FORECASTING WATER-LEVEL FLUCTUATION IN WATER-SUPPLY DAMS OF THE AUCKLAND AND WAIKATO REGIONS

Pramith Waidyaratne
David Phillips

https://doi.org/10.34074/proc.2206010

Business and Infrastructure

This publication may be cited as:

Waidyaratne, P., & Phillips, D. (2022). Forecasting Water-Level Fluctuation in Water-Supply Dams of the Auckland and Waikato Regions. In E. Papoutsaki and M. Shannon (Eds.), *Proceedings: Rangahau Horonuku Hou – New Research Landscapes, Unitec/MIT Research Symposium 2021, December 6 and 7* (pp. 131–160). Auckland: ePress, Unitec, Te Pūkenga. https://doi.org/10.34074/proc.2206010

Contact:
epress@unitec.ac.nz
www.unitec.ac.nz/epress/
Unitec, Te Pūkenga
Private Bag 92025, Victoria Street West
Auckland 1142
Aotearoa New Zealand

ISBN 978-1-99-118340-8

ePress
Research with impact
Unitec
Te Pūkenga

ABSTRACT

Dams play a vital role in supplying fresh water to many cities all over the world. With increasing pressure on and demand for natural resources, water supply remains a scarce resource worldwide. During times of uncertainty, predicting the future availability of water supply by considering various hydrometric and anthropogenic variables will provide a framework for future scenario forecasting and a model-based approach to sustainable water management. To this end, this project proposes a multivariate time-series analysis and forecasting model to both analyse and forecast daily water-level fluctuations in three water-supply dams: Upper Nihotupu, Waitākere and Mangatangi, located in the Tāmaki Makaurau Auckland and Waikato regions of Aotearoa New Zealand.

KEYWORDS

Environment, dams, water supply, forecasting, sustainable water management

INTRODUCTION

Scarcity and increasing demand for natural resources is an inevitable outcome of a growing population faced with the uncertainties of climate change. Municipal water supply represents one such challenge. In April 2020, the total storage in all Tāmaki Makaurau Auckland dams dropped to below 50% for the first time in 25 years (Morton, 2020). With long dry spells and lower than average rainfall, the city enacted Stage 1 water restrictions in the following month, May 2020, which consequently resulted in loss of revenue for businesses and further increased funding towards capital projects aimed at water-supply resilience (Watercare, 2020). Due to the volatility and intermittent characteristics of hydro-climatic variables, reservoir operation and managing water supply is inherently difficult to manage and quantify. Faced with growing climate uncertainties, studying the relationships between hydrological processes, particularly during times of climatic extremes, provides a crucial step in efficiently leveraging finite resources. In the Tāmaki Makaurau Auckland region, smart water networks (e.g., SCADA) are currently used for supply operations, with daily abstraction flows (i.e., outflows for treatment) from reservoirs being carefully managed to balance supply with existing reservoir availability. While these smart networks aim to efficiently model existing supply to meet demand, a reliable forecast of future supply is required for more proactive resource allocation.

This research project focused on three separate water-supply dams supplying the Tāmaki Makaurau Auckland region – Upper Nihotupu, Waitākere and Mangatangi – and proposes several established statistical and machine learning (ML) models to both analyse and forecast the daily water levels of these dams. Specifically, the proposal aims to explain and forecast dam water levels as a function of hydrological and anthropogenic variability.

The hydrological balance model

The dam level at any time-step (t) is an outcome of complex and related interactions between natural hydro-meteorological and anthropogenic variables. Therefore, in a multivariate forecasting task to predict water level, a range of variables is required to be used as inputs to the models. Forecasting daily fluctuations will therefore involve the careful selection of these variables (or features) and, subsequently, obtaining time-series datasets for each feature in the form of separate time-series datasets.

Consider a generalised water balance equation for a given watershed (Equation 1):

$$P = R + ET + \Delta S (1)$$

Where P is the precipitation, R is the runoff, ET is the evapotranspiration and ΔS is the change in storage (Sposób, 2011). Equation 1 is a general representation of the water cycle for a closed watershed. In a dammed reservoir, this equation can be further summarised into Equation 2:

$$\Delta SR = P + \Delta R - ET + Ab (2)$$

Where ΔSR is the change in reservoir level, P is the precipitation, ET is the evapotranspiration, ΔR is the change in runoff and Ab is the abstraction of water for consumption. This causal relationship between in hydrological and various anthropogenic variables is summarised below in Figure 1. This hydrological balance model will serve as the basis for selecting input features that will be used for the prediction task.

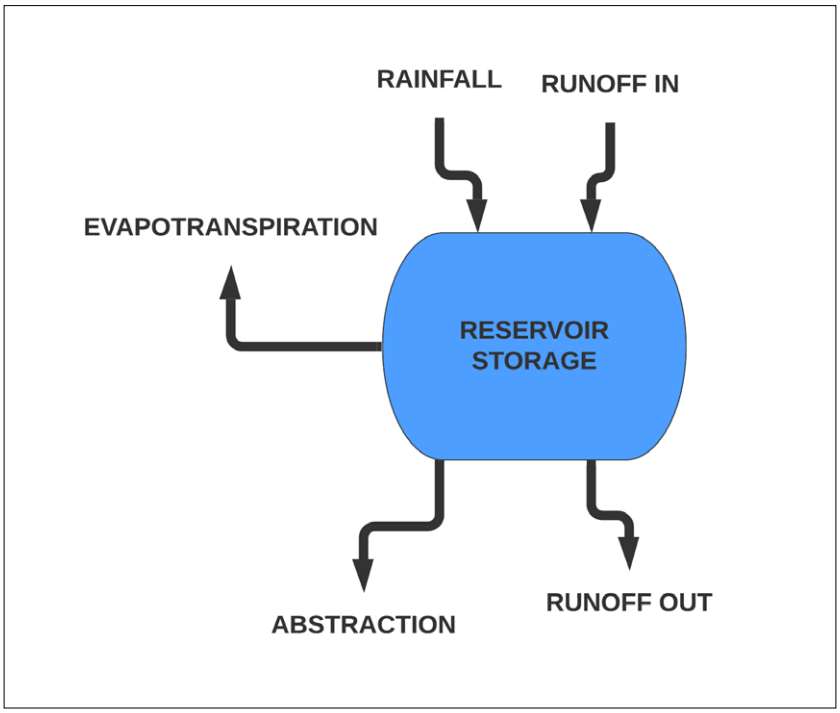

Figure 1. Hydrological balance model.

Machine learning (ML) and artificial neural networks (ANNS)

A broad range of research has been conducted on ML and deep learning (DL), and their applicability to hydrological modelling. Bowden et al. (2005) analysed the structure of ANNs and the validity of the input variables to model hydrological outputs, while more recently Anindita et al. (2016), Filho et al. (2020) and Hu et al. (2018) have all used either ML and/or DL models for the purpose of predicting reservoir water levels and runoff. Perhaps the most relevant pieces of literature on the use of recurrent neural networks (RNNs) for hydrological modelling have been made by Choi et al. (2018) and Kratzert et al. (2019), in which each have successfully used a variation of RNNs, a long short-term memory (LSTM) model, for water level and rainfall run-off predictions. Both papers show reasonable degrees of statistical accuracy with predicting water levels or runoff as a function of temporal hydrological variability. Kratzert et al. (2019) in particular found that a single LSTM model calibrated regionally can perform better than a single LSTM model trained specifically for each catchment.

A commonly used linear multivariate forecasting model is the vector autoregression (VAR) model. VAR is a classical statistical model that was initially used for modelling econometric time-series (Sims, 1980); however, VAR models have since been used as multivariate models in climate and hydrological sciences. Kumar et al. (2009) used a VAR to model the interdependence among O_3, NO, NO_2, and various other volatile organic compounds in Delhi, while Zhao et al. (2019) used a VAR model to measure the runoff response with respect to various contributing parameters in the Loess Plateau, China. Abdallah et al. (2020) presented a short-term weather forecasting model using VAR, and found VAR models to be generally more accurate than more-traditional forecasting models.

Classical hydrological process models depend, to a large degree, on formulating a relationship between catchment characteristics (i.e., area, slope, soil class, vegetation cover, etc.) and the various hydrometric variables (precipitation, solar radiation, wind, etc.). However, most catchments are highly heterogeneous and variations in catchment characteristics are often unknown or not recorded (e.g., soil characteristics, soakage performances, etc.). Moreover, hydrological interactions often take place underground, where, to a large extent detailed information is unavailable (Kratzert et al., 2019). In this context the classical process-based model attempts to describe a system that is relatively unknown.

In contrast, ML and artificial neural network (ANN) models aim to predict an output based on available historical data (e.g., precipitation, temperature, radiation, etc.) and do not necessarily rely on describing a catchment and its underlying processes.

Fundamentally, ML and ANN models are data-driven models that 'learn' patterns and variations in those data based on statistical modelling. To that end, the model utilises a learning algorithm that compares predicted value (ŷ) with the actual values (y) and aims to reduce the difference between the two. This is often achieved by reducing a loss (or cost) function. ANNs achieve this through a process known as back-propagation, which involves adjusting certain parameters (e.g., weights and biases) in the algorithm via the chain rule. This is achieved by minimising the gradient of the cost function through a process known as gradient descent (Goodfellow et al., 2016). Figure 2, below, provides a visual understanding of a typical ANN architecture.

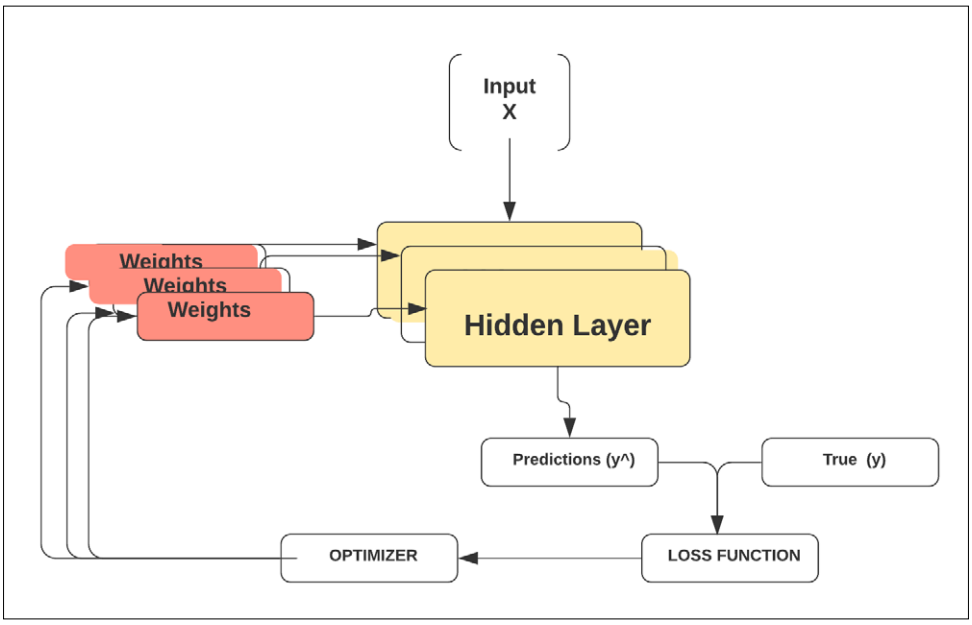

Figure 2: Architecture of a general ANN.

ANNs are unique in that they have hidden layers built into the algorithm. Each hidden layer takes an input from the previous layer, adjusts its weighting and, through an activation function, passes the output to either another hidden layer or an output layer (Gèlron, 2019). ANNs comprise of the following main objects:

- Layer(s) – Which transform input data, usually in the form of tensors, and output a single or multiple value(s).
- Weights – Internal parameters that will be adjusted during each training step.
- Loss function – Which determines the training performance; in other words, quantifies the error between the predicted output and the true output.
- Optimiser – Determines the way neural networks are trained, usually by a gradient descend step to update the weights based on the loss function.

Recurrent neural networks (RNNs)

Based on the above, the main goal of training a neural network is to identify the appropriate parameters to build and compile the model. While a detailed description of ML or ANNs is not the objective of this project, the reader is encouraged to refer to Geìron (2019) for further understanding of the topic.

The proposed ANN model for this research project will be an RNN, which is an ANN that processes data sequentially while retaining a memory (or cell state). It is recurrent because the output of the previous sequence becomes the input of the next sequence (Fausett, 1994).

This ability to process data sequentially while retaining a cell state at each recurring step makes RNNs particularly useful in the context of predicting sequential data, including time series that have a sequential dependency. Unlike many ML and ANN methods that are well suited for regression analyses, the ability to handle sequential dependencies makes RNNs a suitable choice for this project.

An RNN architecture includes a hidden cell state, where the output at a previous step is fed back into the model, which updates the model's internal parameters (or state). Although this process theoretically allows RNNs to have an internal 'memory,' many traditional RNNs, however, suffer from long-term dependencies in which their architecture causes the information to decay or 'explode' after each training iteration, often called the 'vanishing gradient' problem (Bengio et al., 1994). LSTM and gated recurrent units (GRUs) solve this problem by introducing a 'gated' architecture to each recurring cell.

Originally proposed by Hochreiter & Schmidhuber (1997), LSTMs have specialised 'gates,' commonly known as input, forget and output gates, which together control and filter information at each time-step by utilising specific algorithms built into them (Geìron, 2019). Each gate has a specific activation function which either allows or restricts the flow of information at each time-step. GRUs, introduced by Cho et al. (2014), function like LSTMs, however with only two gates, an 'update' and a 'reset' gate. GRUs are therefore less complex than LSTMs; however, their functionality is similar and the often have faster computing times (Yang et al., 2020).

Building an ML model

There are few software frameworks available to build and train ML and ANN models. The most common framework is TensorFlow, which has several inbuilt libraries and tools to create models (Abadi et al., 2016). The most common of these is the Keras library, which has been developed specifically for building and running ANN models (Chollet et al., 2015).

The process of building ML and ANN models involves selecting the appropriate hyper-parameters. Hyper-parameters typically control how the model 'learns' by adjusting various parameters such as its learning rate and number of hidden units, as well as specifying the types of loss (cost) function and optimisers used. Selecting the correct hyper-parameters is often an iterative process, but it can be done by using a grid search method in which multiple combinations of parameters are inputted into the model to select the most appropriate parameter that has the best performance (Geìron 2019).

METHODOLOGY

Study area

The utility owner Watercare currently operates 12 water-supply reservoirs in the Tāmaki Makaurau Auckland and Waikato regions. Five of these dams are in the Waitākere Ranges in Northwest Tāmaki Makaurau Auckland, while four are in the Hunua Ranges (Southeast Auckland and the Waikato region). From these, three dams were chosen for this project: the Upper Nihotupu, Waitākere and Mangatangi Dams. The Upper Nihotupu Dam is a concrete dam built in 1923, with a current lake area of approximately 12.5 hectares and a capacity of approximately 2.2 gigalitres. The catchment area is approximately 980 hectares, with more than 99% of the area covered by native bush. The Waitākere Dam was completed in 1910, and currently has a lake area of approximately 25.1 hectares and a capacity of approximately 1.76 gigalitres. The catchment area is approximately 810 hectares, which is almost exclusively covered by bush. Located in the Waikato region, and completed in 1972, the Mangatangi Dam is the largest of the three dams, with a lake area of 185 hectares and a capacity of approximately 35.3 gigalitres (Watercare, 2021).

Data collection

Two main sources were used to obtain time-series datasets to represent the hydro-meteorological and reservoir operational (anthropogenic) features previously described in the hydrological balance model.

1. Watercare Services Ltd
2. National Institute of Water and Atmospheric Research (NIWA)

Watercare Services Ltd is an Auckland Council-controlled organisation that owns, manages and operates the water supply for the Tāmaki Makaurau Auckland region, including the Upper Nihotupu, Waitākere and Mangatangi Dams (Watercare, 2021). Watercare maintains a historical database and actively monitors and records various metrics related to dam levels and reservoir operation.

NIWA is a Crown Research Institute responsible for monitoring and recording hydroclimatic data for all of Aotearoa New Zealand (National Institute of Water and Atmospheric Research, 2021). NIWA maintains a Virtual Climate Station Network (VCSN), a database that records and estimates hydroclimatic data on a virtual grid by spatially interpolating measurements at each grid point using actual measurements at various automatic weather stations located across the country (Cichota et al., 2008). User access to climate and weather data are provided via the NIWA data portal, which grants the user non-exclusive and non-transferable licence to access the datasets. Access to the data is provided to a user account for a limited time and the data have solely been used for this study, with no commercial use intended.

Table 1 shows the co-ordinates of the virtual grid points used to collect hydrometric data. These have been selected to be representative of the climate data for each catchment as best as possible.

TABLE 1: CO-ORDINATES OF THE NIWA VIRTUAL CLIMATE STATION NETWORK.

Site	Co-ordinates of VCSN
Upper Nihotupu	36.925°S, 174.575°E
Waitākere	-36.925°S, 174.525°E
Mangatangi	-37.1151°S, 175.2080°E

Data preparation

Raw data obtained from each source were sampled and prepared in the Python programming language using the Python Library-Pandas (McKinney et al., 2010). The time-series datasets of each feature were imported to Python and converted to a Data Frame – a Pandas class that allows manipulation of tabular data. Several samples obtained were historically measured at varying time intervals that ranged from daily and hourly to every 15 and 10 minutes; these data were upsampled into daily values either by taking the mean, sum or the last measurement for the day as the daily value. The resample function in the Pandas library was used for this purpose. For example, data obtained for the dam level ('Level') were upsampled by using the final measurement of each day to represent the daily reservoir level. Flow data over the spillway ('Spill') were aggregated to obtain a daily sum of the total volume of discharge.

Following the resampling process, any missing values in the dataset were handled by linear interpolation using the interpolate function in Pandas. It is important to note that two features – 'Abstraction' and 'Compensation' had large quantities of data missing prior to the year 2009 for all three dams. 'Abstraction' refers to the water abstracted to the treatment plant, while 'Compensation' refers to the discharge released to maintain the downstream environment and ecology. Given the importance of these features in forecasting reservoir levels, and since model inputs require data ranges of equal length, for the forecasting component of this project, the historical time-steps only extend back to the year 2009. Additionally, since linear interpolation or other data imputation methods are only statistically meaningful for a limited number of missing values, interpolating large amounts of data was not seen as a feasible option.

Table 2 displays the date ranges of the time-series datasets from all features to be used as inputs to the forecasting models.

TABLE 2. DATE RANGES OF TIME-SERIES DATA USED IN THE FORECASTING MODELS.

	Upper Nihotupu	Waitākere	Mangatangi
Date Range	02/12/2009– 04/03/2021	01/03/2009–04/03/2021	01/01/2009–22/03/2021

ANALYSIS

Prior to the forecasting task, the prepared data was analysed to observe historical trends, and inform decisions related to the importance of each feature (variable) during the modelling process. The analysis part involved the following:

1. Trend analysis
2. Feature engineering
    a. Feature selection and feature creation
    b. Correlation tests to identify correlations between features
    c. XGBoost machine learning (ML) test to measure feature importance

Trend analysis

Broadly speaking, a time series can fall under two main classifications:

1. Stationary
2. Non-stationary

A time-series dataset is assumed to be stationary if the underlying statistically observable properties such as mean, variance and autocorrelation do not change over time (Hyndman & Athanasopoulos, 2018). Mean and variance are fundamental properties that describe the distribution of a given sample. During trend analysis, values can be grouped into segments (e.g., by using moving averages) and the mean and variance between each segment can then be computed and analysed for stationarity. There are several statistical tests that can perform this function. The Augmented Dickey-Fuller (ADF) test has been widely used for trend analysis and was used here to analyse trend (Hamilton, 1994). An ADF test was carried out using the Adfuller function in Statsmodels, an open-source Python library that provides functions and classes for a range of statistical models (Seabold & Perktold, 2010).

Visual representations of the trend, including monthly mean and percentile distributions, were obtained using standard time-series plots and box plots using the Matplotlib and Seaborn packages (Hunter, 2007).

Normalcy tests

An important sub-task of trend analysis is to understand the underlying distribution of the time series. Understanding the distribution also allows the selection of appropriate tests, since statistical tests are typically dependent on whether a dataset has a parametric or non-parametric distribution. Parametric datasets assume an underlying normal (or Gaussian) distribution, while non-parametric datasets do not fit a known or underlying distribution (Hyndman & Athanasopoulos, 2018).

A Shapiro-Wilk Test was used to test the likelihood a given sample was taken from a normal distribution (Shapiro & Wilk, 1965). The Shapiro function in the SciPy Python package was used to obtain two values for each feature, a test statistic (W) and a p-value. The null hypothesis (denoted by the p-value) of this test is that the sample is normally distributed; by convention, the null hypothesis H0 is rejected when the p-value is <0.05, and failed to reject if the p-value is >0.05 (Mishra et al., 2019). The underlying equation for the test statistic is displayed in Equation 3.

$$W = \frac{\left(\Sigma_{i=1}^{n} a_i x_{(i)}\right)^2}{(\Sigma_{i=1}^{n}(x_i - \bar{x})^2} \quad (3)$$

Where $x_{(i)}$ is the value of x at the $(i)th$ order, n is the sample size, $\bar{x}$ is the mean of the sample and a_i are normalised constants based on the scale and sample size of the data (Shapiro & Wilk, 1965).

In addition to the statistical tests, visual representations of the data were displayed using the Matplotlib package. Visual representations are important in exploratory data analysis, and complement purely numerical tests by graphically showing distributions and trends in the dataset. These visual representations complement numerical results and provide an intuitive understanding of the time-series data.

Autocorrelation

Autocorrelation refers to the property of a time series where its values at a specific time-step (t) are correlated with its values in the past (t-1....t-k). Autocorrelation measures the strength of the relationship between an observation and the observations at previous time-steps (or lags) (Hyndman & Athanasopoulos, 2018).

A strong autocorrelation in a time series implies that the value of the observation at the current time-step is strongly related to its value in previous time-steps, which is an important indicator in forecasting. The VAR model, which is used in this project, is an autoregression-based statistical model that uses lag orders (number of previous time-steps) of multiple variables to predict future values of each individual variable. Autocorrelation was presented graphically through correlograms displayed using the Statsmodels library.

Feature engineering

In the field of ML, feature engineering is the process of using domain knowledge to select and/or create new features (variables) to use as inputs to train a model (Geìron, 2019). As multivariate forecasting models rely on multiple input features to make predictions by mapping the relationship between the input and output, feature engineering is an important step to ensure only the most useful features are used in the model. The feature engineering process in this project contains two main steps:

1. Feature selection
2. Feature creation

Feature selection

A simple theoretical relationship between features has already been established using the water balance model. With this theoretical grounding, the basic idea of feature selection is to identify the features that will affect the daily reservoir levels at each subject site. The input features used for this project are displayed in Table 3, below. Based on hydrological intuition and domain knowledge, a general hypothesis can be applied to the relationship of each variable to the target variable – daily water level at time-steps t+ 7 and t+ 30.

TABLE 3. INPUT FEATURES AND THE INITIAL HYPOTHESIS OF ITS EFFECT ON DAILY WATER LEVELS.

Feature	Description	Hypothesis	Data Source
Level	The target variable Units: m	N/A	Watercare
Spill	Flow rate over the dam spillway Units: m3/s	Can be positively and negatively correlated to the water level. Higher spillway flows are caused by high water levels, but high flows will reduce the water level to a baseline above the mouth of the spillway.	Watercare
Rainfall (WSL)	Rainfall depth over 24 hours Units: mm/day	Positive correlation to the water level.	Watercare
Rainfall (NIWA)	Rainfall depth over 24 hours Units: mm/day	Positive correlation to the water level.	NIWAData
Abstraction	Abstraction flows to the treatment plant Units: m3/day	Negative correlation to the water level.	Watercare
Compensation	Flow release to meet downstream demands Units: m3/day	Negative or positive correlation to the water level.	Watercare
Soil moisture	Soil moisture deficit Calculated using rainfall and outgoing daily potential evapotranspiration (PET) and a fixed available water capacity of 150mm (NIWA, 2021). Units: kg/m2	Positive correlation to the water level.	NIWAData
Max Temp	Maximum daily temperature Units: °C	Negative correlation to the water level.	NIWAData
Penman evaporation	PET value using the Penman formula (Burman & Pochop, 1994) Units: kg/m2	Negative correlation to the water level.	NIWAData
Solar radiation	Solar radiation over 24 hours Units: MJ/m2	Negative correlation to the water level.	NIWAData

Feature creation

The features mentioned in the previous section provide useful hydrometric and anthropogenic variables that will be useful in the forecasting task. For a forecasting model, the input features are merely vector or matrix representations of numbers without the added context. Unless specifically designed, an ML or statistical model does not have the theoretical understanding of the data it is trained on. While this is an inherent drawback of these mathematical models, a solution is to introduce new features or representations of existing real-world phenomena as inputs.

One such feature is the representation of time. Since hydrological process are, to a large extent, seasonal, it is important to have features that represent seasonal periodicity. To make the model more time-aware, simply passing the date, month or year at each specific time-step is not particularly useful to a model, as the pure numerical representation of date is meaningless without the underlying context. A work-around is to represent periodicity as sine and cosine waves with a defined seasonal interval. With the intuitive assumption that the target output (i.e., reservoir water-level) has a seasonal cycle of 12 months, periodicity can be represented by a time-of-year signal in a unit circle with a repeating pattern at every 365 time-steps (TensorFlow, 2021). Figure 3 graphically presents the values of sine and cosine inputs that correspond to the time of year at each time-step in the time series. These sine and cosine values were then used as two separate features to represent periodicity.

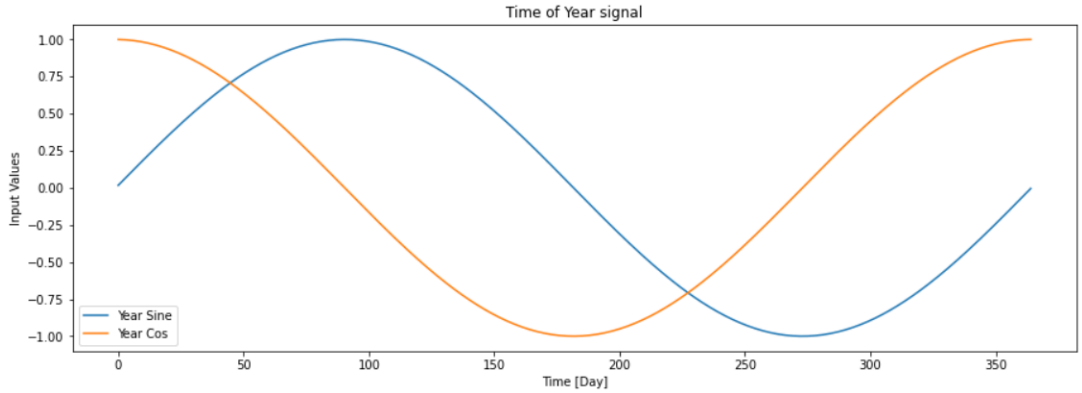

Figure 3. Cosine and sine waves representing the time-of-year inputs.
Source: colab.research.google.com.

Figure 4, below, shows the first 10 and last 10 values for all 13 features in the time-series DataFrame for the Mangatangi Dam. The dataset for Mangatangi, for example, has 13 features and 4464 time-steps, written in matrix form as a 4464 x 13 matrix, where each time-step is represented as a row vector and each feature is represented as a column vector.

Date	Level	Spill	Rainfall (WSL)	Abstraction	Compensation	Rainfall (NIWA)	Soil Moisture	Max Temp	Penman Evaporation	Vapour Pressure	Solar Radiation	Year sin	Year cos
2009-01-01	52.137	0.0	2.35	124727.6250	15000.0000	2.5	-77.3	25.5	3.2	21.2	14.8	0.0172	0.9999
2009-01-02	52.094	0.0	7.99	125076.5000	15000.0000	11.2	-72.4	25.1	4.5	20.8	16.6	0.0344	0.9994
2009-01-03	52.057	0.0	13.63	124547.2500	15000.0000	0.0	-77.4	21.9	6.0	12.5	31.9	0.0516	0.9987
2009-01-04	52.013	0.0	0.00	125002.0000	15000.0000	0.0	-81.4	22.7	5.1	14.1	28.4	0.0688	0.9976
2009-01-05	51.958	0.0	0.00	125034.8750	15000.0000	0.0	-84.6	22.9	5.5	15.0	29.9	0.0859	0.9963
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2021-03-18	43.938	0.0	6.44	62259.6897	24150.0007	0.0	-100.1	21.7	3.9	11.5	16.7	0.9698	0.2437
2021-03-19	43.887	0.0	0.00	53520.3622	24480.8511	0.0	-101.9	21.1	3.4	11.4	17.2	0.9739	0.2270
2021-03-20	43.851	0.0	0.00	39548.6537	24593.3167	0.0	-106.8	22.0	3.3	11.2	17.7	0.9777	0.2102
2021-03-21	43.815	0.0	0.00	39530.8389	24489.1570	0.0	-108.5	22.5	3.5	12.6	18.7	0.9811	0.1933
2021-03-22	43.781	0.0	0.00	39513.0961	24449.8240	0.0	-106.6	22.8	3.4	12.7	16.8	0.9843	0.1764

4464 rows × 13 columns

Figure 4. Mangatangi time-series DataFrame. Source: colab.research.google.com.

Correlation tests

Statistical correlations tests provide insight into the relationship between two features. While they do not directly imply causation, correlation between the data can be a useful indicator of prediction performance and feature relationships.

With the assumption that the data distribution of features is non-parametric (i.e., not normally distributed), the Spearman's rank correlation test was chosen to calculate the correlation between features (Hyndman & Athanasopoulos, 2018). This correlation test describes the strength of the linear relationship between the ranked values of each feature and is achieved by ranking the values at each time-step (based on scale) and calculating the rank coefficient using Equation 4, below.

Where, ρ is the Spearman's rank correlation coefficient, d_i is difference in rank between the two observations and n is the number of observations (Spearman, 1904). Spearman's correlation test provides a coefficient result between 1 and -1 between two variables, where coefficients closer to 1 will be strongly positively correlated, while coefficients closer to -1 will be strongly negatively correlated with each other.

$$\rho = 1 = \frac{6\sum d_i^2}{n(n^2 - 1)} \quad (4)$$

Feature importance and Shapley values using XGBoost

Extreme gradient boosting (XGBoost) is a decision-tree-based supervised ML algorithm that uses gradient boosting techniques for regression and classification tasks. XGBoost was proposed by Chen and Guestrin (2016) to be more computationally efficient than regular boosting methods, and has since proven to be more accurate at prediction tasks than many traditional ML models. Although XGBoost is not used for any direct prediction tasks in this project, the architecture of decision-tree-based algorithms allows extraction of feature importance. Bouktif et al. (2018) used XGBoost to determine feature importance to forecast electricity load, while Zheng & Wu (2019) used XGBoost to forecast wind power using various hydro-meteorological features as inputs.

To this end, Shapley values were used to plot feature importance scores, showing the relative contribution of each feature to the final prediction outcome. Shapley Additive exPlanations (SHAP), proposed by Lundberg and Lee (2017) describe the contribution of a feature to the prediction compared to the average prediction value of the dataset. A description of Shapley values and its uses in the ML context is given by Molnar (2021).

The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalition game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the 'payout' (= the prediction) among the features.

The reader is encouraged to refer to Chen and Guestrin (2016) and Molnar (2021) for further theoretical explanations on XGBoost and Shapley values. For this project, the XGBoostRegressor class in the SHAP library was used to compute the Shapley values for the features (X) in predicting the target (y), which is the dam level at each time-step.

FORECASTING

The forecasting component of this project used two separate forecasting methods: VAR and RNNs, to forecast daily dam levels at a 7- and 30-day horizon. The predictions from both models were made using a walk-forward validation method in which new inputs are given to the model at each time-step. This process is illustrated below in Figure 5 for the +7-day forecast.

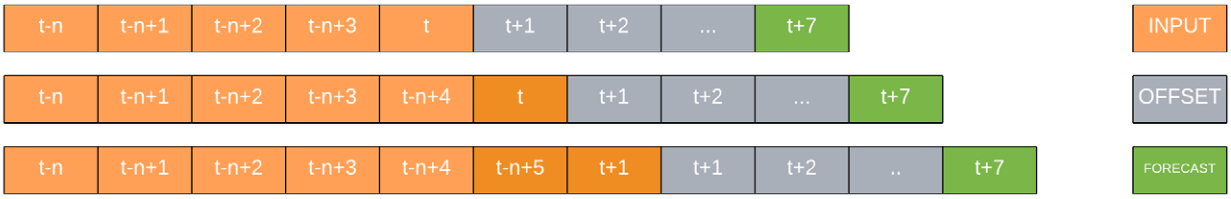

Figure 5. Walk-forward validation.

To accurately gauge model performance, a baseline persistence model was used to further compare and evaluate the performances of the models. The persistence model simply used the most recent observation at the current time-step (y_t) as the forecast for the 7- and 30-day target ($y_{t+7}y_{t+30}$). Due to its simplicity, this naïve forecasting model acted as a baseline to compare the performances of more sophisticated models (Hyndman & Athanasopoulos, 2018).

Model performance was measured for each dam using the root mean squared error (RMSE) and mean absolute error (MAE) between the observed (y) and forecasted ($\hat{y}$) values for the target variable at each future time-step (i.e., at 7 and 30 days).

$$RMSE = \sqrt{\frac{\boldsymbol{\Sigma}_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} (5)$$

$$MAE = \sqrt{\frac{\boldsymbol{\Sigma}_{i=1}^{n}|\hat{y}_i - y_i|}{n}} (6)$$

RMSE and MAE are shown above in Equations 5 and 6 respectively. Where $\hat{y}$ is the predicted value, y is the true value and n is the sample size. All models are evaluated on a test set that comprises the forecasts for the final 365 time-steps of the dam level.

VAR model

The VAR model was introduced by Sims (1980) and is still widely used in an econometric context to model multivariate time series. The VAR model was used to model macroeconomic variables but has since been adopted in many different fields.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t (7)$$

Consider an autoregressive (AR) model of the order P. The order represents its lag component.

Where c is the intercept term, ϕ is an adjustable parameter (constant), and ε is the error. This linear function for a univariate time series indicates that the value of y at time-step t is a function of the intercept, a constant and an error term (Hyndman & Athanasopoulos, 2018).

$$y_{k,t} = c_k + \phi_{1k} yk_{t-1} + \phi_{2k} y_{t-2} + \cdots + \phi_k y_{kt-p} + \varepsilon_{k,t} (8)$$

The VAR model expands this idea to include multiple time series, where each variable is modelled as a linear function of the past values of itself and the other variables (McKinney et al., 2018). The vectorised AR function for k number of variables with P lags is shown in Equation 8, above.

Equation 8 contains multiple equations for each variable up to k number of equations. In simple terms, each variable (feature) in a VAR model, has a specific equation that describes its temporal distribution based on three main components:

1. Its own lagged values
2. Lagged values of other variables
3. An error component

Unlike most traditional statistical forecasting models, VAR assumes the total variables presented to the model are endogenous in nature. This means the variables themselves are dependent and interrelated with one another (Hyndman & Athanasopoulos, 2018).

For the forecasting task in this project, k is be the number of variables (13), while the lag order (P) was selected by using three information criteria to select the optimal lag order (P). Information criteria measures the goodness of fit for a set of given parameters. The select_order function in Python's Statsmodels library was used to select the most suitable lag order from three different information criteria: Akaike information criteria (AIC), Bayesian information criteria (BIC) and the Hannan–Quinn information criterion (HQIC) (Hatemi-J & Hacker, 2019).

A custom function was built to predict the 7- and 30-day forecasting result at each time-step using the walk-forward validation method, where new inputs are given to the model at each time-step.

RNN models

Two separate sets of LSTM and GRU models were created to forecast the 7- and 30-day time horizon for each dam using all 13 features as inputs with a target output as the water level at 7 and 30 days. An additional set of models (no 'Level' input, or NoL) were created, in which the inputs to the model did not contain the 'Level' feature. Instead, the 'Level' feature was only provided as the target variable (y) and the model was trained on the remaining 12 features. While this model would not be useful for the main prediction task, it would be important in further determining whether there was a 'learnable' relationship between the remaining features and the water level, thereby giving an indication as to how useful the remaining 12 features were in predicting the dam water levels. Table 4, below, summarises the RNN models used for this project.

TABLE 4. RNN MODELS USED IN THIS PROJECT.

+7 Day Forecast		+30 Day		No 'Level' input (NoL)	
GRU+7	LSTM+7	GRU+30	LSTM+30	GRU-NoL	LSTM-NoL

Additional data preparation was required prior to training the model. Namely, a 'Target' variable (y) was created to represent the true forecast values at the 7- and 30-day horizon. This was achieved by creating a copy of the 'Level' column and shifting the data up by the specific time-step (7 or 30). The dataset was then split into training, validation and testing sets. The last 365 values were used for the test set, while the rest were split into 85% training

and 15% validation sets. The aim was to train and fit the model on the training set while using the validation set to prevent overfitting to the training data. Once trained, model performance was evaluated by performing predictions on the test set. The data partitioning used for training, validation and test sets for each dam is shown in Table 5, below.

TABLE 5. PARTITIONING OF TRAINING, VALIDATION AND TESTING DATA.

	Upper Nihotupu		**Waitākere**		**Mangatangi**	
	+7 Day	+30 Day	+7 Day	+30 Day	+7 Day	+30 Day
Training Data	3172	3133	3406	3367	3472	3433
Validation Data	560	553	602	625	620	606
Test Data	365	365	365	365	365	365

Once the data was partitioned, the next stage was to scale the input data to within acceptable ranges for the model. As an example, the raw feature data for the Upper Nihotupu Dam had a range between -139.9 and 34101.375. Since the features had different scales/units associated with them, the range of values was large, and large ranges often result in unstable model performance. Since ML models rely on adjusting weights, large data ranges will cause models to train and perform poorly (Goodfellow et al., 2016). To standardise the data the MinMaxScaler object in Scikit-Learn was used to scale the feature values between 0 and 1, while still preserving the underlying shape of the original distribution. The RNNS models (LSTM and GRU) were created, compiled, and run on TensorFlow (Abadi et al., 2016) and Keras (Chollet et al., 2015). The models were created using the Sequential class in Keras. Table 6 summarises the steps taken to create, compile, train and predict each model, including the hyper-parameters used at each stage.

TABLE 6. MODEL HYPER-PARAMETERS USED.

Steps	**Hyperparameters/description**
Create and define model	Model type: Sequential Model layers Input layer Hidden layer with 512 units Dense output layer
Compile model	Optimiser: *Adam* Loss Function: *Huber* Learning rate: 0.001
Fitting the model to the training data	Epochs: *100* Steps per epoch: *30*

RESULTS

Analysis

Trend

Results from the Augmented Dickey-Fuller (ADF) tests are shown in Table 7, below. The statistical test shows a p-value <0.05 for all features in all three dams, which rejects the null hypothesis and gives an indication that the time-series data for all features of the three dams have no observable trend for the chosen time period.

TABLE 7. P-VALUES FROM THE ADF TEST.

	Upper Nihotupu	**Waitākere**	**Mangatangi**
	P-value	**P-value**	**P-value**
Level	4.02E-05	7.22E-06	0.010597
Spill	1.54E-12	3.57E-30	1.41E-26
Rainfall (NIWA)	0	0	0
Rainfall (WSL)	0	0	0
Vapour pressure	6.54E-05	2.77E-05	9.19E-06
Penman Evaporation	6.67E-04	6.65E-04	0.000117
Abstraction	4.13E-11	1.54E-14	2.16E-06
Compensation	1.92E-04	3.48E-04	0.000169
Max temp	1.82E-02	0.010292	3.59E-04
Soil moisture	2.40E-04	1.27E-04	0.000179
Solar radiation	0.002754105	8.26E-03	7.03E-04

Normalcy test

The results from the Shapiro-Wilks normalcy tests reject the null hypothesis for all features in all three dams, which indicates the data is non-parametric and may not be normally distributed.

Box plots displaying the yearly and monthly distribution of dam levels show clear seasonality with changing means during different months of the year. Moreover, the yearly box plots clearly show historical periods with low dam levels that correspond to lower yearly means and greater percentile distributions. Figure 6, below, displays the box plots (both yearly and monthly) for the Waitākere Dam.

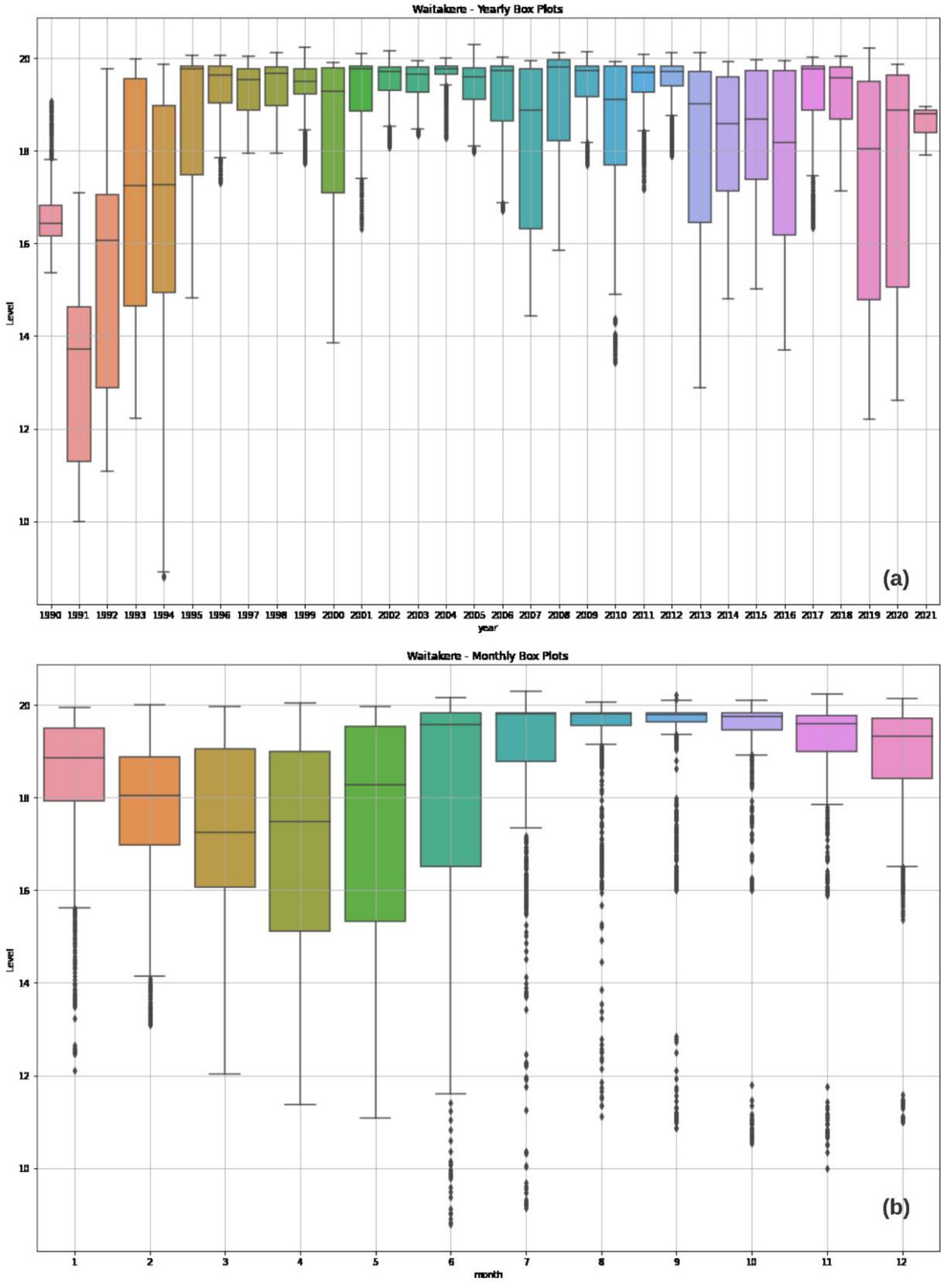

Figure 6. Box plots for the Waitākere Dam level. Showing (a) yearly and (b) monthly aggregations.
Note: Level shown in metres.

Autocorrelation plots

Autocorrelation plots are displayed in Figure 7, below. The autocorrelation is displayed at a lag of 3000 time-steps. The y-axis shows the autocorrelation coefficient, where a coefficient of +1 displays strong positive autocorrelation, while a coefficient of -1 displays strong negative autocorrelation. The light-blue shaded area shows the 95% confidence intervals; values outside of this box can be assumed to have a strong autocorrelation.

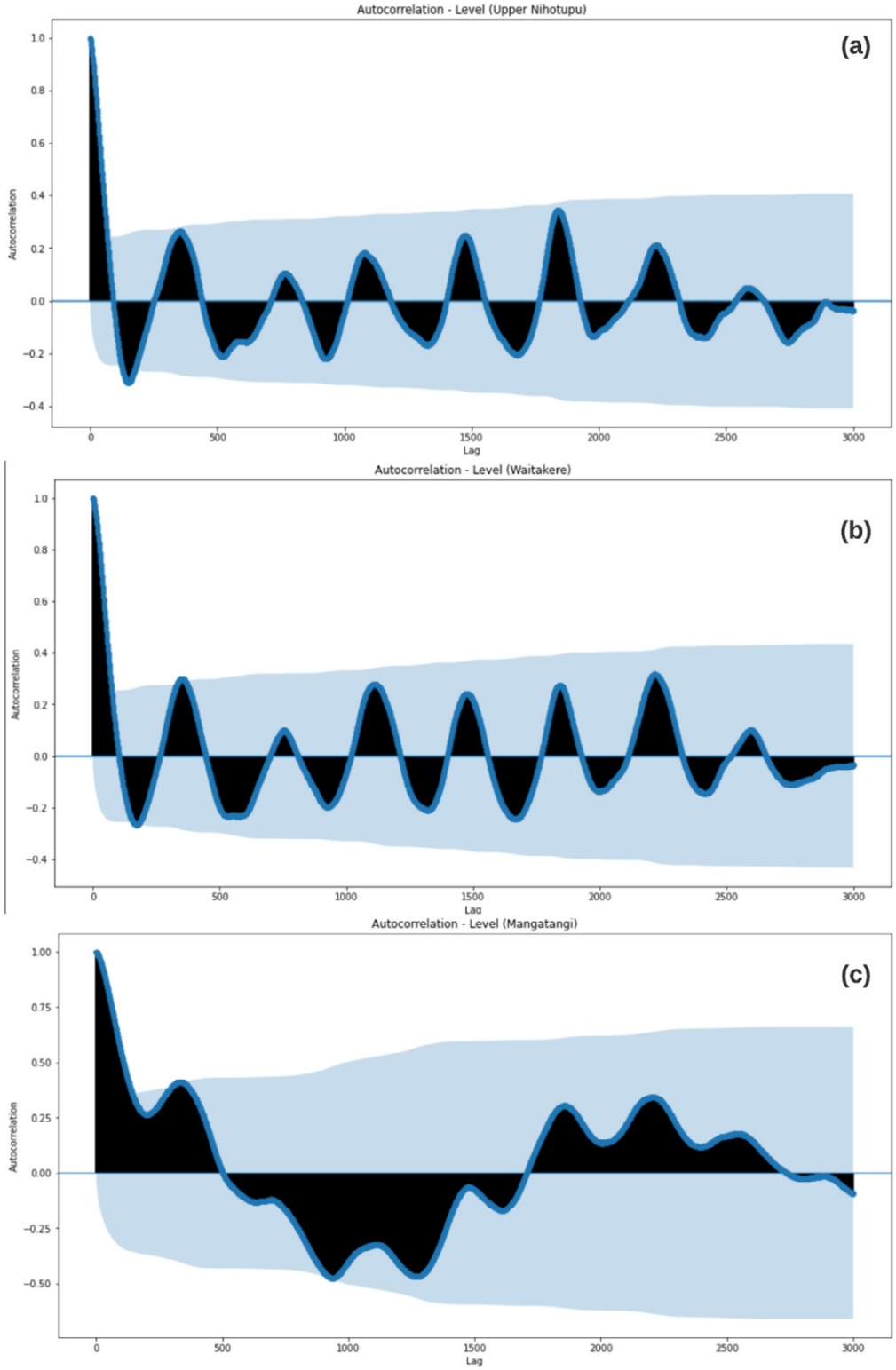

Figure 7. Autocorrelation plots for the (a) Upper Nihotupu, (b) Waitākere and (c) Mangatangi Dam levels. Results are displayed at a lag of 3000.

Feature engineering – correlation tests

Spearman's rank correlation coefficients heatmaps are shown in Figure 8. Dam levels at Upper Nihotupu and Waitākere show a high degree of positive correlation with 'Spill' and 'Soil Moisture.' The dam level at Mangatangi show a high degree of positive correlation with 'Abstraction' and 'Spill.' Strong negative correlations can be observed in the Upper Nihotupu and Waitākere Dam levels for 'Max temp' and 'Vapour pressure.'

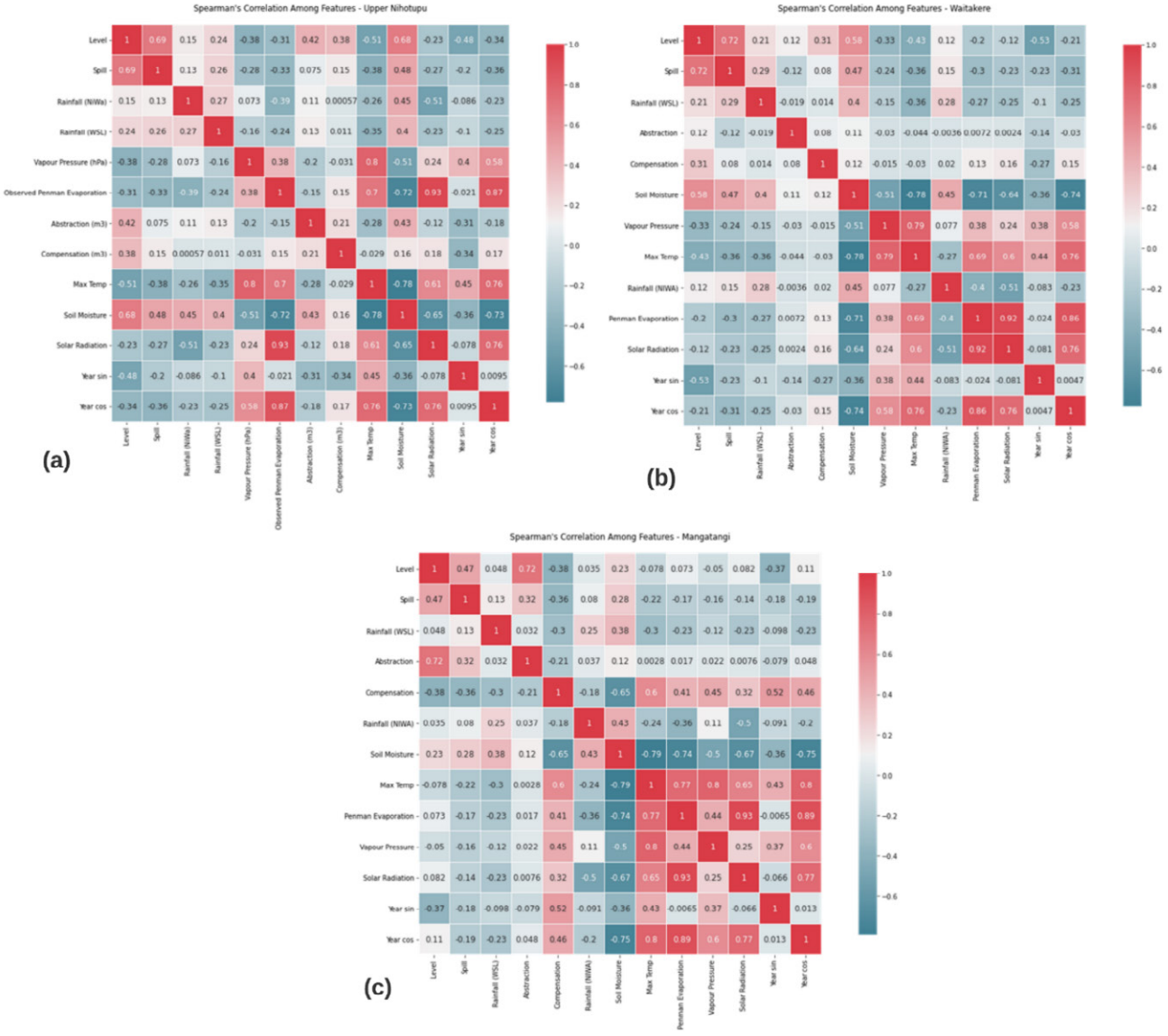

Figure 8. Heatmap of Spearman's correlation among features: (a) Upper Nihotupu, (b) Waitākere, (c) Mangatangi.

The Shapley values for the features obtained from the XGBoost tests are shown in Figure 9, below.

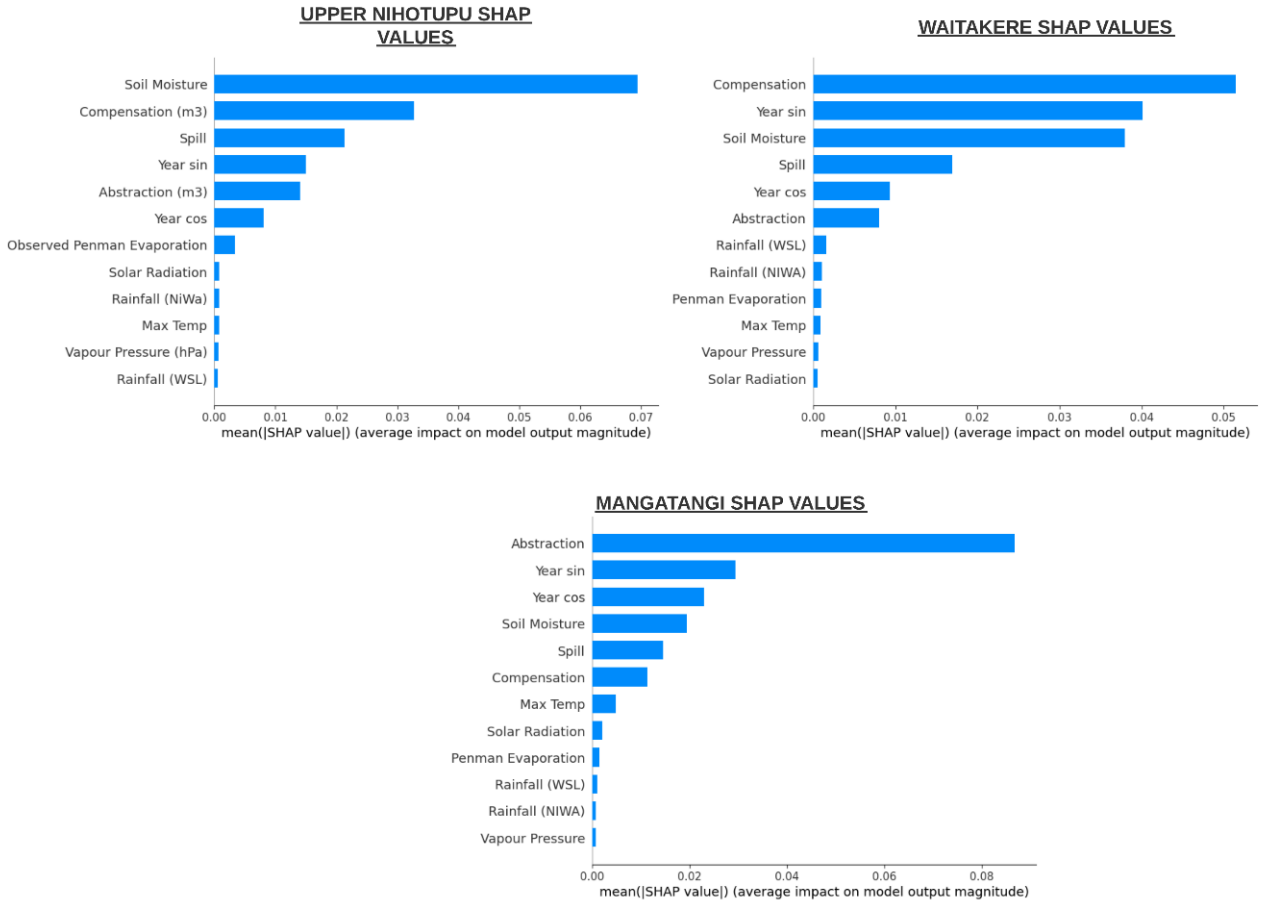

Figure 9. Partitioning of training, validation and testing data.

Forecasting

The overall RMSE and MAE results for the Upper Nihotupu, Waitākere and Mangatangi Dams water-level forecast for the 7- and 30-day periods are shown in Tables 8, 9 and 10, below.

TABLE 8. UPPER NIHOTUPU, FORECASTING RESULTS (MAE AND RMSE).

Upper Nihotupu								
	VAR(3)+7	**VAR(3)+30**	**GRU+7**	**GRU+30**	**LSTM+7**	**LSTM+30**	**Pers+7**	**Pers+30**
RMSE	1.189	2.692	1.115	2.668	1.306	2.362	1.419	3.919
MAE	0.686	1.928	0.684	1.955	0.914	1.762	0.834	2.981

TABLE 9. WAITĀKERE FORECASTING RESULTS (MAE AND RMSE).

Waitākere								
	VAR(5)+7	**VAR(5)+30**	**GRU+7**	**GRU+30**	**LSTM+7**	**LSTM+30**	**Pers+7**	**Pers+30**
RMSE	0.533	1.227	0.518	1.268	0.547	1.167	0.622	1.670
MAE	0.255	0.843	0.195	1.040	0.316	0.966	0.291	1.063

TABLE 10. MANGATANGI FORECASTING RESULTS (MAE AND RMSE).

Mangatangi								
	VAR(2)+7	**VAR(2)+30**	**GRU+7**	**GRU+30**	**LSTM+7**	**LSTM+30**	**Pers+7**	**Pers+30**
RMSE	1.199	2.684	1.115	2.668	1.306	2.362	0.407	1.455
MAE	0.667	1.894	0.684	1.955	0.914	1.762	0.334	1.326

Both VAR, GRU and LSTM models performed better than the persistence model for the 7- and 30-day prediction task at Upper Nihotupu and Waitākere, with only slight performance variations between them. Perhaps surprisingly, the persistence model outperformed the more sophisticated models for both prediction tasks for the Mangatangi Dam. Figures 10 and 11 display the forecasting-error distribution plotted for each month. These results show certain months are prone to larger errors in forecasting. The Upper Nihotupu and Waitākere results show greater prediction errors in the months of April, May, June and July, while the Mangatangi results showed the largest errors in the months of February, April and October.

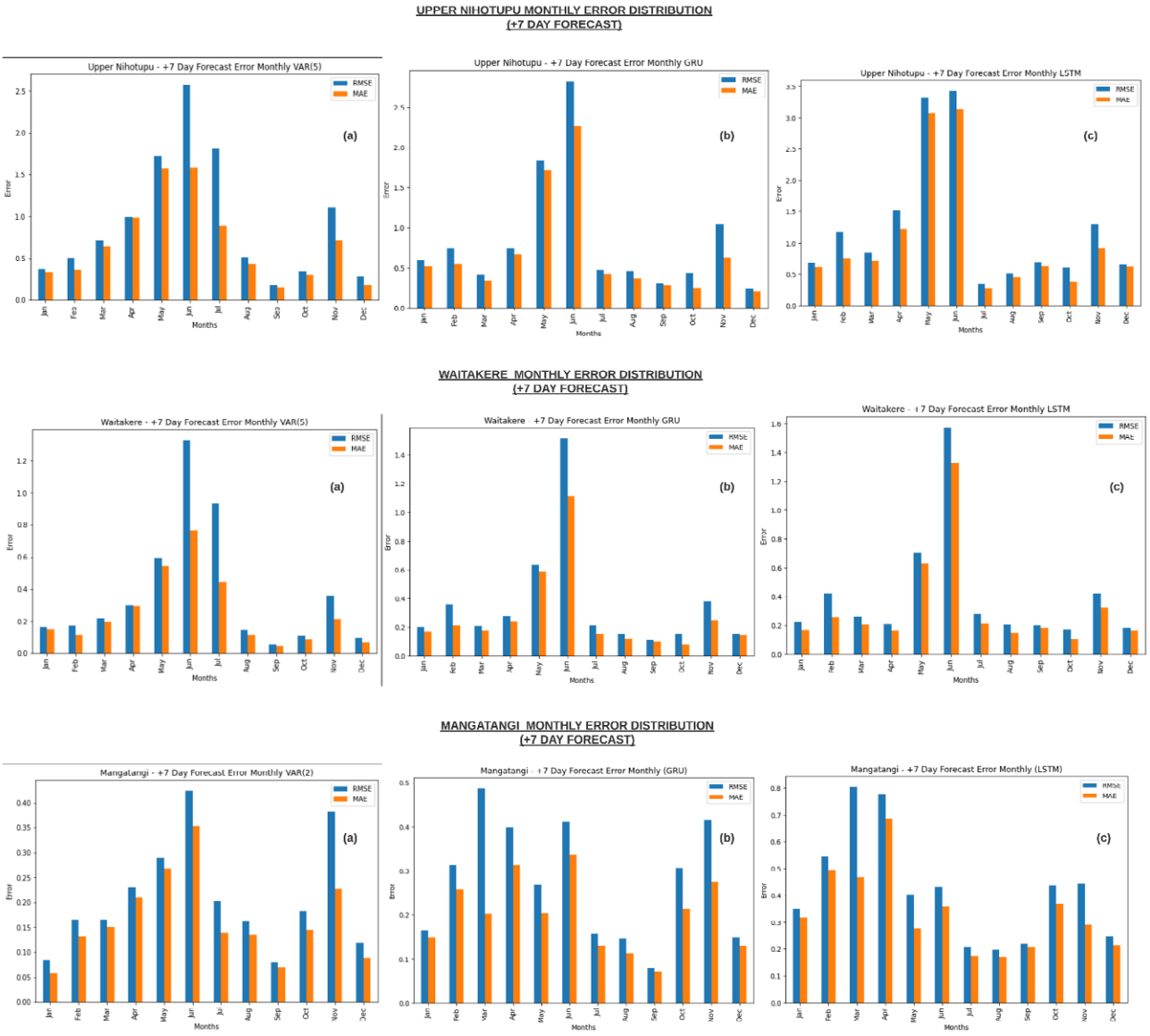

Figure 10. +7-day Forecast, monthly error distribution: (a) Upper Nihotupu, (b) Waitākere and (c) Mangatangi.

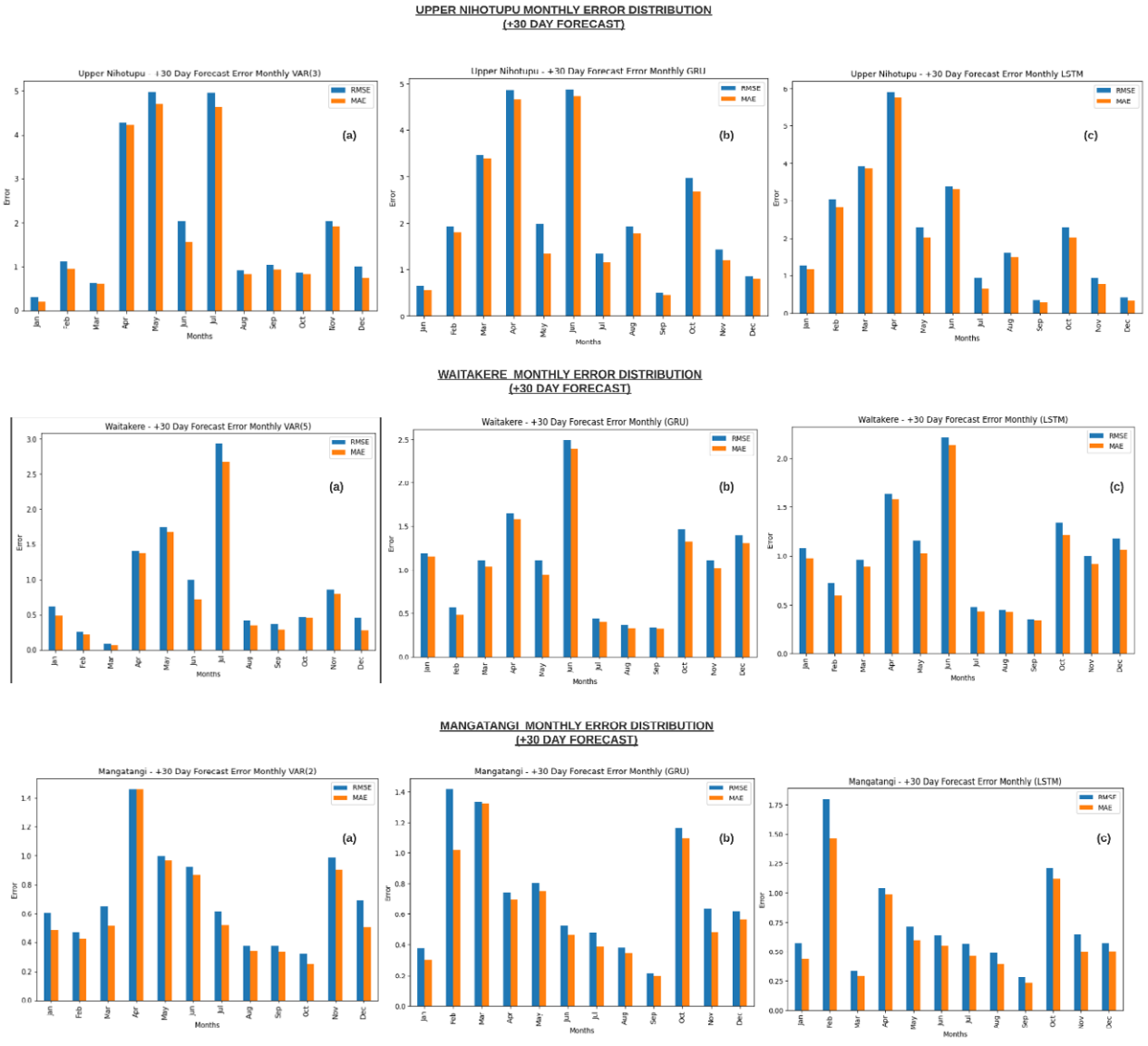

Figure 11. +30-day forecast, monthly error distribution: (a) Upper Nihotupu, (b) Waitākere and (c) Mangatangi.

The +7-day and +30-day predictions for each model are plotted in Figures 12 and 13, below. The plot results show relatively poor forecasting performances in the 30-day forecasting task (Figure 13), with the Upper Nihotupu and Waitākere models failing to accurately forecast the lower extremes.

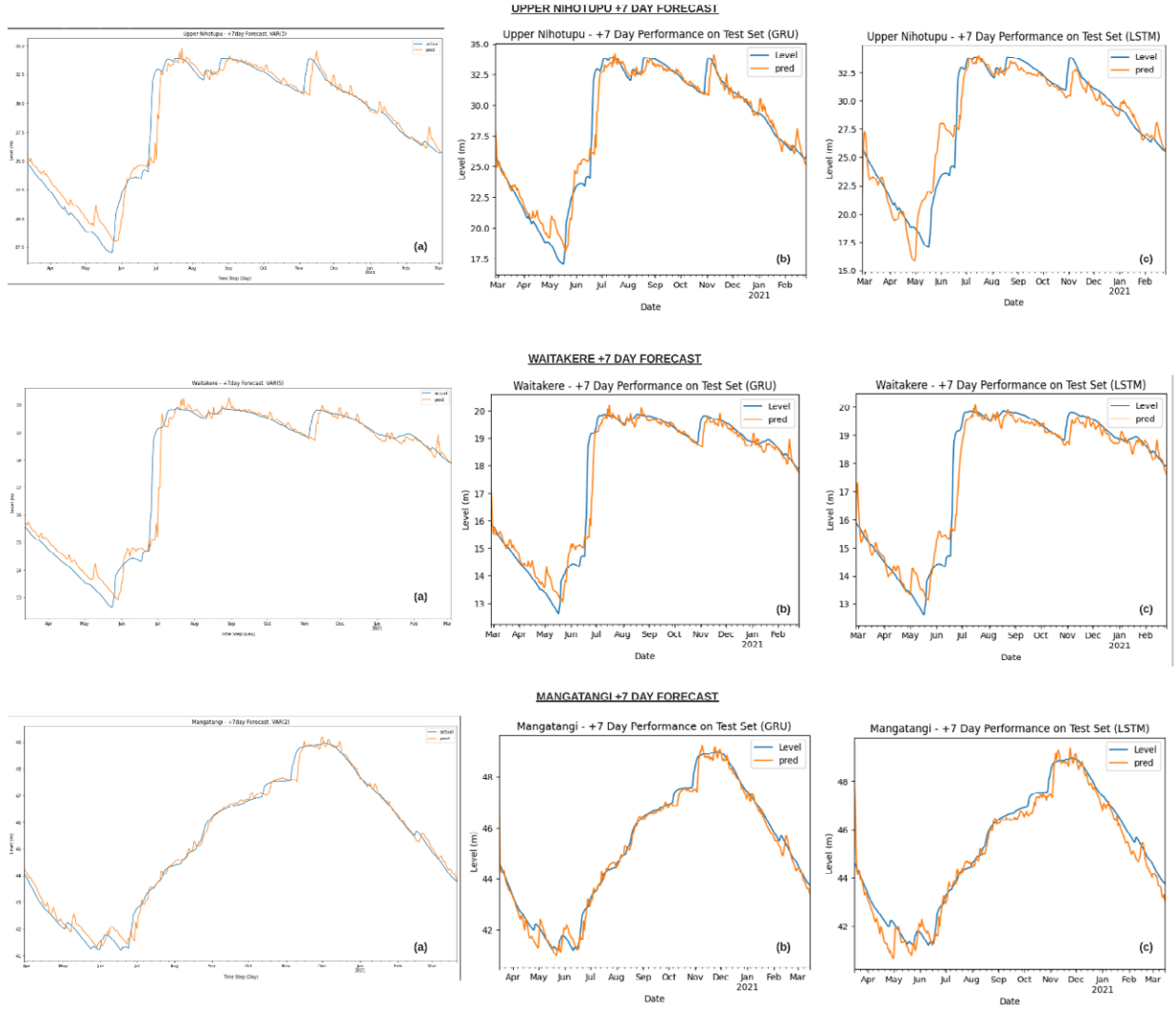

Figure 12. +7-day prediction results for each dam: (a) VAR, (b) GRU and (c) LSTM.

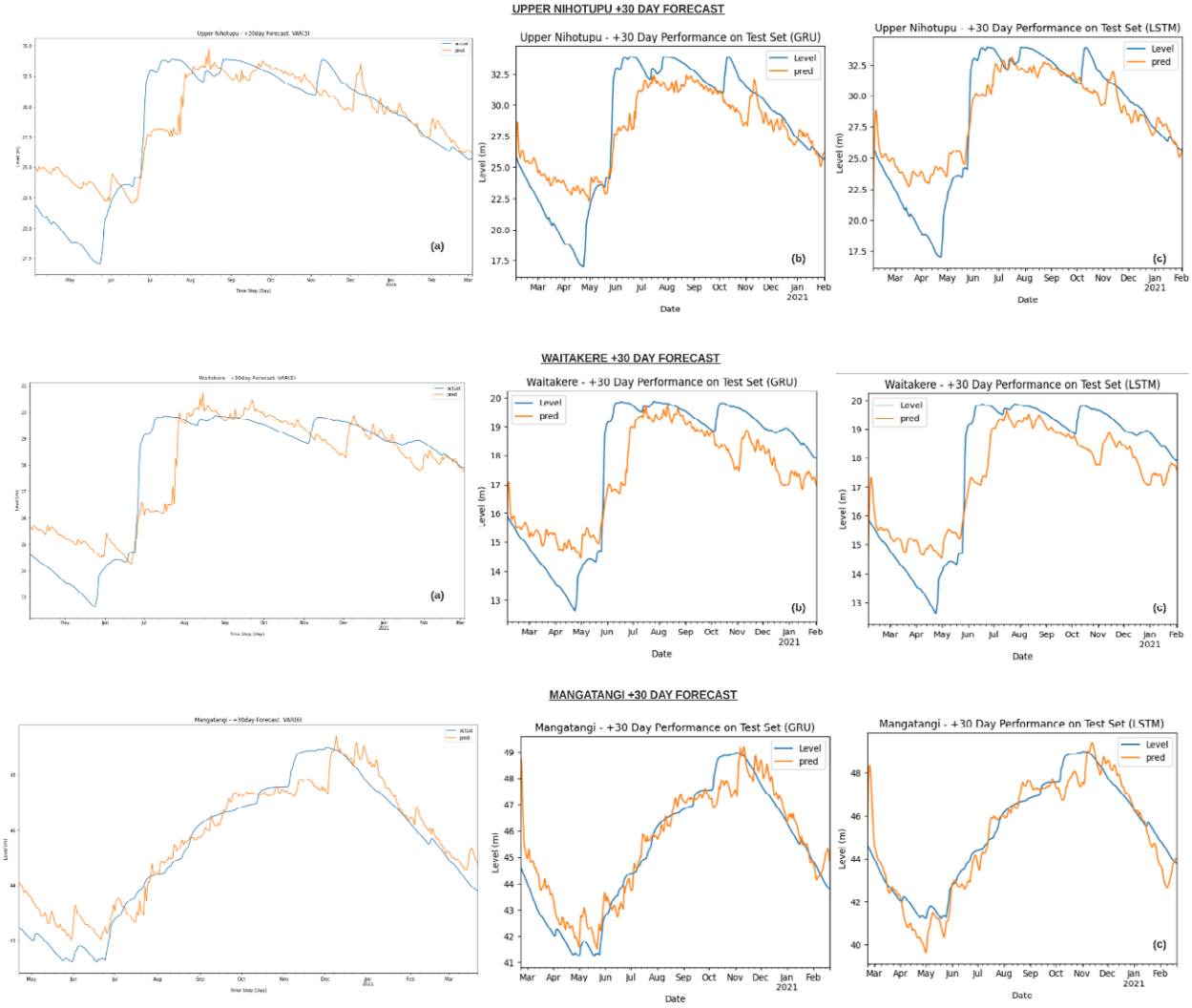

Figure 13. +30-day prediction results for each dam: (a) VAR, (b) GRU and (c) LSTM.

The RMSE and MAE results from the NoL models (GRU-NoL and LSTM-NoL) are shown in Table 11, below. Figure 14 shows the NoL predictions for each dam. The Upper Nihotupu and Waitākere Dams show relatively acceptable predictions while the predictions for the Mangatangi Dam had poor and noisy prediction results.

TABLE 11. MAE AND RMSE RESULTS FOR THE NOL MODELS.

	Upper Nihotupu		**Waitākere**		**Magatangi**	
	GRU NoL	**LSTM NoL**	**GRU NoL**	**LSTM NoL**	**GRU NoL**	**LSTM NoL**
RMSE	2.754	3.724	1.516	1.326	3.152	3.407
MAE	2.499	2.930	1.316	1.194	2.446	3.121

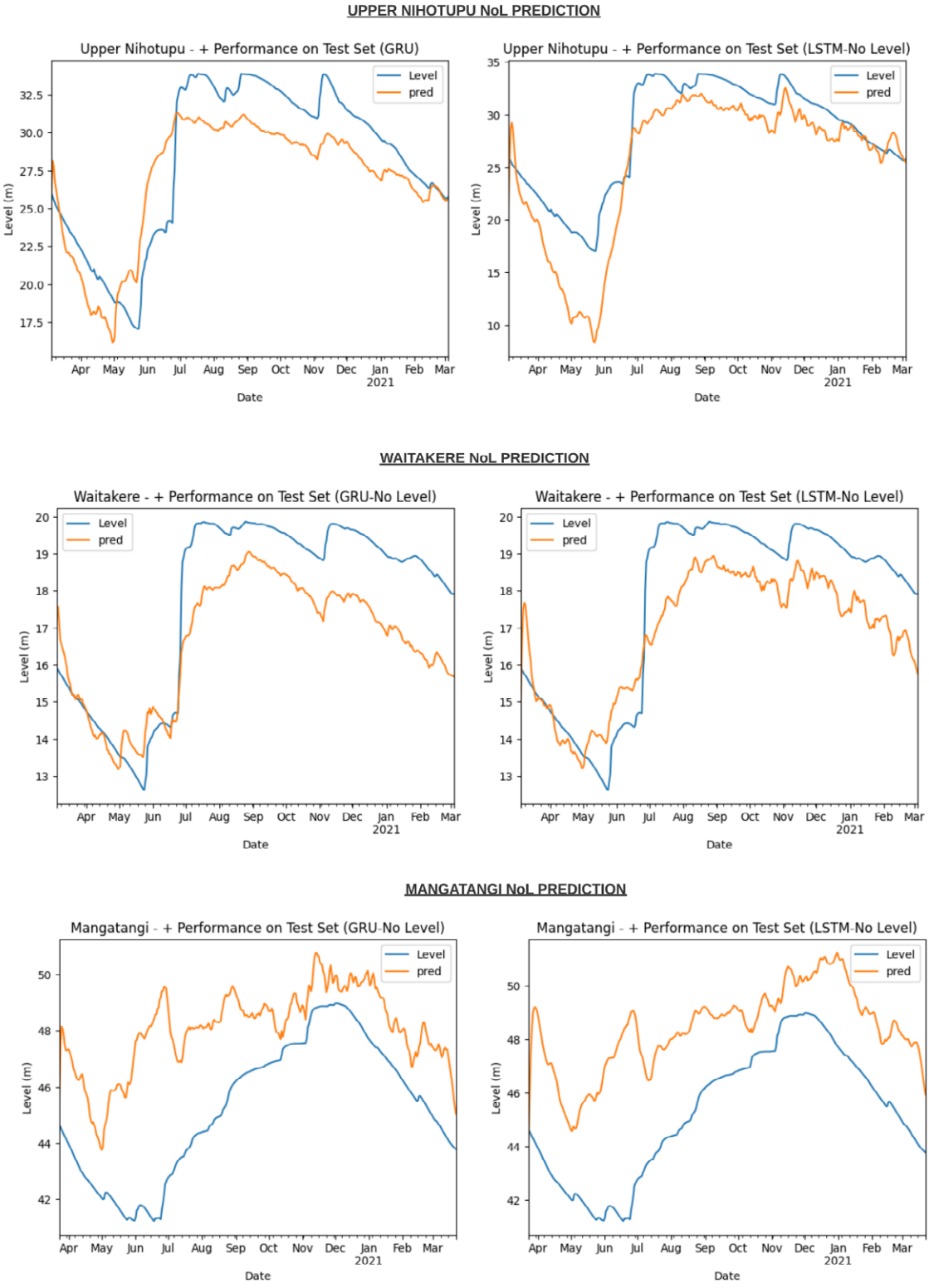

Figure 14. Prediction results from the GRU-NoL and LSTM-NoL models.

DISCUSSION

Analysis

The analysis task focused on three main components of time-series data. Trend, distribution and correlation. Inspection of the individual time-series plots for all three dams does not show a clearly recognisable trend for any of the listed features. This is further evidenced by the ADF results, which indicate a lack of a clearly discernible trend. Further inspection of the time-series plots, however, does show seasonality, with intermittent cycles corresponding to historical droughts. As an example, a historical decrease in water levels for all three dams can be seen in in the years 1993–1994. This is consistent with the Tāmaki Makaurau Auckland drought of 1994, which saw historical lows for dams in the city (Fowler & Adams, 2004).

Further investigation into the box plots shows monthly seasonality, where lower mean and percentile values are evident in the months of April and May for all three dams. Additionally, the yearly box plots show the mean and distribution of the 25th percentile dam levels since 2013 have been comparatively lower than previous years. This may suggest a decreasing trend.

The autocorrelation plots (Figure 7) indicate strong autocorrelation and seasonality of dam levels for the Upper Nihotupu and Waitākere Dams. The dam levels for the Mangatangi Dam seem to show cyclical behaviour as opposed to yearly seasonality.

The Spearman correlation heat maps (Figure 8) show strong correlation between dam levels and 'Soil Moisture,' 'Abstraction' and 'Spill.' Stronger correlation results can be observed between features, especially the yearly cosine signal ('Year cos') which has strong positive correlations to 'Max Temp,' 'Penman Evaporation,' 'Vapour Pressure' and 'Solar Radiation,' and a strong negative correlation to 'Soil Moisture.' This suggests further evidence of seasonality.

The Shapley values of the features for each dam (Figure 9) show 'Soil Moisture,' 'Compensation,' 'Abstraction,' 'Year sin' and 'Year cos' as having stronger feature importance scores for predicting the dam level using XGBoost. This is fairly consistent with the results obtained from the correlation analysis.

Looking at these results, the lack of significant correlation and poor feature importance values for the rainfall features could suggest that rainfall may not have an immediate effect on the dam levels, and soil moisture may have a stronger immediate effect in dictating dam levels. This could be due to low antecedent moisture conditions of the soil, which results in delayed runoff responses to rainfall. During prolonged drought conditions, low antecedent moisture conditions may lead to delayed rainfall responses to meet soil saturation before runoff can occur. Moreover, native bush coverage in all three catchments is high, which may indicate greater interception by vegetation during rainfall events, which can lead to less through-fall of precipitation into the surface. A technical report by Manaaki Whenua Landcare Research for the Ministry for the Environment (MfE) shows approximately 33–37% of interception loss can be observed for native Aotearoa New Zealand bush (Rowe et al., 2002). Which means, for an average yearly rainfall depth of 1000mm, approximately 350mm can be lots due to interception.

However, care should be taken when using data-driven models to explain physical processes. The data used in this project came from two separate sources, and while the measuring variables are different, both organisations may have vastly different measurement accuracies and procedures. Additionally, derived features, such as 'Penman Evaporation,' 'Soil Moisture,' etc., that are not directly measured add another layer of uncertainty and error to the analysis and forecasting tasks. Furthermore, the accuracy of modelled data from the VCSN grids may have varying degrees of accuracy based on topography and the time of year. According to Tait and Woods:

> An error analysis at 20 validation sites has shown that the average RMSE of the daily PET interpolations for the validation sites varies between about 1 mm in summer and 0.4 mm in winter. The average relative error (the RMSE divided by the average daily PET) is between 22% and 26% throughout the important growing season months (October–April), but increases to 60%–70% over winter when the average daily PET at the validation sites is only around 0.6 mm. (2007, p. 436)

Forecasting

The forecasting results at the 7- and 30-day time-steps for both VAR and RNN models show promising results for the Upper Nihotupu and Mangatangi Dams. While the RNN models (in particular the GRU variant) had better performance for the 30-day forecast, the difference between the two methods are marginal at best, suggesting traditional linear models such as VAR can perform as well as more complex neural network-based models for the forecasting task.

Comparatively, the results for the Mangatangi Dam favour the naive persistence model, which suggests, based on the forecast at the 7- and 30-day horizon, the dam level may not fluctuate much from the measurement at the current time-step. The VAR and RNN models output a noisier signal at each time-step, which may suggest models are potentially overfitting to the training data, which results in poor generalisation on the test set. Additionally, relatively low fluctuations in dam levels in Mangatangi are further demonstrated by the relatively small standard deviation 3.78 (with mean: 50.24). By comparison, both Upper Nihotupu and Waitākere Dams show greater fluctuations, as shown in their relatively high standard deviations from the mean. These might indicate that VAR and RNN models might be better at prediction tasks where fluctuations and standard deviations are relatively high, while for lower rates of fluctuation, a simple naive model may work better.

The NoL models had poor performance; however this was expected as the model inputs did not contain the 'Level' feature (i.e., the dam level at the current time-step). This meant the NoL models had to 'learn' to predict an output for the dam level, given only the 12 hydrometric and anthropogenic parameters. The NoL results for the Upper Nihotupu and Waitākere Dams show reasonable prediction accuracy in this regard and this might indicate that there is, in fact, an underlying relationship between the features and the water level.

CONCLUSION

This project presents a statistical analysis and forecast of the dammed water-level in three water-supply dams in the Tāmaki Makaurau Auckland and Waikato region: Upper Nihotupu, Waitākere and Mangatangi Dams. An analysis of the time-series data reveals no underlying trend component; however, there is strong indication of seasonality. The correlation and feature importance tests show certain variables have a stronger correlation with the dam levels, while others show little to no correlation. However, interpreting these results must be done with caution due to the uncertainties in the accuracy of data. Seasonality, in particular, had a strong effect on results, as greater prediction errors were found for the months of April, May, June and July.

The forecasting component of this project used two models, VAR and RNN, to forecast the model water level at a 7- and 30-day time horizon. An additional persistence model to base performance on was utilised in the assessment. Both VAR and RNN models performed better than the persistence model for the Upper Nihotupu and Waitākere Dam level predictions, however the persistence model had better performance for the Mangatangi Dam. Prediction accuracy between VAR and RNN models was nearly identical, with RNN models having a slightly better prediction accuracy as shown by the RMSE and MAE results. An additional model – NoL RNN – was created, in which the model was trained to predict the water level by only using the hydrometric and anthropogenic features as inputs. The performance of these models showed that these features might contain learnable parameters that can be used to forecast water levels.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI 16) (pp. 265–283). https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Abdallah, W., Abdallah, N., Marion, JM. (2020). A vector autoregressive methodology for short-term weather forecasting: Tests for Lebanon. *SN Applied Sciences, 2*, 1555. https://doi.org/10.1007/s42452-020-03292-y

Anindita, A. P., Laksono P., & Nugraha, I. G. B. B. (2016). Dam water level prediction system utilizing Artificial Neural Network Back Propagation: Case study: Ciliwung watershed, Katulampa Dam. *2016 International Conference on ICT For Smart Society (ICISS)*, 16–21. https://doi.org/10.1109/ICTSS.2016.7792862

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Networks, 5(2)*, 157–166. https://doi.org/10.1109/72.279181

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, *11(7)*, 1–20. https://doi.org/10.3390/en11071636

Bowden, G. J., Dandy, G. C., & Maier, H. R. (2005). Input determination for neural network models in water resources applications. Part 1 – background and methodology. *Journal of Hydrology, 301*(1–4), 75–92. https://doi.org/10.1016/j.jhydrol.2004.06.021

Burman, R., & Pochop, L. O. (1994). *Evaporation, evapotranspiration and climatic data*. Elsevier.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 785–794. https://doi.org/10.1145/2939672.2939785

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734). Association for Computational Linguistics. https://aclanthology.org/D14-1179

Choi, C., Kim, J., Han, H., Han, D., & Kim, H. S. (2019). Development of water level prediction models using machine learning in wetlands: A case study of Upo Wetland in South Korea. *Water, 12*. https://doi.org/10.3390/w12010093

Chollet, F. (2015). *Keras*. GitHub. Retrieved from https://github.com/fchollet/keras

Cichota, R., Snow, V. O., & Tait, A. B. (2008). A functional evaluation of virtual climate station rainfall data. *New Zealand Journal of Agricultural Research*, *51*(3), 317–329. https://doi.org/10.1080/00288230809510463

Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. Prentice-Hall.

Filho, A. R. G., Silva, D. F. C., de Carvalho, R. V., de Souza Lima Ribeiro F., & Coelho, C. J. (2020). Forecasting of water flow in a hydroelectric power plant using LSTM recurrent neural network. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, (1–5). https://doi.org/10.1109/ICECCE49384.2020.9179373

Fowler, A., & Adams, K. (2004). Twentieth-century droughts and wet periods in Auckland (New Zealand) and their relationship to ENSO. *International Journal of Climatology.* https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.1100

Gèlron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hamilton, J. D. (1994). *Time series analysis*. Princeton University Press.

Hatemi-J, A., & Hacker, R. S. (2009). Can the LR test be helpful in choosing the optimal lag order in the VAR model when information criteria suggest different lag orders? *Applied Economics*, *41*(9), 1121–1125. https://doi.org/10.1080/00036840601019273

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9(8)*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep Learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, *10*(11), 1543. https://doi.org/10.3390/w10111543

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., & Klambauer, G. (2019). NeuralHydrology – Interpreting LSTMs in hydrology. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, & K-L. Müller (Eds.), *Explainable AI: Interpreting LSTMs in hydrology* (pp. 347–362). Springer. https://doi.org/10.1007/978-3-030-28954-6_19

Kumar, U., Prakash, A., & Jain, V.K. (2009). A multivariate time series approach to study the interdependence among O3, NOx, and VOCs in ambient urban atmosphere. *Environmental Modeling and Assessment, 14*, 631–643. https://doi.org/10.1007/s10666-008-9167-1

Lundberg, S., & Lee, S-I. (2017). A unified approach to interpreting model predictions. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4768–4777. https://dl.acm.org/doi/10.5555/3295222.3295230

McKinney, W., et al. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56. https://doi.org/10.25080/Majora-92bf1922-00a

McKinney, W., Perktold, J., & Seabold, S. (2011). Time series analysis in Python with Statsmodels. *Proceedings of the 10th Python in Science Conference* (107–113). https://doi.org/10.25080/Majora-ebaa42b7-012

Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of Cardiac Anaesthesia*, *22*(1), 67–72. https://doi.org/10.4103/aca.ACA_157_18

Molnar, C. (2019, September 26). *Interpretable machine learning. A guide for making black box models explainable*. https://christophm.github.io/interpretable-ml-book/

Morton, J. (2020, June 23). Dry 2020 forecast: Auckland's water woes turn 'critical'. *New Zealand Herald*. https://www.nzherald.co.nz/nz/dry-2020-forecast-aucklands-water-woes-turn-critical/4LKDZGZTELR2K3TOV2JTHQ4DMA/

National Institute of Water and Atmospheric Research. (2021). *Virtual Climate Station data and products*. https://niwa.co.nz/climate/our-services/virtual-climate-stations

Rowe, L. K., Jackson, R. J., Fahey, B. D. (2002). *Land-use and water resources: Hydrological effects of different vegetation covers*. SMF2167 Report 5. Landcare Research Contract Report LC0203/027 for the Ministry for the Environment.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *9th Python in Science Conference*. https://doi.org/10.25080/Majora-92bf1922-011

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3-4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, *48(1),* 1–48. https://doi.org/10.2307/1912017

Spearman, C. (1904). The proof and measurement of association between two things. *American Journal of Psychology,* 1, 72–101.

Sposób J. (2011). Water balance in terrestrial ecosystems. In J. Gliński, J. Horabik, & J. Lipiec (Eds), *Encyclopedia of agrophysics*. Encyclopedia of Earth Sciences Series. Springer. https://doi.org/10.1007/978-90-481-3585-1_267

Tait, A., & Woods, R. (2007). Spatial interpolation of daily potential evapotranspiration for New Zealand using a Spline model. *Journal of Hydrometeorology*, *8*(3), 430–438.

TensorFlow. (2021). *Time series forecasting*. https://www.tensorflow.org/tutorials/structured_data/time_series

Watercare. (2020). *Waikato Water Treatment Plant*. https://www.watercare.co.nz/About-us/Projects-around-Auckland/Waikato-Water-Treatment-Plant

Watercare. (2021). *Dams*. https://www.watercare.co.nz/Water-and-wastewater/Where-your-water-comes-from/Dams

Yang, S., Yu X., & Zhou, Y. (2020). LSTM and GRU neural network performance comparison study: Taking Yelp review dataset as an example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, (98–101). https://doi.org/10.1109/IWECAI50956.2020.00027

Zheng H., & Wu, Y. (2019). An XGBoost model with weather similarity analysis and feature engineering for short-term wind power forecasting. *Applied Sciences, 9*(15), 3019. https://doi.org/10.3390/app9153019

Zhao, J., Mu, X., & Gao, P. (2019). Dynamic response of runoff to soil and water conservation measures and precipitation based on VAR model. *Hydrology Research*, *50*(3), 837–848. https://doi.org/10.2166/nh.2019.074

AUTHORS

Pramith (Brian) Waidyaratne is an engineer who completed his Bachelor of Engineering Technology degree at Unitec, Te Pūkenga, in 2021 while working at Watercare, and is now a Flood Risk Engineer at Systra in England.

David Phillips is an Associate Professor at Unitec, Te Pūkenga, in the School of Engineering and Construction. He has an interest in engineering infrastructure, stormwater flow quantities and quality, and impacts on the environment, and holds qualifications in civil, environmental and coastal engineering.