# Flood control of reservoir systems

## Learning-based explicit and switched model predictive control approaches

Koo, Ja-Ho; Moradvandi, Ali; Abraham, Edo; Jonoski, Andreja; Solomatine, Dmitri P.

**DOI**
10.1371/journal.pwat.0000361

**Publication date**
2025

**Document Version**
Final published version

**Published in**
PLOS Water

**Citation (APA)**
Koo, J.-H., Moradvandi, A., Abraham, E., Jonoski, A., & Solomatine, D. P. (2025). Flood control of reservoir systems: Learning-based explicit and switched model predictive control approaches. *PLOS Water*, *4*(5), Article e0000361. https://doi.org/10.1371/journal.pwat.0000361

PLOS WATER

RESEARCH ARTICLE

# Flood control of reservoir systems: Learning-based explicit and switched model predictive control approaches

**Ja-Ho Koo**[1,2,3]*, **Ali Moradvandi**[1,4], **Edo Abraham**[1], **Andreja Jonoski**[2], **Dimitri P. Solomatine**[1,2,5]

**1** Department of Water Management, Delft University of Technology, Delft, The Netherlands, **2** Department of Hydroinformatics and Socio-Technical Innovation, IHE Delft, Delft, The Netherlands, **3** Korea Water Resource Public Corporation, Daejeon, Republic of Korea, **4** Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, **5** Department of River Basins Hydrology, Water Problems Institute of RAS, Moscow, Russia

* J.H.Koo@tudelft.nl



OPEN ACCESS

**Citation:** Koo J-H, Moradvandi A, Abraham E, Jonoski A, Solomatine DP (2025) Flood control of reservoir systems: Learning-based explicit and switched model predictive control approaches. PLOS Water 4(5): e0000361. https://doi.org/10.1371/journal.pwat.0000361

**Editor:** Bimlesh Kumar, Indian Institute of Technology Guwahati, INDIA

**Received:** January 16, 2025

**Accepted:** March 21, 2025

**Published:** May 9, 2025

**Peer Review History:** PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pwat.0000361



**Data availability statement:** The hydrological and operational data were acquired from Korea

## Abstract

Effective reservoir flood control demands real-time decision-making that balances multiple objectives. However, traditional optimization approaches are often too computationally intensive and become intractable when considering dynamically changing preferences of operators, modelled as weights of different objectives. This study aims to develop tractable real-time flood control strategies that maintain performance while reducing computational complexity. We propose two data-driven approaches based on Model Predictive Control (MPC): (1) an explicit MPC using deep neural networks to directly determine optimal outflow schedules, and (2) a switched MPC that produces optimal weights of objectives based on hydrological conditions. Both methods leverage offline learning from an online Parameterized Dynamic MPC framework incorporating state-dependent weights. We tested these approaches on South Korea's Daecheong multipurpose reservoir using historical flood events with various patterns. The explicit MPC demonstrated reliable performance under conditions similar to its training data. However, it showed frequent changes in outflow schedules and constraint violations for scenarios outside training data. In contrast, the switched MPC maintained robustness across all test scenarios due to a linear optimization process in a receding horizon manner, though with slightly reduced performance compared to the explicit MPC under scenarios inside the range of training data. Most significantly, both approaches reduced computation time from approximately 10 minutes to less than one second, making real-time implementation feasible. This dramatic improvement enables prompt decision-making during rapidly evolving flood events while maintaining near-optimal control performance.

Water Resources Public Corporation's website (http://kwater.or.kr). The data and code of this research are available at https://doi.org/10.4121/b6dd9d97-118d-406e-867d-b821fb6d08d4 under the CC-BY-4.0 license.

**Funding:** The authors received no specific funding for this work.

**Competing interests:** The authors have declared that no competing interests exist.

## Introduction

Floods are among the most destructive natural disasters, resulting in substantial economic and environmental damage across affected communities. Therefore, flood control is one of the main purposes of most reservoirs. Reservoir flood control, which significantly impacts watershed management, has attracted research attention for decades [1,2]. However, rapidly changing hydrological conditions, highly uncertain forecasts, and changes in decision makers' preferences [3,4] make reservoir flood control challenging. Many researchers have suggested optimization-based control methods because flood control is a control problem, as its name implies [5]. Various optimization-based methodologies have been proposed to address challenges of real-time optimal flood control, particularly subject to uncertainty, either in rainfall or inflow forecasts [6–8]. The complexity of reservoir flood control arises from the system's nonlinearity, which stems from intricate relationships among different objectives that operators want to achieve, including minimizing changing outflow schedules, minimising peak outflow, minimising Reservoir Water Level (RWL), and minimising opening and closing of spillway gates [9].

The flood control problem can be formulated as a multi-objective problem involving outflow schedules, RWL, and other factors. Various approaches have been proposed to effectively solve this multi-objective optimization problem of reservoir flood control. In classical Model Predictive Control (MPC) approaches, the relative importance of objectives is assumed to be constant throughout the flood event [7,10]. Therefore, optimal weights, as significant performance factors, should be determined properly and in advance. However, the relative importance of objectives can vary with the hydrological conditions in practice. Under this dynamic preference scenario, reflecting the operators' preferences in the weights of the objectives [9,11], the optimal outflow sequence can be chosen from the Pareto front [12]. This approach is flexible in the decision-making process, either by operators or through methods such as multi-criteria decision-making at each time step. The performance of this method, however, depends on the effective derivation of the Pareto fronts in real-time. Nonetheless, generating a Pareto front for the real-time application can be intractable when a nonlinear and nonconvex multi-objective problem has to be solved. Because the search spaces for exploration can grow exponentially [13,14], getting optimality guarantees for solutions can also be difficult, and it can take longer than the available decision window during flood events.

In our previous work in [9], a Parameterized Dynamic Model Predictive Control (PD-MPC) framework to derive optimal outflow schedules in reservoir flood control problems has been investigated. The PD-MPC formulation explicitly considers the dynamic characteristic of the relative importance of objectives depending on the situation via time-varying relative weights for the different objectives. In this approach, we separated the weight optimization and control action optimization in an alternating fashion until convergence. A Genetic Algorithm (GA) was employed to optimize objective weights and system parameters at the same time using the nonlinear system model and objectives, while the linear MPC problem was solved to optimality for a given weight set. The approach utilizes a single indicator, i.e., the absolute performance evaluator, to obtain optimal weights and parameters, along with optimal control inputs, at each time step. To the best of our knowledge, it was the first attempt to decide weights at each time step depending on hydrological conditions in real-time MPC-based reservoir flood control. Nonetheless, the approach still suffers from the inefficiency of online implementation due to the employed gradient-free algorithm requiring considerable time to achieve sufficient convergence. This can be limiting for the PD-MPC approach in real-time control.

In recent years, data-driven models like Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL) have emerged as effective tools to build efficient meta-(surrogate) models and thus reduce online computational complexity [15–17]. These models can be trained offline to map the nonlinear relationships between inputs and desired outputs [18,19], enabling efficient online implementation in various fields. For example, an Artificial Neural Network (ANN) was trained offline using the input and output data of a nonlinear MPC controller for a power converter and a robot, allowing the ANN to be implemented on hardware as an efficient explicit MPC that directly maps states to optimal control inputs [20,21]. In [22,23], hybrid RL-MPC approaches are proposed where the original MPC non-linear program is truncated to be one step ahead, with cost-to-go functions of the MPC replaced with the value function of the following state from RL. In [24], Multilayer Perceptron (MLP) policies are learned offline from MPC to automatically choose hard-to-optimize decision parameters, and so to parameterize a linear MPC to be solved online.

In real-time reservoir operations, these data-driven approaches are particularly valuable because they can address a practical operational challenge, i.e., computational time under a multi-objective optimization problem. Despite many studies in various fields, it is difficult to find a study on adopting data-driven models for an optimization problem with many objectives and dynamically changing weights of objectives. Due to the complex relationship among objectives, dynamically changing weights can make it difficult to learn the relationship between states and control inputs or optimal weights in a data-driven model. The conventional approach, which collects training data directly from simulation environments and trains an ANN or RL model, is not suitable for this case. This is because, for example, there can be many weight sets which can produce an optimal control input.

In this work, we leverage data-driven approaches to avoid difficulties in dealing with changing weights and solving a highly nonlinear problem online. From a control point of view, this article investigates explicit MPC and switched linear MPC approaches using data-driven models trained on the outputs of PD-MPC. The switched MPC framework allows the MPC system models to switch based on system states and predefined switching rules [25], requiring prior knowledge of these rules [26]. This allows simpler linear MPC problems to be solved online by selecting the appropriate model based on the initial state and the switching rules as a function of the states. In contrast, explicit MPC approaches compute control actions offline for all states by solving an optimal control problem for each state space partition [27], establishing either a mapping table or piecewise affine functions to relate states and control actions.

For our reservoir optimal flood control problem, we develop explicit MPC and switched MPC approaches from the PD-MPC results computed under a large set of inflow scenarios. A DNN model is trained offline to directly map states to control inputs as an explicit MPC controller. Additionally, a data-driven model is used to learn a mapping from system states to optimal parameters (i.e., state-dependent switching rule for optimal objective weights and other system parameters to use in a linear MPC online); this latter approach can be formulated as a switched linear MPC problem.

The novel contributions of this article are: i) we can obtain a consistent optimal weight set for a specific state by adopting the $\ell_1$-norm, ii) we present approaches for adopting explicit and switched MPC approaches using data-driven models, and iii) we demonstrate the advantages and disadvantages of switched and explicit MPC approaches for a multi-objective optimization problem with dynamically changing weights. To the best of the authors' knowledge, this is the first application and comparison of learning-based explicit and switched MPC for reservoir management.

The article is organized as follows. In Sect 1, the PD-MPC framework and the study area are discussed. The control objectives and system constraints used in this research are also formulated. Sect 2 presents the methodology and experimental setup. Focusing on approaches for data acquisition using PD-MPC, we describe the proposed data-driven surrogate models for explicit MPC and switched MPC. Sect 3 presents results, discussion, assessing the efficiency and characteristics of the surrogate models based on the two considered approaches. Finally, conclusions are drawn.

## 1 Problem formulation

### 1.1 Parameterized dynamic model predictive control

Model Predictive Control (MPC) utilizes a system model to predict states of a control system and produces a control input vector which optimizes objectives while explicitly considering constraints over a prediction horizon [28,29]. By implementing the MPC process in a receding horizon manner, the first control input in a sequence of optimal control inputs is implemented, and then the control inputs are updated based on the new system state, which is generally observed, as described in Fig 1.

The flood control problem can be formulated as a Parameterized Dynamic Model Predictive Control (PD-MPC) problem [9], where the dynamic preferences of operators are represented by changing the relative importance of objectives [9,31]. As depicted in Fig 2, the PD-MPC problem solves a multi-objective optimal control problem at each time step, where the weights are also co-optimized to reflect these changing preferences of the operators. The relative importance of the objectives can depend on the state of the system and projected inflows. For example, although frequent changes in schedules may not be desired from one time step to another, they become far less important if the potential for flooding increases due to a high water level driven by *inaction*.

The diagram in Fig 2 depicts how the PD-MPC approach solves the optimal control problem at the $k^{th}$ step of a receding horizon implementation:

$$\{\mathbf{z}^{k*}, \mathbf{u}^{k*}\} = \arg \min_{\mathbf{z}^k, \mathbf{u}^k, \mathbf{x}^k} E(\mathbf{z}^k, \mathbf{u}^k, \mathbf{x}^k | \mathbf{x}_0^k), \tag{1}$$

by alternatingly solving the two sets of problems:

$$\{\mathbf{z}^{k*}\} = \arg \min_{\mathbf{z}} \mathbf{E}(\mathbf{z} | \mathbf{x}^k, \mathbf{u}^k), \text{ and} \tag{2a}$$

$$\{\mathbf{u}^{k*} | \mathbf{z}^k\} = \arg \min_{\mathbf{x}^k, \mathbf{u}^k} \mathbf{J}(\mathbf{x}^k, \mathbf{u}^k | \mathbf{x}_0^k, \mathbf{z}^k), \tag{2b}$$

respectively, where $\mathbf{x}_0^k$ is the initial state at time $k$, $\mathbf{J}(\cdot)$ denotes the multi-objectives of a linear MPC and $\mathbf{u}$ expresses control inputs for a given weight set, $\mathbf{z}$.

What we call the absolute performance evaluator, $E(\cdot)$, is used to improve the optimal weight set, $\mathbf{z}^{k*}$ by solving a nonlinear simulation-based heuristic optimization, where the corresponding control actions $\mathbf{u}^{k*}|z$ are generated by the linear MPC and the evaluation is done with a nonlinear system model and objectives. Here, this evaluator assesses additional nonlinear performance criteria that represent the operators' selection criterion but are not included in $\mathbf{J}(\cdot)$. Using this method, we can decouple finding an optimal weight set and the control action set, which is represented in (1), thereby reducing the complexity of the co-design problem. In other words, in each time step, solving (2b) provides control actions associated with the given weight set, and by solving (2a), we are able to improve the optimal weight set, and

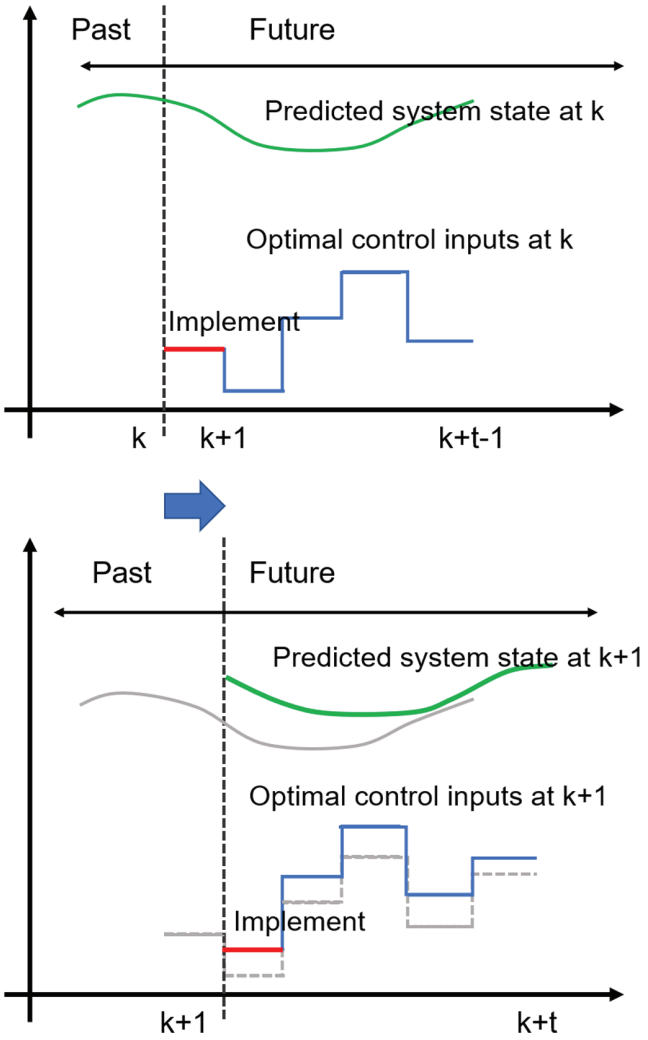



**Fig 1. The concept diagram of receding horizon MPC, in which the first one among the optimal control inputs generated at time step *k* is implemented.** Then, the whole sequence of the control inputs is regenerated at $k+1$, reflecting updated states. (Modified from [30]).

https://doi.org/10.1371/journal.pwat.0000361.g001

the corresponding control action set using the evaluators expressed by (2a). The evaluator can be solved using heuristic optimization methods such as Bayesian Optimization (BO) and GA.

The key aspect of this approach is the dynamic adjustment of the relative importance (weights $\mathbf{z}^*$) of objective functions based on hydrological conditions, allowing reservoir operators to respond more effectively to changing flood situations. For example, if the reservoir water level approaches a critical threshold, the weight for the objective to maintain RWL in the safe range increases.

In our previous research [9], the GA was applied, which evaluates and explores a set of potential solutions (the population) over each iteration until it meets convergence or time criteria. Another popular approach is BO algorithms, such as the Tree-structured Parzen Estimator (TPE) algorithm [32,33]. This algorithm is considered efficient because the TPE algorithm explores solutions based on a probabilistic model and is easy to parallelize [34].

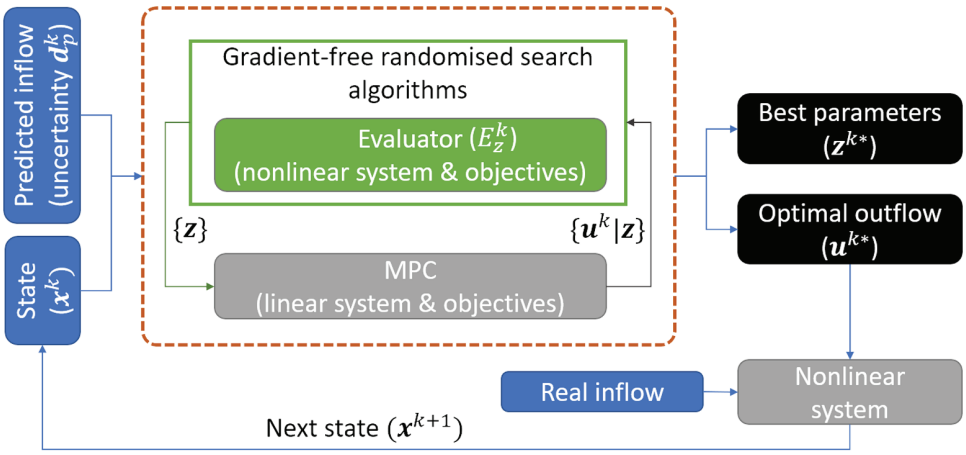



**Fig 2. The schematic diagram of PD-MPC [9].**

https://doi.org/10.1371/journal.pwat.0000361.g002

In this manuscript, we employ the TPE algorithm, a BO algorithm that makes use of tree-structured search spaces. At each time step, the TPE algorithm derives the optimal parameter vector by learning a surrogate probabilistic model for parameters and their associated objective values, i.e., $\mathbf{z}$ and $E$ online. The TPE algorithm utilizes a Kernel Density Estimator, especially a Parzen estimator, as the surrogate model. It employs the Expected Improvement (EI) and weighted random sampling as the acquisition method to explore promising regions of parameters $z$ and update the surrogate model. The exploration terminates when either the specified number of searches is reached, or the evaluation index falls into a predefined threshold. Thereafter, the best parameters and their associated evaluation value are returned. The corresponding control inputs, $\mathbf{u}^{k*}$, can then be computed. Therefore, the TPE algorithm is utilized at each time step in order to determine $\mathbf{z}^*$, minimizing the return (penalty) value from the absolute performance evaluator, $E$.

## 1.2 Reservoir flood control

A total of 13 linear and nonlinear objectives are suggested to describe the practical reservoir flood control problem, including minimizing peak spillway outflow, gate operations, and changes in outflow schedules [9]. Our approach here aims to approximate this rather complex PD-MPC problem derived from these considered objectives. To do so, the switched MPC and explicit MPC approaches are taken into account. To simplify the problem, three objectives for linear MPC and six nonlinear objectives to formulate the evaluator are then considered. This simplification is introduced because it may be difficult to show the performance of our approaches due to the highly complicated relationships among the objectives, when considering all 13 of them. Moreover, it would be reasonable to focus on objectives only directly relating to flood control in practice, as other objectives, such as minimizing environmental effects and securing water for demands, naturally receive lower priorities during flood events.

The three objectives for linear MPC are related to RWL, peak outflow, and outflow schedule changes. Reservoir operators should maintain RWL within a target range not only to ensure the dam's safety but also to secure enough water to satisfy demands. Therefore, three target water levels are defined: (i) The lower target level, to secure water, so it constrains RWL from decreasing a certain level; (ii) The upper level, to secure the flood control capacity between this level and the FWL; and (iii) the highest level for dam safety, considering

uncertainty. Furthermore, to reduce downstream flood risk, minimising peak outflow is also an essential factor, as the high outflows directly affect downstream water levels. Meanwhile, because each outflow schedule is shared with other flood control organizations, the previously shared outflow schedule should be maintained consistently with the minimum number of changes possible. Moreover, limiting the frequency of operations of spillway gates is advantageous to prevent wear and malfunction. This can be achieved implicitly by minimizing changes in consecutive outflow schedules. Additionally, a soft constraint is added to constrain opening spillway gates before the total outflow exceeds the turbine capacity.

Therefore, the overall objective function can be written as follows:

$$\min_{\mathbf{u}} J^k = \min_{\mathbf{u}} z_1 J_1 + z_2 J_2 + z_3 J_3 + z_4 J_4, \tag{3}$$

in which

$$J_1 := \max_{t \in k,\ldots,k+N-1} O_{spill,t} \times N \times \alpha_1^{MAVE}, J_2 := \sum_{t=k}^{k+N-1} (\Delta S_t^1 + \Delta S_t^2 + \Delta S_t^3) \times \alpha_2^{MAVE}, \tag{4a}$$

$$J_3 := \sum_{t=k}^{k+N-1} (\Delta O_t^I + \Delta O_t^D) \times \alpha_1^{MAVE}, \tag{4b}$$

$$J_4 := \sum_{t=k}^{k+N-1} \Delta O_t^{com} \times \alpha_1^{MAVE}, \tag{4c}$$

in which $z_1$, $z_2$, $z_3$, and $z_4$ are the objective weights. To scale the different objective values, constants, i.e., $\alpha^{MAVE}$ here, based on the Maximum Allowed Value Estimate (MAVE) values of the associated objectives are applied [35,36]. The MAVE values are the spillway outflow capacity $C_{spill}$ for $J_1$, $J_2$, and $J_4$, the storage amount between FWL and LWL for $J_2$, respectively. The optimization of these objectives is subject to the constraints:

$$O_{spill,t} + O_{turb,t} - O_{total,t} = 0, \tag{5a}$$

$$O_{total,t} - C_{turb} + \Delta O_t^{com} - O_{spill,t} = 0, \tag{5b}$$

$$O_{total,k}^k - O_{total,k}^{k-1} = 0, \tag{5c}$$

$$\Delta S_t^1 + S_U - S_t \geq 0, \tag{5d}$$

$$\Delta S_t^2 + S_t - S_L \geq 0, \tag{5e}$$

$$\Delta S_t^3 + S_H - S_t \geq 0, \tag{5f}$$

$$\Delta O_t^I - \Delta O_t^D + (O_{total,t}^k - O_{total,t}^{k-1}) \times \sigma_t = 0, \tag{5g}$$

$$\Delta O_t^I, \Delta O_t^D, \Delta O_t^{com}, \Delta S_t^1, \Delta S_t^2, \Delta S_t^3 \geq 0, \tag{5h}$$

in which $k$ represents each time step for the defined receding horizon MPC, where $t$ at time step $k$ ranges from $k$ to $k+N-1$. Therefore, the optimal control input vector obtained from the proposed linear MPC over the time span $t = k, k+1, ..., k+N-1$ can be written as follows:

$$\begin{aligned} \mathbf{u}^k = \{ & O_{total,k}^k, \ldots, \\ & O_{total,k+N-1}^k, O_{spill,k}^k, \ldots, O_{spill,k+N-1}^k, \\ & O_{turb,k}^k, \ldots, O_{turb,k+N-1}^k, \}, \end{aligned} \tag{6}$$

where $C_{turb}$ denotes the turbine outflow capacity, and $O^k_{total,t}$, $O^k_{spill,t}$, and $O^k_{turb,t}$ express the total outflow, the spillway gate outflow, and the turbine outflow at time $t$ and time step $k$, respectively. We refer to $\mathbf{u}^k$ as an outflow schedule at time step $k$. The control horizon is $N$. For simplicity, we omit a superscript $k$, e.g., $J^k_1$ at time step $k$ denotes $J_1$, except when it is essential, e.g., in (5c) and (5g) because these equations compare current outflows with outflows at previous time steps. $\Delta O^I_t$, $\Delta O^D_t$, $\Delta S^1_t$, $\Delta S^2_t$, $\Delta S^3_t$, and $\Delta O^{com}_t$ are considered as slack variables. $\Delta O^I_t$ and $\Delta O^D_t$ express the amounts of increase and decrease between consecutive outflow schedules at time $t$, respectively. $\Delta S^1_t$, $\Delta S^2_t$, and $\Delta S^3_t$ are also introduced to penalize the violations of target reservoir storage, i.e., the highest target water level, $S_H$, the upper target water level, $S_U$, and the lower target water level, $S_L$.

To address the complementarity between outflows via turbines and spillway gates, $\Delta Q^{com}_t$ is also taken into account. In general, this constraint is formulated by a nonlinear complementarity operator that can be written as [37]:

$$O_{spill} \times (O_{turb} - C_{turb}) = 0. \tag{7}$$

This nonlinear constraint can be linearised by adding a slack variable $\Delta O^{com}_t$ in (5b) and using the penalty function approach to integrate constraint violations in the objective $J_4$. As an example, if $O_{total,k} \leq C_{turb}$, then $O_{spill,t} = 0$ and $\Delta O^{com}_t$ is minimized by the difference between the total outflow and the turbines' capacity outflow. Conversely, if $O_{total,k} > C_{turb}$, then $O_{turb,t} = C_{turb}$ and $\Delta Q^{com}_t$ becomes zero. Since this is a soft constraint and this value is independent of other objectives, i.e., there is no trade-off with other objectives; we therefore deem $J_4$ as not dynamically responding to the relative preferences among objectives, and so $z_4$ is set to a constant.

To assign a higher penalty for changes in the near future, a time-dependent weight function, $\sigma_t$, is defined as follows:

$$\sigma_t = \begin{cases} 1/(t-k) & \text{, if } t \geq k+1, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

Even though there are nonlinear elements in the reservoir system, such as reservoir storage volume-water level relationship and water level-spillway outflow capacity relationship, we consider only the linear parts of the reservoir system by using reservoir volume directly as state instead of RWL and by assuming that errors from nonlinear factors can be included as additive errors within inflow uncertainty. These can be written as follows:

$$S_{t+1} = S_t + I_t - O_{total,t} \tag{9a}$$

$$\text{s.t. } 0 \leq O_{spill,t} \leq C_{spill}, \tag{9b}$$

$$0 \leq O_{turb,t} \leq C_{turb}, \tag{9c}$$

$$O_{min} \leq O_{turb,t} + O_{spill,t}, \tag{9d}$$

$$LWL \leq RWL_t \leq FWL, \tag{9e}$$

where $S_t$, $I_t$, and $O_t$ denote the reservoir storage, inflow, and outflow at time $t$, respectively. $C_{spill}$ expresses the outflow capacity via spillway gates. The constraints are set for the spillway and turbine discharge capacity, the LWL, and the FWL. The minimum water supply should be more than the demands, $O_{min}$, as expressed in (9d). In fact, the total demand varies monthly;

however, in this study, a constant value of 52 $m^3 s^{-1}$ is used, which is the annual average. Nonlinear elements, such as the relationship between RWL and storage amount, are used for the absolute performance evaluator.

## 1.3 Online selection of the best weight set

Since the weight for the soft constraint does not need to change dynamically because its value is independent of other objectives and constraints, as mentioned earlier, we set $z_4$ to a constant with a reasonably small value for simplicity. Therefore, the optimal set to be determined consists of three weights, as follows:

$$\mathbf{z}^* = \{z_1^*, \quad z_2^*, \quad z_3^*\}. \tag{10}$$

Using TPE, we can explore search spaces, such as those in Table 1, at each time step of the receding horizon control implementation. The search space is quantized in the normalised range [0,1] with steps of 0.01 interval to sufficiently explore all the objectives' space. Reasonably, $\mathbf{z} = \{0, 0, 0\}$ is excluded from the search space. As mentioned, the evaluator plays an important role in providing the best weight set at each time step according to (2a) and Fig 2. We formulate the absolute performance evaluator consisting of six nonlinear objective functions as follows:

$$\min E = \min(E_1 + E_2 + E_3 + E_4 + E_5 + E_6), \tag{11}$$

where,

$$E_1 = \exp(p \times \max O_{spill,t}/\mathrm{C}_{spill}) - 1, \tag{12a}$$

$$\begin{aligned} E_2 = {} & q \times [\max \mathrm{RWL} - \mathrm{NHWL}]^+ + q \times [\mathrm{RWL}_{t+N-1} \\ & -\mathrm{RWL}_t]^+ + v \times [\max \mathrm{RWL} - S_H]^+ + (q+v) \\ & \times [S_L - \min \mathrm{RWL}]^+ + q \times [\mathrm{RWL}_{t+N-1} - \mathrm{RWL}_t]^+, \end{aligned} \tag{12b}$$

$$E_3 = \max \gamma_t \times \frac{w}{\exp(2 \times t)},$$

$$\gamma_t = \begin{cases} 0 & \text{if } |O_{spill,t}^k - O_{spill,t}^{k-1}| = 0, \\ 1 & \text{otherwise}, \end{cases} \tag{12c}$$

$$E_4 = \begin{cases} \exp(2 \times p \times \max_t \quad (O_{spill,t+1} - O_{spill,t})) - 1 \\ \qquad\qquad\qquad\qquad \text{if } O_{spill,t} \geq \max I_t, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise}, \end{cases} \tag{12d}$$

**Table 1. Search spaces for weights in this research.**

| | $z_1$ | $z_2$ | $z_3$ |
|---|---|---|---|
| Search space | [0,1] | [0,1] | [0,1] |
| Search step | 0.01 | 0.01 | 0.01 |

https://doi.org/10.1371/journal.pwat.0000361.t001

$$E_5 = p \times \sum_t \kappa_t,$$

$$\kappa_t = \begin{cases} 1 & \text{if } O_{spill,t} > 0 \ \& \ O_{spill,t+1} = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{12e}$$

$$E_6 = \begin{cases} l & \text{if } \exists t \text{ where } O_{spill,t} \geq 0 \ \& \ O_{turb,t} < \mathrm{C}_{turb}, \\ 0 & \text{otherwise,} \end{cases} \tag{12f}$$

where $[.]^+$ is the ramp function, i.e., $[A]^+ = \frac{A+|A|}{2}$, and $p$, $q$, $v$, and $w$ are constants to be used to normalize each objective value between zero and 10 over typical situations such as when RWL is lower than $S_H$ and $O_{spill}$ is within the peak inflow of the target flood, with values of 2.4, 10, 5, and 75, respectively. $l$ should be very large, chosen here as 5000, to avoid opening the gates when the outflow capacity of turbines is not fully utilized. $\gamma_t$ and $\kappa_t$ are used for the values from conditional expressions.

Among the objectives in (12), $E_1$ to $E_3$ have the same meaning as $J_1$ to $J_3$, and $E_6$ as $J_4$ in (4), but they are more freely configured using exponential functions and conditional equations. Moreover, $E_4$ and $E_5$ are considered in the evaluator to account for preferences for i) minimising outflow schedule changes inside a prediction horizon and ii) minimising the number of gate closures in a prediction horizon after opening action. These objectives may not be as critical as $E_1$, $E_2$, and $E_3$ but are related to operators' preferences [9].

It is worth noting that the objective values for $J_1$ and $J_3$ in the linear MPC can be normalized to be of the same scale by multiplying $N$ for $J_1$. Additionally, changes are smaller than peaks, i.e., $\Delta O \leq \max O$. Furthermore, the effect of long-term changes is diminished by $\sigma_t$ in (8). Therefore, the MAVE values can not scale the objective values of the linear MPC exactly. In contrast, the evaluator's objective $E_3$ in (12d) holds equal importance to the other objectives within the evaluator, as various exponential and conditional equations are used to scale objective values, ensuring they reach the same values in worst-case scenarios. Moreover, $E_3$ has the maximum value when there is at least one change in outflow schedules.

## 1.4 Daecheong reservoir: Description of a case study

Daecheong Reservoir is a multipurpose reservoir located in the upper reaches of the Geum River, flowing through the central region of South Korea, as depicted in Fig 3.

The Daecheong reservoir serves as a vital water source, delivering 922,000 $m^3$ of drinking water daily to approximately 2 million residents in surrounding areas, while also supporting irrigation, hydroelectric power generation, and flood control. Geographically, it features a complex elongated structure with a dendritic drainage pattern, reaching a maximum width of around 1 km. The watershed flowing into Daecheong Dam covers 3204 $km^2$, and at full capacity, the reservoir holds $1490 \times 10^6 m^3$ of water, spans a surface area of 72.8 $km^2$, and reaches a maximum depth of 55 m [38]. Annual precipitation and water flow predominantly occur during summer months (June-September), with water levels fluctuating significantly—up to 15 m—during monsoon seasons. The hydraulic residence time averages about 145 days, characteristic of typical river-type reservoirs [39]. Annual average rainfall is 1230 mm and average inflow is 102 $m^3 s^{-1}$. The inflow for a five-year frequent flood is 5000 $m^3 s^{-1}$, which is 50 times the annual average inflow.

Although it impacts the flood control in the Geum River basin, Daecheong Reservoir has a relatively limited flood control capacity in comparison to its extensive watershed area.

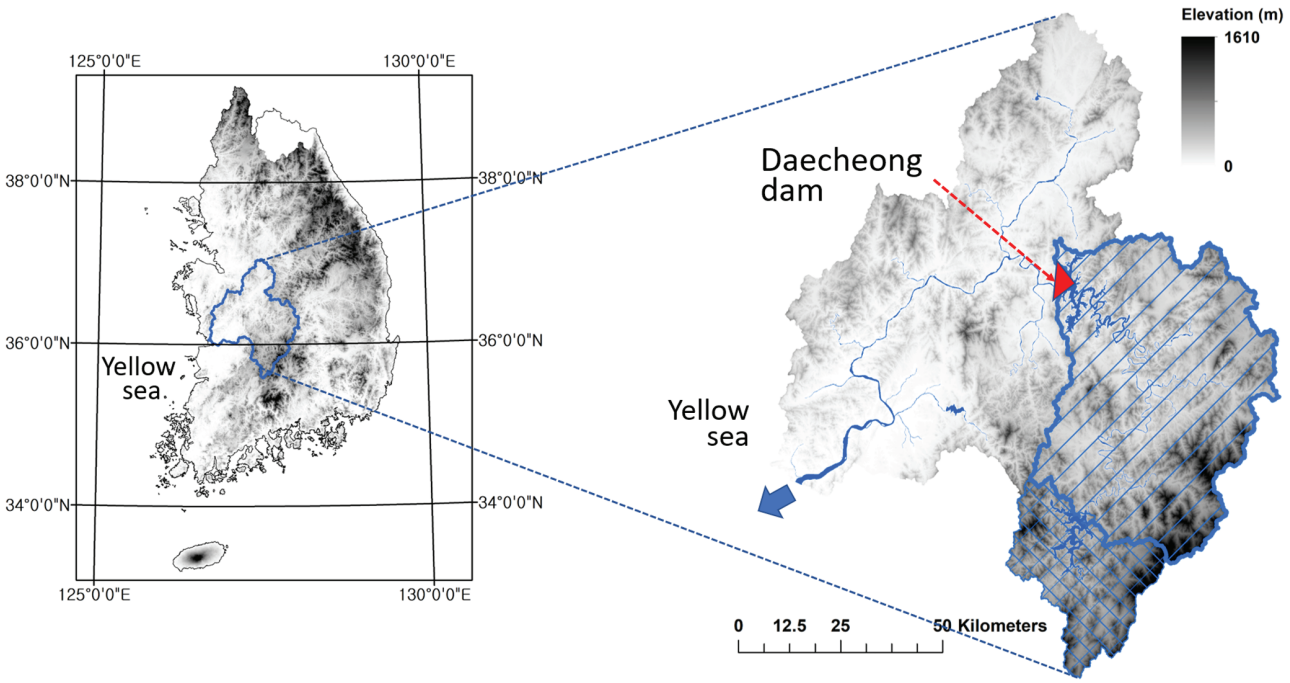



**Fig 3. Location of the Daecheong reservoir.** The left figure shows the location of the Geum river basin, and the right figure illustrates the location of the Daecheong Reservoir. The base maps are the digital elevation model (DEM) and the river shape file of Korea from the National Geographic Information Institute (https://www.ngii.go.kr) and the basin shape file from VWorld (https://www.vworld.kr) published by the Han River Flood Control Office (https://www.hrfco.go.kr).

https://doi.org/10.1371/journal.pwat.0000361.g003

For instance, the 200-year flood inflow is almost 10,700 $m^3 s^{-1}$, which can fill the flood control capacity in almost 6.5 hours. The characteristics of Daecheong Reservoir are provided in Table 2.

Based on the characteristics of the Daecheong reservoir, the threshold $S_H$ is set to Elevation Level (EL) 78.5 m, which is the 99.5th percentile of observed RWL during the experiment flood events, and $S_U$ and $S_L$ are set to EL 76.5 m (the NHWL), and EL 76.0 m in (5), respectively. The control horizon, $N$ in (4), is set to 6 h inspired by 6 h rainfall forecast reported by the meteorological agency of South Korea [9].

Hourly inflow data were collected for 24 years from 1997 to 2020, including 28 observed flood events from the National Water Resources Management Information System (WAMIS) operated by the Ministry of Environment, Korea, and the Korea Water Resources Public Corporation (K-water). These inflow data were calculated data based on measured outflows

**Table 2. Characteristics of the Daecheong reservoir.**

| Type | Value |
|---|---|
| Flood Water Level (FWL) | EL 80.0 m |
| Normal High Water Level (NHWL) | EL 76.5 m |
| Spillway crest level | EL 64.5 m |
| Low Water Level (LWL) | EL 60.0 m |
| Total storage | $1490 \times 10^6$ $m^3$ |
| Spillway capacity ($C_{spill}$) | 11,680 $m^3 s^{-1}$ |
| Turbine capacity ($C_{turb}$) | 264 $m^3 s^{-1}$ |

https://doi.org/10.1371/journal.pwat.0000361.t002

and changes in measured RWL over time. K-water records RWLs and outflows on an hourly basis, allowing inflows to be calculated using the water balance (Eq 9a) presented in Sect 1.2. The dataset is noise-corrupted due to measurement errors. By applying a wavelet filter [40], noises can be detected and removed from the dataset [41,42]. To account for uncertainty in inflow prediction, random normal values, in which the deviation is increased over the prediction time, are added to the filtered inflow data. A moving average filter is then used to prevent significant fluctuations. Details of this process can be found in [9], and a summary of it is given in Appendix A.

## 2 Simulation study

### 2.1 Data-driven models for explicit and switched MPC

Taking dynamical parameters into account based on real-time hydrological conditions used in PD-MPC is computationally expensive. For instance, with three dynamical weights and a search space of 100 quantized values per weight, each iteration of PD-MPC requires an average of 8.48 minutes to complete(CPU: AMD EPYC™ 9654). As the number of dynamical weights increases and the search space grows exponentially, the computational burden becomes too large. Therefore, two well-known methodologies, namely the explicit MPC and the switched MPC approach, are utilized to address this computational challenge of PD-MPC.

In conventional explicit linear MPC, it is assumed that for each predefined partition in the entire state space, there exists an affine function that maps the states to the optimal control inputs [27]. The state space may consist of several polyhedral partitions. Therefore, because of the high dimensionality of either state or control spaces, implementation of explicit MPC may be challenging [27,43]. In conventional switched MPC approaches, the dynamic system states are first partitioned into predefined parts. Then, based on the given system states, control inputs are calculated by MPC with the corresponding dynamical system [26]. Ensuring the continuity of control inputs in line with the well-defined optimized system partitions and switching rules in this approach is challenging [44].

For flood control, i.e., short-term operation, two specific challenges should be addressed: (i) unavailability of a comprehensive dataset including as many potential operational scenarios like flood events with several high peaks as possible; (ii) the smoothly changing states and the implicit relationship between the states and weights, which make it difficult to formulate switching rules for weights explicitly. Therefore, the implementation of explicit MPC and switched MPC is not straightforward.

In this work, we take a pragmatic approach, using a DNN model as a surrogate model for explicit MPC, trained based on PD-MPC results. In other words, the relationship between the states and the optimal control inputs is modeled using DNN based on the dataset obtained by PD-MPC, as shown in Fig 4. Similarly, for the switched MPC approach, the relationship between states and optimal weights from the PD-MPC results is modeled utilizing a Random Forest classifier and regressor as surrogate models for switching between different linear MPC controllers (i.e., switching between different weights). The detailed process illustrated in Fig 4 will be explained in the following section.

### 2.2 Data generation for learning surrogate models

**2.2.1 Data consistency.** At each time step of a receding horizon implementation, PD-MPC can propose a unique optimal outflow schedule $\mathbf{u}^{k*}$. However, there may be many combinations of the optimal parameter set, i.e. $\mathbf{z}^{k*} = \{z_1^*, z_2^*, z_3^*\}$, that could result in the same schedule. For instance, when a flood is going to occur, the RWL is between $S_L$ and $S_U$, and

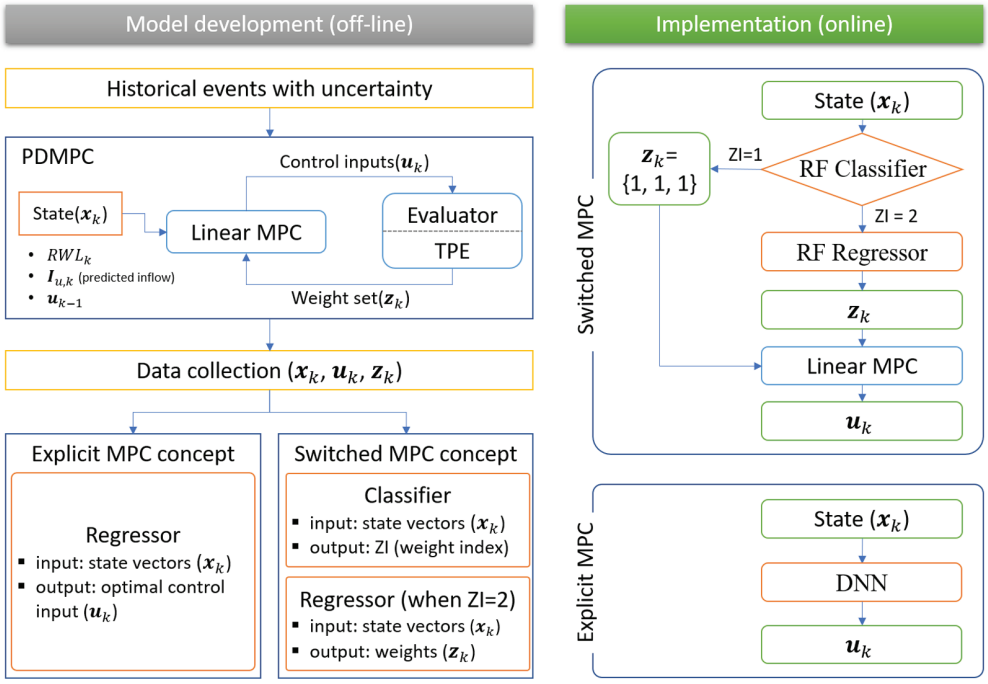



**Fig 4. A schematic diagram of the approach taken in this research.** (a) Model development (b) Implementation.

https://doi.org/10.1371/journal.pwat.0000361.g004

flood control can be achieved by regulating turbine outflow itself, while some objectives are inactive (are zero) in this scenario, resulting in the same optimal outflow schedule with different weight sets corresponding to the zero objectives (i.e., inactive constraint violations). In such cases, PD-MPC would randomly select a set among several available candidates at that time step, as there would be no one-to-one mapping between $z^{k*}$ and $u^{k*}$.

This can be a problem for a surrogate to learn a mapping between states and optimal weights. Therefore, we would want to somehow project the weights corresponding to the objectives relating to inactive constraints to a lower dimension. An $\ell_1$ regularization could, for example, be used to enforce these non-unique weights to have the same value (in this case, the mean of the weights for the objectives that are non-zero). An $\ell_1$ regularisation is typically employed to prevent overfitting in DNN models by adding the $\ell_1$-norm of model parameters to the original loss function. However, we incorporate the $\ell_1$-norm for weights into the objective function of the TPE algorithm to construct preferences for the same weights. The objective function expressed by (11) is then rewritten as follows:

$$\min E + \rho \left\| z - \bar{z} \right\|_1, \tag{13}$$

where $\rho$ denotes the weight for the preference for the same weights and small parameter value, and $\|.\|_1$ expresses the $\ell_1$-norm. To normalize a weight set, the average of weights $\bar{z}$ is subtracted.

It can be implied from (13) that, when there are multiple choices for weights, a set whose elements are as equal as each other should be chosen. Therefore, PD-MPC suggests $\mathbf{z}^*$ where its elements are the same, i.e., $z_1 = z_2 = z_3$, when $\mathbf{u}^*$ is not sensitive to $\mathbf{z}$ and it returns the

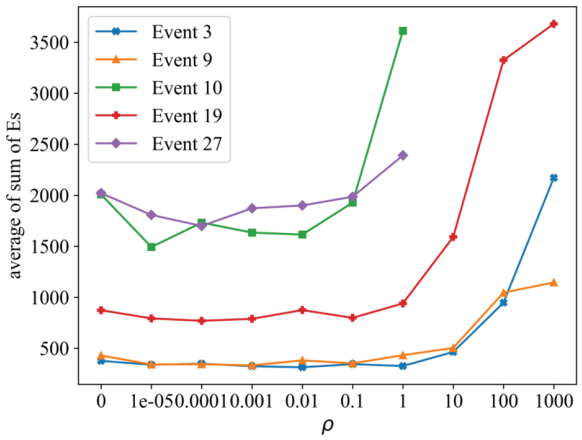
unique optimal $\mathbf{z}^*$ when it is highly sensitive. It should be noted that if $\{z_1 = z_2 = z_3\}$, whatever each value is, then it is trivial that the optimal control inputs are always equal to the control inputs when $\{z_1 = 1, z_2 = 1, z_3 = 1\}$. Thus, we can set $\mathbf{z}^*$ to $\{z_1 = 1, z_2 = 1, z_3 = 1\}$ when $z_1 = z_2 = z_3$ without trying to find the exact value of each weight. Basically, in the case of large $\rho$, PD-MPC proposes a suboptimal $\mathbf{z}$ that tends to be closer to the equal weight preference than $\mathbf{z}^*$. On the other hand, in the case of small $\rho$, it always finds an optimal solution for $\mathbf{z}^*$; however, it may not distinguish situations where $\mathbf{u}^*$ is insensitive to $\mathbf{z}$, thereby failing to ensure data consistency. To select an appropriate $\rho$, a sensitivity analysis is conducted using five training flood events, including some extreme cases. A total of 10 candidates are considered for $\rho$, ranging from $1 \times 10^{-5}$ to $1 \times 10^{3}$ with an additional case where $\rho = 0$ (meaning $\rho = 1 \times 10^{-\infty}$). Moreover, for each event, five uncertain inflow forecasts are applied. To consider a sufficiently diverse set of scenarios, we examine three initial water levels, i.e., EL 76.0 m, EL 76.5 m, and EL 77.0 m, covering a total of 750 cases ($5 \times 10 \times 5 \times 3$).

The sensitivity analysis results are depicted in Fig 5. Each point represents the average or the maximum penalty of the five uncertain inflow scenarios. Points not marked in the figure, e.g., those for Event No.10 in Fig 5a with $\rho = 1 \times 10^{3}$, $1 \times 10^{2}$, and $1 \times 10^{1}$, indicate the cases where online PD-MPC iterations are infeasible, meaning that constraint violations occurred at least once. In this figure, as $\rho$ decreases, the performance of PD-MPC is improved, i.e., penalty values decrease, and no further improvement is observed for smaller $\rho$ than $1 \times 10^{-3}$.

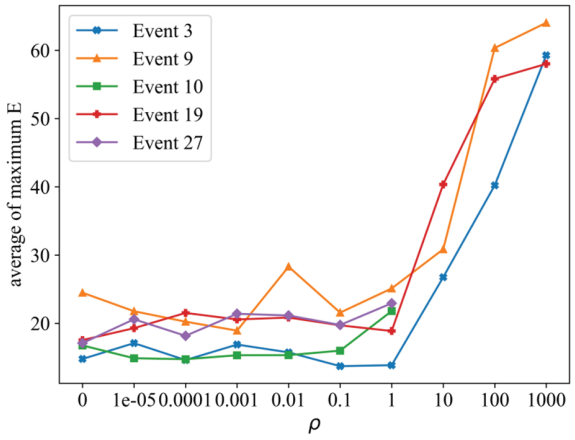
In Fig 6, ZI is an index for weights, where ZI= 1 corresponds to $\mathbf{z}^* = \{z_1 = z_2 = z_3 = 1\}$ and ZI= 2 is for all other cases. when ZI= 1, it also means that the same $\mathbf{u}^*$ is computed for various $\mathbf{z}$s, indicating that $\mathbf{u}^*$ is insensitive to $\mathbf{z}$. As shown in the figure, in most cases, such as the early stage of a flood event, $\mathbf{u}^*$ is insensitive to $\mathbf{z}$, resulting in several instances with ZI= 1. Therefore, ZI= 1 and ZI= 2 are considered differently in this analysis. In this figure, as $\rho$ decreases, a more diverse set of weights, i.e., ZI= 2 cases, appears. This indicates that even when the control inputs are less sensitive to weights, more weight sets closer to $\mathbf{z}^*$ can be obtained. Conversely, for larger values of $\rho$, ZI= 2 cases are only suggested when control inputs are highly sensitive to weights. When $\rho = 0$, most cases are ZI= 2, but it causes the inability to find the consistent weight set when the optimal control inputs are insensitive to weights, as discussed above. For instance, at time steps where multiple $\mathbf{z}$s lead to the same optimal control inputs and none of these $\mathbf{z}$s are in ZI= 1, with $\rho = 0$, the optimal weight set is randomly selected among ZI= 2 cases. However, for $\rho$ greater than zero, a case with less difference between each weight is preferably chosen. Therefore, based on the sensitivity analysis and the discussion above, $\rho$ can be set to $1 \times 10^{-4}$.

**2.2.2 Simulating multiple events for generating control learning data.** Among the total of 28 flood events, 22, 3, and 3 events are assigned for model training, validation, and testing, respectively. These validation and testing events may not represent all practical cases, but we carefully choose them to cover various inflow patterns, such as multiple peak inflows, significant peak magnitudes, and so on. For this case study, we do not anticipate very different conditions compared to the range of extreme and average scenarios considered. For other applications where different conditions are anticipated, data-driven models can be trained using outputs of simulation models, which can, for example, produce future inflows based on changes in land use and land cover patterns, and climate change scenarios.

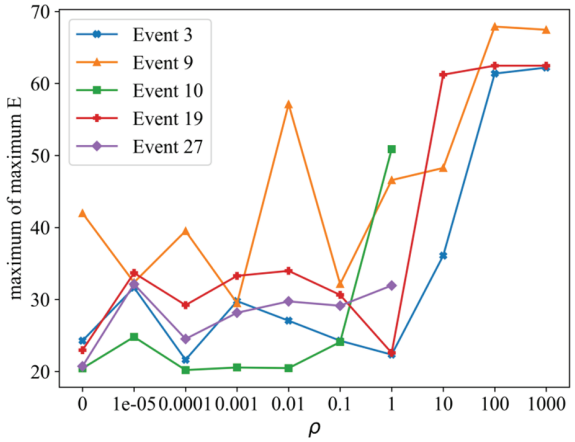
Data are collected through PD-MPC simulations [9] under the following conditions. The initial water levels were set from NHWL-0.5 m to +0.5 m at intervals of 0.1 m, using five uncertain inflow predictions. Setting much higher or lower initial water levels than our initial levels allows PD-MPC to stabilize RWL quickly by increasing or decreasing outflows abruptly. Therefore, a narrow range of water levels around the NHWL is set as initial RWLs. Consequently, the training data consists of 1210 cases (21 events × 11 initial RWLs × five uncertain




(a)




(b)




(c)

**Fig 5. Results of the sensitivity analysis in terms of the penalty values from the evaluator.** (a) average values of the sum of penalty values for five different uncertain forecasts. (b) average values of the maximum penalty value of each uncertain forecast. (c) maximum values of the maximum penalty value of each uncertain forecast.

https://doi.org/10.1371/journal.pwat.0000361.g005

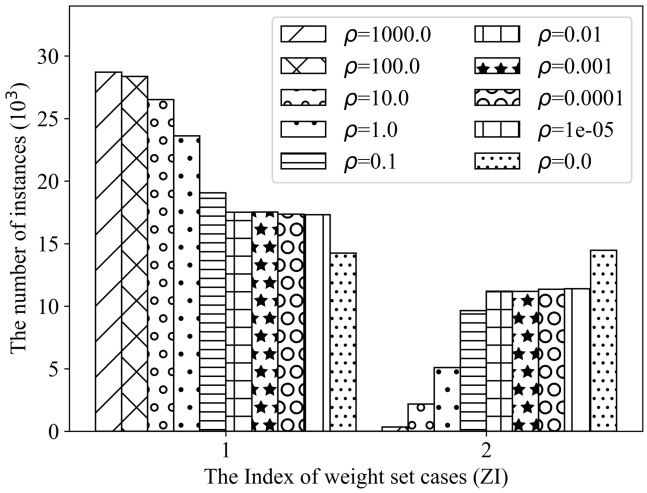



**Fig 6. The number of data for each ZI.**

https://doi.org/10.1371/journal.pwat.0000361.g006

inflow cases), and each validation and testing data have 165 cases (three events × 11 initial RWLs × five uncertain inflow cases) as presented in Table 3.

It is worth noting that historically observed initial RWLs are lower than our simulation, e.g., Event No.3, which occurred near the end of the flood season, started with an RWL at elevation 72.2 m. This is because operators want to secure enough flood control capacity in case of unexpected huge inflows, even though lower RWLs increase risks to water supply. Thus, lower initial RWLs in historical data often result from the absence of an optimal flood control system, which is what this study aims to address. Therefore, we set the initial RWL in this research to the NHWL, representing the likely initial condition when operators have access to optimal reservoir flood control inputs at each time step.

## 2.3 Optimal outflow schedule by DNN controller

The efficiency of DNNs can be affected by hyperparameter settings, such as the number of nodes and layers, the learning rate, the activation function, and the batch size. To find the best hyperparameters, a grid search based on mean squared errors is carried out [45]. However, considering the extensive use of ramp functions in the evaluator, the Rectified Linear Unit (ReLU) is selected as the activation function. In addition, the number of layers is also fixed to six based on a six-hour prediction horizon. We utilize the mean square error (MSE) loss function [46–48] with the Adam optimizer [48–51], which is widely used in developing surrogate inflow prediction models. The MSE loss function is particularly suitable for flood inflow

**Table 3. Study flood events.**

| Purpose | Events | Cases | Event No. |
|---|---|---|---|
| Training | 22 | 1210 | All except validation and testing events |
| Validation[a] | 3 | 165 | #22–#24 |
| Testing[b] | 3 | 165 | #1, #2, and #25 |

[a] one high (≈ 5000 $m^3 s^{-1}$, #23), two (#24) and three inflow peaks (#22).
[b] one high (> 5000 $m^3 s^{-1}$, #1), two high (≈ 5000 $m^3 s^{-1}$, #2), and three inflow peaks (#25).

https://doi.org/10.1371/journal.pwat.0000361.t003

prediction due to the huge variance between peak and low inflows. Additionally, it helps in accurately estimating the peak values, which are critical when targeting flood prediction.

To reduce computational burden and searching time and to prevent overfitting during the process of finding optimal hyperparameters, 330 cases for six events among the training events are utilized along with the early stopping technique [52], and the five-fold cross-validation approach. Through the grid search method, we found that batch size had the greatest impact on model performance, followed by learning rate and the number of nodes, respectively. The best hyperparameters are presented in Table 4.

As the results shown in Fig 7 and summarized in Table 5 indicate, the DNN model, mapping the states to the control inputs, is able to mimic accurately the PD-MPC control. Note that the DNN model is trained only for total outflows, i.e., $O_{total}$, for simplicity because spillway outflows can be obtained by abstracting the turbine capacity from total outflows when total outflows are larger than the turbine capacity. The Nash-Sutcliffe model efficiency coefficient, which is commonly used to evaluate the prediction accuracy of hydrological models [53], is consistently more than 0.95, which demonstrates that the model is reliable enough to replace PD-MPC [54]. The Root Mean Squared Error (RMSE) also varies between 42 $m^3 s^{-1}$ and 167 $m^3 s^{-1}$. Considering that the initial inflows for the three flood events in the validation set are 227.6 $m^3 s^{-1}$, 224.4 $m^3 s^{-1}$, and 116.6 $m^3 s^{-1}$, and their respective peak inflows are 4985.8 $m^3 s^{-1}$, 3655.2 $m^3 s^{-1}$, and 2590.4 $m^3 s^{-1}$, we can say that the model is accurate. It should be noted that because $O_t^k$ is fixed to the value determined in the previous time step, i.e., $k$–1, the DNN model output has $N$–1 values when the prediction horizon is $N$ hours. As expected, predictions for shorter time horizons are more accurate, while performance gradually deteriorates as the prediction time increases.

## 2.4 Training machine learning models for mapping states to preference (weights of objectives)

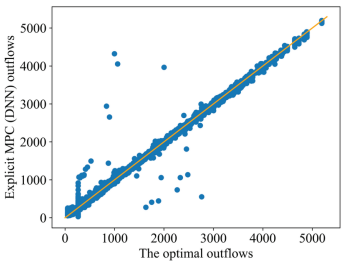
To simulate the changes in weights based on the states of the system, machine learning models are used to map them together. Given the predominance of ZI= 1 cases (shown in Fig 6) and the challenges of identifying a linear relationship between the states and the weights (shown in Fig 8), constructing a linear relationship to directly assign the best weights based on the states seems to be infeasible. Therefore, a binary classification model is employed to distinguish between the cases of ZI= 1 and ZI= 2. Additionally, a regression model is utilized to estimate the weights for ZI= 2.

Among the available criteria for selecting a binary classification algorithm, accuracy, precision score, and recall score are taken into account in this article. Accuracy represents the ratio of correctly classified instances to the total number of predictions. Precision score refers to the proportion of correctly classified instances for a specific class divided by the total instances classified as a given class by the model. Recall score measures the ratio of instances correctly classified as a particular class to the total instances of that class. When a classifier classifies a case with ZI= 1, the weight set is determined; however, when ZI= 2, the optimal weight set should be estimated using the regressor. In other words, a case needs to be classified as ZI= 2

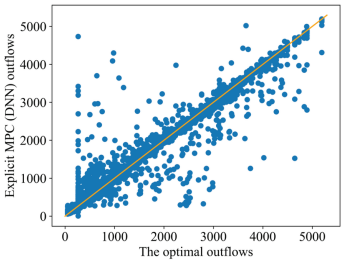
**Table 4. Hyperparameters of derived DNN models.**

| Learning rate | Layers | Nodes | Activation | Batch size | Epochs | Loss |
|---|---|---|---|---|---|---|
| 0.0005 | 6 | 128 | Relu | 10 | 1000 | MSE |

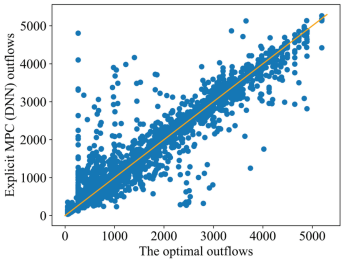
https://doi.org/10.1371/journal.pwat.0000361.t004



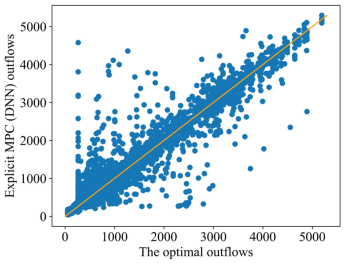

(a)



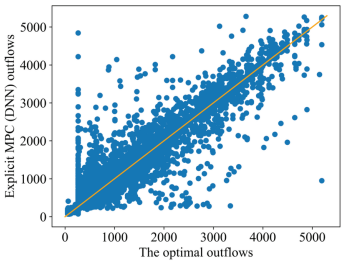

(b)




(c)




(d)




(e)

**Fig 7. Validation results of the DNN models for mapping states to outflow schedules; (a):** $O_{total,t+1}$**, (b):** $O_{total,t+2}$**, (c):** $O_{total,t+3}$**, (d):** $O_{total,t+4}$**, (e):** $O_{total,t+5}$. Performance diminishes with increases over time.

https://doi.org/10.1371/journal.pwat.0000361.g007

**Table 5. Nash–Sutcliffe coefficient and RMSE for validation events.**

| | $O_{total,t+1}$ | $O_{total,t+2}$ | $O_{total,t+3}$ | $O_{total,t+4}$ | $O_{total,t+5}$ |
|---|---|---|---|---|---|
| NSC | 0.998 | 0.986 | 0.982 | 0.980 | 0.963 |
| RMSE ($m^3\ s^{-1}$) | 42.1 | 105.5 | 119.2 | 124.2 | 167.6 |

https://doi.org/10.1371/journal.pwat.0000361.t005

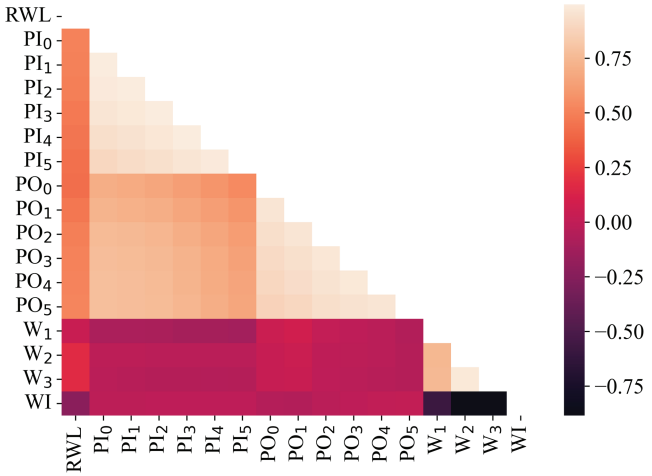



**Fig 8. The correlation among state elements and weights, where $PI_1$: predicted inflow at t+1, $PO_1$: previously decided outflow at t+1.**

https://doi.org/10.1371/journal.pwat.0000361.g008

when there is uncertainty and ZI= 1 when no uncertainty exists. Therefore, ZI= 1 and ZI= 2 are selected with a high precision score and a recall score, respectively.

Furthermore, to address the skewness of data in ZI, we adopt both under-sampling and over-sampling algorithms. The Edited Nearest Neighbours (ENN) is an under-sampling method that removes the nearest neighbors from the majority class data [55]. The Synthetic Minority Over-sampling Technique (SMOTE) [56] is an over-sampling technique where synthetic data are generated by randomly selecting one data point from the minority class and its nearest neighbor in the minority class. The synthetic data are classified by a classification model. Through the application of these techniques, the number of cases for ZI= 1 and ZI= 2 becomes balanced. The number of cases changes to 158134 for ZI= 1 from 187587 in the original data and 150783 from 89833 for ZI= 2.

Classifiers are constructed based on *k*–nearest Neighbors, Random Forest, and Support Vector algorithms using balanced data. As summarized in Table 6, the validation results indicate that the Random forest model is the best regarding precision and recall scores. For the implementation, a python library imbalanced learning [57] is used for the SMOTEENN algorithm, and the scikit-learn package [58] with default parameters for the classifiers.

**Table 6. Binary classification results on ZI.**

| Classifier | Precision ZI=1 | Precision ZI=2 | Recall ZI=1 | Recall ZI=2 | Accuracy |
|---|---|---|---|---|---|
| Random forest | *0.952* | 0.741 | 0.898 | *0.865* | 0.890 |
| K neighbors | 0.930 | 0.708 | 0.888 | 0.802 | 0.866 |
| Support vector | 0.941 | 0.676 | 0.864 | 0.839 | 0.857 |

https://doi.org/10.1371/journal.pwat.0000361.t006

To map the nonlinear relationship between the states and the weights when ZI= 2, a Random forest regressor is employed, which uses an ensemble approach with decision trees, which can effectively map the nonlinear relationship between inputs and outputs [18,59]. This model uses the average value at the final (decision-making) node for regression.

The parameters, i.e., the number of trees and the maximum depth of trees, are selected by grid search based on RMSE values for the validation events, as shown in Fig 9. The parameters are selected at the point where the RMSE curve starts to level off in order to simplify the model. The selected number of trees is then 150, and the maximum depth of each tree is 30. However, as can be seen in Fig 9, no significant difference is observed with different parameter values.

In addition, the Piecewise Affine (PWA) modelling approach, which is known for effectively approximating switched discrete-time nonlinear systems [60,61], is also employed. They are widely used to strike a balance between simplicity and accuracy [18]. This PWA approximation does not show considerable improvement in terms of the Peak RWL and the penalty values of the evaluator compared to the Random Forest regressor. The Random Forest regressor demonstrates even better performance in terms of the peak outflow and peak evaluator value. Linear Random Forest regressor [62], which uses the Random Forest algorithm, but the final value is calculated by linear regression, showing similar results with the Random Forest regressor. The details on the PWA model and Linear Random Forest regressor can be found in Appendix B.

## 3 Results and discussion

In this section, the efficiency of data-driven models based on the two aforementioned MPC approaches on the test dataset for flood events is presented and discussed. In this regard, the flood events selected for testing, as shown in Table 3, include cases with two and three peaks of inflow and a large peak. Therefore, except for extreme cases like the Probable Maximum Flood for designing reservoirs, these events could represent situations similar to those encountered by operators during flood control operations.

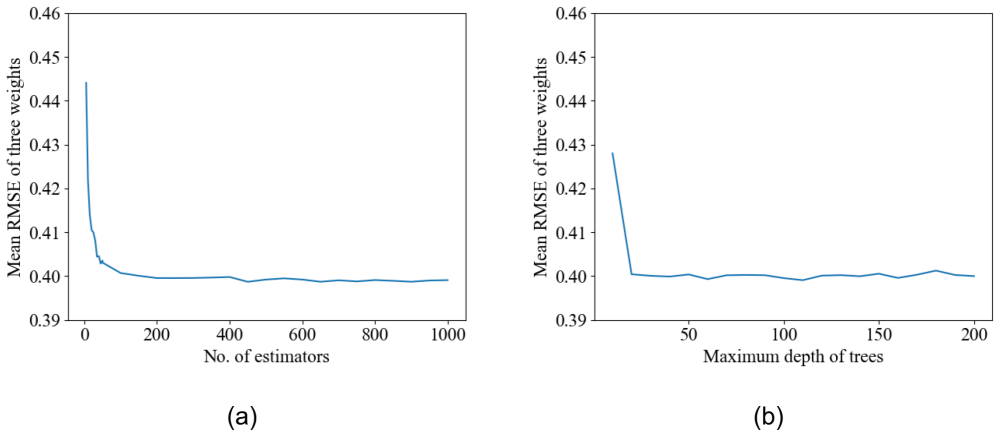



**Fig 9. Hyperparameter grid search results (to compare hyperparameters by the same scales, the scale of the y-axis is fixed to [0.40, 0.45]), (a) the number of trees (when the maximum depth of each tree is 30), (b) the maximum depth of each tree (when the number of trees is 150).**

https://doi.org/10.1371/journal.pwat.0000361.g009

## 3.1 Analyses of test events

The average of result indexes for each test event, i.e., the peak outflows, peak RWLs, minimum RWLs, the total evaluator values, and peak evaluator values, are presented in Table 7, Table 8, and Fig 10. 'Fixed 1', 'Fixed 2', and 'Fixed 3' refer to predefined weights, which are $\mathbf{z} = \{1, 1, 1\}$, $\{0.125, 0.625, 0.25\}$, and $\{0.0625, 0.625, 0.3125\}$, respectively. 'Fixed 2' and 'Fixed 3' represent the cases with a high weight for $J_2$ in (4), which emphasizes maintaining the RWL in the target water level. $J_2$ with a high weight forces the linear MPC controller to maintain the reservoir water level at the minimum possible level, ensuring operational stability even over extreme conditions. In addition, 'Fixed 3' highlights the importance of $J_3$, minimizing changes between outflow schedules.

'E-MPC' refers to the explicit MPC implemented using a DNN model, which is directly mapping the states and the outflow schedules. On the other hand, the switched MPC approach, which employs a random forest classifier and regressor as surrogate models for the weight vector, is referred to as 'S-MPC'.

For cases with predefined weights, the system violates at least one constraint more than once, and the peak RWL reaches the FWL sometimes. For example, although Fixed 3 has a high weight to the water level, in which the weight for $J_2$ is ten times higher than $J_1$ and two times higher than $J_3$, it shows the highest peak RWL in Event No.1 and 25. This result implies that fixing the weights may not respond adequately, i.e., being sensitive enough, to changing hydrological situations, thereby failing to satisfy the constraints and produce reliable results from a long-term perspective. In contrast, E-MPC, which incorporates the DNN model, and S-MPC, which applies the Random Forest models, show stable operation for the test events.

**Table 7. Average results for all test events.**

| | Peak outflow | Peak RWL | Minimum RWL | Total *E* | Peak *E* | No. changes |
|---|---|---|---|---|---|---|
| PD-MPC | 4137.3 | 78.27 | 76.07 | 934.5 | 17.3 | 49.4 |
| E-MPC | 4167.1 | 78.09 | 76.03 | 3207.7 | 30.6 | 926.9 |
| S-MPC | 4650.1 | 79.61 | 75.97 | 2667.7 | 52.3 | 23.9 |

https://doi.org/10.1371/journal.pwat.0000361.t007

**Table 8. Average results for each test event.**

| | Event | Peak out. | Peak RWL | Min. RWL | Total *E* | Peak *E* | No. changes |
|---|---|---|---|---|---|---|---|
| PD-MPC | 1 | 5487.7 | 78.4 | 76.1 | 1007.9 | 18.7 | 38.9 |
| Fixed 1 | 1 | 5304.8 | 79.9 | 76.0 | 2415.8 | 62.4 | 50.3 |
| Fixed 2 | 1 | 5304.8 | 79.9 | 76.0 | 2408.7 | 62.4 | 49.3 |
| Fixed 3 | 1 | 5474.9 | 80.0 | 76.0 | 2496.7 | 58.6 | 43.0 |
| E-MPC | 1 | 5340.0 | 78.4 | 76.0 | 2778.6 | 33.1 | 702.7 |
| S-MPC | 1 | 5225.7 | 79.5 | 76.0 | 1791.2 | 50.4 | 14.5 |
| PD-MPC | 2 | 5024.9 | 78.3 | 76.1 | 908.8 | 20.8 | 64.8 |
| Fixed 1 | 2 | Constraint violation | | | | | |
| Fixed 2 | 2 | Constraint violation | | | | | |
| Fixed 3 | 2 | 4324.7 | 79.9 | 76.0 | 3454.7 | 57.6 | 46.8 |
| E-MPC | 2 | 5056.3 | 78.4 | 76.0 | 3392.5 | 29.6 | 1006.1 |
| S-MPC | 2 | 4154.6 | 79.4 | 76.0 | 2353.1 | 41.9 | 30.1 |
| PD-MPC | 25 | 1899.3 | 78.0 | 76.0 | 886.7 | 12.5 | 44.5 |
| Fixed 1 | 25 | 1533.1 | 80.0 | 76.0 | 4571.4 | 59.4 | 35.0 |
| Fixed 2 | 25 | 1533.1 | 80.0 | 76.0 | 4586.2 | 59.4 | 34.1 |
| Fixed 3 | 25 | Constraint violation | | | | | |
| E-MPC | 25 | 2104.9 | 77.4 | 76.1 | 3452.2 | 29.2 | 1071.9 |
| S-MPC | 25 | 4570.1 | 79.9 | 76.0 | 3858.9 | 64.7 | 27.1 |

https://doi.org/10.1371/journal.pwat.0000361.t008

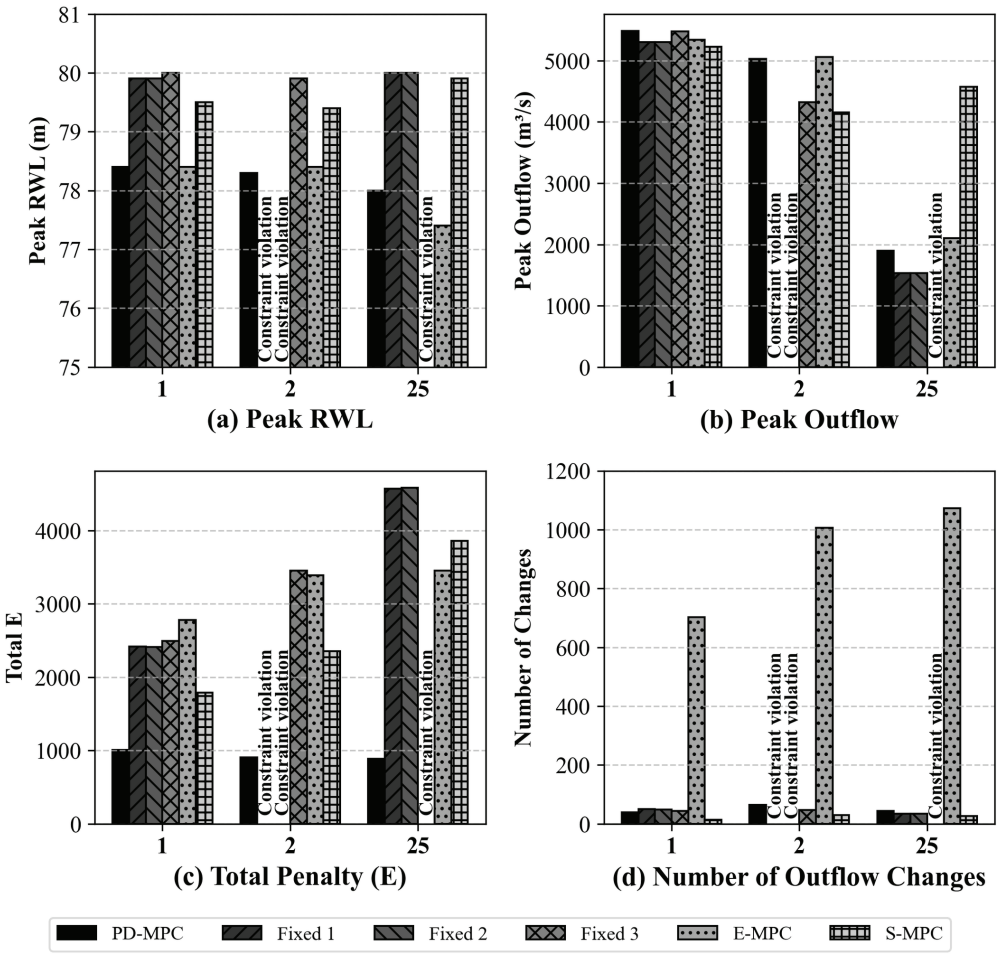



**Fig 10. Average results for each test event in terms of (a) Peak RWL, (b) Peak Outflow, (c) Total E, and (d) Number of changes in outflow schedules.**

https://doi.org/10.1371/journal.pwat.0000361.g010

In this scenario, the maximum outflow capacity, minimum water supply, and RWL constraints are all satisfied. Even though the best performance belongs to PD-MPC, they also outperform the fixed weight cases with respect to the absolute performance evaluator value.

The Absolute Performance Evaluator quantifies what reservoir operators want to achieve during flood events. While conventional MPC approaches are limited to linear or quadratic objective formulations, making it challenging to express complex operational goals mathematically, the PD-MPC framework overcomes this fundamental limitation by enabling the use of diverse nonlinear equations, including exponential functions and conditional formulations. Therefore, this evaluator can serve as a definitive performance index, providing a single quantitative measure. As presented in Table 7, S-MPC shows a lower average penalty in terms of the evaluator compared to E-MPC. Fixed weight cases generally present higher penalties than both S-MPC and E-MPC, with one exception: in Event No.1, E-MPC exhibits the highest evaluator value. However, regarding the highest penalty during an event period, 'Peak E', E-MPC has a lower average penalty than S-MPC. The results indicate that the relative performance of S-MPC and E-MPC is contingent upon specific event characteristics. Note that 'Total *E*' in Tables 7 and 8 represents the sum of penalty values accumulated during an

event period. Here, $E$ is interpreted as a penalty since the evaluator's objectives are formulated as minimization problems.

Considering the peak RWL and peak outflow, E-MPC mimics PD-MPC results accurately. It can be concluded that E-MPC is able to respond adequately to changes in the states. However, E-MPC tends to change the outflow schedule frequently, leading to very high values for $E_3$ in (11). This is because even small changes in the outflow schedule yield an increase in $E_3$ value, and the DNN regressor is less sensitive to minor deviations due to the effect of a loss function, which is the mean squared error.

On the other hand, S-MPC derives state-dependent weights, and the linear MPC, incorporating these weights, generates the optimal outflow schedule in real-time. As a result, in terms of the absolute performance evaluator, switched MPC shows lower penalty values and fewer changes in outflow schedules - an area emphasized by the evaluator, as mentioned in Sect 1.3 - compared to E-MPC. Since the PD-MPC framework is designed to select optimal control inputs based on the evaluator, not the linear MPC's objectives, we can conclude that switched MPC better replicates the original multi-objective optimization results compared to E-MPC.

However, this improvement is obtained by compromising the peak outflow and peak RWL, since the number of changes, peak outflow, and peak RWL are correlated. This can be explained by errors in these two surrogate models, indicating that the models are not sufficiently sensitive to a given state.

Some errors in the classification of ZI and the regression of weights can be expected for the surrogate models. Firstly, if the classifier fails to identify situations where ZI= 2, i.e., when weights need to change, the model maintains RWLs between the target RWLs and outflows at the minimum amount. It yields subsequent higher RWLs and larger outflows. Furthermore, if the regressor fails to provide the optimal weights, linear MPC may struggle to accurately determine the appropriate adjustments to the outflow.

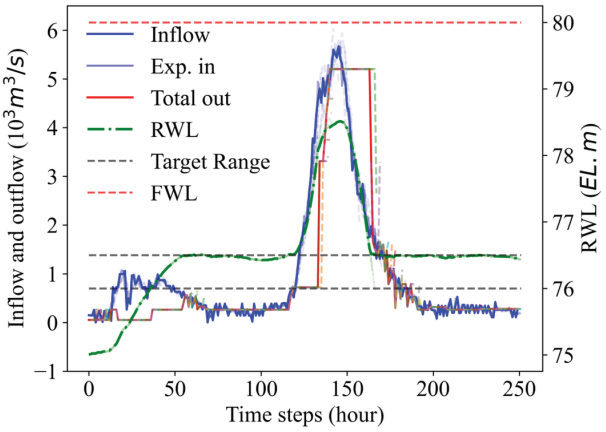
Moreover, when RWLs exceed EL 78.89 m, which is the maximum RWL in training data, switched MPC surrogate models need to provide optimal weights for states that fall outside the range of the training data. It can result in errors.

Despite errors in the surrogate models (RMSEs are approximately 0.45 in Fig 9) in optimal decision timings and outflows, the switched MPC approach enables linear MPC to produce robust results concerning constraints. Even if the surrogate models cannot present the optimal weights at a certain time step and the system state worsens in terms of the evaluator values, they can promptly respond to the slightly deteriorated new state due to the receding horizon control. Thus, linear MPC makes appropriate decisions with the weights provided by the surrogate model.

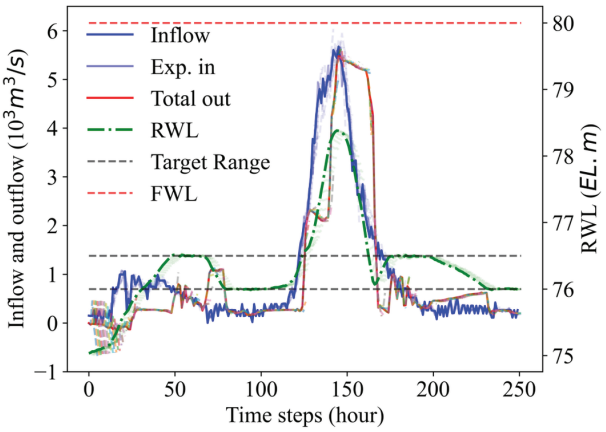
## 3.2 Comparison of explicit MPC and switched MPC

For a deeper understanding of the characteristics of surrogate models within the frameworks of explicit MPC and switched MPC, we define scenarios that these models have not previously been trained on. In general, data-driven models often fail to produce reliable outputs when the inputs are beyond the range of the training data [63,64]. A scenario with an initial RWL of EL 75.0 m is defined, while, in the training data, the lowest water level is EL 75.95 m. This scenario is simulated for Event No.1 with one peak and a significant peak inflow.

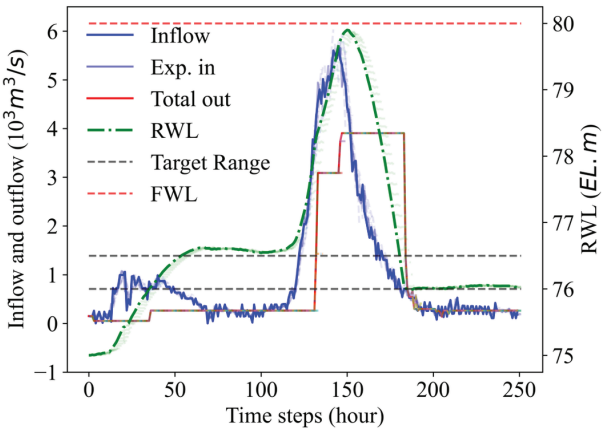
The simulation results of the data-driven models for the aforementioned scenario are shown in Fig 11 and summarized in Table 9. As expected, PD-MPC outperforms other methods in this case. In addition, S-MPC does not violate any constraints. However, E-MPC does not satisfy constraints compared to the other three cases, particularly when the RWL is low; E-MPC suggests negative outflow, e.g., $-844\ m^3\ s^{-1}$, to increase the reservoir level to the target




(a)




(b)




(c)

**Fig 11. Hydrographs for Event No.1 with unseen initial RWL, (a) PD-MPC, (b) Explicit MPC (DNN), (c) Switched MPC (Random Forest classification and regression).**

https://doi.org/10.1371/journal.pwat.0000361.g011

**Table 9. Results for unseen initial RWL for Event No. 1.**

| C | Peak out. | Min. out. | Peak RWL | Min. RWL | Total *E* | No. changes |
|---|---|---|---|---|---|---|
| PD-MPC | 5202.0 | 52.0 | 78.51 | 74.99 | 1282.4 | 49 |
| E-MPC | 5473.0 | **-844.0** | 78.41 | 74.98 | 3091.2 | 714 |
| S-MPC | 3905.1 | 52.0 | 79.90 | 75.00 | 2377.6 | 20 |

https://doi.org/10.1371/journal.pwat.0000361.t009

range (from EL 76.0 m to EL 76.5 m), even lower than the minimum outflow for water supply, i.e. $O_{min}$. This is because S-MPC explicitly incorporates constraints through linear MPC, whereas E-MPC only maps the states and the outflow schedules.

## 3.3 Discussions

Considering the limitation of data-driven models to model complex relationships between states and outflow schedules, or states and weights, which are embedded in the training data, the PD-MPC framework outperforms in all the cases. However, the explicit MPC and the switched MPC outperform in the fixed weight cases, demonstrating applicable results without violating constraints for the test events.

The DNN model based on the explicit MPC approach results in small and frequent changes in the outflow schedule, as shown in Table 8 and Fig 11b. This causes physically impossible outflows and constraint violations for the unseen cases. Despite these shortcomings, they could potentially be addressed by introducing heuristic rules based on an operator's experience. For instance, operators can design a simple heuristic algorithm in which a proposed outflow is replaced with the minimum outflow when the proposed outflow is less than the constrained minimum outflow. Negligible fluctuations between outflow schedules can also be ignored using these kinds of heuristic rules. Furthermore, as the number of training flood events increases and various uncertain inflow scenarios and initial RWLs are integrated, a surrogate DNN model can cover all potential states to avoid infeasible control inputs.

The model using the switched MPC approach also demonstrates reliable results for flood control. However, the complex relationship between the states and optimal weights poses a challenge due to the selection of a single weight set from multiple sets that produce the same optimal control inputs. As a result, using a conventional data-driven model often leads to significant testing errors. For instance, although not presented here, a DNN model does not outperform our proposed classifier and regressor. These errors cause an overall performance degradation, particularly delaying the timing of changes in the outflow schedule. Moreover, this degradation can cause a switched MPC surrogate model to encounter states far outside the training data. Consequently, the peak RWL is higher, and the peak outflow is larger than the PD-MPC results. The errors can also be mitigated by adopting knowledge-based heuristic rules. As an example, we can design different default weight sets for different states outside training data based on the operator's experience. The most crucial point to be considered is that online implementation of linear MPC can guarantee that the optimal outflow is stable, not violating constraints, even for unseen cases.

In summary, although PD-MPC outperforms other methods, it requires solving computationally complex nonlinear optimal control problems online for reservoir flood control, which makes it difficult to implement in practice. The switched MPC approach replaces the gradient-free randomized search algorithm, which was time-consuming for solving PD-MPC online. It thus can significantly reduce complexity. In addition, since linear MPC optimization is conducted every time step using updated information, results are robust to errors and

avoid constraint violations. The explicit MPC approach has the lowest complexity for implementing online as it directly maps states to control, but a DNN model can present infeasible control inputs and needs more training data to cover every possible situation. The complexity and accuracy of the discussed methods are schematically described in Fig 12.

Even though we can convert an expensive online nonlinear optimization problem into expensive offline training and cheap online implementation, it will be better to make the online nonlinear optimization, i.e., the GA or TPE algorithm itself, cheap. This means the online implementation of the PD-MPC framework becomes more practical. The warm-starting strategy [33,65] can be a good method to start. Adaptive learning techniques present promising alternatives to traditional data-driven models. For example, reinforcement learning [66] and learning-based MPC approaches [67,68] have gained significant attention in recent research. These methods can effectively reduce the errors inherent in surrogate models, ultimately near optimality with lower computational complexity.

Another important consideration is how to scale this approach to a multi-reservoir system. In standard MPC approaches, some challenges in scaling the receding horizon MPC approach to complicated multi-reservoir systems have been reported [7,69]. In our framework, expanding the surrogate switched and explicit MPC approaches to a multi-reservoir system is technically straightforward, given that obtaining the optimal outflow schedules and/or the optimal weights is feasible. In such extensions, the number of system states will grow linearly with the number of reservoirs in the system. Since the switched approach uses linear MPC, the computational complexity of the real-time control will remain manageable. For the explicit MPC, it would be even more computationally efficient. However, the most complex task in such an approach is designing a sufficient number of scenarios for the PD-MPC to generate a sufficient range of control data sets for learning the explicit and switched models.

The uncertainty of inflow forecasts also needs to be considered. Even though we consider the uncertainty of inflows by adding noise to observed inflow data when generating the training data, we deal with it in a deterministic way. However, it is advisable to explore stochastic/robust MPC [70] to consider uncertainty explicitly. In addition, we acknowledge that we cannot present the comprehensive impacts of the inherent inaccuracy of data-driven

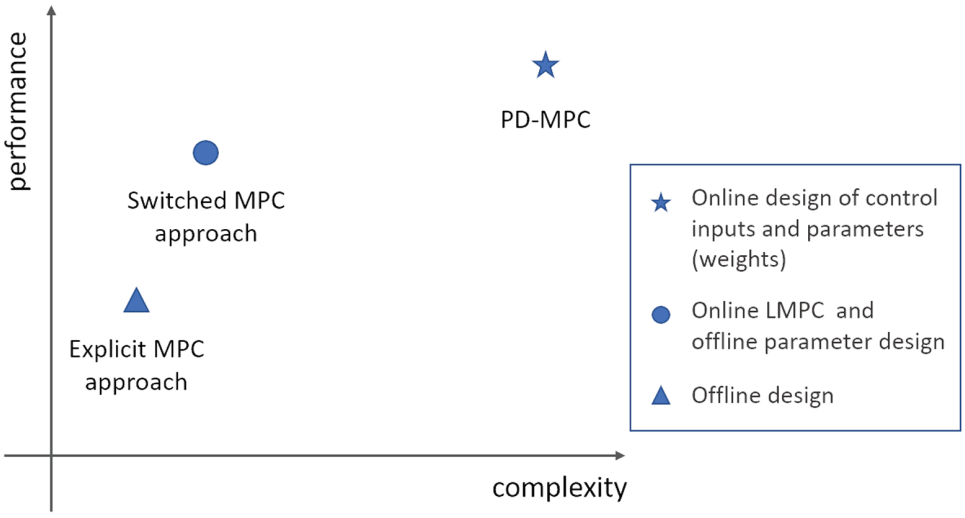



**Fig 12. Conceptual diagram of explicit and switched MPC approaches.**

https://doi.org/10.1371/journal.pwat.0000361.g012

models in reservoir flood control, despite our detailed explanations of how S-MPC and E-MPC respond to the errors of surrogate models within the studied flood scenarios. Many methodologies for uncertainty quantification of data-driven models, e.g., sensitivity analysis [71], Bayesian-based approaches, bootstrap-based approaches, and the Monte-Carlo method, have been studied [72]. However, incorporating these methodologies for comprehensive uncertainty analysis is beyond the scope of this research.

## 4 Conclusions

In this article, we propose data-driven surrogate models in the form of explicit and switched Model Predictive Control (MPC) approaches for a flood control problem with dynamically changing weights of multi-objectives, which is the first study based on the author's best knowledge. These approaches have the advantage of low computational complexity for online implementation compared to a Parameterized Dynamic Model Predictive Control (PD-MPC) framework proposed in our previous work [9]. The PD-MPC formulation explicitly considers the dynamic characteristic of the relative importance of objectives depending on the situation via time-varying relative weights for the different objectives, which are optimized online at each time step of the MPC by solving a highly nonlinear optimization problem. Through numerical experiments on the Daecheong multipurpose reservoir in South Korea, we demonstrate that both approaches outperform linear MPC approaches with predefined fixed weights and produce compatible results with PD-MPC. Especially, the switched MPC approach is robust in terms of model errors despite some constraint violations of the explicit MPC DNN model under previously unseen events. However, because the highly complex relationship between states and weights leads to the performance degradation of data-driven models for the switched MPC, testing these strategies for other case studies where there may be a larger size of training data may be useful.

Overall, we expect this study to contribute to widening the possibility of the application of real-time optimal reservoir flood control and lay the foundation for practical implementation in the field. The switched MPC approach, for example, offers a computationally efficient method for real-time reservoir operation to effectively balance multiple objectives, such as minimizing flood risk downstream, maintaining reservoir levels, and minimizing outflow schedule changes, even under uncertain inflow conditions we simulated. Additionally, the insights into the dynamic weighting of objectives based on real-time hydrological conditions and states can aid in developing more adaptive and robust flood control strategies.

Even though our research focuses on a single reservoir in South Korea, we believe our approach could be applicable to reservoirs with various hydrological conditions. The methodology presented in this study may be adaptable to different reservoir systems with their own unique characteristics and operational requirements, as the fundamental concepts of model predictive control and surrogate modelling potentially extend to diverse hydrological settings. The framework potentially offers sufficient flexibility to accommodate region-specific constraints and objectives, suggesting its potential utility for addressing water resource management challenges in different geographical contexts.

## Supporting information

**S1 Appendix**
(PDF)

**S2 Appendix**
(PDF)

## Author contributions

**Conceptualization:** Ja-Ho Koo, Ali Moradvandi, Edo Abraham, Dimitri P. Solomatine.

**Data curation:** Ja-Ho Koo.

**Formal analysis:** Ja-Ho Koo, Ali Moradvandi.

**Investigation:** Ja-Ho Koo.

**Methodology:** Ja-Ho Koo.

**Software:** Ja-Ho Koo.

**Supervision:** Edo Abraham, Andreja Jonoski, Dimitri P. Solomatine.

**Validation:** Ja-Ho Koo.

**Visualization:** Ja-Ho Koo.

**Writing – original draft:** Ja-Ho Koo.

**Writing – review & editing:** Ali Moradvandi, Edo Abraham, Andreja Jonoski, Dimitri P. Solomatine.

## References

1. Giuliani M, Lamontagne JR, Reed PM, Castelletti A. A state-of-the-art review of optimal reservoir control for managing conflicting demands in a changing world. Water Resour Res. 2021;57(12). https://doi.org/10.1029/2021WR029927
2. Labadie JW. Optimal operation of multireservoir systems: state-of-the-art review. J Water Resour Plan Manag. 2004;130(2):93–111. https://doi.org/10.1061/(Asce)0733-9496(2004)130:2(93)
3. Lee S, Kang T, Lee K. An operational model of a reservoir system simulation for real-time flood control in the Han River Basin. J Flood Risk Manag. 2017;10(4):499–510. https://doi.org/10.1111/jfr3.12159
4. Cheng C, Chau KW. Fuzzy iteration methodology for reservoir flood control operation 1. JAWRA J Am Water Resour Assoc. 2001;37(5):1381–8.
5. Castelletti A, Ficchì A, Cominola A, Segovia P, Giuliani M, Wu W, et al. Model predictive control of water resources systems: a review and research agenda. Annu Rev Control. 2023;55:442–65. https://doi.org/10.1016/j.arcontrol.2023.03.013
6. Jain SK, Shilpa L, Rani D, Sudheer K. State-of-the-art review: operation of multi-purpose reservoirs during flood season. J Hydrol. 2023;618:129165. https://doi.org/10.1016/j.jhydrol.2023.129165
7. Myo Lin N, Tian X, Rutten M, Abraham E, Maestre JM, van de Giesen N. Multi-objective model predictive control for real-time operation of a multi-reservoir system. Water. 2020;12(7):1898. https://doi.org/10.3390/w12071898
8. Breckpot M, Agudelo OM, De Moor B. Flood control with model predictive control for river systems with water reservoirs. J Irrigat Drainage Eng. 2013;139(7):532–41. https://doi.org/10.1061/(Asce)Ir.1943-4774.0000577.
9. Koo JH, Abraham E, Jonoski A, Solomatine DP. Balancing operator's risk averseness in model predictive control of a reservoir system; 2024. Available from: https://arxiv.org/abs/2407.04506
10. Chiang PK, Willems P. Combine evolutionary optimization with model predictive control in real-time flood control of a river system. Water Resour Manag. 2015;29:2527–42.
11. Kong Y, Mei Y, Wang X, Ben Y. Solution selection from a pareto optimal set of multi-objective reservoir operation via clustering operation processes and objective values. Water. 2021;13(8):1046. https://doi.org/10.3390/w13081046
12. Moridi A, Yazdi J. Optimal allocation of flood control capacity for multi-reservoir systems using multi-objective optimization approach. Water Resour Manag. 2017;31:4521–38.
13. Liu D, Bai T, Deng M, Huang Q, Wei X, Liu J. A parallel approximate evaluation-based model for multi-objective operation optimization of reservoir group. Swarm Evolution Comput. 2023;78:101288. https://doi.org/10.1016/j.swevo.2023.101288

14. Pan I, Babaei M, Korre A, Durucan S. Artificial neural network based surrogate modelling for multi-objective optimisation of geological CO2 storage operations. Energy Procedia. 2014;63:3483–91. https://doi.org/10.1016/j.egypro.2014.11.377
15. Karg B, Lucia S. Efficient representation and approximation of model predictive control laws via deep learning. IEEE Trans Cybern. 2020;50(9):3866–78. https://doi.org/10.1109/TCYB.2020.2999556 PMID: 32574145
16. Nubert J, Kohler J, Berenz V, Allgower F, Trimpe S. Safe and fast tracking on a robot manipulator: robust mpc and neural network control. IEEE Robot Automat Lett. 2020;5(2):3050–57.
17. Hertneck M, Kohler J, Trimpe S, Allgower F. Learning an approximate model predictive controller with guarantees. IEEE Control Syst Lett. 2018;2(3):543–8. https://doi.org/10.1109/lcsys.2018.2843682
18. Bemporad A. A piecewise linear regression and classification algorithm with application to learning and model predictive control of hybrid systems. IEEE Trans Automat Control. 2023;68(6):3194–209. https://doi.org/10.1109/TAC.2022.3183036
19. Li L, Rong S, Wang R, Yu S. Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: a review. Chem Eng J. 2021;405:126673. https://doi.org/10.1016/j.cej.2020.126673
20. Wang D, Shen ZJ, Yin X, Tang S, Liu X, Zhang C, et al. Model predictive control using artificial neural network for power converters. IEEE Trans Indust Electron. 2021;69(4):3689–99.
21. Hirose N, Tajima R, Sukigara K. MPC policy learning using DNN for human following control without collision. Adv Robot. 2018;32(3):148–59. https://doi.org/10.1080/01691864.2017.1422188
22. Cai W, Kordabad AB, Esfahani HN, Lekkas AM, Gros S. MPC-based reinforcement learning for a simplified freight mission of autonomous surface vehicles. In: 2021 60th IEEE Conference on Decision and Control (CDC). IEEE; 2021. p. 2990–5.
23. Arroyo J, Manna C, Spiessens F, Helsen L. Reinforced model predictive control (RL-MPC) for building energy management. Appl Energy. 2022;309:118346. https://doi.org/10.1016/j.apenergy.2021.118346
24. Song Y, Scaramuzza D. Policy search for model predictive control with application to agile drone flight. IEEE Trans Robot. 2022;38(4):2114–30.
25. Liberzon D. Switching in systems and control vol. 190. Springer; 2003
26. Zhang L, Zhuang S, Braatz RD. Switched model predictive control of switched linear systems: Feasibility, stability and robustness. Automatica. 2016;67:8–21.
27. Alessio A, Bemporad A. A survey on explicit model predictive control. Nonlinear model predictive control: towards new challenging applications. 2009. p. 345–69.
28. Schwenzer M, Ay M, Bergs T, Abel D. Review on model predictive control: an engineering perspective. Int J Adv Manuf Technol. 2021;117(5):1327–49. https://doi.org/10.1007/s00170-021-07682-3
29. Kouvaritakis B, Cannon M. Model predictive control. Switzerland: Springe; 2016; 38(13-56):7.
30. Alasali F, Haben S, Foudeh H, Holderbaum W. A comparative study of optimal energy management strategies for energy storage with stochastic loads. Energies. 2020;13(10):2596.
31. Wang H, Olhofer M, Jin Y. A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex Intell Syst. 2017;3:233–45.
32. Bergstra J, Bardenet R, Bengio Y, Kégel B. Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst. 2011;24.
33. Watanabe S. Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance. arXiv preprint 2023. https://arxiv.org/abs/2304.11127
34. Bergstra J, Yamins D, Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: International Conference on Machine Learning. PMLR; 2013. p. 115–23.
35. Aydin BE, Essink GHPO, Delsman JR, van de Giesen N, Abraham E. Nonlinear model predictive control of salinity and water level in polder networks: case study of Lissertocht catchment. Agricult Water Manag. 2022;264:107502. https://doi.org/10.1016/j.agwat.2022.107502
36. Xu M, van Overloop PJ, van de Giesen NC. On the study of control effectiveness and computational efficiency of reduced Saint-Venant model in model predictive control of open channel flow. Adv Water Resour. 2011;34(2):282–90.
37. Powell K, Eaton A, Hedengren J, Edgar T. A continuous formulation for logical decisions in differential algebraic systems using mathematical programs with complementarity constraints. Processes. 2016;4(1):7. https://doi.org/10.3390/pr4010007

38. Chung S, Hipsey MR, Imberger J. Modelling the propagation of turbid density inflows into a stratified lake: daecheong reservoir, Korea. Environ Model Softw. 2009;24(12):1467–82. https://doi.org/10.1016/j.envsoft.2009.05.016
39. Park Y, Kim S, Chon K, Lee H, Kim JH, Shin JK. Impacts of heavy rain and floodwater on floating debris entering an artificial lake (Daecheong Reservoir, Korea) during the summer. Desalination Water Treatment. 2021;219:399–404.
40. Nourani V, Baghanam AH, Adamowski J, Kisi O. Applications of hybrid wavelet–artificial intelligence models in hydrology: a review. J Hydrol. 2014;514:358–77. https://doi.org/10.1016/j.jhydrol.2014.03.057
41. Sahay RR, Srivastava A. Predicting monsoon floods in rivers embedding wavelet transform, genetic algorithm and neural network. Water Resour Manag. 2014;28:301–17.
42. Kisi O, Shiri J. Precipitation forecasting using wavelet-genetic programming and wavelet-neuro-fuzzy conjunction models. Water Resour Manag. 2011;25:3135–52.
43. Ferreau HJ, Bock HG, Diehl M. An online active set strategy to overcome the limitations of explicit MPC. Int J Robust Nonl Control: IFAC-Affiliat J. 2008;18(8):816–30.
44. Zhu F, Antsaklis PJ. Optimal control of hybrid switched systems: a brief survey. Discrete Event Dyn Syst. 2015;25:345–64. https://doi.org/10.1007/s10626-014-0187-5
45. Huang IH, Chang MJ, Lin GF. An optimal integration of multiple machine learning techniques to real-time reservoir inflow forecasting. Stochastic Environ Res Risk Assessm. 2022;36(6):1541–61.
46. Shelke M, Londhe S, Dixit P, Kolhe P. Reservoir inflow prediction: a comparison between semi distributed numerical and artificial neural network modelling. Water Resour Manag. 2023;37(15):6127–43.
47. Ahmad M, Al Mehedi MA, Yazdan MMS, Kumar R. Development of machine learning flood model using artificial neural network (ANN) at Var River. Liquids. 2022;2(3):147–60.
48. Maddu R, Pradhan I, Ahmadisharaf E, Singh SK, Shaik R. Short-range reservoir inflow forecasting using hydrological and large-scale atmospheric circulation information. J Hydrol. 2022;612:128153. https://doi.org/10.1016/j.jhydrol.2022.128153
49. Kingma D.P. Ba J. Adam: a method for stochastic optimization arXiv preprint 2014. https://arxiv.org/abs/14126980
50. Kao I-F, Zhou Y, Chang L-C, Chang F-J. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. J Hydrol. 2020;583:124631. https://doi.org/10.1016/j.jhydrol.2020.124631
51. Le X-H, Ho HV, Lee G, Jung S. Application of long short-term memory (LSTM) neural network for flood forecasting. Water. 2019;11(7):1387. https://doi.org/10.3390/w11071387
52. Zhang J, Chen F, Cui Z, Guo Y, Zhu Y. Deep learning architecture for short-term passenger flow forecasting in urban rail transit. IEEE Trans Intell Transp Syst. 2020;22(11):7004–14.
53. Jain SK, Sudheer K. Fitting of hydrologic models: a close look at the Nash–Sutcliffe index. J Hydrol Eng. 2008;13(10):981–6.
54. Moriasi DN, Arnold JG, Van Liew MW, Bingner RL, Harmel RD, Veith TL. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Trans ASABE. 2007;50(3):885–900. https://doi.org/10.13031/2013.23153
55. Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern. 1972;3(3):408–21. https://doi.org/10.1109/tsmc.1972.4309137
56. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57. https://doi.org/10.1613/jair.953
57. LemaA Z˜tre G, Nogueira F, Aridas CK. Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J Mach Learn Res. 2017;18(17):1–5.
58. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.
59. Breiman L. Random forests. Machine learning. 2001; 45: 5–32.
60. Moradvandi A, Lindeboom RE, Abraham E, De Schutter B. Models and methods for hybrid system identification: a systematic survey. IFAC-PapersOnLine. 2023;56(2):95–107.
61. Wang M, Qiu J, Yan H, Tian Y, Li Z. Data-driven control for discrete-time piecewise affine systems. Automatica. 2023;155:111168. https://doi.org/10.1016/j.automatica.2023.111168
62. Ao Y, Li H, Zhu L, Ali S, Yang Z. The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. J Petrol Sci Eng. 2019;174:776–89. https://doi.org/10.1016/j.petrol.2018.11.067
63. Badillo S, Banfai B, Birzele F, Davydov II, Hutchinson L, Kam-Thong T, et al. An introduction to machine learning. Clin Pharmacol Ther. 2020;107(4):871–85. https://doi.org/10.1002/cpt.1796 PMID: 32128792

64. Wang H, Lei Z, Zhang X, Zhou B, Peng J. Machine learning basics. Deep learning. 2016; p. 98–164.
65. Feurer M, Springenberg J, Hutter F. Initializing bayesian hyperparameter optimization via meta-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29; 2015.
66. Li Y Deep reinforcement learning: an overview. arXiv preprint 2017. https://arxiv.org/abs/1701.07274
67. Hewing L, Wabersich KP, Menner M, Zeilinger MN. Learning-based model predictive control: toward safe learning in control. Annu Rev Control Robot Auton Syst. 2020;3(1):269–96. https://doi.org/10.1146/annurev-control-090419-075625
68. Yang S, Wan MP, Chen W, Ng BF, Dubey S. Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Appl Energy. 2020;271:115147. https://doi.org/10.1016/j.apenergy.2020.115147
69. Lin NM, Rutten M. Optimal operation of a network of multi-purpose reservoir: a review. Procedia Eng. 2016;154:1376–84. https://doi.org/10.1016/j.proeng.2016.07.504
70. Saltık MB, Ozkan L, Ludlage JHA, Weiland S, Van den Hof PMJ. An outlook on robust model predictive control algorithms: reflections on performance and computational aspects. J Process Control. 2018;61:77–102.
71. Pianosi F, Beven K, Freer J, Hall JW, Rougier J, Stephenson DB, et al. Sensitivity analysis of environmental models: a systematic review with practical workflow. Environ Modell Softw. 2016;79:214–32. https://doi.org/10.1016/j.envsoft.2016.02.008
72. Kasiviswanathan K, Sudheer K. Quantification of the predictive uncertainty of artificial neural network based river flow forecast models. Stochastic Environ Res Risk Assessm. 2013;27:137–46.