikina River. Firstly, the study underscores the urgent need for effective and accurate early warning systems. Secondly, the integration of artificial neural networks represents a groundbreaking approach that offers several distinct advantages. By utilizing ANNs, the study streamlines the forecasting processes, reducing reliance on insufficient data sources that may be inaccessible or unavailable to the public. Moreover, the findings of the study will be beneficial to the following stakeholders:

- **Local Residents.** Residents benefit from access to timely flood alerts, enabling them to initiate timely evacuations and safeguard lives and property.

- **National and Local Government.** Authorities can optimize resource allocation and emergency response strategies, enhancing overall preparedness and minimizing flood-related risks. Policymakers can also use the results to generate effective policies.

- **Future Researchers.** Future researchers can use the methodological framework and data analysis techniques to further advance water level forecasting models for other frequently flooded waterways in the country. Additionally, the findings of the study can provide insights into how the neural network architectures can be improved and modified for different scenarios.

The significance of this study extends beyond theoretical advancements, aiming to deliver practical solutions that value lives, protect infrastructure, and enhance the overall resilience of communities along the Marikina River and similar vulnerable regions nationally and globally.

1.4 Scope and Limitations of the Study

This study focuses on the Marikina River Basin (MRB), as seen in Fig. 1.1, situated in the Greater Manila Area in Luzon island. More specifically, it looks at the section of the Marikina River between Montalban and Sto. Niño stations, the key stations where hourly water level in 2016 and 2017 were tracked and obtained, and the surrounding mountainous area where different rainfall gauging stations are situated.

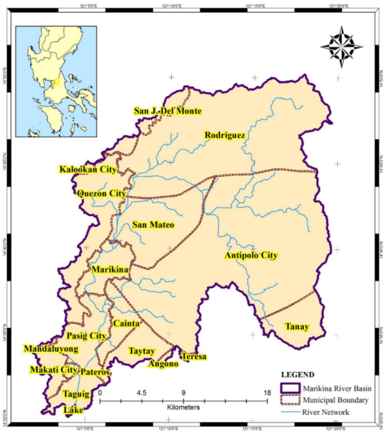


Figure 1.1: Marikina River Basin Map from the Department of Environment and Natural Resources.

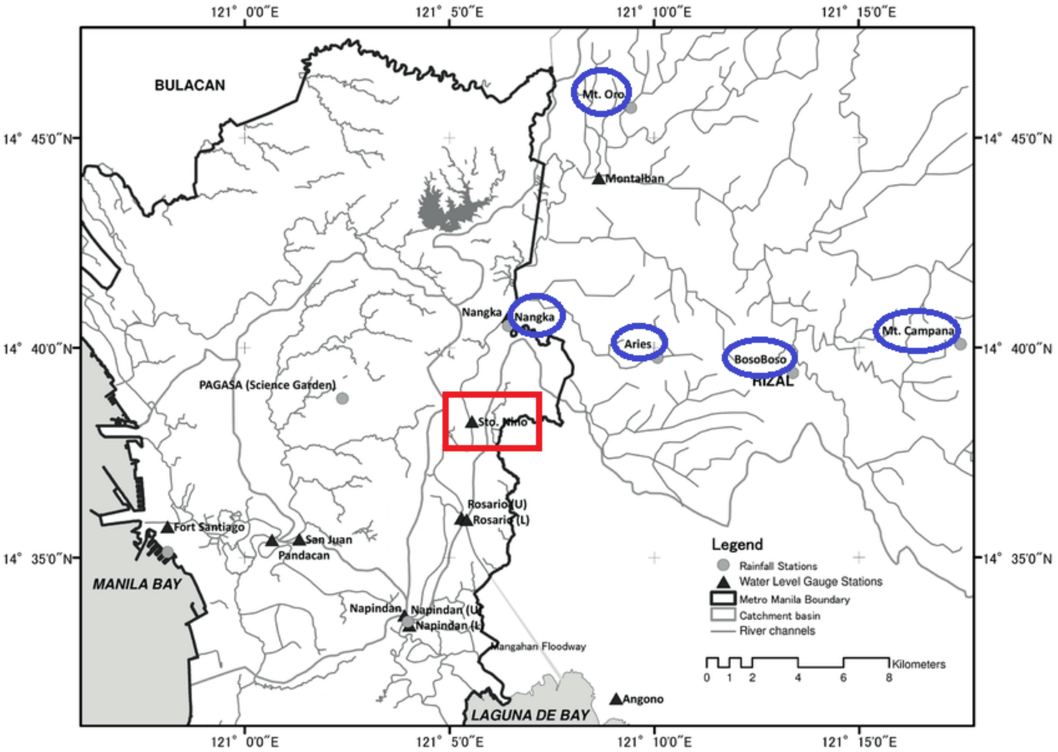


Figure 1.2: Water level and rainfall stations from PAG-ASA and Effective Flood Control Operation System (EFCOS); enclosed in a red rectangle is the Sto. Niño station and in blue circles are the rainfall stations of interest.

Among the numerous rainfall gauging stations found in the MRB, hourly rainfall data in 2016 and 2017 was collected in the Boso-Boso, Mt. Aries, Mt. Campana, Mt. Oro, and Nangka stations, illustrated in Fig. 1.2. The rainfall and water level stations were determined based on previous research on the area [5, 39] and current warnings provided by the Marikina Public Information Office (PIO). Furthermore, the rainfall and water level data were received from the MMDA.

The rainfall in the five aforementioned stations directly causes a rise in water level in the Sto. Niño station as illustrated in Section 3.3.5. However, they are not the only stations in the area that may contribute to a rise in water level. There are more rainfall stations around the MRB, but they have not been used in the Marikina PIO announcements and the researchers were also not able to receive data from the different institutions. Hence, they were excluded from this study, but they still present a possible extension that can be looked into – if their inclusion in the model could improve prediction accuracy.

CHAPTER II

REVIEW OF RELATED LITERATURE

In more advanced flood forecasting, two distinct methodologies are in use: physical models that numerically solve water flow equations [21] and data-driven models that utilize neural networks for making predictions [31]. Physical models employ analytical numerical techniques like automatic differentiation to solve coupled differential equations governing water flow dynamics, sediment transport, and hydraulic interactions within river basins and urban drainage systems [21]. These models integrate factors like topography, land use, soil properties, and rainfall patterns to simulate and predict flood behavior under varying scenarios. Physical models [5, 39] yielded relatively high rates of accuracy. However, physical models require various physical parameters that may not be readily available [37], especially in less developed areas with little to no recording or measuring systems in place such as the Marikina River Basin.

On the other hand, data-driven models can harness the power of machine learning, particularly neural networks, to analyze historical flood data, meteorological observations, satellite imagery, and other relevant datasets [11, 40]. The main difference between these and the physical models is that data-driven models have no knowledge of physical laws or equations that govern a system. Instead, it makes use only of the data provided and draws conclusions purely based on patterns being observed.

2.1 Physical Models

In hydrologic modeling, which includes water level forecasting, the Nash-Sutcliffe Efficiency (NSE) is typically used to evaluate developed models. It compares the predictive ability of hydrologic models with the mean of the observations [16]. The NSE measures three components of the predictions: the correlation, the bias, and the variance. It penalizes both the underestimation and overestimation of streamflows and considers the entire range of streamflow values, as opposed to just looking at the peak flows. The NSE has a range of negative infinity to 1, and a model having an NSE of more than 0.5 is considered to be a satisfactory model, while a negative NSE means that the predictive ability of the model is worse than the mean value of the observations [26]. There have been various studies that have forecasted and modeled flooding dynamics. Previous studies have used models heavily dependent on a lot of physical parameters that were able to satisfactorily simulate water discharge by achieving an NSE of more than 0.5.

In the case of Santillan [39], who conducted research on flood forecasting for the MRB, the software Hydrologic Engineering Center Hydrologic Modeling System (HEC HMS) [18] was used to compute for discharge, which was used as an input into the HEC River Analysis System (HEC RAS) [19] for flow simulation. Different versions of their model achieved NSEs between 0.77 and 0.88, and the HEC HMS was eventually used as a water level forecasting system before it became discontinued. However, this methodology involved providing numerous parameters such as soil and land-cover maps, the initial baseflow, and the river roughness coefficient which are needed in the various components of the models.

Similarly, Badilla made a flood model for the Marikina River Basin [5].

Instead of the software Santillan used, Badilla employed the Hydrologiska Byråns Vattenbalansavdeling (HBV) model from the Swedish Meteorological and Hydrological Institute [45] to simulate discharge and DUFLOW [8] to make the flow calculations. The resulting discharge hydrographs from the models turned out to be accurate, achieving an NSE of 0.9 and only having small differences in terms of peak discharge. But then again, as is the case with physical models, both models used in the study also required a lot of parameters to take different physical phenomena into consideration.

Because of the computational difficulty of these physical-based models and the need for large amounts of physical, topographical, and geometric data [31, 35], other researchers have begun to explore alternative approaches. One such avenue involves harnessing the power of data-driven models, specifically, neural networks and advancements in deep learning, to predict flooding dynamics.

2.2 Artificial Neural Networks

Neural networks are machine learning techniques used in classification, pattern recognition, and a wide variety of other data analysis tasks [6]. For flood forecasting, three types of ANNs are most commonly used: dense neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) [9].

DNNs are the most typical neural networks, composed purely of multiple layers of neurons. As such, it is common practice to utilize this architecture in deep learning research, even for tasks such as time series forecasting and hydrological modeling. Dtissibe et al., for instance, used a multilayer perceptron, a DNN, to predict water discharge along the Gardon d'Anduze River in France using 25 historic flood events [10]. The trained model, after tuning parameters such as the number of layers and neurons to be used through a grid search, was able to accurately predict water discharge and flood peaks with an NSE of 0.98. The suitability of DNNs for a lot of deep learning tasks led to their adoption as one of the architectures used in this study.

CNNs, on the other hand, are typically used for tasks requiring multidimensional data such as object recognition, image processing, and image classification [9, 25]. This is done by mapping a group of features into smaller sizes. Through recent developments, one-dimensional convolution layers were made to accommodate data in sequences, such as temporal data [23]. For temporal data, this meant that features at multiple time steps were taken together to make predictions [46]. As such, CNNs have also become viable for time series forecasting. This was demonstrated by Wunsch et al., who aimed to make forecasts about groundwater levels in central Europe and compare the performance of different types of neural networks, including CNNs [51]. Given inputs of precipitation, temperature, and relative humidity, their CNN achieved an NSE of 0.72 and was found to have only slightly lower predictive powers in comparison with a non-linear autoregressive network with exogenous input, which is a shallow neural network for multivariate time series modeling that have been used successfully for groundwater level forecasting, while achieving this at a significantly faster computation speed. These show that CNNs are viable options for a time series forecasting task. Hence, this architecture will also be used in this study.

Lastly, RNNs are the most commonly used for sequential and time series data due to their ability to remember previous instances in the training set [9, 25]. However, training RNNs is prone to vanishing gradients, wherein the model is not able to update its weights, and exploding gradients, wherein training does not lead to a convergence on the optimal weights [38]. This means that RNNs cannot train on very long sequences. Hence, an improvement over the standard RNN, the long-short-term memory network (LSTM), is now more often used.

Compared to RNNs, the LSTM has mechanisms such as the input, output, and forget gates that control the flow of information between cells, the basic component of an LSTM where the memory is stored. This means that it can remember long-term information better. The LSTM was also used by Wunsch et al., but it had the worst performance among the three architectures used, only achieving an NSE of 0.61. They mentioned that this was unexpected, but that this could have been due to the smaller-sized dataset they used in training the models [51]. Faruq et al. also provided another study where LSTM networks are used for river water level forecasting in Malaysia [11]. After training on multiple years' worth of hourly observations, their model was able to predict the water level of the Klang River with an MSE of 0.042 and coefficient of determination (R^2) of 0.844, although it was not so accurate in forecasting the peak values.

LSTM networks have also been used in water level forecasting in the Philippines. Ibañez et al. used this architecture to forecast reservoir water levels in Angat Dam by training it on 20 years of historical water level data for the univariate model [40]. On the other hand, their multivariate model incorporated rainfall, climate, and irrigation data for training. Furthermore, one-step forecasting, that is, predicting one-time step into the future, and multi-step forecasting, or predicting values multiple time steps into the future, were implemented for their approaches. They found that the two instances of LSTM networks accurately predicted water levels in both short and long terms. Moreover, the univariate model performed better than the multivariate model in one-step forecasting, with an MSE of 0.04 and R^2 of 0.999, while the opposite is true for multi-step forecasting, as the latter achieved an MSE of 10.713 and R^2 of 0.910 for their 30-day forecast. These studies offer a promising basis for the implementation of LSTM networks in forecasting the Marikina River water level as well.

These neural networks are powerful tools for data-driven modeling in forecasting water levels, as they can learn complex patterns and dependencies directly from historical data.

CHAPTER III

METHODOLOGY

3.1 Data Collection and Pre-processing

3.1.1 Dataset

The source of the water level and rainfall for this study was the Metro Manila Development Authority (MMDA). The MMDA handles various rainfall gauging stations around the Philippines. One of its departments, the Effective Flood Control Operation System (EFCOS), regularly collects and reports rainfall and water level data along key stations of the Pasig-Marikina river basin. Hourly rainfall and water level data were requested from MMDA-EFCOS through an electronic freedom of information (eFOI) request. Hourly data was used to maintain the granularity of the data as much as possible and ensure that predictions made would be timely and relevant.

Hourly water level (in m) and rainfall data (in mm) were received from the MMDA in the form of Excel spreadsheet files for each month in each year per station. For rainfall, MMDA sent data from the five rainfall stations: Mt. Oro, Mt. Boso-Boso, Mt. Campana, Aries, and Nangka. All five of these rainfall stations are upstream of the Marikina River Basin and are mostly in the Rizal area. Heavy rainfall from these upstream areas will flow down and may cause flooding at the lower-lying, downstream parts of the MRB, specifically at areas in Marikina City such as Sto. Niño. As for the water level, MMDA sent data for three stations: Sto. Niño, Nangka, and Montalban. This study focused on the water level data from (1) Sto. Niño, as it is the main station tracked by

MMDA-EFCOS and the Marikina City government with regard to their alert level system [41, 5], and from (2) Montalban, as it is also a key station in the upstream that was identified by previous research [5, 39].

3.1.2 Data Pre-processing

The first step in the data cleaning process was to compile the hourly rainfall and water level data for each station into Excel files that contain a year's worth of data so that it can be easily accessed by the neural network models. In the end, each station had two corresponding Excel files — one for 2016 data and another for 2017. The data preprocessing was done using the macros in Excel and the **pandas** library of Python, a fast and powerful package that specializes in structured data manipulation and analysis.

Another thing of note is that 2016 is a leap year, which means that there are 366 days in 2016. To detect seasonality, time series models require a constant time period. To maintain a time period of 365 days, it was decided that February 29, 2016 would be removed.

Upon further inspection of the data, it was found that the water level and rainfall stations in Mt. Aries, Nangka, Mt. Oro, Sto. Niño, and Montalban contained some isolated cases of missing data. Since the duration of the missing data was at most only up to a few hours, linear interpolation was sufficient to estimate the missing data points, as other studies have shown that linear interpolation is the best method for predicting missing data in rainfall datasets [20]. The pandas **interpolate** function was used to automatically perform linear interpolation over all the missing values in the dataset.

Table 3.1: Snippet of the Final Dataset.

(a) First 3 columns with the Sto. Niño water level data.

index	datetime	Waterlevel_Sto_Nino
0	2016-01-01 00:00:00	12.18
1	2016-01-01 01:00:00	12.19
2	2016-01-01 02:00:00	12.19

(b) Last 5 columns with the rainfall data from the 5 stations.

Aries	Boso	Campana	Nangka	Oro
0	1	2	0	0
0	1	1	1	0
1	1	1	0	1

3.1.3 Data Splitting

For the ANNs, the whole dataset was split $50\% - 25\% - 25\%$ into these three sets. While 70% or more of the data is typically used for training, leaving 30% or less for testing [27], a larger percentage of the data in this study was used for evaluating and testing the models. The entire year of 2016 was used for training, the first half of 2017 was used for validation, and the rest of 2017 was used for testing. This split was chosen to ensure that seasonality is taken into account since the second half of the year, the rainy season in the Philippines, is when more frequent extreme rainfall events occur. As such, the whole year of 2016 was used for training the model to take into account both the wet and dry seasons in the Philippines. Note that the data was not randomly split for this study to ensure that the evaluation results were more realistic.

3.2 Univariate Time Series Forecasting by Basic Time Series Models

Hydrologic models employ time series models to analyze and predict changes in river flow, water levels, rainfall, and groundwater levels [17]. Time series models are particularly useful in hydrology because they can capture temporal dependencies and recurring patterns within historical data. By analyzing past observations, these models identify trends and periodic fluctuations that can influence water dynamics over time. Common time series approaches in hydrology include autoregressive (AR) models, moving average (MA) models, and more advanced methods like autoregressive moving average (ARMA) models [28].

Moreover, various hydrologic models used time series models to evaluate volatility such as autoregressive conditionally heteroscedastic (ARCH) and generalized autoregressive conditionally heteroscedastic (GARCH) models. A series $\{Y_t\}$ is an ARCH(m) model if it is of the form

$$\begin{cases} Y_t = \sigma_t \epsilon_t \\ \sigma_t^2 = \alpha_0 + \sum\limits_{i=1}^{m} \alpha_i Y_{t-i}^2, \end{cases}$$

where $\epsilon_t \overset{\text{iid}}{\sim} \text{N}(0,1), \alpha_0 > 0, a_i \geq 0$, and $\sum\limits_{i=1}^{m} \alpha_i < 1$.

While a series $\{Y_t\}$ is a GARCH(m, s) model if it is of the form

$$\begin{cases} Y_t = \sigma_t \epsilon_t \\ \sigma_t^2 = \alpha_0 + \sum\limits_{i=1}^{m} \alpha_i Y_{t-i}^2 + \sum\limits_{j=1}^{s} \beta_j \sigma_{t-j}^2, \end{cases}$$

where $\epsilon_t \overset{\text{iid}}{\sim} \text{N}(0,1), \alpha_0 > 0, a_i, \beta_j \geq 0$, and $\sum\limits_{i=1}^{\max(m,s)} (\alpha_i + \beta_i) < 1$.

In modeling rainfall-runoff relationships, Modarres and Ouarda used GARCH(1,1) and ARCH(2) models [33]. Think of ARCH(2) as looking back at the last two significant "rainfall events" (Y_{t-1} and Y_{t-2}) to understand how much variability (or unpredictability) they contribute to the current runoff. Meanwhile, GARCH(1,1) goes a step further by not only looking at the most recent rainfall event like an ARCH(1) model but also considering the overall trend of variability ($\{\sigma_t^2\}$) over time, in this case, σ_{t-1}^2. It assumes that variability accumulates and persists. In predicting daily streamflow, Wang also used the GARCH(1,1) model [47]. Lastly, in modeling streamflow, Fathian found the ARCH(5) and ARCH(7) models, which consider the 5 and 7 most recent values, respectively, to be the most fitted to their residual series [12].

Because of the scarce studies on water level predictions along the Marikina River Basin, linear time series models will be used as one of the baseline models for this study. The three mean models namely the AR, MA, and the ARMA process will be combined with the volatility models ARCH and GARCH to forecast the water levels at the Sto. Niño station.

3.2.1 Checking Stationarity and Autocorrelation

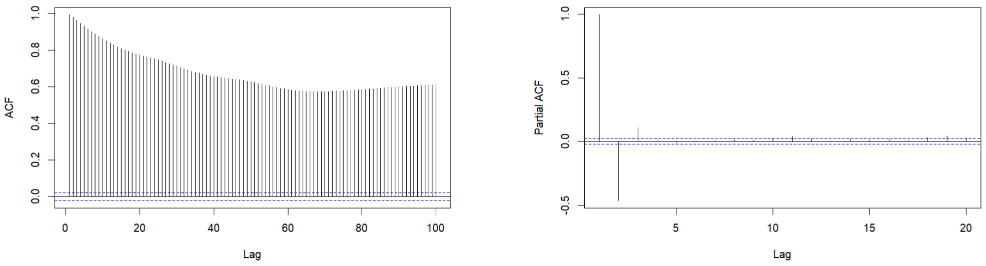
The time series models will use the hourly water level data at the Sto. Niño station for the entire year of 2017. Inconsistencies in mean, variance, and autocorrelation make modeling and interpretation more challenging. Thus, stationarity of the time series data is desired since it offers a stable analytical framework by ensuring that these properties remain consistent over time, simplifying the modeling process [4]. The Augmented Dickey-Fuller test was used through the '`adf.test`' function in `R` to check the stationarity of the data. This tests the null hypothesis that a unit root is in the time series process, which means that the mean of the data is not stationary [32]. Using $\alpha = 0.05$, the obtained $p-$value of 0.01 implies that the null hypothesis is rejected and the conclusion is that the data is stationary.

Meanwhile, autocorrelation reveals patterns within time series data, offering insights into how the variable behaves over time [14]. Recognizing these patterns helps in understanding the underlying influences on the variable, aiding in more accurate predictions and supporting informed decision-making. Autocorrelation is also important in implementing AR and ARMA models. In this study, the Ljung-Box test was conducted using '`Box.test`' in `R` to check whether the data is autocorrelated. This tests the null hypothesis that the time series is not autocorrelated [44]. Using $\alpha = 0.05$, a $p-$value of less than 2.2×10^{-16} was obtained, and so the null hypothesis is rejected. It was concluded that the time series is autocorrelated.

3.2.2 Identifying Candidate Models for the Mean Model

Since the time series data is both stationary and autocorrelated, forecasting can now continue. In identifying candidate mean models, possible lag orders must first be determined. Lag orders determine how many of the previous values will be needed to identify the current value. Determining lag orders is usually done by taking the autocorrelation function (ACF) and partial autocorrelation function (PACF) correlograms of the data [43]. PACF cutting off at lag order p means the AR model to be tested has lag order p. ACF cutting off at lag order q means the MA model to be tested has lag order q. Moreover, combining the results of both correlograms can test an ARMA model with order p, q. The correlograms of the time series data are illustrated in Figs. 3.1a and 3.1b.

It was observed that the ACF does not seem to cut off, while the PACF cuts off at different lag orders. Hence, no MA models can be tested. Furthermore, it was determined that AR(2), AR(3), and AR(11) will be tested. The lag orders 3 and 11 were chosen because the PACF cuts off at those lag orders, and lag order 2 was chosen to check if a more simplified model is possible.



(a) ACF

(b) PACF

Figure 3.1: Correlograms that identify the lag orders for the time series models

Then, the mean models will be evaluated using the ‘`arima`’ function in `R`. The most common metrics in evaluating time series models are the log-likelihood, Akaike information criterion (AIC), corrected AIC (AICc), and Bayesian information criterion (BIC). Best-performing models have the highest log-likelihood and the most negative AIC, AICc, and BIC [43].

3.2.3 Mean Model and Residual Diagnostic

The residuals will then be tested using the Ljung-Box test to determine if they exhibited the ARCH effect, which happens when the standardized residuals are uncorrelated but the squared standardized residuals are autocorrelated. If the mean model residuals exhibit the ARCH effect, then volatility modeling can be performed [43].

3.2.4 Identifying Candidate Models for the Volatility Model

In volatility modeling, a grid search using ‘`garchFit`’ in `R` can be implemented to find suitable values for the lag orders which are the parameters of the GARCH model. In this study, for GARCH(m, s), values of 1 to 4 were assigned to m, while 0 to 4 were assigned to s.

The models will then again be evaluated using the log-likelihood (LLH), AIC, BIC, and two additional metrics, namely, Schwarz information criterion (SIC) and Hannan-Quinn information criterion (HQIC). The best-performing models have the higher LLH and lower AIC, BIC, SIC, and HQIC. Furthermore, the significance of the coefficients can again be checked to see if there is a more simplified model that can be evaluated.

3.2.5 Final Model Diagnostic

After identifying the final model, the residuals can again be tested using the Ljung-Box Test to check if the standardized residuals were autocorrelated or conditionally heteroscedastic, meaning the squares of the residuals were autocorrelated. The standardized and the squared standardized residuals of an ARMA-GARCH model should ideally be both uncorrelated to capture the ARCH Effect and achieve more reliable forecasts [43].

3.3 Artificial Neural Network Modeling

3.3.1 Types of Artificial Neural Networks

For a problem involving time series data such as the one in this study, the models are given data at different time steps t_1, t_2, t_3, $\ldots$, t_n. Using various techniques, the DNN, CNN, and LSTM can all be configured to make predictions for t_{n+1} using the given data. The DNN, illustrated in Fig. 3.2, flattens the two-dimensional input and treats it as a one-dimensional array, which is then the typical input shape of neural networks doing simple regression tasks. On the other hand, the LSTM, as shown in Fig. 3.3, has gate mechanisms that facilitate the forgetting and passing forward of past information as the model goes from time step to time step. Finally, the CNN, visualized in Fig. 3.4, utilizes convolutional kernels to gather information and detect patterns across multiple time steps.

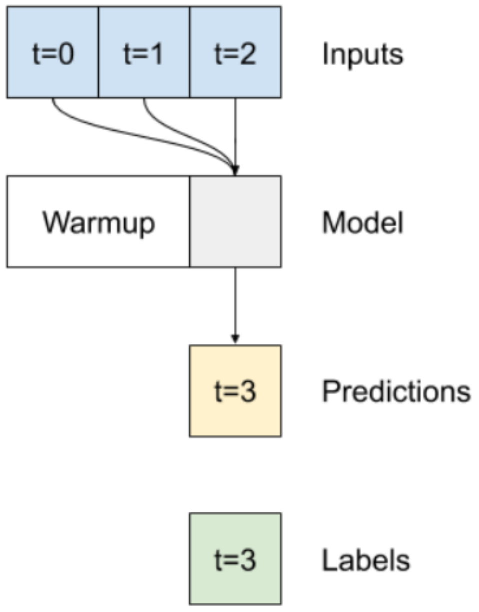


Figure 3.2: Dense Neural Network from [46]

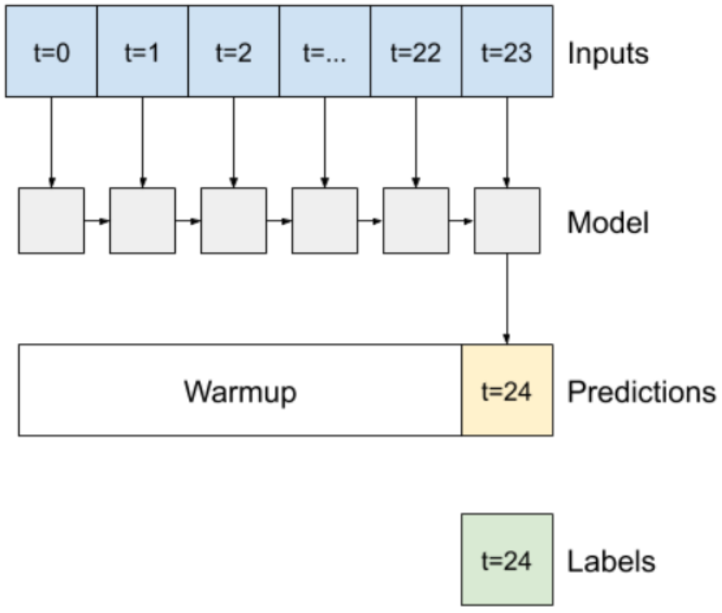


Figure 3.3: Long Short-Term Memory from [46]

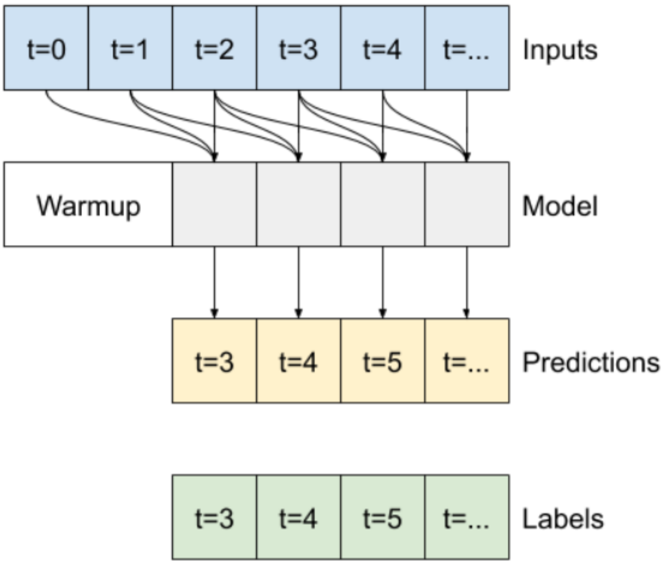


Figure 3.4: Convolutional Neural Network from [46]

In this study, to maintain consistency among the ANNs, each model's hidden layers used the rectified linear unit (ReLU) activation function with 64 neurons in each layer. The DNN model contained three hidden layers, the CNN model contained a single 1-D convolutional layer followed by two hidden layers, and the LSTM model was made up of an LSTM layer followed by two hidden layers as well.

3.3.2 Time Series Forecasting in Neural Networks

Since this study is dealing with time series data, consecutive data points, which represent records across consecutive time steps, are directly related. Specifically, current water level may be dependent on water level and other variables from previous hours. Hence, to take older values into consideration when predicting water level, the inputs for the models must be windows of consecutive time steps of data.

Data windowing was implemented using the **WindowGenerator** class from the TensorFlow documentation [46]. When creating windows, three features must be defined: the input width, label width, and the shift.

1. Input Width - number of consecutive data points used as model input
2. Label Width - number of consecutive data points that the model outputs as its predictions
3. Shift - time difference between the input and the label windows

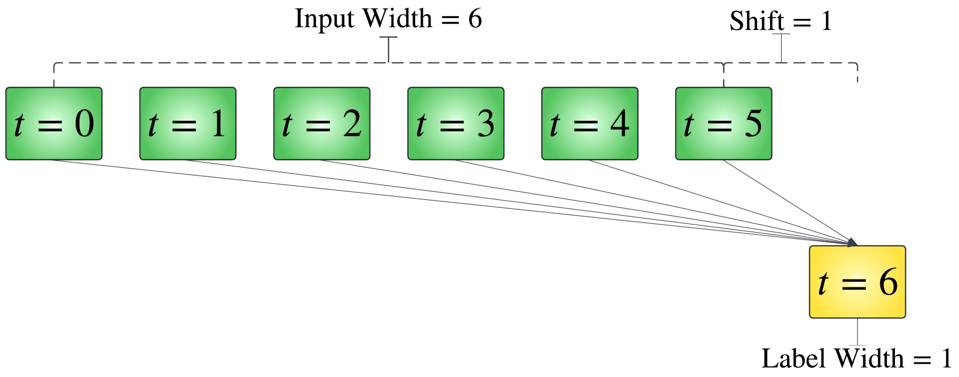


Figure 3.5: A sliding window illustration where the network will take the six previous input data to determine the current value.

To follow the configuration outlined in Section 3.3.1, the windows to be generated has label width and shift of one. This means that one hour of water level one hour later than the most recent data point will be predicted. Initially, an input width of six will be used, but as will be discussed later, this value would be changed. An illustration for this window is shown in Figure 3.5.

3.3.3 Univariate Time Series Forecasting using ANNs

Transitioning from the basic time series models to the different ANNs allows for exploring the latter's ability to model non-linear relationships in time series data. By initially developing univariate models that use the same input (Sto. Niño water levels) as the basic time series models, the results can be comparable, ensuring consistency in the evaluation of performance across the methods.

This approach is justified because while basic time series models rely on predefined statistical assumptions (e.g., stationarity or autocorrelation patterns) [43], ANNs can potentially capture hidden patterns and dynamic relationships without such assumptions. Using a univariate approach at this stage is a controlled way to isolate the performance improvements (if any) due to the modeling architecture itself, rather than differences in input complexity.

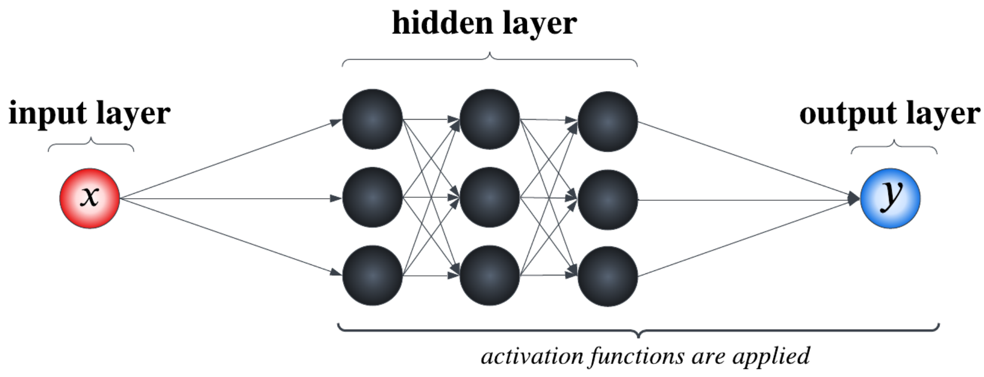


Figure 3.6: A univariate ANN architecture with one input layer, three hidden layers, and one output layer.

The univariate ANN framework focuses on using Sto. Niño water levels as the sole input variable as shown in Fig. 3.6, where x is a vector of consecutive previous water levels at the Sto. Niño Station and y is the predicted water level at the same station after an hour. To enhance the predictive capabilities of the ANN models, the next step involves transitioning to a multivariate ANN framework. Unlike the univariate ANN, which uses only the Sto. Niño water level as input, the multivariate model incorporates additional features. This extension allows the network to use multiple correlated variables that influence water levels. This transition from univariate to multivariate modeling is critical for creating a realistic forecasting model. While the univariate model offers insights into the predictive power of water level alone, the multivariate approach introduces the contextual variables that drive hydrologic processes, thereby improving the adaptability and accuracy of the models.

3.3.4 Granger Causality Test

However, before adding more variables to the time series forecasting model, it can first be determined if other time series are related to the time series of the variable of interest. To this end, the Granger causality is a widely used statistical method for analyzing time series data, specifically the interactions between multiple time series [42]. It utilizes the temporal ordering of data points inherent to time series data. The Granger causality test aims to determine the predictive capabilities of using the lagged values of one time series in predicting the current value of another time series. A time series $\{X_t\}$ is said to be Granger-causal of another time series $\{Y_t\}$ if knowing the history of $\{X_t\}$ reduces the variance in the prediction of $\{Y_t\}$. The Granger causality between two time series processes can either be reciprocal or unidirectional.

Let $\{X_t\}$ and $\{Y_t\}$ be stationary time series. Let $\overline{X_t}$ represent the past history of the time series $\{X_t\}$, where $\overline{X_t} = \{X_{t-i} \mid i = 1, 2, ...\infty\}$. Likewise, let $\overline{Y_t} = \{Y_{t-i} \mid i = 1, 2, ...\infty\}$. Let $\mathcal{P}(X_t|\overline{X_t})$ represent the optimal prediction for X_t given the past history of $\{Y_t\}$.

$\{Y_t\}$ Granger-causes time series $\{X_t\}$ if

$$Var(X_t - \mathcal{P}(X_t|\overline{X_t} \cup \overline{Y_t})) < Var(X_t - \mathcal{P}(X_t|\overline{X_t})).$$

Granger explained that $\{Y_t\}$ Granger-causes time series $\{X_t\}$ if the present value, X_t, can be better predicted using all available past information as compared to if only the history of X_t was used [15].

When performing the Granger causality test, the time series to be considered are modeled using a vector autoregressive model (VAR), where the current value is based on the lagged values of the time series. Since the variances between the predicted value and actual value in the reduced and complete model are compared, an F-test statistic is used [42].

When testing whether the time series $\{Y_t\}$ Granger-causes the time series $\{X_t\}$, the following are the null and alternate hypotheses, and the corresponding test statistic.

H_0 : The time series $\{Y_t\}$ does not Granger-cause the time series $\{X_t\}$.

H_A : The time series $\{Y_t\}$ Granger-causes the time series $\{X_t\}$.

$$F = \frac{RSS_{reduced} - RSS_{full}/(r-s)}{RSS_{full}/(N-r)},$$

where $RSS_{reduced}$ is the residual sum of squares of the reduced model not including the history of $\{Y_t\}$ while RSS_{full} is the residual sum of squares of the full model. The full model has r parameters, the reduced model has s parameters, and the number of observations is N.

Given a level of significance α, if the test statistic F is greater than the critical value F_α with degrees of freedom $r-s$ and $N-r$, then the full model has significantly less residual variance than the reduced model. Thus, the time series $\{Y_t\}$ Granger-causes the time series $\{X_t\}$.

The Granger causality test can be applied to more than two time series. When considering three or more time series, there are two methods for finding the relationship between the time series. The first method is pairwise Granger causality, where bivariate models are fit to all pairwise combinations of two time series [48]. The second method is by doing a conditional Granger causality analysis, where all independent time series are combined in a single vector autoregressive (VAR) model [49].

There is a strict difference between true causality and Granger causality. When X Granger-causes Y, it does not imply that the time series X directly influences time series Y. The definition of Granger causality means that the time series X contains useful information in predicting the values of the time series Y.

Rainfall upstream or within the basin contributes to runoff and, consequently, to variations in river water levels. Hence, to justify the addition of rainfall as input features, both pairwise and conditional Granger causality were tested between the hourly rainfall and water level time series data from 2016 to 2017. The pairwise Granger causality tests were used to determine whether each of the five rainfall time series had a causal or predictive effect on the water level at Sto. Niño. Additionally, the conditional Granger causality test was performed to determine whether the five rainfall time series had a combined causal effect on the water level at Sto. Niño.

The Granger Causality tests were implemented using the **statsmodels** Python package. The **VAR** and **grangercausalitytests** functions from **statsmodels** were used to create and fit a vector autoregressive model and run the F-test for Granger causality. The Granger causality tests aimed to determine if any or all of the five causal time series, the rainfall data, had predictive capability on the dependent time series, the water level at Sto. Niño.

Using the built-in Augmented Dickey-Fuller and Ljung-Box test functions in **statsmodels**, all six of the time series analyzed were found to be stationary and autocorrelated. The pairwise Granger-causality tests between each of the rainfall station's time series data and the water level time series were subsequently performed. Given a significance level of $\alpha = 0.05$, in each of the five Granger-causality tests, the F-test resulted in a $p-$value of < 0.001, which indicates that each of the rainfall time series Granger-causes the Sto. Niño water level time series.

Six time series models were fitted using a vector autoregressive (VAR) model to determine if all five rainfall time series Granger-cause the Sto. Niño water level time series. The level of significance for the Granger test was set at $\alpha = 0.05$. A VAR(2) model was the best-fitting model for the six variables. After performing an F-test, a $p-$value, $p < 0.001 < \alpha$ was obtained, which means that the past data for the rainfall in Mt. Aries, Boso-Boso, Mt. Campana, Nangka, and Mt. Oro Granger-cause the water level at Sto. Niño. The past rainfall data has predictive utility in forecasting the current water level at Sto. Niño. Consequently, a neural network with past rainfall data as input can be created to forecast the water level at Sto. Niño.

3.3.5 Multivariate Time Series Forecasting using ANNs

Adding Rainfall Data as Inputs

Through establishing the Granger causality between the rainfall data and the water level data, the study proceeds to add rainfall data as input features to the neural networks. By incorporating rainfall measurements from these stations, the model gains access to spatially distributed hydrological inputs, improving its ability to capture the complex interactions between rainfall and water levels.

To implement these changes, modifications to the input layer will be made to accommodate multiple features. The **WindowGenerator** will also be adapted to create input-output pairs where each window includes previous water level and rainfall values as inputs, with the current water level as the output.

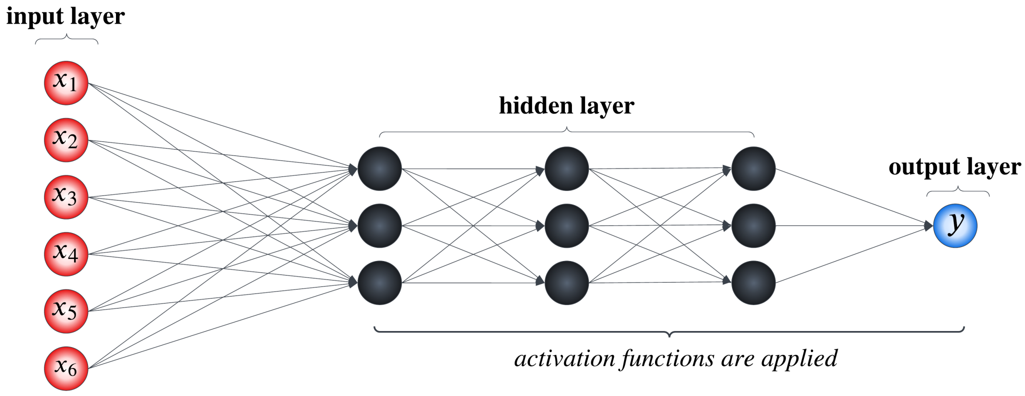


Figure 3.7: Multivariate ANN Architecture with Rainfall Inputs.

The ANN architecture to be used in this multivariate model is shown in Fig. 3.7, where the flow of information is from six input neurons with data labels x_i, $i = 1, 2, \ldots 5$ as the precipitation or rainfall values from the five selected rainfall stations and x_6 as the previous water level at the Sto. Niño Station. The vector of these input data is passed through the input layer, then the hidden layers (denoted in black), and finally to the output layer as current water level (y) in meters.

3.3.6 Dataset Augmentation and Experimenting with Lag Values

In this study, an additional model configuration is explored by incorporating water level data from the Montalban station to increase the data points used for training the models. For the best-performing model that utilizes additional rainfall input features, water level data from the Montalban station is being incorporated to augment the training dataset. Since Montalban is an upstream station within the MRB, its water level data exhibit temporal patterns that are similar to those at the Sto. Niño station. By increasing the number of training data points, the researchers aim to let the model learn a comprehensive range of hydrological behaviors, which is crucial for capturing dynamics and reduce overfitting. This approach will enhance the robustness of the model's predictions and will also forecast water levels for both stations. With the augmented dataset, there will also be another input feature that was added by label encoding the station from where the data point was recorded. Furthermore, to maintain the temporal aspect of the data, the month, day, and hour from the recorded dates and times were extracted and used as input features.

Table 3.2: Snippet of the Augmented Dataset.

(a) First 5 columns; station is 1 for Sto. Niño and 0 for Montalban.

index	month	day	hour	Station
0	1	1	0	0
1	1	1	0	1
2	1	1	1	0

(b) Last 6 columns.

Aries	Boso	Campana	Nangka	Oro	Waterlevel
0	1	2	0	0	21.03
0	1	2	0	0	12.18
0	1	1	1	0	21.03

Augmenting the dataset with the additional station will be beneficial for deep learning models by exposing them to a wider array of scenarios and hydrological variations, improving their ability to generalize predictions. Moreover, this method is scalable and can be extended in future work to include additional stations, further strengthening the forecasting model's potential applications. With the augmented dataset, the study will also experiment with different lag values — specifically 1, 6, and 12 hours — to determine the optimal temporal window for forecasting water levels at the Sto. Niño and Montalban station. Testing these lag values will help assess how short-term versus longer-term historical data influences the model's ability to capture the dynamic patterns in the river's behavior. A 1-hour lag may provide insight into immediate water level trends, while 6- and 12-hour lags could capture broader patterns and cyclical changes due to weather events and hydrodynamics. To explore the impact of different lag values, two approaches will be tested as discussed in the following paragraphs.

Adding Different Lag Values as Input Features

The first method involves adding past water level values as separate input features. In this approach, only historical water level readings are used to forecast the current water level while rainfall data is limited to the most recent hour. For instance, if the lag value is set to 3, the model will receive water level measurements from three hours prior up to one hour prior but will only have the rainfall data from the present hour. This setup aims to determine whether past water levels alone are sufficient indicators of future conditions, simplifying the input structure and reducing the number of variables the model needs to process. While this approach reduces computational complexity and may be effective in cases where water levels exhibit strong trends, it has a notable limitation: it does not account for the delayed effects of rainfall. Since river systems often respond to rainfall with a time lag due to runoff, relying solely on past water level data may lead to less accurate predictions during extreme weather events. However, this method still offers valuable insights, particularly in weather conditions where water level trends are more dynamic but predictable.

Modifying Window Generator

Meanwhile, the second method for implementing different lag values involves modifying the input width of the WindowGenerator, which applies a sliding window technique as discussed in 3.3.2. This method ensures that for a given prediction, the model is trained using past values of all input features, including both water level and rainfall data, based on the specified lag value. For instance, if the lag value is set to 3, the model will receive water level and rainfall values from three hours prior until one hour prior, to predict the current water level. By structuring the data in this way, the model captures temporal patterns across multiple features, allowing it to learn both short-term fluctuations in water levels and delayed effects of rainfall on river conditions. A key advantage of this approach is that it allows the model to recognize patterns in how rainfall influences water levels over time. For example, if a heavy rainfall occurred two hours ago, the model can infer that its impact on water levels may be observed at the present hour due to run off and river flow dynamics. By comparing both approaches, this study wants to determine whether the inclusion of past rainfall data significantly enhances forecasting accuracy or if historical water level patterns alone provide sufficient predictive power.

3.3.7 Model Training

When training the models, the maximum number of epochs was set to 20. The optimization algorithm used was the adaptive moment estimation (Adam) optimizer. The Adam optimizer was used since it has lower memory requirements and has reduced computational time compared to other neural network optimization algorithms [22]. To save computational resources and prevent overfitting, an early stopping condition was included where if the validation loss does not improve over two epochs, the model will already stop training even before it reaches the maximum number of epochs. In this case, the state and parameters of the model at the epoch when the best validation loss was achieved will be saved and used for testing.

3.4 Evaluation Metrics

The performance of the model on the validation and testing set was measured using the following evaluation metrics:

1. Mean squared error (MSE) is the squared error between the predicted and actual values. This is computed using the following formula:

$$\text{MSE} = \frac{1}{2N}\sum_{i=1}^{N}(d_i - y_i)^2,$$

   where N is the number of data points, d_i is the actual value, and y_i is the value predicted by the model.

2. Mean Absolute Percentage Error (MAPE) is the difference between the predicted and actual values represented as a percentage with respect to the latter. This is computed by:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|d_i - y_i|}{|d_i|}.$$

3. Nash Sutcliffe Efficiency (NSE) is a commonly used metric to assess the predictive capabilities of hydrological models. This metric is an improvement over R^2, which measures the linear correlation between the target and output values, as NSE is more sensitive to systematic over and underestimations [3]. The formula of NSE is given by:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{N}(d_i - y_i)^2}{\sum_{i=1}^{N}\left(d_i - \overline{d}\right)^2},$$

where $\overline{d}$ is the mean of the actual values.

In this study, the objective is to minimize the MSE while bringing NSE closer to 1. MSE measures the average squared difference between predictions and observations, which implies that the smaller the MSE, the better. In contrast, NSE assesses how well the model captures variability and patterns compared to the mean of observations. An NSE closer to 1 indicates that the model effectively explains data variability and improves prediction reliability. Moreover, the MAPE will be used to make model performance more interpretable to real-world settings. This will measure the area between the curves to have a better sense of how far the models are from reality. The MAPE metric will only be used for the best models in different set-ups and stations.

3.5 Analyzing Machine Learning Models using SHAP

After model training and evaluation using the augmented dataset, the best models for each station are analyzed and explained using Shapley Additive Explanations (SHAP). SHAP is a general approach to explaining the output of a machine learning model by using its values to describe the importance of each feature to the model's prediction. It uses a game theoretic approach when finding the importance of each feature on the model's prediction [29]. SHAP values are model-agnostic, in that they can be used to interpret any machine learning model, such as linear regression, decision trees, and neural networks.

SHAP values are obtained by allocating credit for a model's output among each of the input features. Using a game theory approach, the change in the model's prediction when the value of a feature is known versus when it is unknown is calculated. The obtained values measure each feature's contribution by calculating the average marginal contribution of a feature across different combinations of known and unknown features. For every individual prediction, a random set of feature values is made unknown to the model. The unknown feature values are replaced by a random sample of feature values from the training set. The model then makes two separate predictions for when a feature is known or unknown, and the difference between the model predictions is calculated. This is done multiple times using different combinations of unknown and known features until the SHAP value for the feature can be estimated by taking the average of the differences.

Two separate SHAP Explainers were created in this study, one for the best-performing model on the Sto. Niño data and another for the Montalban data. The base SHAP Explainer was used in this study to calculate the SHAP values for each feature [30]. To reduce the computational cost, 50 randomly selected batches of time series windows from the training set were used to initialize and estimate the contributions of each of the features on the model output. Since a batch size of 32 was used for the model training, 50 batches is equivalent to a total of 1600 data points. The trained explainer was then applied to the first 50 batches of the unshuffled testing set to obtain the SHAP values for each feature.

Several graphs were then generated using the SHAP values to provide visualizations of the effect of each feature on the model's predictions. A summary plot was created to visualize the positive or negative impact of feature values on the model's predicted output. The relative importance of each feature was also shown using a pie chart, with labels indicating the percentage contribution of each feature on the model. Lastly, for some randomly chosen samples, a waterfall plot was created to show the additive nature of the SHAP values. The SHAP values for each feature are added to the baseline prior expectation of the output, the mean, until it reaches the model output or prediction.

CHAPTER IV

RESULTS AND ANALYSIS

4.1 Univariate Time Series Forecasting by Basic Time Series Models

After modeling the data on the models identified in Section 3.2.2, the metrics for each were recorded and are found in Table 4.1.

Table 4.1: Model performance for candidate models

Model	Log Likelihood	AIC	AICc	BIC
AR(2)	12367.56	-24727.13	-24727.12	-24698.81
AR(3)	12417.99	-24825.97	-24825.96	-24790.58
AR(11)	12431.05	-24836.10	-24836.06	-24744.09

As observed, the AR(2) model performed the worst in all metrics, while the AR(11) model performed the best in all but the BIC. However, given that there is a large difference in the number of coefficients to be estimated between the AR(11) and AR(3) models, i.e. AR(11) will need to estimate 12 coefficients while AR(3) will only need to estimate 4 coefficients, it was determined that the more simplified, less computationally expensive model was to be used. Shumway explained that having higher-order time series models will not exactly significantly improve the forecasts and will more likely result in a model that will run for a long period just to generate the forecasts [43]. Next, the coefficients of the AR(3) model and their corresponding standard errors were obtained. Constructing 95% confidence intervals for each coefficient revealed that all coefficients are significant, and so the AR(3) model is given by

$$X_t = 0.1168 + 1.5035X_{t-1} - 0.6200X_{t-2} + 0.1070X_{t-3} + Y_t,$$

where X_t is the current water level, X_{t-i} is the water level i hour/s ago, and Y_t is the noise term. Furthermore, using the Ljung-Box test on the residuals and an $\alpha = 0.05$, a $p-$value of 0.9028 was obtained, which means that the null hypothesis is not rejected and that the residuals are uncorrelated. The same test and significance level were also used on the squared residuals, which yielded a $p-$value of less than 2.2×10^{-16}, and so the squared residuals are autocorrelated. Hence, the residuals exhibit the ARCH effect and volatility modeling can be performed.

In volatility modeling, the top three models from the grid search are outlined in Table 4.2 and the best model was identified as AR(3)-GARCH(3,1).

Table 4.2: Top 3 AR(3)-GARCH(m, s) models

(m, s)	LLH	AIC	BIC	SIC	HQIC
(3,1)	16104.00	-3.6747	-3.6674	-3.6747	-3.6722
(1,0)	14599.49	-3.3318	-3.3270	-3.3318	-3.3302
(4,2)	13994.79	-3.1926	-3.1838	-3.1926	-3.1896

However, upon checking the significance of the coefficients of the AR(3)-GARCH(3,1), a reduced model AR(1)-GARCH(1,1) can be evaluated. In deciding between the two models, the metrics were again checked as shown in Table 4.3. With a minimal difference in the information criteria, the simplified model AR(1)-GARCH(1,1) was chosen as the final model. Finally, upon checking the significance of the coefficients of the AR(1)-GARCH(1,1), all of the coefficients are now significant.

Table 4.3: Comparison between the original and refined models

Metric	AR(3)-GARCH(3,1)	AR(1)-GARCH(1,1)
LLH	16104.00	14039.95
AIC	-3.6747	-3.2043
BIC	-3.6674	-3.2003
SIC	-3.6747	-3.2043
HQIC	-3.6722	-3.2029

Thus, the final model is AR(1)-GARCH(1,1) with equation:

$$\begin{cases} X_t = 0.0312 + 0.9495X_{t-1} + Y_t \\ Y_t = \sigma_t \epsilon_t \\ \sigma_t^2 = 0.0006 + 0.4580Y_{t-1}^2 + 0.5495\sigma_{t-1}^2 \end{cases}$$

where X_t is the current water level, X_{t-1} is the water level an hour ago, Y_t is the residual term, σ_t^2 is the volatility at time t, and $\epsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$. A forecast horizon of 24 hours or 1 day was used in this study. Using the ‘predict’ function in R, the plot of the forecasts can be seen in Fig. 4.1. The plot also shows confidence intervals of the predictions.

A comparison of the forecasted water level and the actual values taken from the MMDA-EFCOS can be seen in Table B.1. The 1-hour ahead forecast for the water level is 12.4283 meters with a forecasted variance of 0.0015. According to the refined model’s predictions, the water level of Sto. Niño will slowly but continuously decrease in the next 24 hours. Since the predicted values continually decrease, the model may be unable to predict the water level over a long forecast horizon. Furthermore, the model has an MSE value of 0.0116 and an NSE of 0.82 in a 24-hour forecast period.

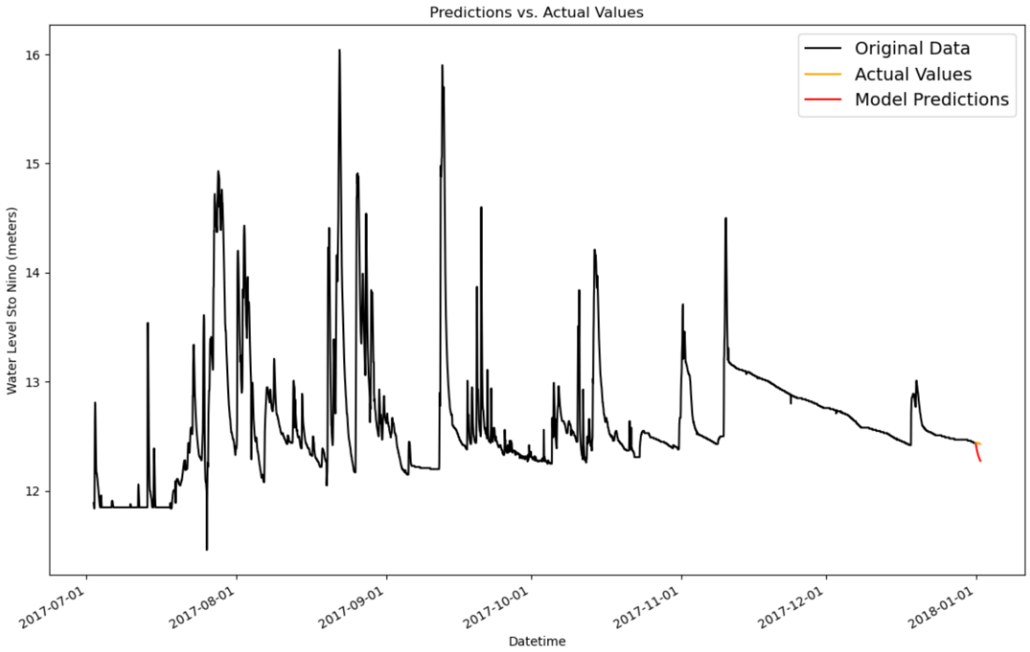


Figure 4.1: A 24-Hour Forecast of the Sto. Niño Water Level

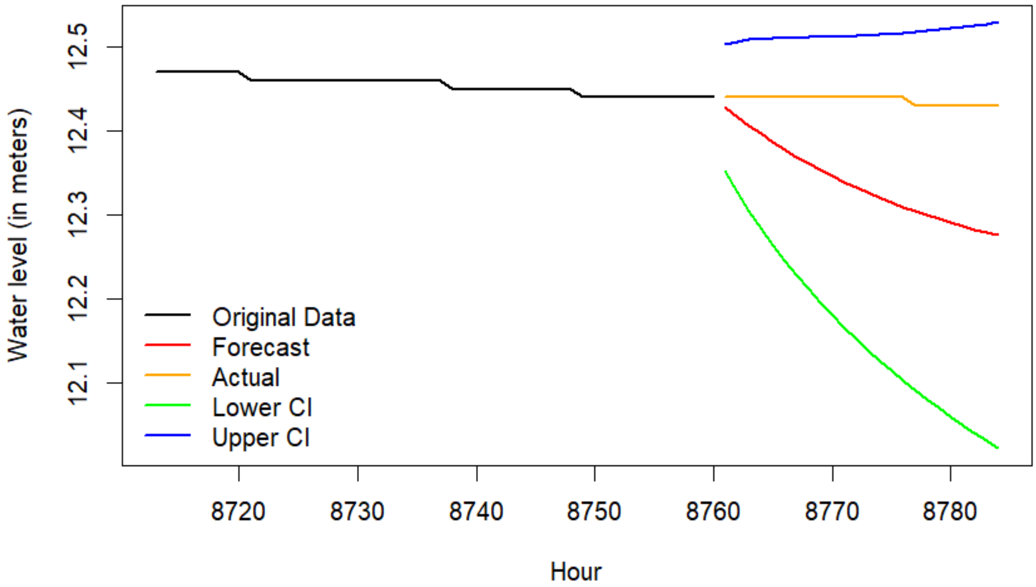


Figure 4.2: Zoomed in 24-Hour Forecast of the Sto. Niño Water Level

Given the final model, the standardized residuals were again tested for ARCH effect. For the standardized residuals, a p-value of less than 2.2×10^{-16} for the Ljung-Box test with $\alpha = 0.05$ was obtained. This means that the standardized residuals are autocorrelated. For the squared standardized residuals using the same test and significance level, a p-value of 0.9178 was obtained. This means that the squared standardized residuals are not correlated.

Since the standardized residuals of the final model were correlated, the GARCH model was unable to fully capture some patterns in the data [43]. The model's residuals were not independent in the final model, so this could lead to less accurate forecasts.

However, this time series model offers an improvement over the Sto. Niño water level alarm system, which only alerts residents and stakeholders once the water reaches a critical threshold. By forecasting water levels, these models enable earlier interventions and preparations. However, time series models face challenges with processing large datasets, such as hourly data with yearly seasonality, where detecting seasonal patterns through ACF and PACF correlograms is difficult. Basic ARMA-GARCH models also struggle to account for the ARCH effects or the volatility of these water levels, which are present in many hydrological datasets. Addressing these limitations may require more advanced models such as the one proposed in this study to strengthen early warning systems.

4.2 Univariate Time Series Forecasting using ANNs

The univariate ANNs which only used the water level at Sto. Niño as its input data was able to achieve overall satisfactory results with all NSE values being greater than 0.5. Out of the three univariate ANN architectures tested, the

Table 4.4: Performance of the Univariate ANNs on the test set

ANN Architecture	MSE	NSE
DNN	0.0462	0.8511
CNN	0.0429	0.8617
LSTM	0.0095	0.9694

LSTM achieved the best performance when evaluated on the test set for both metrics. The LSTM also performed better than the best time series model, the AR(1)-GARCH(1,1), in Section 4.1 given that it has lower MSE and higher NSE. The DNN and CNN, on the other hand, posted higher MSE but lower NSE than the AR(1)-GARCH(1,1). This does not follow the pattern observed in Table 4.4, wherein MSE being lower meant that NSE is higher. Hence, the higher MSE but lower NSE may have happened due to the basic time series model only predicting 24 hours worth of water level values, while the ANNs predicted water level for half a year. These metrics show how ANNs are better water level forecasting models in the MRB than the basic time series models and further improvement of the ANNs are justified.

4.3 Multivariate Time Series Forecasting using ANNs

The results of the three ANN models trained using the rainfall and water level data are shown in Table 4.5. All of the ANN models were able to achieve an NSE of more than 0.6, which means all ANNs are satisfactory models. Overall, the LSTM model performed the best across both metrics, while the second-best-fitting model was the DNN. In comparison with the basic time series model, the DNN and CNN performed worse in both metrics while the LSTM had bet-

Table 4.5: Performance of the ANNs with rainfall inputs on the test set

ANN Architecture	MSE	NSE
DNN	0.1068	0.6556
CNN	0.1179	0.6198
LSTM	0.0251	0.9187

ter NSE but worse MSE. This phenomenon has also been observed in Section 4.2, and the possible explanation on why this happened is the same. As for the comparison between univariate and multivariate ANNs, the metrics achieved by the latter are noticeably worse across all architectures. However, the LSTM suffered the least reduction in performance. This can be attributed to the gate mechanism found in this architecture, where less useful past information can be forgotten and only the more useful information are taken into consideration when making predictions. This can not exactly be done by the other architectures, and so this can partly explain why the multivariate DNN and CNN suffered worse performances.

Since the LSTM performed the best among the three architectures tested, this will be used for training on the augmented dataset which included Montalban station data and added input features.

4.4 Adding Montalban Station and Experimenting with Lag Values

The results in Table 4.6 present the performance of the LSTM model at Sto. Niño and Montalban stations using different lag values, where past water level values were added as input features. At both stations, the model achieves the best performance in both metrics at lag 1. The NSE values are lower and MSE

Table 4.6: Model Performance by Adding Lags as Input Features

	Sto. Niño		**Montalban**	
metric / lag	MSE	NSE	MSE	NSE
1	0.0410	0.8679	0.1011	0.6674
6	0.0535	0.8278	0.1542	0.4930
12	0.0476	0.8467	0.1972	0.3515

values are higher in cases where lag values are higher. This indicates that using additional past water level values as separate features does not necessarily improve the model's performance. Moreover, the model at lag values 6 and 12 are not performing satisfactorily for Montalban with NSE values less than 0.5. The significant drop in NSE values at higher lags at Montalban suggests a negative impact on the model's predictive capabilities. These findings suggest that when using past water levels as separate input features, it is preferable to limit the lag window to avoid unnecessary complexity and performance degradation. This is especially crucial for upstream stations like Montalban, where water levels are influenced by rapid hydrological changes that may not correlate well with older observations.

On the other hand, the results in Table 4.7 illustrate the model performance where past values of all features (rainfall and water level data) were included based on different lag values. Here, the model also performs best in predicting Sto. Niño water levels at lag 1 and even better than in the previous approach. Moreover, the model performs satisfactorily in the Montalban station at all lag values with NSE values being greater than 0.5. This suggests that in

Table 4.7: Model Performance by Modifying Window Generator

	Sto. Niño		**Montalban**	
lag \ metric	MSE	NSE	MSE	NSE
1	0.0399	0.8713	0.1276	0.5806
6	0.0502	0.8380	0.1079	0.6454
12	0.0490	0.8423	0.1095	0.6406

general, the model performs better when both the historical rainfall and water level data are used as input features. This also indicates that rainfall history contains valuable information that enhances model performance beyond using past water levels alone, especially at higher lag values. The key takeaway from this comparison is that windowing all features is generally more effective at handling longer lag values, particularly at Montalban, where higher lags retained good NSE values. Hence, for very short-term predictions, using just the most recent water level as an input feature may be sufficient, whereas past rainfall data proves more useful for longer forecast horizons.

To further compare the models in this section with those that were previously discussed, water level predictions of the best-performing models for each implementation were graphed along with the actual values. The graph for Sto. Niño water level predictions is illustrated in Fig. 4.3.

The Sto. Niño predictions were made by the univariate LSTM in Section 4.2, the multivariate LSTM in Section 4.3, and the model trained on window generators with input width one in Section 4.4. Reflecting the resulting metrics, the predictions from the Section 4.4 model include a number of large

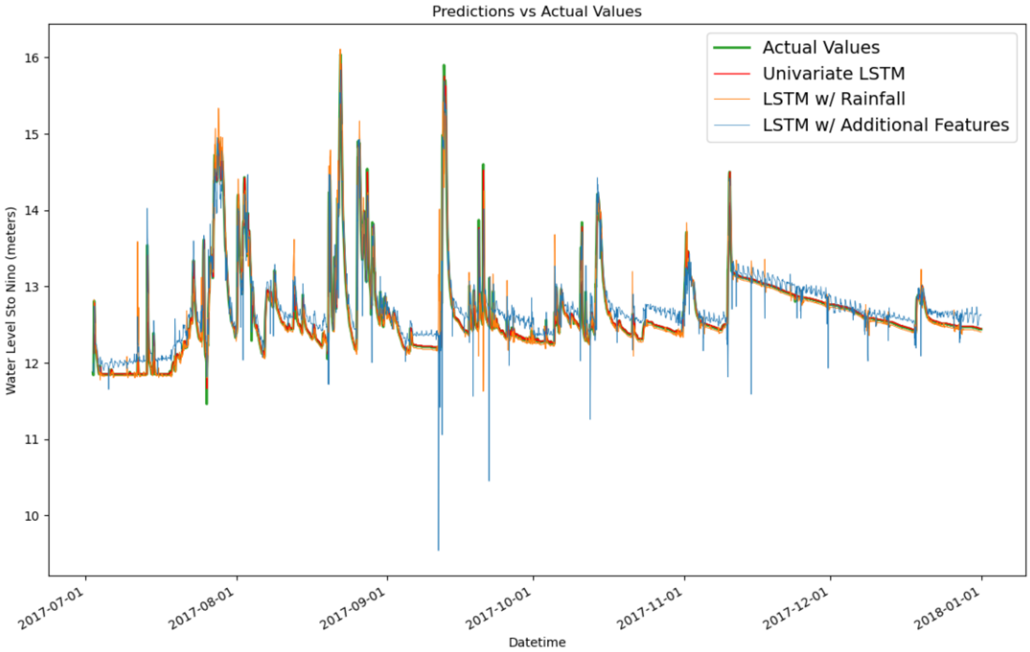


Figure 4.3: Predictions of different models for Sto Niño water level

spikes downwards, which means a lot of severe underestimations, and fluctuations across the graph which do not follow the patterns of the actual data. Meanwhile, the multivariate LSTM trained on rainfall also has some overestimations, but it made better predictions than the former. These over- and under-estimations by the multivariate models can be due to the noise that each new feature introduces to the model. However, it is still worth exploring these as more variables can add to the context and explainability of the model.

A 24-hour snippet including the maximum value attained by Sto. Niño water level was also extracted to observe the ability of the models to predict high water levels. This is crucial as the alarm level system employed by the Marikina City government is based on rising water level under the Sto. Niño bridge. Highlighted in Fig. 4.4 are the highest water level, 16.04m, in the test set and the predictions made by the three models. It can be observed that the multivariate LSTM from Section 4.3 made the closest prediction of 15.995m.

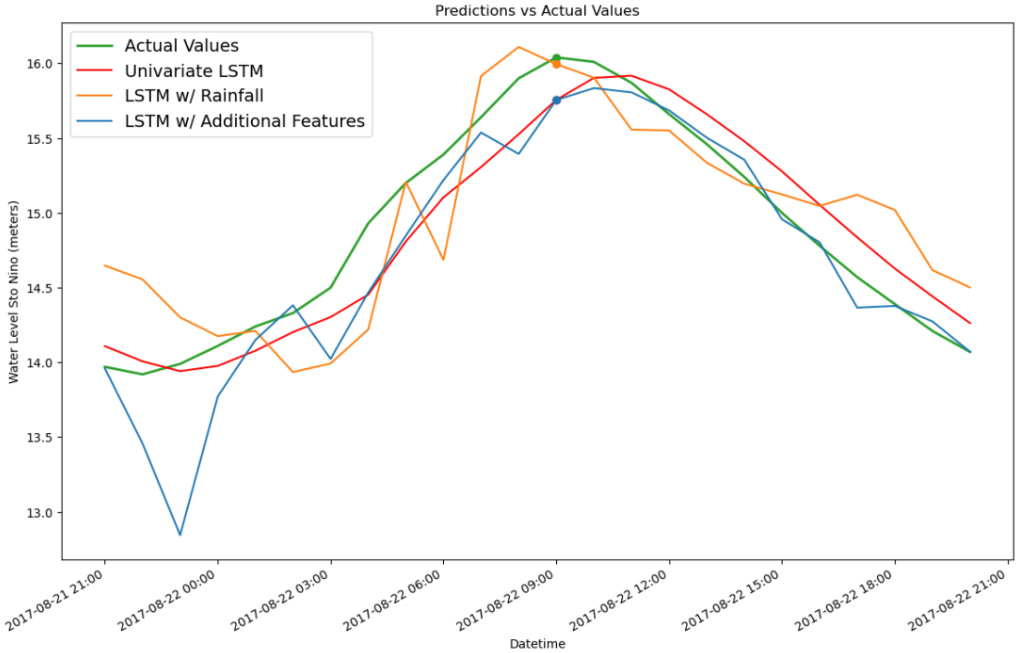


Figure 4.4: A 24-hour snippet from the Sto. Niño predictions

Given that the alarm level 2 instructing residents to evacuate is raised when water level reaches 16m, this prediction could prove to be more useful to various stakeholders compared to the lower predictions of the other models if they determine that this is within an acceptable margin to the threshold.

Table 4.8: The performance of the best models in Sto. Niño based on MAPE

Model	MAPE
Univariate LSTM	0.0029
LSTM w/ Rainfall	0.0056
LSTM w/ Additional Features	0.0123

In Table 4.8 are the MAPE values of the three aforementioned models. Similar to their respective MSE and NSE values in the previous sections, the MAPE for the univariate LSTM was the best, followed by the multivariate LSTM. Overall, the predictions of the three models are, on average, less than 1.5% off from the actual values. Furthermore, the predictions of the univariate LSTM are only 0.3% off, which represents a difference of around 0.03m to 0.05m between the true and predicted values.

Meanwhile, the graph for Montalban predictions is illustrated in Fig. 4.5. The predictions used in the graph were made by the best-performing model from Table 4.6 with a lag of 1 and the best implementation as outlined in Table 4.7 which has an input width of 6.

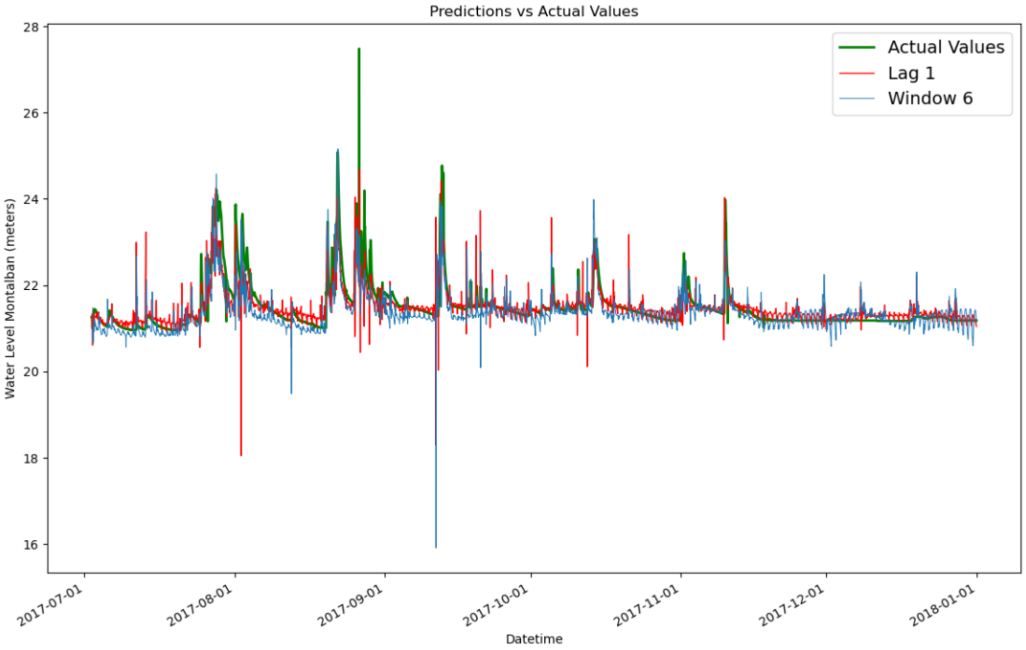


Figure 4.5: Predictions of different models for Montalban water level

From the graphs, it can be observed that both models have a number of large underestimations, with the more severe downward spike coming from the model trained on window generators. Moreover, fluctuations can also be observed from both models, similar to the fluctuations found in the Fig. 4.3. Overall, as can also be inferred from the metrics in Tables 4.6 and 4.7, the models are not able to predict Montalban water level very well. One thing to note in this graph is the large spike in actual water level value at the end of August. Upon closer inspection, it was revealed that water level rose by 5m in one hour, and then went back 5m down in the next. This could point to a data recording issue for that instance, which is why the next highest point was chosen for the 24 hour snippet found in Fig. 4.6.

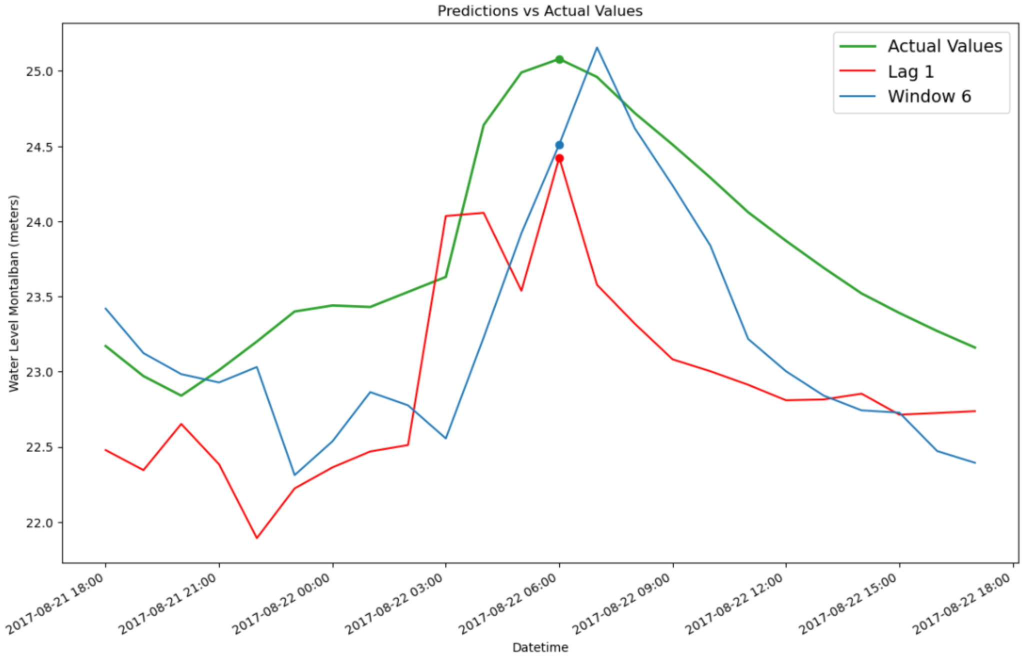


Figure 4.6: A 24-hour snippet from the Montalban predictions

The 24-hour snippet further shows that both models underestimated the peak water level of 25.08m in Montalban. It is interesting to note, however, that the model trained on window generators actually exceeded this value, predicting 25.1566m, but for the following hour. Although there is no alarm level system for the Montalban station, predicting Montalban water level is still crucial as it is in the upstream of the Sto. Niño station and generating accurate forecasts in the former can lead to the development of early warning systems for downstream areas as well.

Table 4.9: The performance of the best models in Montalban based on MAPE

Model	MAPE
Lag 1	0.0080
Window 6	0.0104

Moreover, Table 4.9 shows the MAPE for the two models used in the graphs. It can be observed that while these two models had worse MSE and NSE values compared to those of the three models used in the Sto. Niño water level graph, their MAPE values are comparable. The predictions of both models are around 1% off from actual water level, but this is a 0.2m to 0.25m difference in absolute terms. This illustrates the dependence of MAPE on the scale of the values, which means that it may not be useful when comparing model performance among different stations.

4.5 Feature Importance

The two best-performing models based on both NSE and MSE from Section 4.4 were the model trained on window generators with input width 1 for Sto. Niño and the model with a lag of 1 for Montalban. SHAP values were computed for these models' predictions to enhance model explainability and to identify the features that contribute the most to the models' outputs.

The mean absolute SHAP values of the input features can be seen in Table 4.10. The mean absolute SHAP value represents the absolute contribution of each feature to the model's prediction of the current water level, whether positive or negative. The water level feature in Table 4.10 refers to the previous water level at the station. Since the difference was only one hour for both models, the previous water level feature had the greatest contribution to the prediction of the current water level. Among the rainfall station features, the rainfall at Mt. Oro had the greatest SHAP value, indicating that its rainfall value has the greatest impact on the model prediction compared to the other rainfall stations.

Table 4.10: Mean absolute SHAP values for Sto. Niño and Montalban water level predictions

Feature	Model with window (Sto. Niño)	Model, no windows (Montalban)
Waterlevel	0.4698	0.5258
day	0.0450	0.0318
month	0.0289	0.2461
hour	0.0217	0.0404
Rainfall_Oro	0.0171	0.0669
Rainfall_Aries	0.0135	0.0100
Rainfall_Boso	0.0103	0.0110
Rainfall_Nangka	0.0091	0.0116
Rainfall_Campana	0.0068	0.0126
Station	0.0000	0.0000

The positive or negative contribution of each feature on the model's prediction can be seen through the summary plots in figures 4.7 and 4.8 for Sto. Niño and Montalban, respectively. Each point refers to a single instance in the test set, where red points are above-average feature values while blue points are below-average feature values. For both graphs, the majority of the red points for

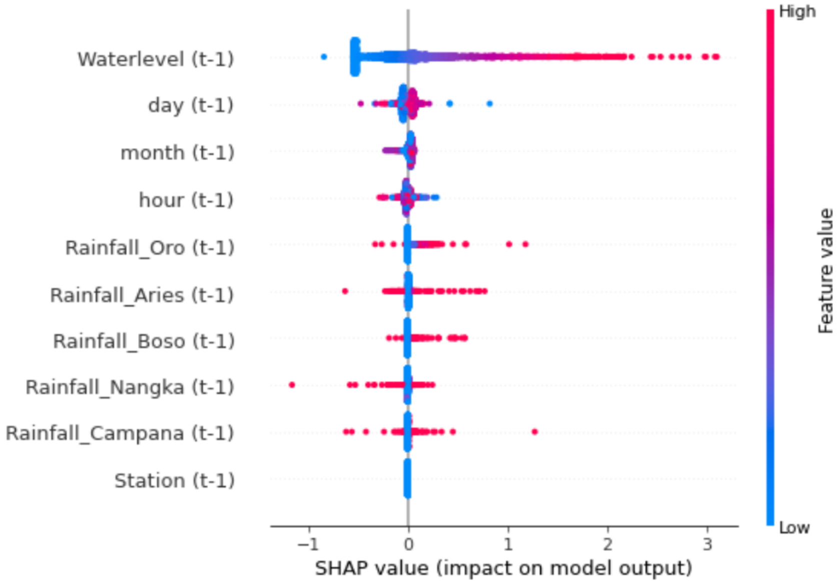


Figure 4.7: Summary Plot of the Individual Predictions of the Model with Window on the Sto. Niño Test Data

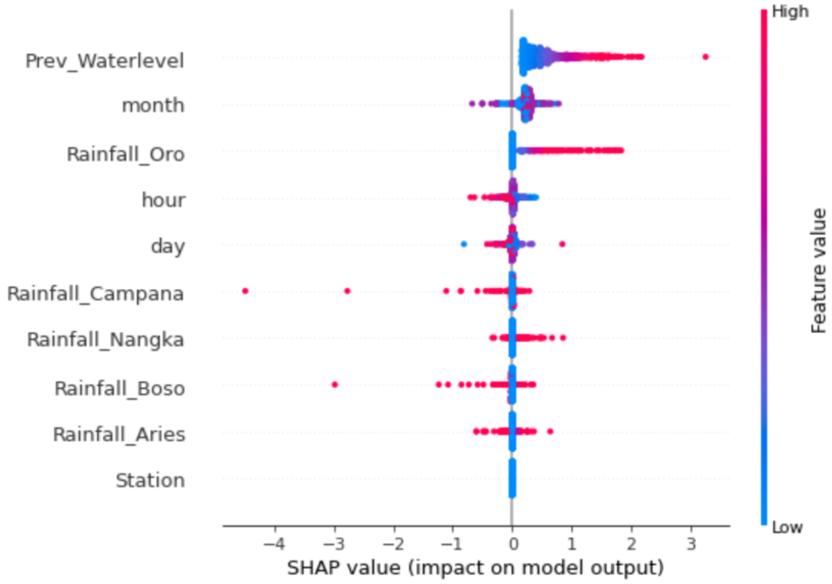


Figure 4.8: Summary Plot of the Individual Predictions of the Model without Windows on the Montalban Test Data

the rainfall features had positive SHAP values, which indicates that the presence of rainfall has a positive contribution to the model's prediction. That is, more rainfall contributes to a rise in water level. Generally, most of the points are in the middle at SHAP value 0 due to the sparsity of the rainfall data.

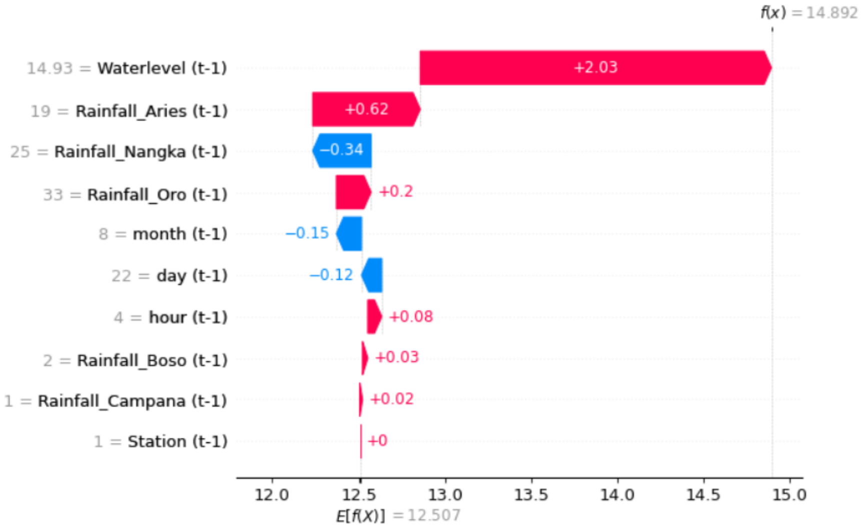


Figure 4.9: Waterfall Plot for the Prediction of Sto. Niño Water Level

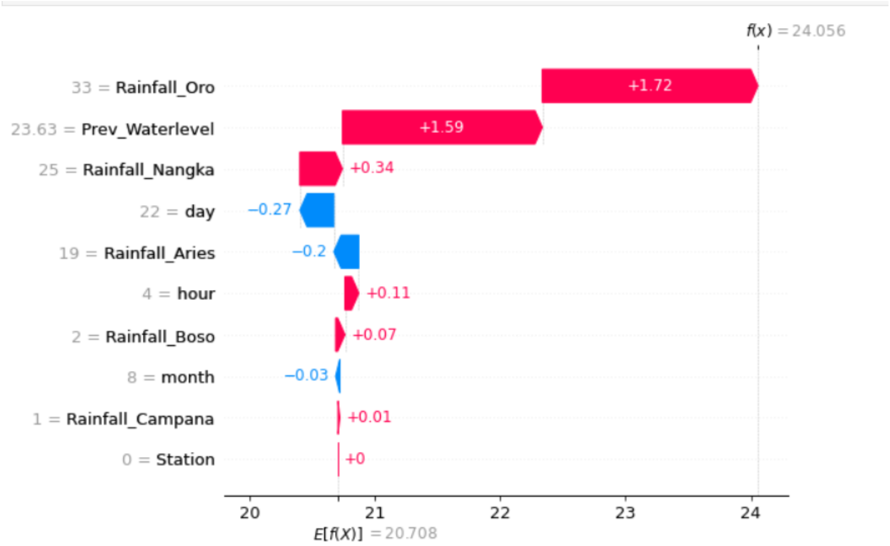


Figure 4.10: Waterfall Plot for the Prediction of Montalban Water Level

Figures 4.9 and 4.10 are waterfall plots for both the Sto. Niño and Montalban predictions during a high rainfall period, selected here as August 22, 2017. The waterfall plot shows which feature values contribute the most to the model prediction. Moreover, the plot shows the additive nature of SHAP values, where the net contribution of each feature is equal to the difference between the model's prediction and the baseline prediction, the mean water level. For the first model, the previous water level contributed the most while rainfall at Mt. Aries had the second largest contribution to Sto. Niño water level prediction. For the second model, rainfall at Mt. Oro surpassed the contribution of previous water level in predicting Montalban water level.

CHAPTER V

CONCLUSION

This study aimed to develop and evaluate predictive models for water level forecasting in the Marikina River Basin using artificial neural networks. A comparative analysis of three ANN architectures – DNN, CNN, and LSTM – demonstrated that the LSTM network consistently outperformed the other models in both univariate and multivariate settings. The LSTM network's ability to capture dependencies in temporal data made it particularly well-suited for hydrological forecasting in this set-up, where water levels are influenced by past observations. Furthermore, the LSTM models outperformed the traditional AR(1)-GARCH(1,1) model, reinforcing the capability of deep learning to model the nonlinear and dynamic nature of water level fluctuations at the Sto. Niño station.

Building upon these initial successes, the study augmented the dataset by incorporating water level data from the upstream Montalban station, effectively increasing the training data available for the model. The results revealed that incorporating 1 hour of past water level and current rainfall data generally yielded better performance. This suggests that LSTM benefits from capturing complex temporal dependencies between rainfall and water levels in the Montalban station.

To further enhance model interpretability, SHAP analysis was employed on the top-performing multivariate models. The analysis confirmed that the previous water level was the most influential predictor for forecasting current water levels, while rainfall data from specific stations (notably Mt. Oro and

Mt. Aries) also played a role, albeit less significant, in shaping the model outputs. The ability of LSTM to integrate these multi-source inputs effectively underscores its suitability for hydrological modeling, as it dynamically adjusts to both immediate and delayed rainfall effects on water level changes. However, the sparsity of the rainfall data meant that its contribution to predictions was lower compared to water level inputs.

While the results establish LSTM as the superior model for this particular forecasting task, its strong performance is likely a function of both the model's architecture and the specific dataset used. The hydrological characteristics of the Sto. Niño and Montalban stations—such as their flow dynamics and response to rainfall—align well with the sequential nature of LSTM, allowing the model to leverage its memory mechanisms effectively. However, the current LSTM configuration may not generalize optimally to other stations within the Marikina River Basin. As a result, further configuration, including feature selection and hyperparameter tuning, would be necessary to adapt the model for other stations.

In summary, this study does not only demonstrate the efficacy of LSTM-based models enhanced with augmented datasets and optimized lag configuration for water level forecasting but also advances our understanding of feature importance within the context of the MRB. The integration of artificial neural networks, data augmentation from additional stations, strategic lag value experimentation, and comprehensive model explainability via SHAP establishes a framework for developing water level forecasting models. These insights are crucial for early warning systems, potentially mitigating flood risks and safeguarding communities along the Marikina River Basin and other river systems in the country.

CHAPTER VI

RECOMMENDATIONS

6.1 Recommendations for Future Work

Our current study has provided valuable insights into implementing time series forecasting for the water level in the Marikina River Basin (MRB). While the ANNs have shown great performance in forecasting water levels in the MRB, there are still additional processes that can be done to further explore and contextualize the ANNs.

In line with this, a different approach to modeling can be undertaken as implemented by other research [31, 35, 10]. Instead of considering the whole dataset as a single entity, significant typhoon and heavy rainfall events from 2016 and 2017 can be extracted separately by using their start and end dates or the dates they enter and leave the Philippine Area of Responsibility. The ANNs could be trained specifically on extreme rainfall events, potentially reducing noise caused by the sparsity of rainfall data. These events can be obtained through documents from NDRRMC and PAG-ASA, and are to be used as training, validation, and testing data.

Further feature selection can also be implemented to determine the best features that can aid in predicting water level. We only used rainfall from five measuring stations as inputs in our models, but there are still other rainfall gauging stations around the MRB such as Pintong Bukawe and Sitio Wawa. These stations can be looked into and tested to determine if they can also be viable features for our models. This targeted approach could reduce noise and improve the model's efficiency by focusing on the most influential inputs. This can also lead to better monitoring and reporting from the Marikina LGU with regard to their early warnings.

Moreover, in addition to the Montalban gauging station, ANNs could be used to predict the water level at other gauging stations upstream of Sto. Niño, such as Tumana and San Jose. SHAP feature importance analysis can help identify which rainfall stations contribute most to the model's predictions. These results can then be compared to the SHAP feature importance findings for Sto. Niño and Montalban presented in this study.

Hyperparameter tuning can still be done for the models. Adjustments such as changing activation functions, network depth, or learning rates may uncover configurations that better capture the complex dynamics of water levels in the MRB. Exploring these avenues will contribute to building a more accurate and reliable forecasting system.

Aside from ANN models, other types of neural networks can be used to forecast the water level at Sto. Niño. An encoder-decoder LSTM was successfully used in a previous study to forecast water level [24]. State-of-the-art deep learning models such as the Hydro-informer, which uses hydrological multi-head attention layers, could be implemented and tested in predicting the water level at Sto. Niño during typhoons [2]. These neural network components could also be compared with the ANNs used in this study to enhance forecasting accuracy.

6.2 Recommendations to Stakeholders

Flooding is a constant threat in the Marikina River Basin, but if we could predict water levels before the danger arrives, then casualties and risks will be mitigated. Our study proves that with the right data and advanced forecasting models, we can do just that. At Sto. Niño, our best-performing model can predict water levels using only the past 6 hours of water level data with accuracy—missing the actual values by just 0.03m to 0.05m (around 0.3%). That's a small difference, but in flood forecasting, those centimeters mean the difference between safety and disaster. This level of precision can empower the city government and local communities to take action before water levels rise.

However, not all areas behave the same way. In Montalban, predictions can be generated using the water level data from the previous hour and the current rainfall data. They were slightly less precise, with an average difference of 0.2m to 0.25m (around 0.8%). This is because the river's flow at this upstream location is more unpredictable, and rainfall from different areas contributes in complex ways. To make forecasts even better, we need more real-time rainfall data from multiple upstream locations—not just the ones we currently use.

While these models offer strong predictive capabilities, they are not perfect. One key limitation is that rainfall data is sparse, and some key stations may not always provide reliable real-time readings. More monitoring stations and better integration of weather radar data could significantly boost accuracy. Another challenge is that our current models are station-specific. This means that they work best for the locations they were trained on, but might not generalize well to other areas along the river. Expanding the study to include more stations across the Marikina River Basin, and fine-tuning the models accordingly, will be crucial in making this forecasting system applicable to a wider area.

Thus, our models are excellent tools for flood preparedness, but they can be game-changers with better data. The Marikina City Government can integrate this forecasting system into their disaster response, sending out timely warnings. Residents, in turn, can use these insights to make informed, life-saving decisions—whether it's moving to higher ground or preparing their homes. With the right investment in data collection and implementation, we can make flood forecasting smarter, faster, and more effective.

BIBLIOGRAPHY

[1] Agonafir, C., Lakhankar, T., Khanbilvardi, R., Krakauer, N., Radell, D., and Devineni, N. A review of recent advances in urban flood research. *Water Security 19* (Aug. 2023). DOI: 10.1016/j.wasec.2023.100141.

[2] Almikaeel, W., Soltesz, A., Cubanova, L., and Barokova, D. Hydro-informer: a deep learning model for accurate water level and flood predictions. *Natural Hazards* (2024).

[3] Althoff, D., and Rodrigues, L. Goodness-of-fit criteria for hydrological models: Model calibration and performance assessment. *Journal of Hydrology 600* (2021). DOI: doi.org/10.1016/j.jhydrol.2021.126674.

[4] Arif, A. Stationary time-series analysis, June 2024. Retrieved November 6, 2024 from https://www.timescale.com/learn/ stationary-time-series-analysis.

[5] Badilla, R. A. Flood modelling in Pasig-Marikina River Basin. Tech. rep., International Institute for Geo-Information Science and Earth Observation, 2008.

[6] Bishop, C. Neural networks and their applications. *Review of Scientific Instruments 65* (1994), 1803–1832. DOI: doi.org/10.1063/1.1144830.

[7] CALIWAN, L. Marikina River reaches 3rd alarm, forced evacuation underway. Retrieved October 28, 2024 from https://www.pna.gov.ph/index.php/articles/1229666.

[8] CLEMMENS, A. J., HOLLY, F. M., AND SCHUURMANS, W. Description and evaluation of program: Duflow. *Journal of Irrigation and Drainage Engineering 119*, 4 (1993), 724–734. DOI: doi.org/10.1061/(ASCE)0733-9437(1993)119:4(724).

[9] CUOMO, S., COLA, V. S. D., GIAMPAOLO, F., ROZZA, G., RAISSI, M., AND PICCIALLI, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing 92*, 88 (2022). DOI: doi.org/10.1007/s10915-022-01939-z.

[10] DTISSIBE, F., ARI, A., TITOUNA, C., THIARE, O., AND GUEROUI, A. Flood forecasting based on an artificial neural network scheme. *Natural Hazards 104* (Aug. 2020), 1211–1237. DOI: 10.1007/s11069-020-04211-5.

[11] FARUQ, A., ABDULLAH, S., MARTO, A., BAKAR, M., SAMIN, AND MUBIN, A. River water level forecasting for flood warning system using deep learning long short-term memory network. *IOP Conference Series Materials Science and Engineering 821* (2020). DOI: doi.org/10.1088/1757-899X/821/1/012026.

[12] Fathian, F., Fakheri Fard, A., Ouarda, T. B., Dinpashoh, Y., and Mousavi Nadoushani, S. Modeling streamflow time series using nonlinear setar-garch models. *Journal of Hydrology 573* (2019), 82–97.

[13] Federal Emergency Management Agency. Freeboard, 2020. Retrieved March 30, 2024 from https://www.fema.gov/glossary/freeboard.

[14] Feldman, K. Exploring the benefits of autocorrelation in time series analysis, Nov. 2024. Retrieved November 11, 2024 from https://www.isixsigma.com/dictionary/autocorrelation/.

[15] Granger, C. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica 37*, 3 (1969), 424–438.

[16] Gupta, H., Kling, H., Yilmaz, K., and Martinez, G. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modeling. *Journal of Hydrology 377* (2009), 80–91. DOI: doi.org/10.1016/J.JHYDROL.2009.08.003.

[17] Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C., and Fenicia, F. Improving hydrologic models for predictions and process understanding using neural odes. *Hydrology and Earth System Sciences 26*, 19 (2022), 5085–5102.

[18] Hydrologic Engineering Center. HEC Hydrologic Modeling System. https://www.hec.usace.army.mil/software/hec-hms.

[19] Hydrologic Engineering Center. HEC River Analysis System. https://www.hec.usace.army.mil/software/hec-ras.

[20] Jafri, I. A. M., Noor, N. M., Ul-Saufie, A. Z., and Suwardi, A. Prediction of missing data in rainfall dataset by using simple statistical method. *IOP Conference Series: Earth and Environmental Science 616* (2020). DOI: 10.1088/1755-1315/616/1/012005.

[21] Jain, S. K., Mani, P., Jain, S., Prakash, P., Singh, V., Tulos, D., Kumar, S., Agarwal, S., and Dimri, A. A brief review of flood forecasting techniques and their applications. *Intl. J. River Basin Management 16*, 3 (2018), 329–344. DOI: doi.org/10.1080/15715124.2017.1411920.

[22] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2017.

[23] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, Daniel J. 1d convolutional neural networks and applications: a survey. *Mechanical Systems and Signal Processing 151* (2021). DOI: 10.1016/j.ymssp.2020.107398.

[24] Kusudo, T., Yamamoto, A., Kimura, M., and Matsuno, Y. Development and assessment of water-level prediction models for small reservoirs using a deep learning algorithm. *Water 14*, 1 (2022).

[25] LeCun, Y., Bengio, Y., and Hinton, G. *Deep learning*, vol. 521. Nature, 2015, pp. 436–444. DOI: doi.org/10.1038/nature14539.

[26] Lin, F., Chen, X., and Yao, H. Evaluating the use of nash-sutcliffe efficiency coefficient in goodness-of-fit measures for daily runoff simulation with swat. *Journal of Hydrologic Engineering 22* (Nov. 2017). DOI: doi.org/10.1061/(ASCE)HE.1943-5584.0001580.

[27] Liu, H., and Haig, E. Semi-random partitioning of data into training and test sets in granular computing context. *Granular Computing 2* (2017), 357–386. DOI: doi.org/10.1007/S41066-017-0049-2.

[28] Liu, Z., Cui, Y., Ding, C., Gan, Y., Luo, J., Luo, X., and Wang, Y. The characteristics of arma (arima) model and some key points to be noted in application: A case study of Changtan Reservoir, Zhejiang Province, China. *Sustainability 16*, 18 (2024).

[29] Lundberg, S. M., and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[30] Lundberg, S. M., and Lee, S.-I. Shap: Explainable machine learning. https://shap.readthedocs.io/en/latest/generated/shap.Explainer.html, 2020. Accessed: 2025-03-10.

[31] MAHESH, R., LIN, Q., AND LEANDRO, J. Physics informed neural network for spatial-temporal flood forecasting. *Climate Change and Water Security* (Jan. 2022), 77–91. DOI: 10.1007/978-981-16-5501-2_7.

[32] MINITAB. Methods and formulas for Augmented Dickey-Fuller Test. Retrieved November 10, 2024 from https://support.minitab.com/en-us/minitab/help-and-how-to/statistical-modeling/time-series/how-to/augmented-dickey-fuller-test/methods-and-formulas/methods-and-formulas.

[33] MODARRES, R., AND OUARDA, T. Modeling rainfall–runoff relationship using multivariate garch model. *Journal of Hydrology 499* (2013), 1–18.

[34] MONJARDIN, CRIS EDWARD, CABUNDOCAN, CLARENCE, IGNACIO, CAMILLE, AND TESNADO, CHRISTIAN JEDD. Impact of climate change on the frequency and severity of floods in the Pasig-Marikina River Basin. *E3S Web Conf. 117* (2019), 5. DOI: doi.org/10.1051/e3sconf/201911700005.

[35] NAZARI, L., CAMPONOGARA, E., AND SEMAN, L. Physics-informed neural networks for modeling water flows in a river channel. *IEEE Transactions on Artificial Intelligence* (Aug. 2022), 1–15. DOI: 10.1109/TAI.2022.3200028.

[36] OLSEN, R., AYYUB, B., WALKER, D., BARROS, A., MEDINA, M., VINSON, T., SAMARAS, C., STAKHIV, E., LEI, W., LOMBARDO, F., TAN, J.,

AND WRIGHT, R. Adapting infrastructure and civil engineering practice to a changing climate. *Adapting Infrastructure and Civil Engineering Practice to a Changing Climate* (Jan. 2015). DOI: doi.org/10.1061/9780784479193.

[37] RAISSI, M., PERDIKARIS, P., AND KARNIADAKIS, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics 378* (Feb. 2019), 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[38] SAK, H., SENIOR, A., AND BEAUFAYS, F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014.

[39] SANTILLAN, J., RAMOS, R., DAVID, G., AND RECAMADAS, S. Development, calibration and validation of a flood model for Marikina River Basin, Philippines and its applications for flood forecasting, reconstruction, and hazard mapping. Tech. rep., University of the Philippines, June 2013.

[40] SEBASTIAN C. IBAÑEZ, CARLO VINCIENZO G. DAJAC, MARISSA P. LIPONHAY, ERIKA FILLE T. LEGARA, JON MICHAEL H. ESTEBAN, AND CHRISTOPHER P. MOTEROLA. Forecasting reservoir water levels using deep neural networks: A case study of Angat Dam in the Philippines. *Water 14*, 1 (2022), 34. DOI: 10.3390/w14010034.

[41] SERAFICA, R. Guide to Marikina River's alarm level system, 2017. Retrieved January 12, 2024 from https://www.rappler.com/environment/disasters/181894-guide-marikina-river-alarm-level-system/.

[42] SHOJAIE, A., AND FOX, E. Granger causality: A review and recent advances. *Annual review of statistics and its application 9 1* (2021), 289–319.

[43] SHUMWAY, R., AND STOFFER, D. S. *Time series analysis and its applications : with R examples*. Springer, 2017.

[44] STATISTICS HOW TO, Oct. 2024. Retrieved November 10, 2024 from https://www.statisticshowto.com/ljung-box-test.

[45] SWEDISH METEOROLOGICAL INSTITUTE. Hydrologiska Byråns Vattenbalansavdeling, 2015. https://www.smhi.se/en/publications/ the-hbv-model-its-structure-and-applications-1.83591.

[46] TENSORFLOW. Time series forecasting, 2024. Retrieved March 28, 2024 from https://www.tensorflow.org/tutorials/struc- tured_data/time_series.

[47] WANG, H., SONG, S., ZHANG, G., AND AYANTOBOC, O. O. Predicting daily streamflow with a novel multi-regime switching arima-ms-garch model. *Journal of Hydrology: Regional Studies 47* (2023), 101374.

[48] WANG, X., CHEN, Y., BRESSLER, S., AND DING, M. Granger causality between multiple interdependent neurobiological time series: Blockwise

versus pairwise methods. *International journal of neural systems 17 2* (2007), 71–8.

[49] WEN, X., RANGARAJAN, G., AND DING, M. Multivariate granger causality: an estimation framework based on factorization of the spectral density matrix. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371*, 1997 (2013), 20110610.

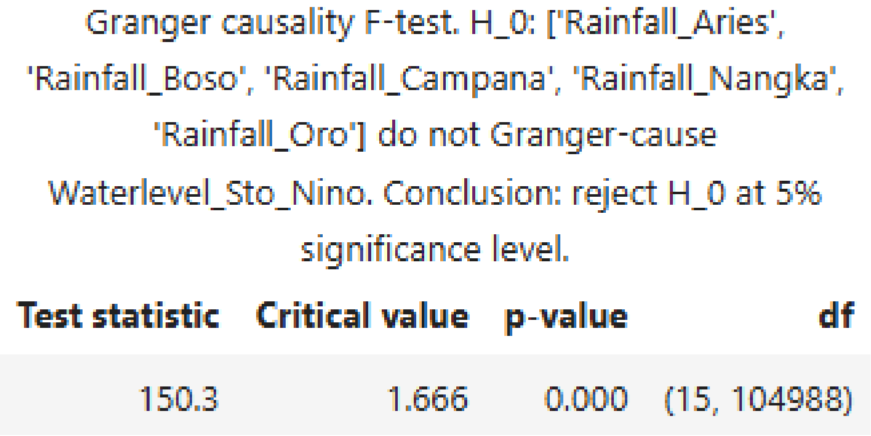
[50] WILLIAMS, L., ARGUILLAS, M., AND ARGUILLAS, F. Major storms, rising tides, and wet feet: Adapting to flood risk in the Philippines. *International Journal or Disaster Risk Reduction 50* (Nov. 2020). DOI: 10.1016/j.ijdrr.2020.101810.

[51] WUNSCH, A., LIESCH, T., AND BRODA, S. Groundwater level forecasting with artificial neural networks: A comparison of long short-term memory (lstm). convolutional neural networks (cnns), and non-linear autoregressive networks with exogenous input (narx). *Hydrology and Earth System Sciences 25*, 3 (2021), 1671–1687. DOI: doi.org/10.5194/hess-25-1671-2021.

[52] YALAO, K. 5,024 families evacuated in Marikina City amid Typhoon Karding, 2022. Retrieved February 16, 2024 from https://mb.com.ph/2022/09/26/5024-families-evacuated-in-marikina-city-amid-typhoon-karding/.

APPENDIX A

Granger-Causality Test



Granger causality F-test. H_0: ['Rainfall_Aries', 'Rainfall_Boso', 'Rainfall_Campana', 'Rainfall_Nangka', 'Rainfall_Oro'] do not Granger-cause Waterlevel_Sto_Nino. Conclusion: reject H_0 at 5% significance level.

Test statistic	Critical value	p-value	df
150.3	1.666	0.000	(15, 104988)

Figure A.1: Granger Causality F-test result

APPENDIX B

AR(1)-GARCH(1,1) Forecasts

Table B.1: AR(1)-GARCH(1,1) Forecasts

DateTime	Actual	Prediction	Mean Error	σ^2
2018-01-01 00:00:00	**12.44**	12.4283	0.0384	0.0015
2018-01-01 01:00:00	**12.44**	12.4173	0.0584	0.0021
2018-01-01 02:00:00	**12.44**	12.4067	0.0760	0.0027
2018-01-01 03:00:00	**12.44**	12.3967	0.0924	0.0033
2018-01-01 04:00:00	**12.44**	12.3873	0.1079	0.0040
2018-01-01 05:00:00	**12.44**	12.3783	0.1228	0.0046
2018-01-01 06:00:00	**12.44**	12.3697	0.1372	0.0052
2018-01-01 07:00:00	**12.44**	12.3616	0.1511	0.0059
2018-01-01 08:00:00	**12.44**	12.3539	0.1646	0.0065
2018-01-01 09:00:00	**12.44**	12.3466	0.1777	0.0072
2018-01-01 10:00:00	**12.44**	12.3396	0.1905	0.0078
2018-01-01 11:00:00	**12.44**	12.3330	0.2029	0.0085
2018-01-01 12:00:00	**12.44**	12.3268	0.2151	0.0091
2018-01-01 13:00:00	**12.44**	12.3208	0.2269	0.0098
2018-01-01 14:00:00	**12.44**	12.3152	0.2386	0.0105
2018-01-01 15:00:00	**12.44**	12.3098	0.2499	0.0112
2018-01-01 16:00:00	**12.43**	12.3047	0.2611	0.0118
2018-01-01 17:00:00	**12.43**	12.2999	0.2720	0.0125
2018-01-01 18:00:00	**12.43**	12.2953	0.2827	0.0132
2018-01-01 19:00:00	**12.43**	12.2909	0.2932	0.0139
2018-01-01 20:00:00	**12.43**	12.2868	0.3036	0.0146
2018-01-01 21:00:00	**12.43**	12.2829	0.3137	0.0154
2018-01-01 22:00:00	**12.43**	12.2791	0.3237	0.0161
2018-01-01 23:00:00	**12.43**	12.2756	0.3336	0.0168