



Delft University of Technology

## Machine learning for real-time reservoir operation simulation: comparing input variables and algorithms for the Sirikit Reservoir, Thailand

Wannasin, C.; Brauer, C. C.; Uijlenhoet, R.; Torfs, P. J.J.F.; Weerts, A. H.

### DOI

[10.2166/hydro.2024.153](https://doi.org/10.2166/hydro.2024.153)

### Publication date

2024

### Document Version

Final published version

### Published in

Journal of Hydroinformatics

### Citation (APA)

Wannasin, C., Brauer, C. C., Uijlenhoet, R., Torfs, P. J. J. F., & Weerts, A. H. (2024). Machine learning for real-time reservoir operation simulation: comparing input variables and algorithms for the Sirikit Reservoir, Thailand. *Journal of Hydroinformatics*, 26(12), 3151-3171. Article jh2024153. <https://doi.org/10.2166/hydro.2024.153>

### Important note

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

### Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

### Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

## Machine learning for real-time reservoir operation simulation: comparing input variables and algorithms for the Sirikit Reservoir, Thailand

C. Wannasin <sup>a,b,c,\*</sup>, C. C. Brauer <sup>a</sup>, R. Uijlenhoet <sup>d</sup>, P. J. J. F. Torfs<sup>a</sup> and A. H. Weerts <sup>a,e</sup>

<sup>a</sup> Hydrology and Environmental Hydraulics Group, Wageningen University and Research, P.O. Box 47, 6700 AA Wageningen, The Netherlands

<sup>b</sup> Faculty of Engineering Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

<sup>c</sup> Marine and Coastal Systems, Department of Hydrodynamics and Forecasting, Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands

<sup>d</sup> Department of Water Management, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands

<sup>e</sup> Operational Water Management, Department of Inland Water Systems, Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands

\*Corresponding author. E-mail: c.wannasin@utwente.nl; mo.wannasin@deltares.nl

 CW, 0000-0002-3311-5205; CCB, 0000-0002-6459-9230; RU, 0000-0001-7418-4445; AHW, 0000-0002-3249-8363

### ABSTRACT

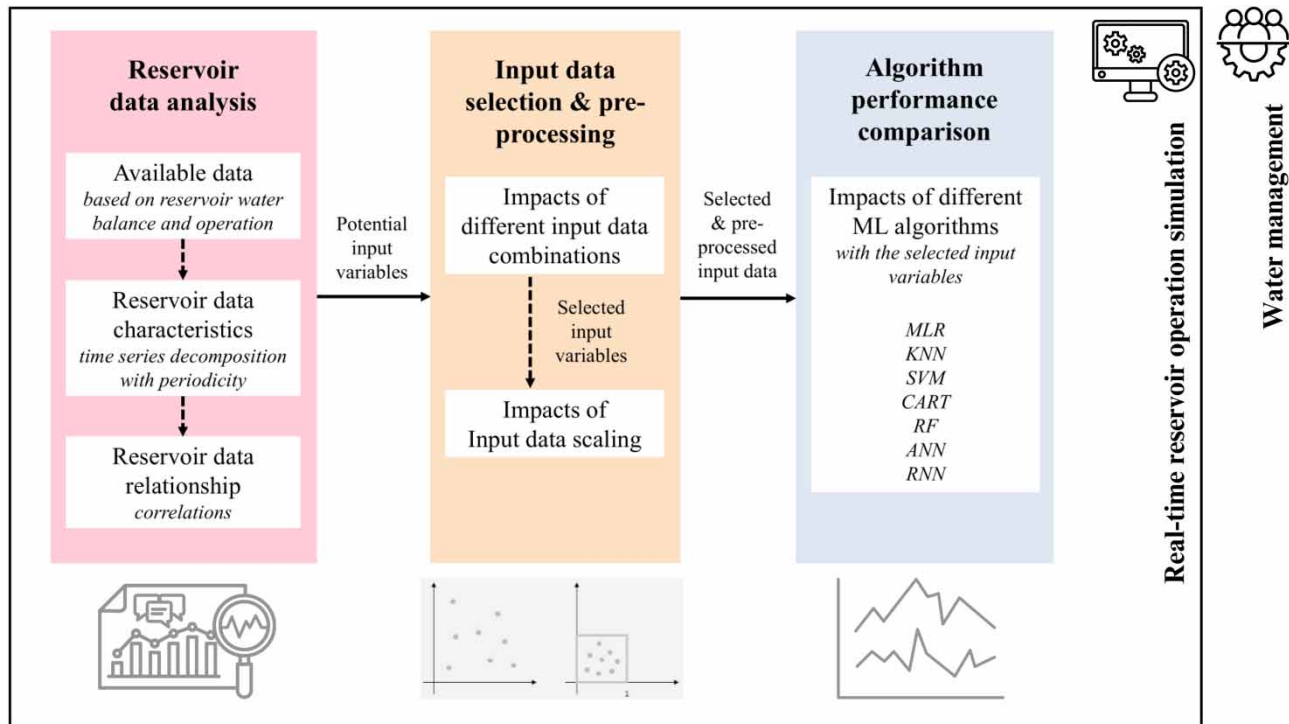
Machine learning (ML) models offer advantages over process-based models for real-time reservoir operation modelling, yet the impact of input variable selection (IVS) and data pre-processing on model performance remains underexplored. This study investigates various input variables for simulating daily reservoir outflow, using the Sirikit reservoir in Thailand as a case study. The datasets include daily Sirikit storage and inflow, outflow of Bhumibol (neighbouring reservoir), downstream discharge, and temporal factors (month and day of the week). Time series decomposition and correlation analyses were used to assess data relationships. We tested seven ML models: multiple linear regression, support vector machine, K-nearest neighbour, classification and regression tree, random forest, multi-layer perceptron, and recurrent neural network (RNN). The optimal input set comprised the previous day's storage, inflow from 2 days before to 2 days after, and month. With these inputs, all ML models simulated outflow adequately ( $KGE_{\text{training}} = 0.42\text{--}1.0$  and  $KGE_{\text{testing}} = 0.46\text{--}0.56$ ), with RNN showing the most potential for improvement. Input scaling significantly enhanced model performance, reducing  $RMSE_{\text{training}}$  by  $44\text{ m}^3\text{ s}^{-1}$  and  $RMSE_{\text{testing}}$  by  $14\text{ m}^3\text{ s}^{-1}$ . This study's novelty lies in its comprehensive insights of IVS and data scaling, highlighting their critical roles in enhancing ML model application for operational reservoir simulations.

**Key words:** input variable selection, input variable scaling, machine learning, multi-purpose reservoir, real-time reservoir operation, upper Chao Phraya River basin

### HIGHLIGHTS

- Machine learning (ML) models can adequately simulate the daily outflow of Sirikit reservoir using (1) the past storage, (2) past and future inflow, and (3) month of the year.
- The month of the year effectively represents the operating rule curves of the Sirikit reservoir in all selected ML algorithms.
- Scaling the input data improves the accuracy of the Sirikit outflow simulations across all selected ML algorithms.

## GRAPHICAL ABSTRACT



## 1. INTRODUCTION

Reservoirs and dams are key infrastructures for water resource management worldwide. They serve various purposes, including flood and drought control, irrigation planning, hydropower generation, and water supply management. The number of reservoirs is increasing, particularly in developing regions (Zarfl *et al.* 2015). However, climate change – through altered precipitation patterns and extreme weather – poses significant challenges to their operation and management (Şen 2021). These challenges are especially pronounced in monsoon-dominated regions, where balancing water supply and demand throughout the year becomes difficult. In addition, reservoirs can have adverse environmental and societal impacts, including ecosystem disruptions and increased vulnerability to floods and droughts due to mismanagement (Di Baldassarre *et al.* 2018; Mei *et al.* 2018). Addressing these issues underscores the urgent need for efficient reservoir operation modelling (the process of developing or setting up the calculation model) and real-time simulation (using the model). However, this remains a complex task, especially for large, multi-purpose reservoirs, requiring advanced approaches to ensure accuracy and efficiency.

Process-based models have been commonly used for reservoir operation modelling, focusing on simulating controlled outflows and storage. Notable models such as CalSim (Draper *et al.* 2004), WEAP21 (Yates *et al.* 2005), and HEC-ResSim (Klipsch & Evans 2006) provide transparent representations and interpretable calculations of reservoir operation. These advantages make them easily trusted by reservoir operators and water managers from both practical and policy perspectives. However, they rely heavily on predefined reservoir operating policies (e.g., rule curves), which may not reflect real-time operational decisions made by reservoir operators in response to immediate conditions (Oliveira & Loucks 1997). This reliance makes it difficult for process-based models to simulate real-time operations and controlled outflows accurately, especially during extreme events, due to their limited complexity and mathematical representation of dynamic decision-making processes.

In the past two decades, data-driven models, particularly those based on machine learning (ML), have gained attention in hydrology, including reservoir operation modelling. These models have the potential to provide higher accuracy in simulating real-time reservoir operations and outflows compared to process-based models. ML algorithms establish data-driven relationships between input variables (e.g., reservoir storage) and output variable (i.e., reservoir outflows) without relying on explicit mathematical representations (Yassin *et al.* 2019). While ML models may lack interpretability due to their black-box nature,

they excel at capturing complex, nonlinear, and high-dimensional relationships between input and output data, which is beneficial for real-time reservoir operation and outflow simulations.

Previous studies have applied various ML models with different learning approaches for reservoir operation modelling, including instance-based learning, decision trees, ensemble learning, and neural networks. Notable algorithms include K-nearest neighbour (KNN; e.g., [Yang et al. 2021](#)) and support vector machine (SVM; e.g., [Niu et al. 2019](#)) for instance-based learning, classification and regression tree (CART; e.g., [Chen et al. 2022](#)) for decision tree, random forest (RF; e.g., [Qie et al. 2022](#)) for ensemble learning, and multi-layer perceptron (MLP; e.g., [Zarei et al. 2021](#)) and recurrent neural networks (RNNs; e.g., [Zhang et al. 2019](#)) for neural networks. Each ML algorithm has distinct characteristics with its own advantages and drawbacks, making the algorithm selection dependent on aims of this study, available data, and computational resources.

Input variable selection (IVS) is crucial for ML modelling, significantly affecting modelling speed and predictive capabilities ([Jain et al. 2020](#)). Recent studies have underscored its importance in environmental and hydrological applications. [Galelli et al. \(2014\)](#) established a framework for evaluating IVS algorithms, emphasizing systematic assessment based on accuracy, computational efficiency, and robustness. [Snieder et al. \(2020\)](#) evaluated several IVS methods, each offering unique advantages for optimizing neural networks for flow forecasting. [Gharib & Davies \(2021\)](#) demonstrated pitfalls in the IVS process, such as overlooking data quality and the feature relevance, and proposed a workflow that includes careful evaluation and validation of selected variables. [Moreido et al. \(2021\)](#) highlighted that incorporating expert knowledge significantly enhances the effectiveness of automated IVS techniques.

In addition to IVS, input data pre-processing can affect ML model accuracy ([Ahsan et al. 2021](#)). In hydrological applications, [Lange & Sippel \(2020\)](#) and [Xu & Liang \(2021\)](#) emphasized the critical role of input data preprocessing, highlighting techniques such as data scaling, data cleaning, and handling missing values to enhance model performance, while stressing the importance of understanding data characteristics and domain knowledge for effective preprocessing strategies.

Several ML model applications specifically for daily reservoir outflow simulations have primarily focused on either comparing the performance of different ML algorithms (e.g., [Yang et al. 2019](#)) or optimizing a single algorithm's performance (e.g., [Yang et al. 2016](#)). In addition, some studies have explored detailed training, fine-tuning, and parameterization processes (e.g., [Zhang et al. 2019](#)). However, there has been limited exploration of the influence of IVS and data preprocessing on ML model accuracy, reliability, and applicability (e.g., [Chen et al. 2018](#)).

In reservoir operation modelling, IVS and data pre-processing can play crucial roles in improving model performance, yet they have not received sufficient attention. This study addresses this gap by systematically investigating how these factors affect the performance of various ML algorithms for daily reservoir outflow simulations. The case study of the Sirikit reservoir in Thailand offers a novel contribution for improving real-time simulations of large, multi-purpose reservoirs with complex operation in monsoon-dominated regions.

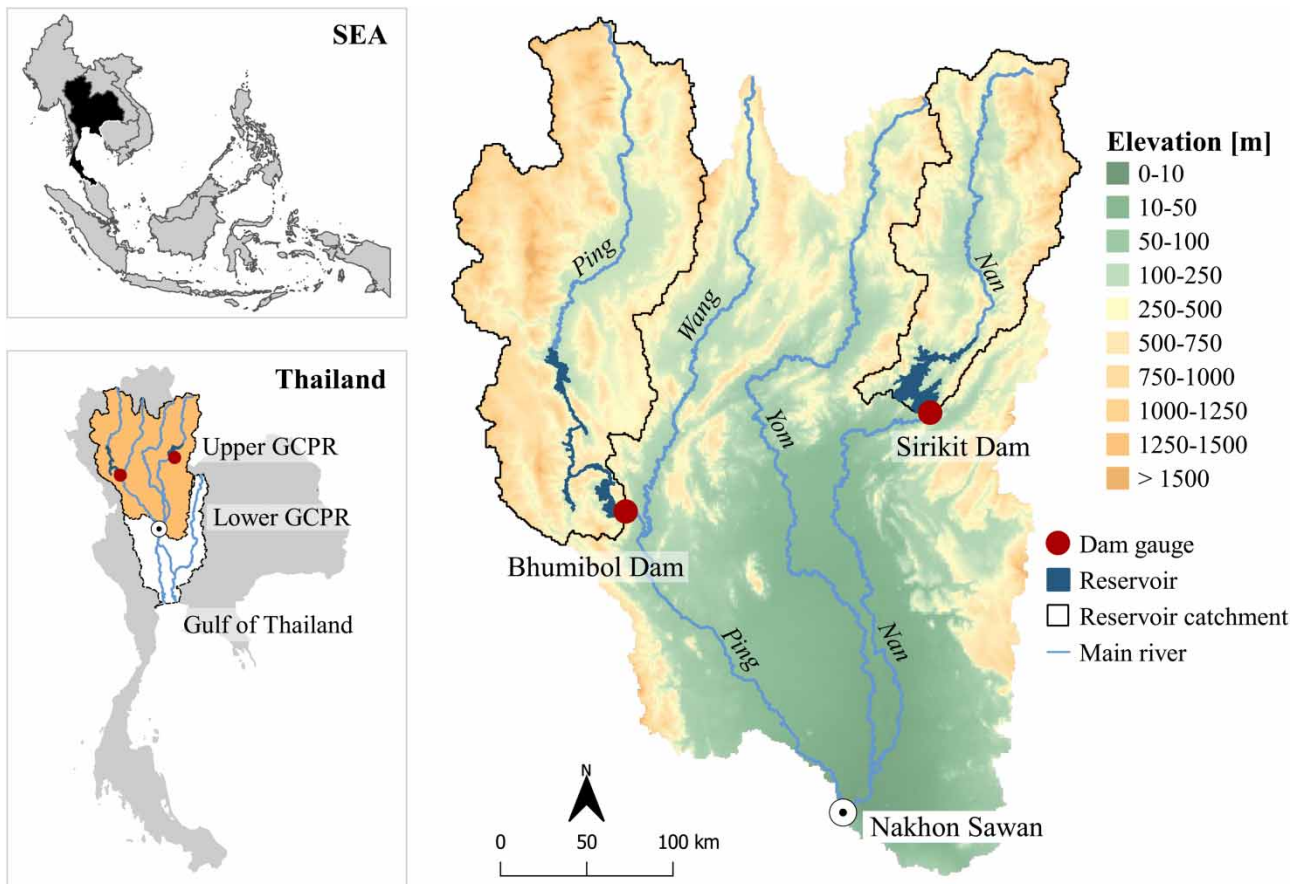
## 2. METHODS

### 2.1. Study area and reservoirs

The Greater Chao Phraya River (GCPR) basin in Thailand covers approximately 158,600 km<sup>2</sup>, equivalent to about 30% of the country's area ([Figure 1](#)). It is divided into an upper region (66%) and a lower region (34%) at Nakhon Sawan. The upper region includes the Ping, Wang, Yom, and Nan Rivers. These rivers converge around Nakhon Sawan into the Chao Phraya River, which continues through the lower region before reaching the Gulf of Thailand. The GCPR basin experiences the tropical climate with monsoons and cyclones. The majority of precipitation occurring from May to October, leading to flood risks, especially around Nakhon Sawan and lower GCPR areas. During the dry season from November to April, the basin heavily depends on artificial water sources due to extensive agricultural, industrial, and residential areas.

Over the past 60 years, seven main dammed reservoirs have been constructed and operated in the upper GCPR region for basin-wide purposes, including irrigation, hydropower, water supply, and flood and drought control. Bhumibol and Sirikit reservoirs (location shown in [Figure 1](#)), the largest reservoirs, have capacities of 13.46 billion m<sup>3</sup> and 9.51 billion m<sup>3</sup>, respectively. They represent 93% of the total reservoir capacity (24.7 billion m<sup>3</sup>) in the upper region. Five other reservoirs were constructed on the Ping River (two), Wang River (two), and Nan River (one).

This study focuses on simulating the operation and outflow of the Sirikit reservoir, which experienced the highest daily outflow (809 m<sup>3</sup> s<sup>-1</sup>) among all reservoirs during flood periods in the basin. The Bhumibol reservoir, which operates concurrently, was also considered in the analysis. The Sirikit reservoir was selected due to its critical role in the GCPR basin's water



**Figure 1** | Overview of the study area and its major reservoirs. The top left panel indicates Thailand's location in Southeast Asia. The bottom left panel shows the GCRP basin in Thailand. The right panel illustrates the spatial distribution of Sirikit and Bhumibol reservoirs in the upper GCRP basin, with solid black lines delineating the drainage areas from the headwaters to the dam locations (adapted from the study by Wannasin *et al.* 2021b).

management. Its complex multi-purpose operation, requiring careful balance of storage and outflow, makes it an ideal case study for advancing real-time reservoir operation modelling in Thailand and other monsoon-dominated regions.

## 2.2. Reservoir data

### 2.2.1. Reservoir operation and reservoir water balance

Real-time operation of a multi-purpose reservoir is a complex task, involving determining outflow and maintaining stored water to fulfil various purposes in ever-changing conditions. In general, the reservoir operation is based on the reservoir water balance, storage zoning, operating rule curves, total downstream water demand (for irrigation, hydropower generation, flood and drought control, water supply and environmental flow), and operators' judgements. The simplified real-time reservoir operation function, adapted from the studies by Lund & Guzman (1999), Jain & Singh (2003), and Yang *et al.* (2019), is described in the Supplementary Material (Equation (S1)).

The reservoir water balance plays an important role in real-time reservoir operation. It includes reservoir outflow, reservoir inflow from surface water and groundwater, reservoir storage, precipitation and evaporation over the reservoir surface, and storage loss (e.g., groundwater seepage). Following the studies by Kumar & Reddy (2007) and Yang *et al.* (2019), a simplified relationship among reservoir outflow, inflow, and storage at the daily timescale is described in Equation (S2).

### 2.2.2. Available data for the Sirikit reservoir

Several available datasets for the Sirikit reservoir align with the reservoir water balance components in Equation (S2). Historical daily time series data, including reservoir outflow, storage, and downstream flow, were collected from the Royal

Irrigation Department (RID) and the Electricity Generating Authority of Thailand (EGAT) over a 10-year period from 2004 to 2013. In addition, we obtained daily reservoir inflow data through a hydrological simulation using the *wflow\_sbm* model.

The *wflow\_sbm* model is a fully distributed rainfall-runoff model (Van Verseveld *et al.* 2024), which was used to simulate the discharge generation on the land surface upstream of the Sirikit reservoir, and therefore the amount of river water flowing into the reservoir. This reservoir inflow is one of the inputs for our ML models. The setup of the (~1 km spatial resolution) *wflow\_sbm* model and the datasets used as input for that model, as well as a complete validation and analysis of the output, are described in detail in our previous study (Wannasin *et al.* 2021b).

Obtaining data and information on the reservoir operation components in Equation (S1) is challenging, but can be deduced from available sources. The downstream water demand and operating rule curves vary by month (seasonality). Long-term monthly reference values for the water demand at Nakhon Sawan can be extracted from historical time series, while the operating rule curves are available from reports (Wannasin *et al.* 2021b). However, using these values directly in ML models may be ineffective due to year-to-year variations in reality. Daily outflow and storage data somewhat reflect operators' decisions and there is a 7-day periodicity (weekly cycle) in outflow for irrigation (Tebakari *et al.* 2012). In addition, Bhumibol reservoir outflow is considered valuable as it operates simultaneously with the Sirikit reservoir.

To incorporate the essential and available reservoir-related data in our analysis, we, therefore, investigated Sirikit reservoir outflow ( $Q$ ), Sirikit reservoir storage ( $S$ ), Sirikit reservoir inflow ( $I$ ), Bhumibol reservoir outflow ( $Q_B$ ), downstream river discharge ( $Q_D$ ), and timing information, including the month of the year ( $M$ ) for the seasonality and the day of the week ( $D$ ) for the weekly cycle. Our goal was to simulate  $Q$  while considering the others as potential input variables of ML models. The statistical information of these datasets is supplied in Table S1.

### 2.3. Input variable selection and data pre-processing

To investigate and select input variables, we chose correlation-based IVS due to the small number of variables and their significant lagged and lead effects on real-time reservoir operation and simulation. This method is straightforward and can reveal both univariate and bivariate relationships. We acknowledged its reliance on linear relationships and sensitivity to outliers. To improve robustness of the IVS process, we followed a three-step approach: analyzing data characteristics, correlations, and input variable combinations.

#### 2.3.1. Analysing reservoir data characteristics

To assess the relevance of the potential reservoir-related data as inputs of the considered ML models for  $Q$  simulations, we performed a comprehensive time series analysis of daily  $Q$ ,  $S$ ,  $I$ ,  $Q_B$ , and  $Q_D$ . This analysis focused on identifying their long-term trends and periodic patterns, including the weekly and monthly cycles, which are valuable signatures for data-driven models. By using time series decomposition, we deconstructed the time series into three components: *trend* (long-term progression), *cycle/seasonality* (recurring behaviours), and *residual* (remaining noise).

First, we decomposed the original time series based on the annual cycle. Subsequently, we decomposed the residual component of the original time series, emphasizing the weekly cycle. By utilizing the residual component, which had already undergone removal of the trend and annual cycle, we could better extract the remaining weekly cycle, which is often less prominent. We employed the classical seasonal decomposition by moving averages function (Kendall 1946) in R (R Core Team 2013). In addition, we conducted the non-parametric Mann-Kendall trend test (Hamed & Rao 1998) to identify long-term trends in the time series data, with a significance level set at 5%.

#### 2.3.2. Analysing reservoir data relationships

We conducted a univariate analysis, using *autocorrelation*, to assess characteristics and time dependencies of individual time series ( $Q$ ,  $S$ ,  $I$ ,  $Q_B$ , and  $Q_D$ ). Autocorrelation measures systematic correlations with lagged values (Jenkins 1968), with the autocorrelation coefficients (ACs) ranging from -1 to 1. The resulting autocorrelation function (ACF) identifies memory effects and periodicity, with high AC values at lags indicating strong linear dependence or potential periodic patterns within the data.

We also performed a *spectral analysis* that can reveal periodicity that may not be clearly visible in the ACF. The spectral density function (SDF), obtained through Fourier transformation, identifies dominant frequencies in the time series (Shumway & Stoffer 2000). The SDF describes total variance over frequency components, showing the correlation with sine/cosine waves (periodogram). While the ACF and SDF are two sides of the same coin, the SDF enhances periodicity visualization, especially after decomposing the residual component.

For the bivariate analysis, we used *cross-correlation* to assess lagged relationships between two time series (Box *et al.* 2015). The cross-correlation function (CCF) identifies the lead or lag between series. The lag with the highest positive or negative cross-correlation coefficient indicates the strongest linear correspondence between the two series.

### 2.3.3. Input variable combinations

After evaluating the potential reservoir-related variables for input into the considered ML models, we examined how different input variable combinations and data scaling affect ML model performance. To simulate  $Q$ , we categorized the potential input variables into Sirikit reservoir variables ( $S$  and  $I$ ), timing variables ( $M$  and  $D$ ), and other related variables ( $Q_B$  and  $Q_D$ ). We created 17 input variable combinations, considering their availability in real-world reservoir operations (Table 1).

### 2.3.4. Data scaling

The potential input variables varied greatly in unit and distribution, with  $S$  ranging from 3.14 to 9.5 million  $\text{m}^3$  and  $I$  from 12 to 4136  $\text{m}^3 \text{s}^{-1}$ . To address this, we investigated how data scaling affects ML model performance. Two common methods are standardization, which scales data to a mean of 0 and standard deviation of 1, and normalization, which scales data between 0 and 1. Since most of the time series had non-Gaussian distributions (Figure S1), we chose normalization (Equation (S3)). In addition, the  $I$  time series was log-transformed to reduce skewness before normalization, while other variables were normalized directly.

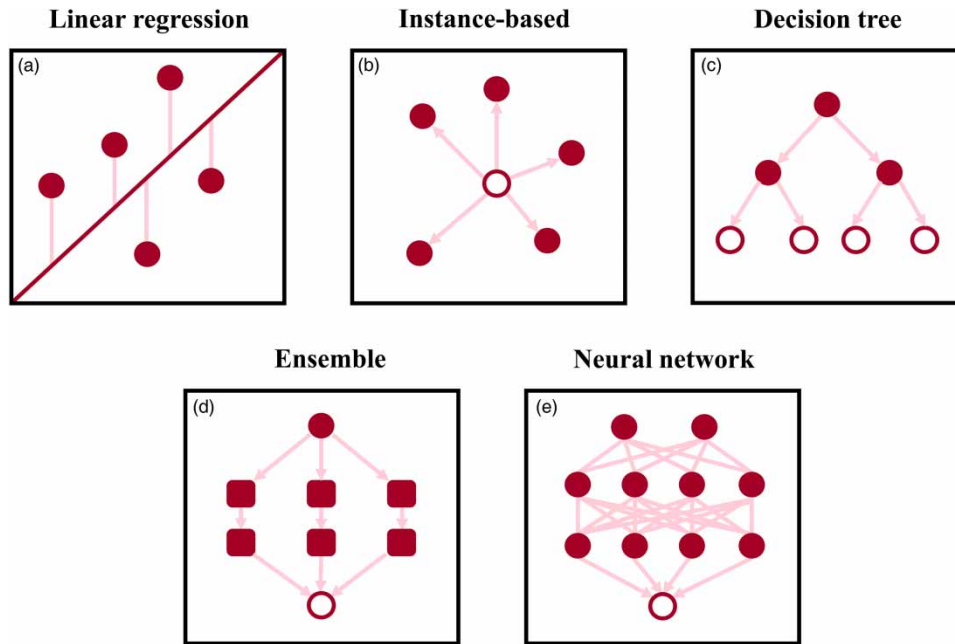
### 2.4. Machine learning algorithms

ML models analyse the input–output relationships to understand their behaviours and calculate output values. An ML model is trained on a dataset to derive a mathematical expression that best fits the data with minimal errors. The trained model can then be tested on unseen data to evaluate their predictive performance.

To simulate  $Q$ , this study explored five regression ML model classes: linear regression, instance-based learning, decision tree, ensemble learning, and neural network, as illustrated in Figure 2. These model classes were selected to represent a diverse range of learning approaches in handling data patterns, ranging from simple linear models to more complex ones.

**Table 1** | Input scenarios for machine learning models to simulate the Sirikit reservoir outflow  $Q$ . The inputs include the Sirikit reservoir storage ( $S$ ), the Sirikit reservoir inflow ( $I$ ), the timing data (month of the year ( $M$ ) and day of the week ( $D$ )), and other related data (the Bhumibol reservoir outflow ( $Q_B$ ) and the downstream discharge ( $Q_D$ )).  $t$  is the current model time step

| Scenarios | Input variables        |                                                             |             |                        | Investigation                   |
|-----------|------------------------|-------------------------------------------------------------|-------------|------------------------|---------------------------------|
|           | Reservoir storage data | Reservoir inflow data                                       | Timing data | Other data             |                                 |
| Baseline  | $S_{t-1}$              | $I_{t-1}$                                                   |             |                        |                                 |
| 1         | $S_{t-2}, S_{t-1}$     | $I_{t-1}$                                                   |             |                        | $S$                             |
| 2         | $S_{t-1}$              | $I_{t-2}, I_{t-1}$                                          |             |                        | $I$                             |
| 3         | $S_{t-2}, S_{t-1}$     | $I_{t-2}, I_{t-1}$                                          |             |                        | $S$ & $I$                       |
| 4         | $S_{t-1}$              | $I_{t-3}, I_{t-2}, I_{t-1}$                                 |             |                        | $I$ (past)                      |
| 5         | $S_{t-1}$              | $I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}$                        |             |                        | $I$ (past)                      |
| 6         | $S_{t-1}$              | $I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}, I_t$                   |             |                        | $I$ (past and present)          |
| 7         | $S_{t-1}$              | $I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}, I_t, I_{t+1}$          |             |                        | $I$ (past, present, and future) |
| 8         | $S_{t-1}$              | $I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$ |             |                        | $I$ (past and present & future) |
| 9         | $S_{t-1}$              | $I_{t-3}, I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$          |             |                        | $I$ (past, present, and future) |
| 10        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   |             |                        | $I$ (past, present, and future) |
| 11        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $M$         |                        | Timing                          |
| 12        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $D$         |                        | Timing                          |
| 13        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $M, D$      |                        | Timing                          |
| 14        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $M$         | $Q_{B,t-1}$            | Other factor                    |
| 15        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $M$         | $Q_{D,t-1}$            | Other factor                    |
| 16        | $S_{t-1}$              | $I_{t-2}, I_{t-1}, I_t, I_{t+1}, I_{t+2}$                   | $M$         | $Q_{B,t-1}, Q_{D,t-1}$ | Other factor                    |



**Figure 2** | The generalized characterization of the selected machine learning classes, including the linear regression algorithm (multiple linear regression (MLR)), instance-based learning algorithms (KNN and SVM), decision tree algorithm (CART), ensemble learning algorithm (RF), and neural network algorithms (MLP and RNN).

Linear regression provides straightforward linear relationships between inputs and outputs. Instance-based learning identifies patterns based on similarities between data points. Decision trees create rules by splitting data based on input features. Ensemble learning combines multiple decision trees to manage complex data. Neural networks capture complex, nonlinear relationships, especially in sequential data. This diversity ensured a comprehensive evaluation and comparison of ML model performance in the reservoir outflow simulation, offering valuable insights into the capability of each model class.

#### 2.4.1. Linear regression: multiple linear regression

Linear regression assumes a linear relationship between input and output variables (Figure 2(a)). This study applied MLR, which uses multiple input variables to predict an output variable. MLR is straightforward, explainable, and computationally efficient, but may have limitations with nonlinear processes and fewer inputs (Helsel *et al.* 2020). It commonly serves as a benchmark for comparing other ML algorithms (e.g., Zhang *et al.* 2018), including in this study.

#### 2.4.2. Instance-based learning: K-nearest neighbour and support vector machine

Instance-based learning algorithms, including KNN and support vector machine (SVM), learn from training data and generalize to new instances based on similarity measures (Figure 2(b)). KNN predicts nonlinear time series by selecting similar past instances (nearest neighbours) and assigning weights based on their distances (Kramer 2013). While easy to implement, KNN may struggle with new events and larger datasets (Halder *et al.* 2024). SVM finds a hyperplane in a multi-dimensional space to fit the data points using decision boundary lines (Vapnik 2000). It captures nonlinearity with promising performance but can be computationally expensive for large datasets (Naganna & Deka 2014). Both KNN and SVM have been used in reservoir operation modelling studies, particularly for operation optimization (e.g., Niu *et al.* 2019; Yang *et al.* 2021).

#### 2.4.3. Decision tree: classification and regression tree

Decision trees are hierarchical algorithms that recursively split data based on decision rules at each node (Figure 2(c)). This study used CART, which uses the Gini index to select branches with the greatest reduction in impurity (Breiman *et al.* 1984). CART is transparent, allowing a clear interpretation of the model and data, but sensitive to noise and prone to overfitting (Loh 2014). It has been applied in reservoir operation modelling studies for understanding reservoir operation (e.g., Chen *et al.* 2022).

#### 2.4.4. Ensemble learning: random forest

Ensemble learning algorithms combine multiple models to improve predictive performance (Figure 2(d)). This study used RF, which fits decision trees on different samples of the same dataset and average their predictions (Breiman 2001). It reduces prediction variance and mitigates overfitting issues, but cannot extrapolate outside the training range (Tyrallis *et al.* 2019). Therefore, RF has gained popularity in reservoir operation modelling studies as it tends to overcome CART limitations (e.g., Qie *et al.* 2022).

#### 2.4.5. Artificial neural networks: multi-layer perceptron and recurrent neural network

Artificial neural networks (ANNs), inspired by the human brain, use interconnected mathematical functions to model complex relationships between inputs and outputs (Figure 2(e)). This study explored MLP and recurrent neural network (RNN). MLP, a common ANN, has one input layer, one or more hidden layers, and one output layer. It processes inputs with weighted connections and activation functions, adjusting weights using backpropagation with stochastic gradient descent, and thus, it may face local optima issues (Zhang *et al.* 2019). RNN, designed for sequential data, incorporates previous information in the sequence as feedback loops. It is advanced but may encounter gradient vanishing and exploding (Sit *et al.* 2020). Nonetheless, both MLP and RNN have been successfully applied in reservoir operation modelling studies (Chaves & Chang 2008; Zhang *et al.* 2019).

#### 2.4.6. Machine learning packages

The seven ML models (MLR, KNN SVM, CART, RF, MLP, and RNN) were implemented in user-friendly and open-source ML frameworks in python, including Scikit-learn (Pedregosa *et al.* 2011), TensorFlow (Abadi *et al.* 2016), and Keras (Ketkar 2017), as listed in Table 2.

### 2.5. Algorithm performance evaluation and comparison

Using the 17 input variable scenarios (Table 1), we compared the  $Q$  simulation results of the seven ML models. To ensure fairness, the models were constructed with their simplest structures and default hyperparameters. The MLP and RNN models had one input layer, one processing layer (64 neurons), and one output layer, trained for 50 iterations using a learning rate of 0.001 with the adaptive moment estimation optimizer. Note that this study focuses on performance comparisons rather than fine-tuning for optimal performance.

The dataset was split into 80% for training (2004–2011) and 20% for testing (2012–2013). The mean square error (MSE) was used to measure error during training and testing, while the root mean square error (RMSE) and Kling Gupta efficiency (KGE) were used to assess and compare model performance. The metric calculations are supplied in Equations (S4)–(S6).

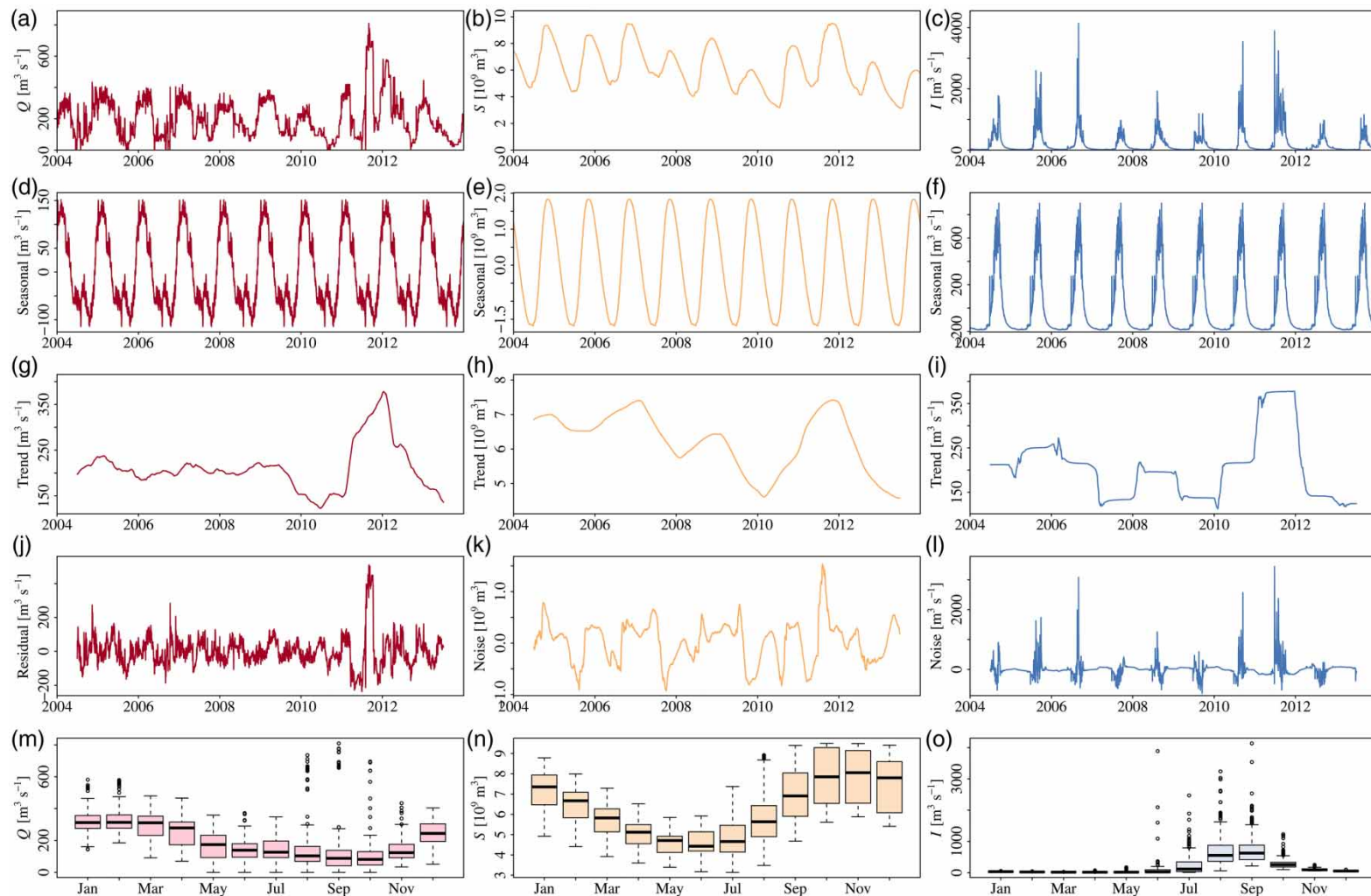
## 3. RESULTS

### 3.1. Reservoir data characteristics

The decomposed time series of the daily Sirikit reservoir data ( $Q$ ,  $S$ , and  $I$ ) are shown in Figure 3. Decomposed time series of other data ( $Q_B$  and  $Q_D$ ) are provided in the Supplementary Material (Figures S2 and S3). Overall, the decomposition reveals trends and periodicity, with the most distinct patterns being the annual cycle (i.e., month of the year).

**Table 2** | Machine learning models and their implemented frameworks and modules

| Algorithm                          | Abbreviation | Framework    | Module                                 |
|------------------------------------|--------------|--------------|----------------------------------------|
| Multiple linear regression         | MLR          | Scikit-learn | sklearn.linear_model.LinearRegression  |
| K-Nearest neighbour                | KNN          |              | sklearn.neighbors.KNeighborsRegressor  |
| Support vector machine             | SVM          |              | sklearn.svm.LinearSVR                  |
| Classification and regression tree | CART         |              | sklearn.tree.DecisionTreeRegressor     |
| Random forest                      | RF           |              | sklearn.ensemble.RandomForestRegressor |
| Multi-layer perceptron             | MLP          | Keras and    | tf.keras.layers.Dense                  |
| Recurrent neural network           | RNN          | TensorFlow   | tf.keras.layers.SimpleRNN              |



**Figure 3** | Decomposition of observed daily Sirikit reservoir time series, including outflow ( $Q$ ; left panels), storage ( $S$ ; middle panels), and inflow ( $I$ ; right panels). The original time series (first row) are decomposed into the seasonal (second row), trend (third row), and residual (fourth row) components. The distribution of the daily time series per month are also shown (last row).

### 3.1.1. Annual cycle

The seasonal components of  $Q$  (Figure 3(d) and (m)) and  $Q_B$  (Figure S2(b) and (e)) exhibit similar patterns. Outflows from both Sirikit and Bhumibol reservoirs peak between November and February (dry season), reflecting water releases for irrigation. Conversely, water is stored between February and October (wet season), with a focus on downstream flood control in October. Accordingly, the seasonal component of  $S$  (Figure 3(e) and (n)) varies oppositely to  $Q$  and  $Q_B$ . The storage increases from June to November, as most rainwater and inflow are stored, and decreases during the entire dry season, as water is continuously released to meet the downstream requirements. In contrast, the seasonality component of  $I$  (Figure 3(f) and (o)) follows a natural pattern, peaking in August or September, before decreasing during the rest of the year. The seasonal component of  $Q_D$  (Figure S3(b) and (e)) reveals the same pattern with a delayed peak in October, implying a rather natural characteristic, despite being influenced by the two reservoirs.

### 3.1.2. Long-term trend

The Mann–Kendall trend test revealed a significant long-term trend (at the 5% significance level with  $p$ -value  $\ll 0.01$ ) for  $Q$  (Figure 3(g)),  $S$  (Figure 3(h)), and  $Q_D$  (Figure S3(c)) over the 10-year period of 2004–2013. This is potentially due to the adjustment of reservoir operation policies, which affected the signatures of  $Q$  between the 1989–1997 and the 2003–2013 period (Wannasin *et al.* 2021a). The trend component of  $S$  tends to increase for 1 year and then decrease for another year, with a notable interruption in 2011 when  $S$  continued to rise due to the substantial inflow ( $I$ ).

Meanwhile, no significant long-term trend was observed for  $I$  (Figure 3(i);  $p$ -value = 0.43) and  $Q_B$  (Figure S2(c);  $p$ -value = 0.21) during this period. The trend component of  $I$  contains humps that reflect the wet and dry years in the basin. It is acknowledged that different time windows and trend-extracting methods may reveal different trends.

### 3.1.3. Weekly cycle

The residuals of the  $Q$  and  $S$  time series (Figure 3(j) and (k)) were further decomposed, focusing on the weekly cycle. The  $Q$  residual shows a pronounced weekly cycle, peaking around Thursday after starting at a lower level on Monday (Figure 4(c)). The SDF (Figure 4(i)) reveals significant spikes at the frequencies of 0.14 and 0.29, corresponding to cycles of approximately 7 days and 3.5 days, respectively. The 3.5-day cycle may be an artefact of the processing. This aligns with the finding of Tebakari *et al.* (2012) that  $Q$  had a 7-day periodicity due to the irrigation requirement.

Conversely, the weekly cycle of  $S$  is less distinct (Figure 4(d) and (j)). The SDF for  $S$  displays several frequency peaks, with periods of approximately 250 days, 17 days, 7 days, and 3 days. This can be because  $S$  does not only depend on reservoir operation but also depend on  $I$ .

## 3.2. Reservoir data relationship

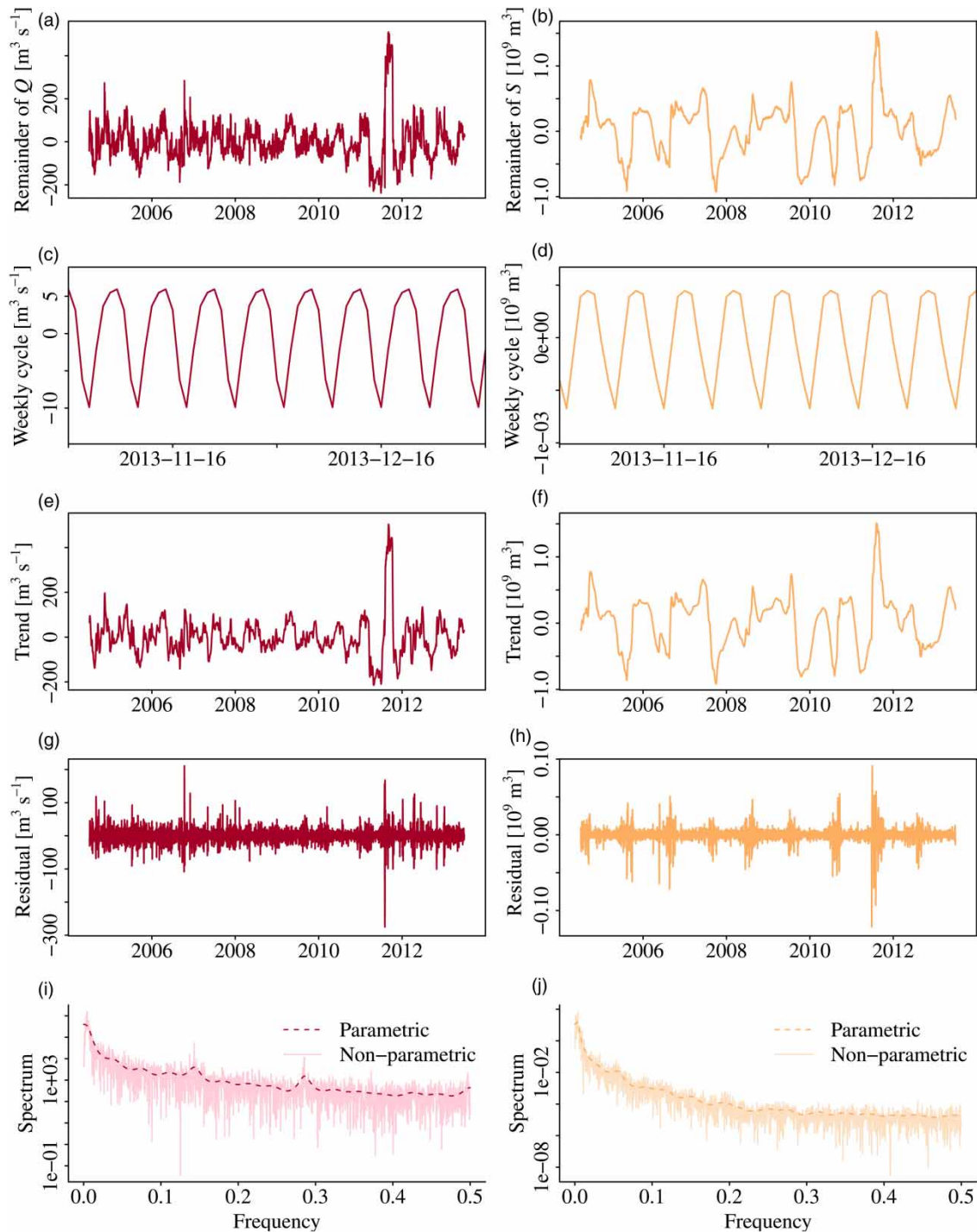
Interpreting the ACFs and CCFs of time series with strong (annual and weekly) cycles can be challenging. Therefore, the ACFs and CCFs were analysed for the original time series, the first decomposition residual (Residual<sub>1</sub>, eliminating long-term trends and annual cycles), and the second decomposition residual (Residual<sub>2</sub>, eliminating long-term trends, annual cycles, and weekly cycles). Interpreting the ACFs and CCFs of the residuals provided insights into the system dynamics.

### 3.2.1. Autocorrelation

The ACFs of the daily  $Q$ ,  $S$ , and  $I$  time series are displayed in Figure 5 and the ACFs of  $Q_B$  and  $Q_D$  in Figure S4.

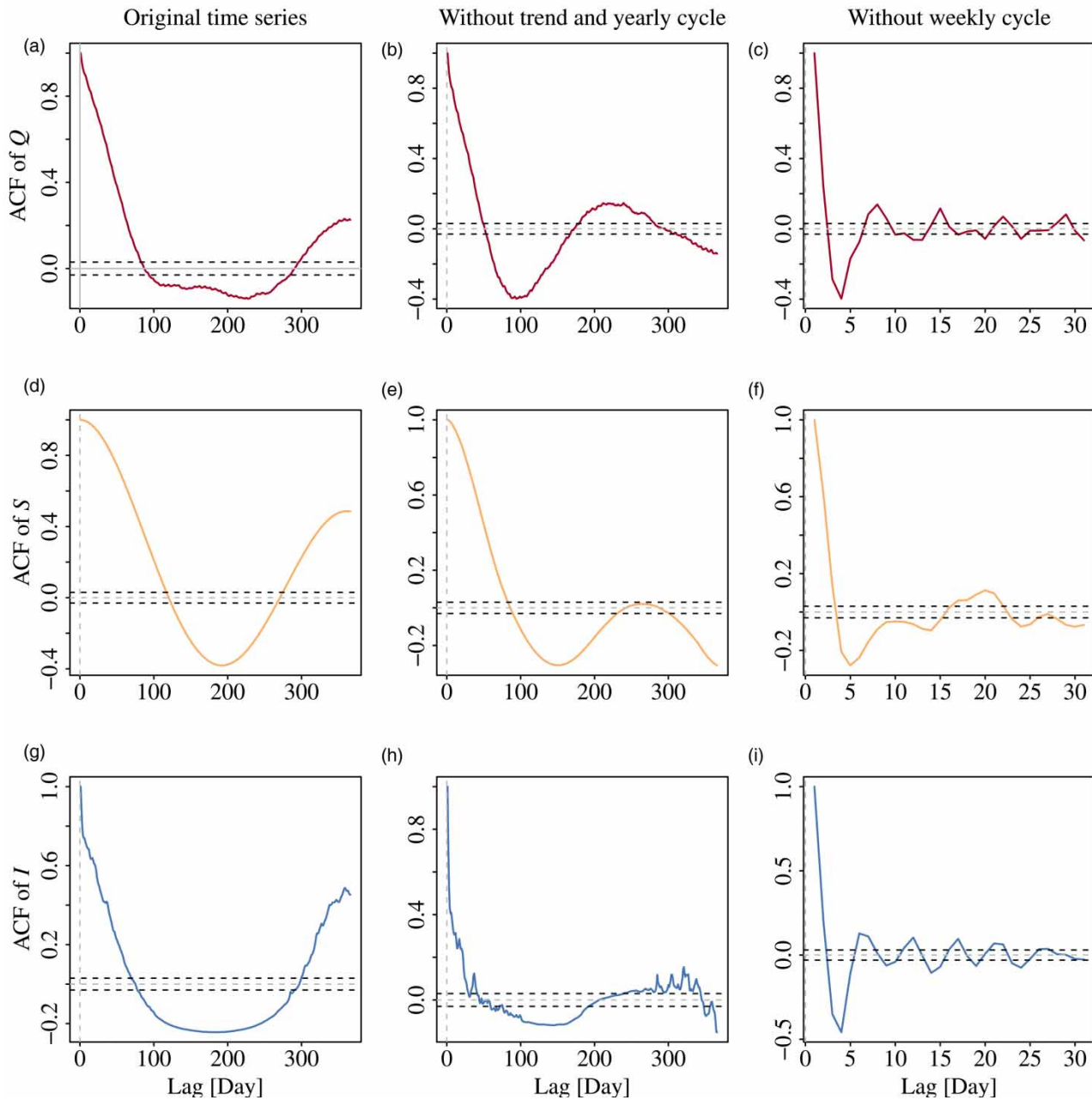
The ACFs of the original time series (left panels in Figures 5 and S4) show slow decays, implying that the variables gradually change and have long-term influences on the hydrological and operation systems. The decreasing correlation of  $Q$  reached the decorrelation threshold ( $1/e \sim 0.37$ ) after 51 lag days,  $S$  after 85 lag days,  $I$  after 39 lag days,  $Q_B$  after 64 lag days (not shown), and  $Q_D$  after 106 lag days (not shown). The shorter decorrelation times of  $I$  reflect the natural seasonality, while the longer decorrelation times of the other regulated variables also reflect the long-term memory effects of the reservoir operation. In other words, the seasonal inflow was stored and released gradually over many months.

The ACFs of the Residual<sub>1</sub> time series (middle panels in Figures 5 and S4) show considerably faster decays to reach the decorrelation threshold compared to the original time series, at 31 days for  $Q$ , 55 days for  $S$ , 7 days for  $I$ , 24 days for  $Q_B$ , and 29 days for  $Q_D$ . This implies the extent of the annual cycle's influence on the reservoir operation and the hydrological system. Although it is invisible in the figures,  $Q$  and  $Q_B$  also contain a 7-day periodicity, aligning with the SDF result in Figure 4(i).



**Figure 4** | Decomposition of the Sirikit reservoir outflow ( $Q$ ) and storage ( $S$ ), focusing on the weekly cycle. The initial time series (a and b) are the residuals of the decomposed original time series (shown in Figure 3(d) and (h)). The weekly component (c and d) focuses on a 2-month period, so that the periodicity is visible. The residual components (g and h) are the residuals of the time series after further excluding the weekly cycle. The SDF (i and j) shows the periodicity estimated with both non-parametric and parametric methods. Note that the y-axis of the SDF is presented on a logarithmic scale.

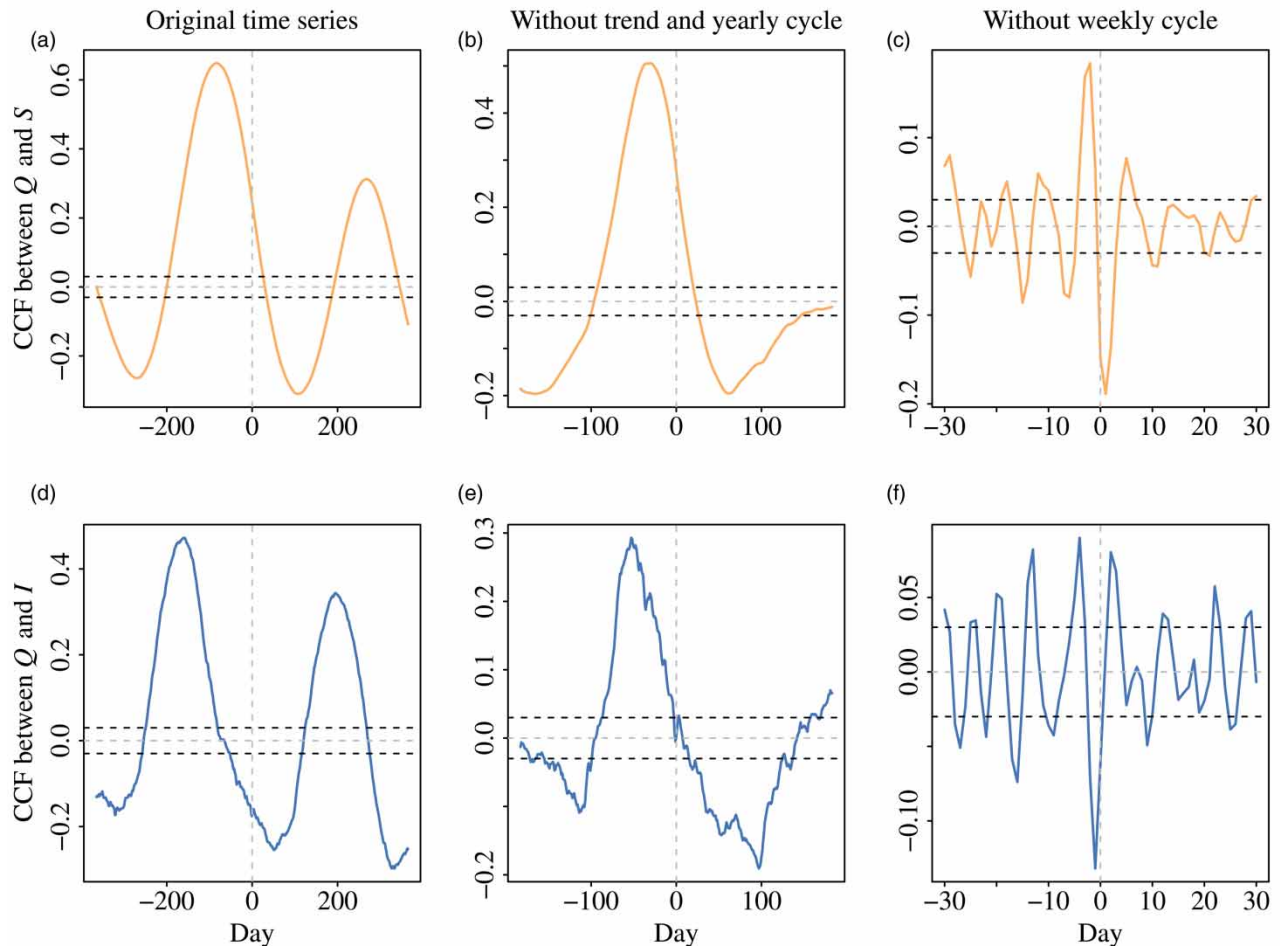
The ACFs of the  $\text{Residual}_2$  time series (right panels in Figures 5 and S4) show a very rapid decay, reaching the decorrelation threshold only after 1–2 days.  $I$  has no correlation left after around 3 months, while the other variables keep showing small oscillations with local positive correlations as the time lag increases.



**Figure 5** | ACF of the daily Sirikit reservoir outflow ( $Q$ ), storage ( $S$ ), and inflow ( $I$ ) data. The left panels show the ACFs of the original time series. The middle panels show the ACFs of the first residual time series (as shown in Figure 3(d), (h), and (m)), representing the data without the long-term trend and annual cycle. The right panels show the ACFs of the second residual time series (as shown in Figure 4(g) and (h)), representing the data without the long-term trend, annual cycle, and weekly cycle. Dashed black lines indicate where the correlation is significantly different from zero at the 95% level.

### 3.2.2. Cross-correlation

Similar to the ACFs, the CCFs were also analysed for (i) the original time series, (ii) Residual<sub>1</sub>, and (iii) Residual<sub>2</sub>. The CCFs of the original time series (left panels in Figures 6 and S5) show that  $S$  and  $I$  lead  $Q$  with a lag time of 83 days ( $\sim 3$  months) and 158 days ( $\sim 5$  months), respectively. Hence, one could expect the highest  $Q$  3 months after the highest  $S$  and 5 months after the highest  $I$ . This is in accordance with the annual cycles of the variables, where the highest  $I$  was found in September, the highest  $S$  in October to November, and the highest  $Q$  in January to February (Figure 3(m)–(o)).  $Q$  and  $Q_B$  are highly correlated at lag zero, indicating that both reservoirs are operated simultaneously in real time.



**Figure 6** | CCF between the daily Sirikit reservoir outflow ( $Q$ ) and storage ( $S$ ) (top panels) and between the outflow and inflow ( $I$ ) (bottom panels). The left panels show the CCFs of the original time series. The middle panels show the CCFs of the first residual time series. The right panels show the CCFs of the second residual time series. Dashed black lines indicate where the correlation is significant to the 95% level.

The CCFs of the Residual<sub>1</sub> time series (middle panels in Figures 6 and S5) show a similar correlation pattern, but with faster response times. The high Residual<sub>1</sub> of  $Q$  could be expected 35 days ( $\sim 1$  month) after the high Residual<sub>1</sub> of  $S$  and 54 days ( $\sim 2$  months) after the high Residual<sub>1</sub> of  $I$ . The Residual<sub>1</sub> of  $Q$  and  $Q_B$  are still highly correlated at lag zero. The Residual<sub>1</sub> of  $Q_D$  positively correlates with  $Q$ , with a lead time of 22 days, partly reflecting the travel time of  $Q$  to the downstream region.

The CCFs of the Residual<sub>2</sub> time series (right panels in Figures 6 and S5) show short response times. The high Residual<sub>2</sub> of  $Q$  tends to occur 2 days after the high Residual<sub>2</sub> of  $S$  and 1 day before the low Residual<sub>2</sub> of  $S$ . On the other hand, the high Residual<sub>2</sub> of  $Q$  was likely to occur 4 days after the low Residual<sub>2</sub> of  $I$  and 2 days before the high Residual<sub>2</sub> of  $I$ . The high Residual<sub>2</sub> of  $Q_B$  took place 1 day before the high Residual<sub>2</sub> of  $Q$ . The Residual<sub>2</sub> of  $Q$  was high on the day that the Residual<sub>2</sub> of  $Q_D$  was low and 4 days before the high Residual<sub>2</sub> of  $Q_D$ .

### 3.3. Selection of input variables

#### 3.3.1. Influence of different input variable combinations

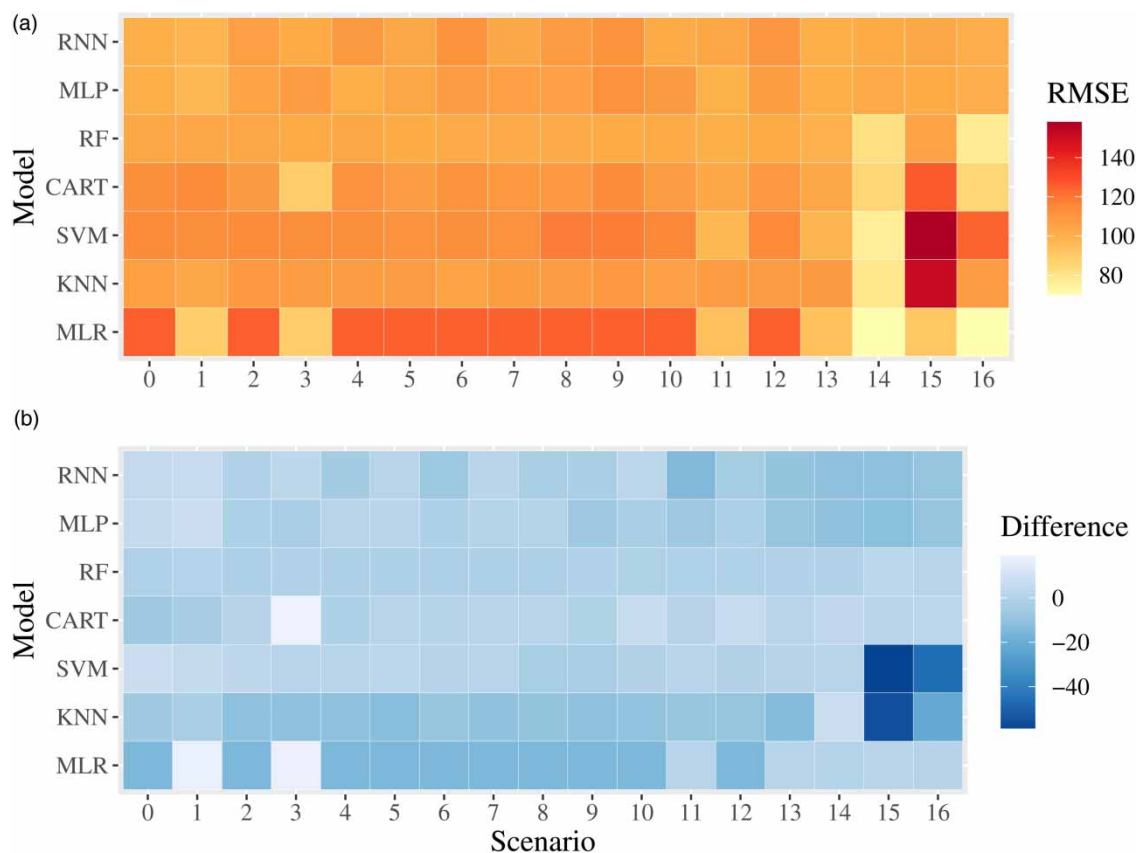
The analysis of the daily reservoir-related time series ( $S$ ,  $I$ ,  $Q_B$ , and  $Q_D$ ) and timing information ( $M$  and  $D$ ) shows their importance and potential as the input variables of the ML models. The CCFs (Figures 6 and S5) indicate that, to simulate  $Q$  on the current day ( $t$ ), the most important values of  $S$  are observed on days  $t-2$ ,  $t-1$ ,  $t$ , and  $t+1$ , while the most important values of  $I$  span from days  $t-4$  to  $t+2$ , particularly reflecting strong correlations in the Residual<sub>2</sub> time series. While past data are typically accessible through observations and simulations, caution is needed with current and future data. Future  $I$  values can be estimated from hydrological forecasts, but future  $S$  data, dependent on other reservoir water balance components (Equation

(S2)), are often unavailable. Therefore, only the past  $S$  data at  $t-2$  and  $t-1$  were considered in the input scenarios, along with past values of  $Q_B$  and  $Q_D$  from  $t-1$ .

With 17 input variable combinations (Table 1), performances of the seven ML models (MLR, KNN, SVM, CART, RF, MLP, and RNN) during the testing period (years 2012–2013) are illustrated in Figure 7(a). The results for the training period are supplied in Figure S6(a). The baseline scenario contains  $S_{t-1}$  and  $I_{t-1}$  as the inputs. Adding  $S_{t-2}$  (Scenarios 1 and 3) improved performance only for MLR and CART. Likewise, although adding  $I_{t-4}-I_{t+2}$  (Scenarios 2–10) improved the model performance, their effects were not remarkable, with no clear indication of the relative importance of timing of the inflow. Therefore, we selected  $S_{t-1}$  and  $I_{t-2}-I_{t+2}$  to include a comprehensive timing range of inflow data while maintaining computational efficiency. Further exploration of timing information ( $M$  and  $D$ ; Scenarios 11–13) showed that  $D$  slightly reduced the model performance, while  $M$  improved the model performance (Scenario 11). Adding  $Q_{B,t-1}$  (Scenario 14) significantly enhanced the performance of many models, except for MLP and RNN. In contrast, including  $Q_{D,t-1}$  (Scenario 15) negatively impacted several models (KNN, SVM, and CART), while models with both  $Q_{B,t-1}$  and  $Q_{D,t-1}$  (Scenario 16) exhibited no performance difference.

### 3.3.2. Influence of data scaling

Overall, running the ML models with the scaled (normalized) input data increased the resulting accuracy, as shown in Figure 7(b). Compared to their performance with the raw inputs in Figure 7(a), the MLR and KNN model results improved



**Figure 7** | Performance of the seven ML models on the Sirikit reservoir outflow simulations for the 17 input combination scenarios during the testing period (years 2012–2013), using RMSE as the criterion. (a) The RMSE of the model simulations with the raw input data. (b) The difference in RMSE between the model simulations with the raw input data and with the normalized input data, reflecting the effect of input data scaling on the model performance. Differences between the input combination scenarios are explained in Table 1 and the ML models in Table 2. (a) A lower RMSE value presented in a lighter colour indicates a better model performance. (b) A higher difference presented in a darker colour indicates an improvement in the model performance with the scaled input data. Lighter to white colours indicate that the model performance did not improve or even decrease. The models' performance in the training period is supplied in Figure S6.

the most in the scenarios that focus on  $S$  and  $I$  data (Scenarios 0–10). Meanwhile, the MLP and RNN model results improved the most in the scenarios that include timing information and other data ( $M$ ,  $D$ ,  $Q_B$ , and  $Q_D$ ; Scenarios 11–16), of which RNN improved the most in Scenario 11. The SVM, CART, and RF models did not show as much improvement as other models. The most distinct improvements were found in Scenarios 15 and 16 (where  $Q_D$  was included) for the KNN and SVM models. The difference in the RMSE results after the data scaling during the training period is supplied in Figure S6(b).

### 3.4. Performance comparison of different algorithms

Based on the analysis results of the input data scenarios and data scaling, we selected the two best-performing input datasets: Scenario 11 ( $S_{t-1}$ ,  $I_{t-2}$  to  $I_{t+2}$  and  $M$ ) and Scenario 16 ( $S_{t-1}$ ,  $I_{t-2}$  to  $I_{t+2}$ ,  $Q_{B,t-1}$ ,  $Q_{D,t-1}$ ,  $M$  and  $D$ ). The simulated  $Q$  by the seven ML models in these two scenarios during the testing period (years 2012–2013) are displayed in Figure 8. As can be seen, the KGE values are higher in Scenario 16 for most models, except for the RNN model, which shows no improvement. Interestingly, the MLR and RNN models had the best performance in Scenario 11, while MLR also outperformed other models in Scenario 16. The performance comparison for the training period is supplied in Figure S7.

## 4. DISCUSSION

### 4.1. Available data to reflect real-time reservoir operation

In this study, the potential input variables of the ML models for simulating  $Q$  were selected based on their availability, including the daily reservoir-related time series ( $S$ ,  $I$ ,  $Q_B$ , and  $Q_D$ ) and timing information ( $M$  and  $D$ ). While these variables were specifically investigated for the Sirikit reservoir, they are also general enough to apply to ML modelling for other reservoirs with similar operations and the same climatological setting.

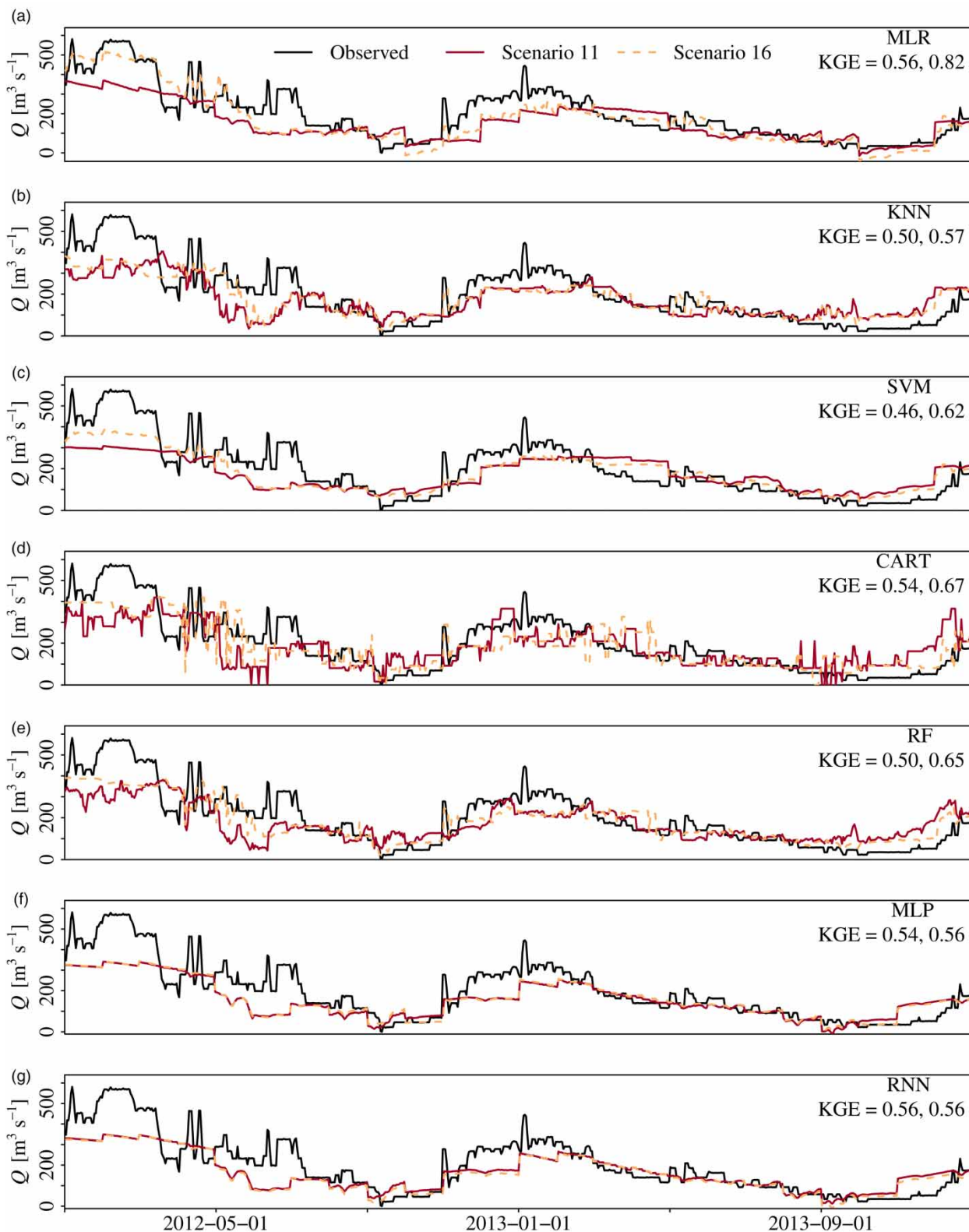
The aforementioned variables represent most components in the simplified reservoir water balance (Equation (S2)). However, the real-world reservoir water balance is more complex due to precipitation and evaporation over the reservoir surface. Previous studies have shown conflicting results regarding the influence of meteorological variables on ML model performance. Some suggested meteorological data are more significant than timing information (e.g., Zhang *et al.* 2019), while others argued they are negligible if captured by timing indicators (e.g., Yang *et al.* 2016). These conflicting findings reflect the differences in reservoir characteristics and operations. For the Sirikit reservoir, we deduced that timing information is more critical, as its operation is more driven by downstream water demands than weather. In addition, groundwater inflows and infiltration losses, though important in the real-world water balance (Fowe *et al.* 2015), are often excluded in ML modelling due to limited data availability.

The aforementioned variables cannot fully represent the real-time reservoir operation function (Equation (S1)) due to a lack of operational data. Important factors influencing decision-making, such as hydroelectricity generation and water allocation policies have proved to notably increase ML model performance (e.g., Yang *et al.* 2016; Zhang *et al.* 2019). Unfortunately, these data are often difficult to obtain without authority support or collaboration.

### 4.2. Importance of input variable selection

This study found that past and future values of  $S$  and  $I$  were closely correlated with the current  $Q$  value after eliminating the long-term trend and cycles (Figure 6). However, incorporating these variables with different time lags and lead times did not significantly enhance the ML model performance (Figure 7(a)). This aligns with the finding by Khatun *et al.* (2024) that an increase in time lag introduced redundant information and adversely reduced ML model performance for real-time stream-flow forecasting.

Although incorporating all input variables ( $S_{t-1}$ ,  $I_{t-2}$  to  $I_{t+2}$ ,  $Q_{B,t-1}$ ,  $Q_{D,t-1}$ ,  $M$ , and  $D$ ), yielded the best performance for most ML models (Scenario 16 in Figures 7(a) and 8), it is not ideal for real-time application. While  $Q$  displayed annual and weekly cycles (Figures 3 and 4), only including  $M$  notably improved the model accuracy. Similarly, although  $Q_B$  and  $Q_D$  showed short-term correlations with  $Q$  (Figure S5),  $Q_D$  negatively affected some ML models.  $Q_{B,t-1}$ , which strongly correlates with  $Q_{t-1}$ , added unnecessary complexity. These results align with Chen *et al.* (2018), who demonstrated that an excessive number of input variables increased noise and computational complexity, leading to overfitting. Similarly, Shen *et al.* (2022) found that certain correlated variables did not significantly reduce errors in a hybrid ML-process-based hydrological model. Moreover, using such a complex input set for real-time  $Q$  prediction would require real-time predictions of  $I$ ,  $Q_B$ , and  $Q_D$ , introducing additional input uncertainties, particularly for  $Q_B$  and  $Q_D$ , thus increasing the complexity and reducing reliability. Therefore, balancing model simplicity with accuracy is crucial for effective ML-based reservoir simulations.



**Figure 8** | Simulated daily outflows of the Sirikit reservoir by the seven ML models compared to the observations in the testing period (years 2012–2013). The ML models are described in Table 2. The simulation results of two selected and scaled input scenarios are compared: Scenario 11 (in solid red colour) and Scenario 16 (in dashed orange colour), which are the best input variable set. The details of each scenario can be found in Table 1. The KGE values are also indicated for Scenarios 11 and 16.

After careful consideration, we suggest that the most suitable input variable set for simulating  $Q$  comprises  $S_{t-1}$ ,  $I_{t-2}$  to  $I_{t+2}$ , and  $M$  (Scenario 11), striking a balance between data importance, availability, model performance, and computational demand. The  $Q$  simulation results were satisfactory across all ML models, showing minimal differences from Scenario 16, especially for the MLP and RNN models (Figure 8). This input variable set effectively captures key components in real-time reservoir operation function.  $M$  sufficiently accounts for downstream demand and rule curves, reducing input complexity. This confirmed the finding of Marton & Knoppová (2019) that hydrological and reservoir models focusing on key input variables, such as inflow, perform better under climate change uncertainties as they avoid unnecessary complexity while maintaining model flexibility.

As a next step, we suggest incorporating real-time operator decisions by using previous  $Q$  values as inputs, an approach shown to improve accuracy in several ML studies (e.g., Chen *et al.* 2018; Zhang *et al.* 2019). This approach enhances linear relationships between the inputs and output, but it is only practical when the  $Q$  input data come from real-time predictions rather than historical observations. A realistic application would involve using an ML model to predict real-time  $Q$  and then feeding the predicted  $Q$  result into the model as an input for the subsequent  $Q$  calculation.

This study utilized correlation-based IVS together with time series decomposition and stepwise variable combinations. Correlation analysis has been proved effective in ML-based hydrological modelling when time lags and lead times play a critical role in simulations, as supported by previous studies (Sushanth *et al.* 2023; Khatun *et al.* 2024). We acknowledge that other IVS methods, including model-free approaches such as mutual information and principal component analysis, and model-based approaches such as input omission, neural pathway strength, stepwise selection (forward, backward, or bidirectional), and recursive feature elimination (RFE), have been explored and suggested (Ssegane *et al.* 2012; Snieder *et al.* 2020; Gharib & Davies 2021; Reis *et al.* 2021). In this study, the results from RFE and built-in IVS methods in the MLR, CART, and RF algorithms aligned with the correlation-based results, providing similar rankings of variable importance, reinforcing the relevance of our selected inputs. Since it is evident that the suitability of input variables can vary across different ML algorithms (Figure 7(a)), we recommend future research to combine both correlation analysis and model-based IVS approach to gain insights into input variables and ensure appropriate input selection for ML-based hydrological predictions.

#### 4.3. Importance of data scaling

This study found that many ML algorithms are sensitive to data distributions and can achieve better performance with scaled input data (Figure 7(b)). For example, MLR uses a weighted sum of inputs, KNN uses distance measures based on input scales, and RNN initializes the input weights to small random values. This finding is consistent with Cabello-Solorzano *et al.* (2023), who showed that an appropriate scaling method can contribute significantly to improving the accuracy of many ML algorithms, especially for time series simulations.

An important concern when scaling the input data for ML models is data leakage. It occurs when an ML model is already aware of some unknown or future data in the training process, causing model overfitting with an overly optimistic training result but poor testing or validating results (Gharib & Davies 2021). To prevent data leakage, we first normalized the training dataset (years 2004–2011) and used the statistical values (minimum and maximum) to normalize the testing dataset (years 2012–2013). We strongly recommend this approach to obtain the realistic performance of the ML models.

When using scaled input data, careful attention must be given to the training and testing periods to minimize overfitting. If the training dataset lacks representation of future extreme events, the scaling factors (e.g., minimum and maximum values) from the training period may inadequately represent scaled extreme values in the testing data, leading to reduced model accuracy. For example, including the historical flood year of 2011 in the testing period instead of the training period could significantly reduce the model performance in this study. Therefore, as also demonstrated by Frame *et al.* (2022), we recommend that future research assess ML model performance through cross-validation, incorporating different data periods to ensure robust data selection and scaling, and overall model performance.

#### 4.4. Importance of algorithm selection

The model performance comparison (Figure 8) shows that simpler algorithms, such as MLR, outperformed more complex algorithms. This can be attributed to two main factors. First, the testing period excluded extreme events. In Scenario 11, the KGE value for MLR (0.52) was lower than those of KNN (0.90) and RF (0.95), and similar to MLP (0.54) and RNN (0.52) during the training period (Figure S7). If the testing period had included more extreme outflow values, MLR's performance would likely have decreased, while other models could maintain their performance levels (although CART exhibited

signs of overfitting with a KGE of 1). Second, the ML models were configured using their basic structures and parameterization for fair comparison. While the MLR model achieved optimal performance due to its linear nature, the other ML models can still be improved from enhanced architectures and hyperparameter tuning. This is especially the case for the RNN model, which was limited to a single hidden layer with 64 neurons, only 50 training iterations, and no recurrent memory. In line with the study by Sit *et al.* (2020), this study suggests that RNNs have a significant potential for enhancement in hydrological and reservoir modelling.

## 5. CONCLUSION AND OUTLOOK

This study explores the utilization of ML models, with insight into the available reservoir-related data as the inputs, for simulating daily operation and outflow of the Sirikit reservoir in Thailand in real time. Key findings are as follows:

- (1) The recent past and future values of the potential input variables, especially reservoir storage and inflow, were closely correlated with the current value of the reservoir outflow, indicating the short-term memory effect of real-time decision-making by the reservoir operators. However, including the input data at different time lags and lead times did not always significantly improve the model performance, possibly due to added complexity and noise.
- (2) While including all potential data as the inputs provided the best performance in most ML models, they added redundancy and complexity. Considering also data importance, data availability, and computational effort, the suggested input variable set for the Sirikit reservoir includes reservoir storage, inflow, and month of the year, excluding the Bhumibol reservoir outflow, downstream river discharge, and day of the week.
- (3) Consistent scaling and distribution of input data significantly enhance ML model performance, emphasizing the importance of proper data pre-processing in the model training process.

This study underscores the potential of advanced ML applications for multi-day forecasting of real-time reservoir operation and outflow, especially for reservoirs in Thailand and other monsoon-dominated regions. As the selection of appropriate input variables and their pre-processing are influenced by reservoir characteristics, study areas, and ML algorithms, individualized consideration is essential. Beyond case-by-case considerations, it is crucial for practitioners to conduct a comprehensive IVS and data pre-processing prior to setting up any operational reservoir modelling system. We recommend starting with a streamlined set of input variables – reservoir storage, inflow, and month of the year – to evaluate ML model performance while minimizing data complexity. Subsequently, an insightful analysis, combining both correlation-based and model-based approaches, should be conducted.

Ultimately, this study highlights the strengths and limitations of ML models in real-time reservoir modelling, specifically for the Sirikit reservoir. ML models offer significant advantages in terms of adaptability, ability to handle complex datasets, and quick processing. However, they can become overly complex and sensitive to noise and require large training datasets. Process-based models, in contrast, offer valuable physical insights and stability, especially when data availability is limited. Neither ML models nor process-based models should be seen as inherently superior. The future of reservoir operation modelling and management could lie in a hybrid approach. Future efforts should focus on integrating the capability of ML models for real-time updating with the interpretability and robustness of process-based models.

## ACKNOWLEDGEMENTS

The authors thank the Royal Irrigation Department of Thailand (RID) and the Electricity Generating Authority of Thailand (EGAT) for providing observational data.

## FUNDING

The first author received the Royal Thai Government Scholarship as the financial support for this study, which was conducted as part of the doctoral research project.

## DATA AVAILABILITY STATEMENT

All relevant data are included in the paper or its Supplementary Information.

## CONFLICT OF INTEREST

The authors declare there is no conflict.

## REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. & Kudlur, M. (2016) 'TensorFlow: A system for large-scale machine learning', *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. Savannah, GA: USENIX Association, pp. 265–283.
- Ahsan, M. M., Mahmud, M. P., Saha, P. K., Gupta, K. D. & Siddique, Z. (2021) *Effect of data scaling methods on machine learning algorithms and model performance*, *Technologies*, **9**, 52. doi: 10.3390/technologies9030052.
- Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2015) *Time Series Analysis: Forecasting and Control*. NJ: John Wiley & Sons.
- Breiman, L. (2001) *Random forests*, *Machine Learning*, **45**, 5–32. doi: 10.1023/A:1010933404324.
- Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984) *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group. doi: 10.1201/9781315139470.
- Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M. & Correia, L. (2023) 'The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis', *International conference on soft computing models in industrial and environmental applications*. Springer, pp. 344–353. doi: 10.1007/978-3-031-42536-3\_33.
- Chaves, P. & Chang, F.-J. (2008) *Intelligent reservoir operation system based on evolving artificial neural networks*, *Advances in Water Resources*, **31**, 926–936. doi: 10.1016/j.advwatres.2008.03.002.
- Chen, K., Guo, S., He, S., Xu, T., Zhong, Y. & Sun, S. (2018) *The value of hydrologic information in reservoir outflow decision-making*, *Water*, **10**, 1372. doi: 10.3390/w10101372.
- Chen, Y., Li, D., Zhao, Q. & Cai, X. (2022) *Developing a generic data-driven reservoir operation model*, *Advances in Water Resources*, **167**, 104274. doi: 10.1016/j.advwatres.2022.104274.
- Di Baldassarre, G., Wanders, N., AghaKouchak, A., Kuil, L., Rangelcroft, S., Veldkamp, T. I., Garcia, M., van Oel, P. R., Breinl, K. & Van Loon, A. F. (2018) *Water shortages worsened by reservoir effects*, *Nature Sustainability*, **1**, 617–622. doi: 10.1038/s41893-018-0159-0.
- Draper, A. J., Munévar, A., Arora, S. K., Reyes, E., Parker, N. L., Chung, F. I. & Peterson, L. E. (2004) *CalSim: generalized model for reservoir system analysis*, *Journal of Water Resources Planning and Management*, **130**, 480–489. doi: 10.1061/(ASCE)0733-9496(2004)130:6(480).
- Fowe, T., Karambiri, H., Paturel, J.-E., Poussin, J.-C. & Cecchi, P. (2015) *Water balance of small reservoirs in the Volta basin: A case study of Boura reservoir in Burkina Faso*, *Agricultural Water Management*, **152**, 99–109. doi: 10.1016/j.agwat.2015.01.006.
- Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shalev, G., Gilon, O., Qualls, L. M., Gupta, H. V. & Nearing, G. S. (2022) *Deep learning rainfall-runoff predictions of extreme events*, *Hydrology and Earth System Sciences*, **26**, 3377–3392. doi: 10.5194/hess-26-3377-2022.
- Galelli, S., Humphrey, G. B., Maier, H. R., Castelletti, A., Dandy, G. C. & Gibbs, M. S. (2014) *An evaluation framework for input variable selection algorithms for environmental data-driven models*, *Environmental Modelling & Software*, **62**, 33–51. doi: 10.1016/j.envsoft.2014.08.015.
- Gharib, A. & Davies, E. G. (2021) *A workflow to address pitfalls and challenges in applying machine learning models to hydrology*, *Advances in Water Resources*, **152**, 103920. doi: 10.1016/j.advwatres.2021.103920.
- Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S. & Khraisat, A. (2024) *Enhancing k-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications*, *Journal of Big Data*, **11**, 113. doi: 10.1186/s40537-024-00973-y.
- Hamed, K. H. & Rao, A. R. (1998) *A modified Mann-Kendall trend test for autocorrelated data*, *Journal of Hydrology*, **204**, 182–196. doi: 10.1016/S0022-1694(97)00125-X.
- Helsel, D., Hirsch, D., Ryberg, K., Archfield, S. & Gilroy, E. (2020) *Statistical Methods in Water Resources: U.S. Geological Survey Techniques and Methods*. U.S. Geological Survey. Reston, VA. doi: 10.3133/tm4a3.
- Jain, S. K. & Singh, V. P. (2003) *Water Resources Systems Planning and Management*. Amsterdam, the Netherlands: Elsevier Science B.V..
- Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S., Mujumdar, S., Afzal, S., Sharma Mittal, R. & Munigala, V. (2020) 'Overview and importance of data quality for machine learning tasks', In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. NY, pp. 3561–3562. doi: 10.1145/3394486.3406477.
- Jenkins, G. M. (1968) *Spectral Analysis and Its Applications*. San Francisco, CA: Holden-Day, Inc.
- Kendall, M. G. (1946) *The Advanced Theory of Statistics*, 2nd edn. London, United Kingdom: Charles Griffin and Co., Ltd.
- Ketkar, N. (2017) *Introduction to Keras*. In: *Deep Learning With Python: A Hands-On Introduction*. Berkeley, CA: ApressSEP, pp. 97–111. doi: 10.1007/978-1-4842-2766-4\_7.
- Khatun, A., Nisha, M., Chatterjee, S. & Sridhar, V. (2024) *A novel insight on input variable and time lag selection in daily streamflow forecasting using deep learning models*, *Environmental Modelling & Software*, **179**, 106126. doi: 10.1016/j.envsoft.2024.106126.
- Klipsch, J. D. & Evans, T. A. (2006) 'Reservoir operations modeling with HEC-ResSim', In: *Proceedings of the 3rd federal interagency hydrologic modeling conference*. NV.
- Kramer, O. (2013) *K-nearest neighbors*. In: (Kacprzyk, J. & Jain, L. C., eds) *Dimensionality reduction with unsupervised nearest neighbors*. J. Kacprzyk, Warsaw, Poland L. C. Jain, Adelaide, Australia Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 13–23. doi: 10.1007/978-3-642-38652-7\_2.
- Kumar, D. N. & Reddy, M. J. (2007) *Multipurpose reservoir operation using particle swarm optimization*, *Journal of Water Resources Planning and Management*, *ASCE*, **133**, 192–201. doi: 10.1061/(ASCE)0733-9496(2007)133:3(192).
- Lange, H. & Sippel, S. (2020) *Machine Learning Applications in Hydrology*. In: Levia, D. F., Carlyle-Moses, D. E., Iida, S., Michalzik, B., Nanko, K., Tischer, A. (eds) *Forest-Water Interactions*. Ecological Studies, vol 240. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-26086-6\_10.

- Loh, W.-Y. (2014) Fifty years of classification and regression trees, *International Statistical Review*, **82**, 329–348. doi: 10.1111/insr.12016.
- Lund, J. R. & Guzman, J. (1999) Derived operating rules for reservoirs in series or in parallel, *Journal of Water Resources Planning and Management*, **125**, 143–153. doi: 10.1061/(ASCE)0733-9496(1999)125:3(143).
- Marton, D. & Knoppová, K. (2019) Developing hydrological and reservoir models under deep uncertainty of climate change: robustness of water supply reservoir, *Water Supply*, **19**, 2222–2230. doi: 10.2166/ws.2019.102.
- Mei, X., Dai, Z., Darby, S. E., Gao, S., Wang, J. & Jiang, W. (2018) Modulation of extreme flood levels by impoundment significantly offset by floodplain loss downstream of the Three Gorges Dam, *Geophysical Research Letters*, **45**, 3147–3155. doi: 10.1002/2017GL076935.
- Moreido, V., Gartsman, B., Solomatine, D. P. & Suchilina, Z. (2021) How well can machine learning models perform without hydrologists? Application of rational feature selection to improve hydrological forecasting, *Water*, **13**, 1696. doi: 10.3390/w13121696.
- Naganna, S. R. & Deka, P. C. (2014) Support vector machine applications in the field of hydrology: A review, *Applied Soft Computing*, **19**, 372–386. doi: 10.1016/j.asoc.2014.02.002.
- Niu, W.-J., Feng, Z.-K., Feng, B.-F., Min, Y.-W., Cheng, C.-T. & Zhou, J.-Z. (2019) Comparison of multiple linear regression, artificial neural network, extreme learning machine, and support vector machine in deriving operation rule of hydropower reservoir, *Water*, **11**, 88. doi: 10.3390/w11010088.
- Oliveira, R. & Loucks, D. P. (1997) Operating rules for multireservoir systems, *Water Resources Research*, **33**, 839–852. doi: 10.1029/96WR03745.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011) Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, **12**, 2825–2830. doi: 10.48550/arXiv.1201.0490.
- Qie, G., Zhang, Z., Getahun, E. & Mamer, E. A. (2022) Comparison of machine learning models performance on simulating reservoir outflow: A case study of two reservoirs in Illinois, USA, *Journal of the American Water Resources Association*, **12**, JAWR-21-0019-P. doi: 10.1111/1752-1688.13040.
- R Core Team (2013) *R: A Language and Environment for Statistical Computing*. Austria: R Foundation for Statistical Computing Vienna. ISBN 3-900051-07-0.
- Reis, G. B., da Silva, D. D., Fernandes Filho, E. I., Moreira, M. C., Veloso, G. V., de Souza Fraga, M. & Pinheiro, S. A. R. (2021) Effect of environmental covariable selection in the hydrological modeling using machine learning models to predict daily streamflow, *Journal of Environmental Management*, **290**, 112625. doi: 10.1016/j.jenvman.2021.112625.
- Şen, Z. (2021) Reservoirs for water supply under climate change impact—A review, *Water Resources Management*, **35**, 3827–3843. doi: 10.1007/s11269-021-02925-0.
- Shen, Y., Ruijsch, J., Lu, M., Sutanudjaja, E. H. & Karssenbergh, D. (2022) Random forests-based error-correction of streamflow from a large-scale hydrological model: Using model state variables to estimate error terms, *Computers & Geosciences*, **159**, 105019. doi: 10.1016/j.cageo.2021.105019.
- Shumway, R. H. & Stoffer, D. S. (2000) *Spectral analysis and filtering*. In: *Time Series Analysis and Its Applications*. George Casella, Stephen Fienberg, and Ingram Olkin Cham, Switzerland: Springer, pp. 213–300. doi: 10.1007/978-1-4757-3261-0\_3.
- Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y. & Demir, I. (2020) A comprehensive review of deep learning applications in hydrology and water resources, *Water Science and Technology*, **82**, 2635–2670. doi: 10.2166/wst.2020.369.
- Snieder, E., Shakir, R. & Khan, U. (2020) A comprehensive comparison of four input variable selection methods for artificial neural network flow forecasting models, *Journal of Hydrology*, **583**, 124299. doi: 10.1016/j.jhydrol.2019.124299.
- Ssegane, H., Tollner, E., Mohamoud, Y., Rasmussen, T. & Dowd, J. (2012) Advances in variable selection methods I: Causal selection methods versus stepwise regression and principal component analysis on data of known and unknown functional relationships, *Journal of Hydrology*, **438**, 16–25. doi: 10.1016/j.jhydrol.2012.01.008.
- Sushanth, K., Mishra, A., Mukhopadhyay, P. & Singh, R. (2023) Real-time streamflow forecasting in a reservoir-regulated river basin using explainable machine learning and conceptual reservoir module, *Science of the Total Environment*, **861**, 160680. doi: 10.1016/j.scitotenv.2022.160680.
- Tebakari, T., Yoshitani, J. & Suvanpimol, P. (2012) Impact of large-scale reservoir operation on flow regime in the Chao Phraya River basin, Thailand, *Hydrological Processes*, **26**, 2411–2420. doi: 10.1002/hyp.9345.
- Tyralis, H., Papacharalampous, G. & Langousis, A. (2019) A brief review of random forests for water scientists and practitioners and their recent history in water resources, *Water*, **11**, 910. doi: 10.3390/w11050910.
- Van Verseveld, W. J., Weerts, A. H., Visser, M., Buitink, J., Imhoff, R. O., Boisgontier, H., Bouaziz, L., Eilander, D., Hegnauer, M., Ten Velden, C. & Russell, B. (2024) Wflow\_sbm v0.7.3, a spatially distributed hydrological model: from global data to local applications, *Geoscientific Model Development*, **17**, 3199–3234. doi: 10.5194/gmd-17-3199-2024.
- Vapnik, V. (2000) *The Nature of Statistical Learning Theory*. NY: Springer Science & Business Media. doi: 10.1007/978-1-4757-3264-1\_1.
- Wannasin, C., Brauer, C. C., Uijlenhoet, R., Van Verseveld, W. J. & Weerts, A. H. (2021a) Daily flow simulation in Thailand Part II: Unraveling effects of reservoir operation, *Journal of Hydrology: Regional Studies*, **34**, 100792. doi: 10.1016/j.ejrh.2021.100792.
- Wannasin, C., Brauer, C. C., Uijlenhoet, R., van Verseveld, W. J. & Weerts, A. H. (2021b) Daily flow simulation in Thailand Part I: Testing a distributed hydrological model with seamless parameter maps based on global data, *Journal of Hydrology: Regional Studies*, **34**, 100794. doi: 10.1016/j.ejrh.2021.100794.

- Xu, T. & Liang, F. (2021) [Machine learning for hydrologic sciences: An introductory overview](#), *Wiley Interdisciplinary Reviews: Water*, **8**, e1533. doi: 10.1002/wat2.1533.
- Yang, T., Gao, X., Sorooshian, S. & Li, X. (2016) [Simulating California reservoir operation using the classification and regression-tree algorithm combined with a shuffled cross-validation scheme](#), *Water Resources Research*, **52**, 1626–1651. doi: 10.1002/2015WR017394.
- Yang, S., Yang, D., Chen, J. & Zhao, B. (2019) [Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model](#), *Journal of Hydrology*, **579**, 124229. doi: 10.1016/j.jhydrol.2019.124229.
- Yang, T., Zhang, L., Kim, T., Hong, Y., Zhang, D. & Peng, Q. (2021) [A large-scale comparison of artificial intelligence and data mining \(AI&DM\) techniques in simulating reservoir releases over the Upper Colorado Region](#), *Journal of Hydrology*, **602**, 126723. doi: 10.1016/j.jhydrol.2021.126723.
- Yassin, F., Razavi, S., Elshamy, M., Davison, B., Sapriza-Azuri, G. & Wheeler, H. (2019) [Representation and improved parameterization of reservoir operation in hydrological and land-surface models](#), *Hydrology and Earth System Sciences*, **23**, 3735–3764. doi: 10.5194/hess-23-3735-2019.
- Yates, D., Sieber, J., Purkey, D. & Huber-Lee, A. (2005) [WEAP21—A demand-, priority-, and preference-driven water planning model: Part 1: model characteristics](#), *Water International*, **30**, 487–500. doi: 10.1080/02508060508691893.
- Zarei, M., Bozorg-Haddad, O., Baghban, S., Delpasand, M., Goharian, E. & Loáiciga, H. A. (2021) [Machine-learning algorithms for forecast-informed reservoir operation \(FIRO\) to reduce flood damages](#), *Scientific Reports*, **11**, 1–21. doi: 10.1038/s41598-021-03699-6.
- Zarfl, C., Lumsdon, A. E., Berlekamp, J., Tydecks, L. & Tockner, K. (2015) [A global boom in hydropower dam construction](#), *Aquatic Sciences*, **77**, 161–170. doi: 10.1007/s00027-014-0377-0.
- Zhang, Z., Zhang, Q. & Singh, V. P. (2018) [Univariate streamflow forecasting using commonly used data-driven models: Literature review and case study](#), *Hydrological Sciences Journal*, **63**, 1091–1111. doi: 10.1080/02626667.2018.1469756.
- Zhang, D., Peng, Q., Lin, J., Wang, D., Liu, X. & Zhuang, J. (2019) [Simulating reservoir operation using a recurrent neural network algorithm](#), *Water*, **11**, 865. doi: 10.3390/w11040865.

First received 11 May 2024; accepted in revised form 11 November 2024. Available online 22 November 2024