

Article

Advancing Hydrological Prediction with Hybrid Quantum Neural Networks: A Comparative Study for Mile Mughan Dam

Erfan Abdi ^{1,2} , Mohammad Taghi Sattari ^{1,2,3} , Saeed Samadianfard ^{1,2,4}  and Sajjad Ahmad ^{5,*} 

¹ Department of Water Engineering, Faculty of Agriculture, University of Tabriz, Tabriz 51666-16471, Iran; erabdi78@gmail.com (E.A.); mtsattar@gmail.com (M.T.S.); samadianfard@gmail.com (S.S.)

² Water Sciences and Hydroinformatics Research Center, Khazar University, Mahsati Str. 41, AZ 1096 Baku, Azerbaijan

³ Department of Agricultural Engineering, Ankara University, Ankara 06100, Türkiye

⁴ Department of Environmental Engineering, Izmir Institute of Technology, Izmir 35433, Türkiye

⁵ Department of Civil and Environmental Engineering and Construction, University of Nevada Las Vegas, Las Vegas, NV 89154, USA

* Correspondence: sajjad.ahmad@unlv.edu

Abstract

Predicting dam inflow is critical for human life safety, water resource management, and hydroelectric power generation. While machine learning (ML) models address complex, nonlinear hydrological problems, quantum machine learning (QML) offers greater potential to overcome classical computational limits. This study compares a hybrid quantum neural network (HQNN) with the following two classical models: bidirectional CNN-LSTM and support vector regression (SVR). These models were evaluated to predict monthly inflow to the Mile Mughan Dam, a transboundary hydroelectric and irrigation dam located on the Aras River between Azerbaijan and Iran, using a 14-year dataset (2010–2023) under two scenarios. In total, 70% of data was used for training and 30% for testing. The first scenario encompassed meteorological variables plus three months of inflow lags, and the second included inflow lags only. Model performance was assessed using Coefficient of Determination (R^2), Root Mean Squared Error (RMSE), Nash–Sutcliffe efficiency (NSE), Mean Absolute Percentage Error (MAPE), and graphical plots. HQNN showed superior performance across all metrics. In Scenario 1, HQNN achieved $R^2 = 0.915$, RMSE = 37.318 MCM, NSE = 0.908, MAPE = 8.343%; CNN-BiLSTM had $R^2 = 0.867$, RMSE = 46.506 MCM, NSE = 0.858, MAPE = 10.795%; SVR had $R^2 = 0.846$, RMSE = 52.372 MCM, NSE = 0.821, MAPE = 12.772%. In Scenario 2, HQNN maintained strong performance ($R^2 = 0.855$, RMSE = 48.56 MCM, NSE = 0.845, MAPE = 9.979%) and outperformed CNN-BiLSTM ($R^2 = 0.810$, RMSE = 56.126 MCM, NSE = 0.793, MAPE = 11.456%) and SVR ($R^2 = 0.801$, RMSE = 60.336 MCM, NSE = 0.761, MAPE = 12.901%). In Scenario 1 and Scenario 2, HQNN increased the prediction accuracy by 19.76% and 13.47%, respectively, compared to the CNN-BiLSTM model. These results confirm HQNN's reliability in both multivariate and univariate modeling.

Keywords: quantum machine learning; hybrid quantum neural network; hydrological prediction; dam inflow prediction; ensemble predicting; convolutional neural network-bidirectional LSTM



Academic Editor: Stefano Alvisi

Received: 18 November 2025

Revised: 7 December 2025

Accepted: 15 December 2025

Published: 18 December 2025

Citation: Abdi, E.; Sattari, M.T.; Samadianfard, S.; Ahmad, S. Advancing Hydrological Prediction with Hybrid Quantum Neural Networks: A Comparative Study for Mile Mughan Dam. *Water* **2025**, *17*, 3592. <https://doi.org/10.3390/w17243592>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Accurately predicting dam inflow is a public safety and resource management challenge. By better predicting the volume of water flowing into the reservoir from rainstorms

or snowmelt, water managers can gain time to safely release stored water and increase storage capacity for the management of floods. This can moderate the peak inflow of the flood and protect downstream structures from damage. This can turn a natural disaster into a manageable event, protecting lives and property. In areas where reservoir storage is important, forecasting inflows enables reliable planning of drinking water supply and agricultural irrigation for future seasons [1]. For hydroelectric dams, inflow forecasting is directly related to optimizing power generation. This process enables operators to maximize production when demand is high and maintain grid stability [2]. This function ensures good water management for homes, farms, and industries.

Traditional methods struggle with complex, nonlinear time series due to their reliance on assumptions of stationarity and linearity, leading to poor capture of shifts, trends, and temporal dependencies [3,4]. Machine learning (ML) excels at finding complex, nonlinear patterns within large datasets that are often impossible for physical models or human analysts to discern. In general, machine learning improves performance in most fields, including transportation [5], agriculture, and hydrology [6]. By training on historical hydrologic data—including rainfall, snowpack, soil moisture, temperature, and past inflow rates—ML models can learn the unique features of a watershed's response to weather events [7]. This allows ML models to make highly accurate inflow forecasts by processing real-time data from weather forecasts and sensor networks. Unlike complex physical hydrological models that require detailed knowledge of the terrain and soil properties, ML models can often achieve robust predictions directly from the data, making them particularly valuable in data-scarce or rapidly changing environments. The practical benefits of this are transformative for reservoir management. ML models can process vast amounts of data in near-real time, providing forecasts with a longer and more reliable lead time. This gives operators a crucial window to make proactive decisions, such as pre-releasing water to create flood storage or conserving water in anticipation of a drought. Furthermore, ML enables ensemble prediction, where multiple scenarios are run simultaneously, quantifying the probability and uncertainty of different inflow outcomes [8]. This shift from a single “most likely” forecast to a probabilistic range empowers operators to assess risks more comprehensively and make smarter, more informed choices.

The practical value of these advanced models is confirmed through successful real-world implementations. For instance, ensemble ML methods and deep learning models are increasingly being deployed for operational forecasting at major dams, directly enhancing water management decisions. A case study from the Gezhouba Dam on the Yangtze River by Zhang et al. [9] provides compelling evidence, where machine learning models delivered highly accurate short-term inflow forecasts. Lee et al. [10] introduced a multi-inflow prediction ensemble (MPE) model for dam inflow forecasting, leveraging auto-sklearn to integrate ensemble models tailored for high- and low-inflow regimes. By assigning datasets based on flow conditions, the MPE model significantly improves prediction accuracy compared to conventional ensemble approaches. It achieves notable reductions in RMSE (22.1%) and MAE (24.9%) for low inflows, while boosting R^2 and NSE by 21.9% and 35.8%, respectively. Latif and Ahmed [11] explored the generalization of support vector regression (SVR) models for reservoir inflow forecasting using daily, weekly, and monthly inflow and rainfall data. Four SVR kernels—RBF, linear, normalized polynomial, and sigmoid—were tested across two climate-diverse case studies: Dokan Dam (Iraq) and Warragamba Dam (Australia). Results showed that daily data and the RBF kernel yielded the best performance at Dokan Dam ($RMSE = 145.7$, $R^2 = 0.85$), but failed to generalize effectively to Warragamba Dam. Deb et al. [12] evaluated the predictive capabilities of five machine learning models—Bidirectional Long Short-Term Memory (Bi-LSTM), Convolutional Neural Networks (CNNs), eXtreme Gradient Boosting (XGBoost), Light

Gradient Boosting Machine (LGBM), and Random Forest (RF) —for daily reservoir inflow forecasting at the Sri Ram Sagar Project, Telangana, India. Using 15.5 years of hydrological and climate data, seven input combinations (S1–S7) were tested, with hyperparameter tuning via the Tree-Structured Parzen Estimator. Bi-LSTM with input set S7 achieved the highest performance (Kling–Gupta efficiency = 0.92 training, 0.87 testing).

Quantum Machine Learning (QML) holds significant promise for time series prediction by addressing fundamental computational limitations of classical methods. Time series data, especially in finance or hydrology fields, often involves modeling highly nonlinear patterns and interactions between a vast number of variables. Classical models, such as deep neural networks, struggle with the curse of dimensionality, where computational costs grow exponentially with data complexity [13]. QML offers a pathway to overcome this by leveraging the inherent properties of quantum mechanics. The core importance lies in quantum computing's ability to process information in a massively parallel way. Through quantum superposition, a system of qubits can represent a multitude of possible data states simultaneously. Furthermore, quantum feature maps can naturally encode classical data into an exponentially large feature space, allowing QML models to identify complex, hidden patterns in the data that are computationally prohibitive for classical computers to uncover. This intrinsic capability makes QML exceptionally suited for capturing the intricate dependencies and stochastic nature inherent in real-world time series. Niu et al. [14] proposed a hybrid ELM–QPSO model for hydrologic time-series prediction, addressing limitations of traditional single-layer feedforward networks (SLFNs) and gradient-based learning. Extreme Learning Machine (ELM) offers fast training and strong nonlinear mapping, but suffers from random parameter initialization. To overcome this, the model integrates quantum-behaved particle swarm optimization (QPSO) to optimize ELM's input-hidden weights and biases, while output weights are computed analytically via the Moore–Penrose inverse. Applied to daily runoff data from China's Xinfengjiang Reservoir (2000–2014), the ELM–QPSO model demonstrated superior generalization performance, validating QPSO as an effective alternative for ELM parameter tuning. Grzesiak and Thakkar [15] applied quantum machine learning (QML) for enhancing flood forecasting, focusing on daily flood events along Germany's Wupper River in 2023. They integrated a hybrid framework of classical models (Support Vector Machines (SVM), K-Nearest Neighbors (KNN), regression, and Autoregressive (AR)) with QML techniques (SVM, KNN, Adaboost, Quantum Variational Circuits, QBoost, and QSVC_ML), leveraging quantum properties such as superposition and entanglement. Comparative analysis revealed that QML models deliver competitive training times and superior prediction accuracy. Vajpayee et al. [16] proposed a quantum optimization framework integrated with Geographic Information Systems (GISs) to enhance flood risk management. By reviewing existing literature, analyzing case studies, and comparing classical and quantum optimization methods, the research demonstrated that quantum algorithms offer faster and more accurate solutions for complex flood scenarios. The approach improves resource allocation, prediction accuracy, and overall resilience to flooding events. The findings underscore the value of merging emerging quantum technologies with traditional disaster management strategies to address the growing challenges posed by climate change and urbanization.

Although accurate inflow forecasting is vital for dam safety and water resource management, comparative analyses of quantum-inspired and advanced classical models for this purpose remain scarce. To address this gap, the primary aim of this study is to conduct a comprehensive comparative analysis of three distinct machine learning architectures—a Hybrid Quantum Neural Network (HQNN), a Convolutional Neural Network–Bidirectional Long Short-Term Memory (CNN–BiLSTM) model, and Support Vector Regression (SVR)—for predicting inflow into the Mile Mughan Dam. The investigation

is designed to evaluate model performance under two key forecasting scenarios: a univariate framework that utilizes only historical inflow lags, and a multivariate framework that incorporates both inflow lags and meteorological parameters. By benchmarking the nascent HQNN model against established classical deep learning and statistical benchmarks, this research seeks to elucidate the potential advantages and limitations of quantum-inspired machine learning in capturing the complex, nonlinear temporal dynamics inherent in hydrological systems.

2. Materials and Methods

2.1. Overview of the Study Area

The Mile Mughan Dam, also known as the Mile Mughan reservoir, is a transboundary hydroelectric and irrigation complex located along the Aras River, forming part of the boundary between Azerbaijan and Iran. Situated near the town of Horadiz in Azerbaijan's Fuzuli District, the dam lies at approximately $39^{\circ}25'59''$ N and $47^{\circ}22'15''$ E, about 150 m above sea level. The dam system consists of two embankment dams, a spillway, and intake structures that distribute water to the Main Mil Canal (Azerbaijan) and Main Mughan Canal (Iran) (Figure 1). The reservoir supports the irrigation of about 400,000 hectares of farmland across both countries, including more than 100,000 hectares of new agricultural development in Azerbaijan alone [17]. This joint water system has been central to the regional economy, enabling agricultural expansion in the Mil and Mughan plains, known for their cotton, grain, and horticultural production. With a total water volume of over 1.3 billion cubic meters, the dam is used for both water supply and flood control. The area of Mil Mughan Dam is 996 hectares, with an average depth of 5 m and a dam length of 2800 m, including a 125 m concrete section. Its hydroelectric system has a capacity to transfer 2700 cubic meters of water per second during floods or inundations. The dam supports the irrigation of approximately 200,000 hectares of land, of which about 120,000 hectares is in Azerbaijan alone.

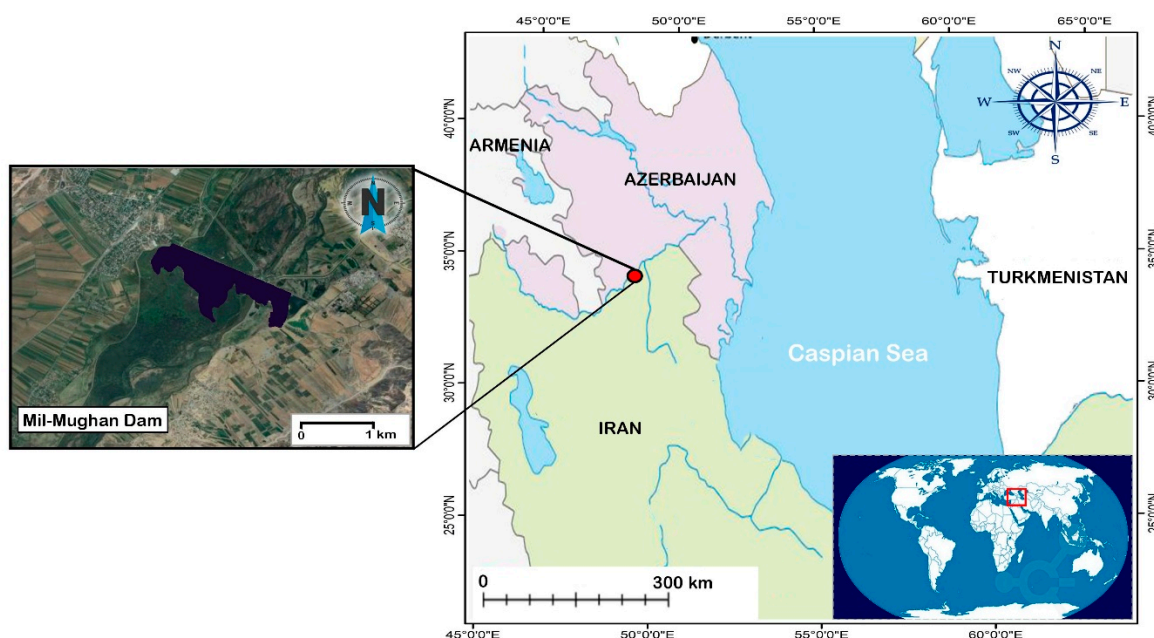


Figure 1. The location of the Mil Mughan Dam.

The climate in the region of the Mil Mughan Dam is characterized by a semi-arid to arid continental climate. Summers are hot, dry, and sunny, with average temperatures in the warmest months reaching around 30 to 32 °C. Winters tend to be relatively cold with

average temperatures dropping to around 2 to 8 °C. Annual precipitation in this area is low, typically ranging between 250 and 350 mm, mostly falling during the spring and autumn seasons. Sunshine duration is high, with around 2200 to 2400 h of sunlight annually, which contributes to the dry conditions in summer and moderate temperatures in other seasons. Relative humidity drops significantly in summer, often to 30–40%, with moderate levels (50–65%) in the colder months.

Predicting the inflow to the Mil Mughan Dam is important for efficient water resource management, optimizing irrigation schedules, and enhancing hydroelectric power generation. Accurate inflow forecasts help in planning how much water to release for agricultural demand, managing reservoir storage to prevent flooding, and ensuring water availability during dry periods. Given that the dam supports substantial irrigation across Azerbaijan and Iran, inflow prediction directly impacts regional food security, water allocation, and cross-border water sharing agreements. Figure 2 presents the 14-year time series of monthly inflow data for the Mil Moghan Dam, measured in million cubic meters (MCM).

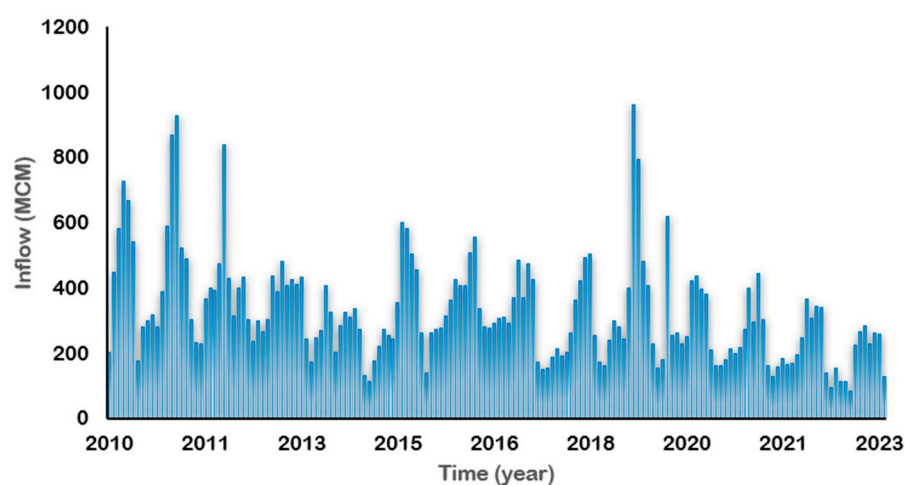


Figure 2. Time series plot of monthly inflow to Mil Mughan Dam.

2.2. Overview of the Data Collection and Data Preprocessing

To predict and analyze the inflow to the Mil Mughan Dam, monthly inflow records were sourced from the Water Department. Complementary meteorological data were retrieved from the Mathematica 13.3 software database [18]. The analysis spans 14 years, covering January 2010 through June 2023. Given the advantages of multivariate over univariate modeling, the study incorporated three key meteorological variables: precipitation (PP), mean temperature (Tmean), and mean humidity (Hmean), along with three-month lags in inflow data. Table 1 presents the statistical summary of the selected parameters.

Table 1. Statistical characteristics of the parameters used for the modeling.

Statistical Criteria	Maximum	Average	Minimum	Standard Deviation	Number of Zero
Inflow (MCM)	962.26	327.694	83.22	159.642	0
Mean Temperature (°C)	30.07	15.502	1.74	8.688	0
Mean Humidity (%)	88.90	69.22	44.60	10.21	0
Precipitation (cm)	19.09	2.003	0	2.484	10

Preparing raw data for modeling and prediction tasks requires thorough preprocessing. This step involves applying various methods to enhance the quality and analytical readiness of the data. Key phases in preprocessing typically include data cleaning, transformation, and dimensionality reduction. Effective preprocessing not only boosts the accuracy

of machine learning models but also reduces training time and leads to more reliable predictions. Without this critical step, even sophisticated algorithms may yield flawed results due to substandard input data. As such, investing effort in preprocessing is fundamental to developing robust machine learning solutions. Following the collection of data, outliers were detected based on their substantial deviation from the general data distribution using a statistical criterion. Specifically, values exceeding three standard deviations (3σ) from the mean in each dataset were considered outliers, following the common 3-sigma rule. To maintain uniformity, missing values were filled in using the mean of the respective feature. The vast majority of outliers occurred sporadically, with no concentration in a specific season, and the proportion of removed data in any single month remained below 1%. A before-and-after comparison of monthly inflow climatology confirmed that the seasonal cycle (mean and standard deviation of monthly inflow) changed very little in all months, indicating that the outlier treatment did not materially distort the seasonal signal. Following this, Z-score normalization was applied to all variables, standardizing them to have a mean of zero and a standard deviation of one. In total, around 3% of the dataset was altered due to anomalies or missing data. Given that a small amount of data was outliers or missing, simple methods were used for data preprocessing. Hence, completing and correcting these data did not affect the model's ability to predict peak flows.

Once the dataset was finalized, two modeling scenarios were developed to predict monthly inflow in million cubic meters (MCM). The first scenario incorporated variables such as average temperature, relative humidity, precipitation, and a three-month lag of inflow data. In contrast, the second scenario relied solely on the three-month inflow lag. Incorporating these temporal delays, particularly in monthly inflow forecasting, proved beneficial by capturing the short-term memory characteristics of the hydrological system. This strategy provided valuable insights into how past inflow trends influence current inflow behavior. In this research, monthly data covering 14 years from 2010 to June 2023 were divided into training and testing subsets. The training set comprised 70% of the data, spanning January 2010 to June 2019, while the remaining 30%, from July 2019 to June 2023, was allocated for testing purposes. The predictive modeling was conducted using three distinct approaches: SVR, a hybrid CNN-BiLSTM architecture, and an HQNN. Following the modeling and prediction of monthly inflow, the outcomes were evaluated using visual plots and performance metrics. All computational experiments were executed in Python version 3.13.3 (released in April 2025) on a Windows 11 system equipped with 16 GB of RAM. The analysis utilized several updated Python libraries, including NumPy 1.25.0, Pandas 2.0.1, Scikit-learn 1.3.0, TensorFlow 2.13.0, and Matplotlib 3.8.0. Leveraging these recent versions ensured compatibility and benefited from the performance enhancements introduced in 2024 and 2025. Figure 3 illustrates the workflow used to model monthly wind energy generation through machine learning techniques.

2.3. Machine Learning Model

2.3.1. An Overview of the SVR Model

Support Vector Regression (SVR) is a variant of the Support Vector Machine (SVM) algorithm designed specifically for regression problems, where the objective is to estimate continuous outputs rather than assign class labels. SVR constructs a predictive function—typically a hyperplane—that captures the relationship between input variables and target values, while allowing for a tolerance margin (ϵ) within which deviations are not penalized [19]. This margin forms a tube around the regression function, and the algorithm strives to include as many data points as possible within this boundary, balancing simplicity and predictive accuracy. Two critical parameters govern SVR's behavior: the regularization constant (C), which manages the trade-off between model complexity and training error,

and the epsilon threshold, which defines the zone of acceptable error. Given training data (x_i, y_i) for $i = 1, \dots, n$, where $x_i \in \mathbb{R}^p$ are predictors and $y_i \in \mathbb{R}$ are target values, SVR seeks a function [20]:

$$f(x) = w^T \phi(x) + b \quad (1)$$

where w is the weight vector, $\phi(x)$ is a (possibly nonlinear) mapping of the input features into a higher-dimensional space, b is the bias term. The coefficients w and b are estimated by minimizing a risk function that balances the model's complexity and the degree to which predictions deviate from the true values by more than ϵ .

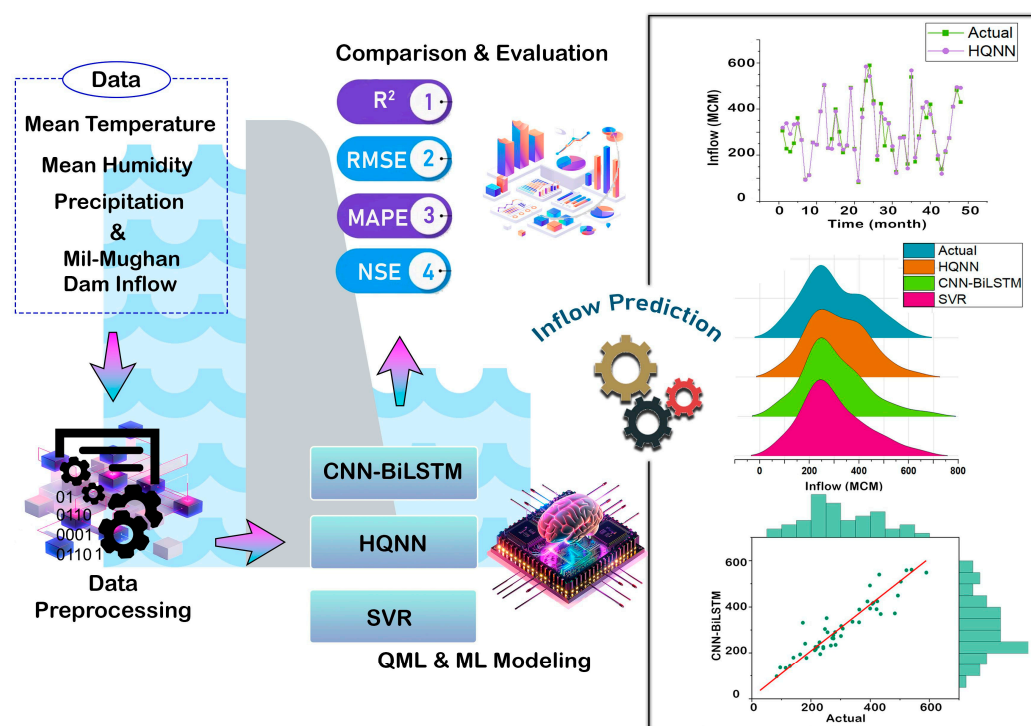


Figure 3. Flowchart of the steps taken to predict the reservoir inflow with three machine learning models.

Formally, SVR minimizes

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2)$$

subject to

$$\begin{cases} y_i - w^T x_i - b \leq \epsilon + \xi_i \\ w^T x_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (3)$$

where ϵ is the margin of tolerance (epsilon-insensitive zone), ξ_i, ξ_i^* are slack variables for errors outside the margin, and C controls the trade-off between model flatness and tolerance of deviations.

Using kernels $K(x_i, x_j)$, the prediction becomes

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (4)$$

where α_i, α_i^* are Lagrange multipliers learned during training.

This simple form captures the core idea: fit a function as flat as possible while ignoring small errors within ϵ , and penalizing larger deviations linearly.

In contrast to traditional regression techniques that aim to minimize overall prediction error, SVR focuses on maintaining predictions within a defined epsilon-insensitive margin, which enhances its resilience to noise and outliers. The support vectors—those data points nearest to the margin boundaries—are instrumental in shaping the regression model [20]. SVR is capable of modeling both linear and complex nonlinear relationships by employing kernel functions that project data into higher-dimensional feature spaces. This adaptability makes SVR a valuable method for tasks such as time series analysis and financial modeling, where achieving a balance between precision and generalization is essential.

The model used in this research adopts a radial basis function (RBF) kernel, which efficiently captures nonlinear relationships between the input parameters and the target variable by mapping the data into a higher-dimensional space. The SVR model parameters were carefully selected to balance model complexity and generalization capability. The penalty parameter (C) was set to 300, which controls the trade-off between maximizing the margin and minimizing training error—larger values of C allow fewer violations but can increase overfitting risk. The kernel coefficient (γ) was assigned a value of 0.5, determining the influence of individual training samples; smaller γ values result in smoother decision functions, whereas larger values capture more localized relationships. The epsilon (ϵ) parameter, which defines the width of the insensitive zone around the regression line, was set to 0.01 to ensure high sensitivity to small prediction errors. The hyperparameters were selected via grid search cross-validation (GridSearchCV) on the training data to optimize model performance metrics such as the R^2 score and mean absolute percentage error (MAPE), ensuring a balance between underfitting and overfitting.

2.3.2. An Overview of the CNN-BiLSTM Hybrid Model

Convolutional Neural Networks (CNNs) are widely employed in time series analysis for their ability to capture temporal relationships through layered feature extraction mechanisms [21]. The model architecture begins with convolutional layers, where filters slide across the input data to identify significant local patterns. These initial features are then refined by pooling operations, which reduce dimensionality through downsampling—retaining essential information while minimizing redundancy. The condensed representations are passed to fully connected layers that apply nonlinear transformations, converting spatially encoded inputs into abstract representations suitable for prediction. By progressively adjusting weights during training, CNNs learn to associate input sequences with target outcomes, enabling accurate forecasting and pattern recognition in temporally dynamic datasets.

In CNNs designed for time series analysis, kernel size serves as a pivotal hyperparameter. It determines the spatial extent of the filter that traverses the input sequence. Larger kernels are advantageous for capturing extended temporal correlations and overarching structural patterns, while smaller kernels excel at isolating short-term, granular features. The optimal kernel size is contingent upon the complexity of the time series and the specific characteristics intended for extraction. The output size of a convolutional layer is commonly calculated using the following expression [22]:

$$\text{output size} = \left\lfloor \frac{\text{input size} - \text{filter size} + 2 \times \text{padding}}{\text{stride}} + 1 \right\rfloor \quad (5)$$

Within this framework, the input size corresponds to the length of the time series data, while the filter size specifies the scope of the convolutional kernel that processes the sequence. Padding refers to the addition of zero values at the edges of the input, which helps maintain the original spatial dimensions during convolution. The stride determines the step size at which the kernel shifts along the sequence. Enlarging the kernel or increasing

padding expands the receptive field, enabling the network to learn more extensive temporal relationships. In contrast, a larger stride compresses the output by skipping over more data points during the convolutional pass.

Bidirectional Long Short-Term Memory (BiLSTM) networks incorporate two LSTM layers operating in opposite temporal directions—one traversing the input sequence forward and the other backward. This bidirectional structure enables the model to simultaneously capture information from preceding and succeeding time steps, enhancing its ability to model dependencies where future context contributes to predictive accuracy [23]. The architecture consists of key components: an input layer that receives sequential data, hidden layers composed of LSTM units responsible for updating temporal representations, and memory cells that store long-range dependencies. These cells facilitate the retention of historical patterns while assimilating forward-looking signals. The final output layer synthesizes the dual hidden states to produce forecasts for subsequent time points in the sequence.

Accurate time series forecasting using BiLSTM models hinges on careful tuning of key hyperparameters, such as the number of LSTM units, network depth, hidden layer dimensions, dropout rate, and learning rate. These settings play a crucial role in shaping the model's learning dynamics and its capacity to generalize effectively across unseen data [24]. To identify optimal configurations, practitioners often employ systematic search techniques like grid search and random search, which balance computational cost with predictive accuracy. A more advanced architecture integrates CNNs with BiLSTM layers, forming a CNN-BiLSTM hybrid that exploits both spatial and temporal feature hierarchies [25]. In this framework, CNN layers first extract salient patterns from the raw time series, which are subsequently processed by BiLSTM units operating bidirectionally to capture contextual dependencies from both past and future time steps. This synergistic design enhances both predictive precision and robustness, making it particularly suitable for tasks such as time series classification and forecasting. Figure 4 presents the structural configuration of the model applied to lake water level prediction.

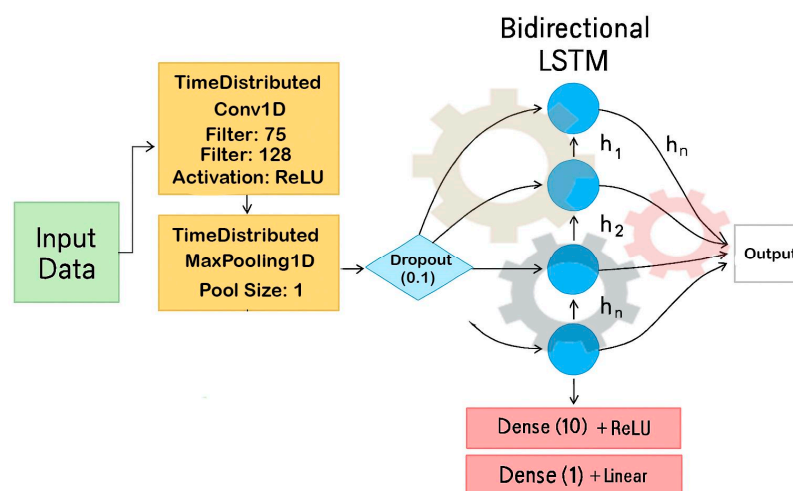


Figure 4. The structure of the CNN-BiLSTM model used in this study.

In CNN-BiLSTM architectures for time series modeling, a range of hyperparameters critically influence predictive performance. Within the CNN module, essential settings include the number and size of convolutional filters, kernel dimensions, and pooling configurations—all of which govern the network's capacity to extract spatial representations. For the BiLSTM component, key parameters encompass the count of hidden units, the number of stacked BiLSTM layers, and the dropout rate, which helps prevent overfitting. Additional factors such as learning rate, batch size, and the structure of subsequent dense

layers also shape the model's learning behavior. Other impactful hyperparameters include the activation function (commonly ReLU in convolutional layers), the choice of optimizer (often Adam), the input sequence length or window size, and regularization techniques like dropout and batch normalization to improve generalization and stability.

The CNN-BiLSTM model was constructed using the Keras Sequential API. The convolutional component consists of two Conv1D layers, the first with 75 filters and the second with 128 filters, both using a kernel size of 1 and the ReLU activation function. These layers automatically extract spatial and temporal patterns from the input sequences by learning local dependencies between input parameters. After the convolutional feature extraction, a MaxPooling1D layer with a pool size of 1 was used to slightly reduce data dimensionality while retaining critical features. The output was then passed into a Bidirectional LSTM layer with 64 units, which processes information in both forward and backward directions, allowing the model to capture long-term dependencies and temporal correlations in dam inflow behavior. To prevent overfitting, a Dropout layer with a dropout rate of 0.1 was included, randomly disabling a small portion of neurons during training. The extracted temporal features were then fed into a Dense layer with 10 neurons and ReLU activation, followed by a Dense (1) output layer with linear activation to predict the continuous inflow values. The model was compiled using the Adam optimizer with a learning rate of 0.00039, which was selected through a Random Search hyperparameter tuning procedure. The model was trained to minimize the mean squared error (MSE) loss function, which measures the average squared difference between observed and predicted inflow values. The training process was carried out for 131 epochs with a batch size of 56, and 20% of the training data were used for validation to monitor generalization performance. This CNN-BiLSTM structure effectively combines the feature extraction ability of CNN with the temporal learning capability of BiLSTM, enabling the model to learn complex spatiotemporal patterns governing dam inflow variations accurately.

2.4. Quantum Machine Learning Model

An Overview of the HQNN Hybrid Model

The Hybrid Quantum Neural Network (HQNN) architecture seamlessly combines classical and quantum computational components within a unified framework. It begins with a classical embedding layer that processes the input data and extracts relevant features. These features are then transformed into quantum states, which serve as input to the quantum layer. This layer comprises parameterized quantum circuits, incorporating elements such as feature mapping and entangling gates, and functions as the final stage for classification or regression tasks [26]. The model undergoes end-to-end training, simultaneously refining both classical and quantum parameters to improve predictive accuracy.

This integrated framework harnesses the complementary capabilities of classical and quantum computing. The classical embedding layer is responsible for extracting nonlinear patterns and contextual information from the input data, while the quantum layer utilizes principles such as superposition and entanglement to capture intricate dependencies that may elude conventional neural networks [27]. These architectures typically incorporate variational quantum circuits composed of several qubits (commonly 4 or 9), and employ modules like PennyLane's 35QAOAEmbedding and StronglyEntanglingLayer to encode and manipulate quantum states. The quantum outputs are measured and translated into predictions, with the entire model trained jointly through classical optimization techniques.

A typical mathematical formulation of the HQNN hybrid model involves three main components: data encoding (feature map) F , a parameterized quantum circuit (ansatz) $A(\vec{\theta})$, and measurement of an observable $\hat{O}$. Classical input data $\vec{x}$ is first encoded

into an n -qubit quantum state $|\varphi(\vec{x})\rangle$ by a feature map F . The quantum circuit $A(\vec{\theta})$, parameterized by $\vec{\theta}$, acts on this state to produce the output quantum state $|\psi(\vec{x}, \vec{\theta})\rangle$.

The prediction or output of the HQNN is obtained by measuring an observable operator $\hat{O}$ on the output state, represented as the expectation value [28]:

$$\langle \hat{O} \rangle = \langle \psi(\vec{x}, \vec{\theta}) | \hat{O} | \psi(\vec{x}, \vec{\theta}) \rangle \quad (6)$$

This expectation value serves as the analog to the output of a classical neural network layer.

Training the HQNN entails optimizing the parameters $\vec{\theta}$ to minimize a loss or cost function, often defined by the difference between predicted and true labels or energies, using classical optimization methods. The model typically incorporates nonlinearities induced by measurement and repetition of quantum-classical layers [29]. The cost function f for energy calculations can be written:

$$f = \sum_j \langle \phi_j | H_j | \phi_j \rangle \quad (7)$$

where $|\phi_j\rangle$ is the quantum state output for input j , and H_j is the associated Hamiltonian operator.

In this study, a HQNN model was developed for predicting dam inflow by integrating quantum computation with classical deep learning. The quantum circuit was implemented using the PennyLane library with the default.qubit state vector simulator as the backend. In this study, the number of qubits was chosen to match the number of input features in the training dataset (e.g., 6 features mapped to 6 qubits) for simplicity of design, while amplitude embedding was used to encode the normalized classical feature vector into the quantum state, a standard practice in HQNNs that maps each input dimension to a qubit for efficient quantum state preparation without additional ancillary qubits. This one-to-one correspondence preserves the dimensionality of the input data in the quantum Hilbert space, allowing the variational quantum circuit to process feature relationships through quantum gates while minimizing circuit depth and simulation overhead on PennyLane's default qubit simulator.

Input data were encoded into quantum states using the Angle Embedding technique, and four Strongly Entangling Layers were applied to introduce entanglement among the qubits, enabling the circuit to capture complex dependencies among hydrological parameters. The output of the quantum circuit was the expectation value of the Pauli-Z operator on the first qubit, producing a single quantum feature per sample that represents a nonlinear transformation of the classical inputs. A custom QuantumLayer was constructed in the TensorFlow/Keras environment to embed the quantum circuit into the deep learning structure. The quantum weights, initialized from a normal distribution with the shape (4, num_features, 3), were treated as trainable parameters and updated jointly with the classical weights during training. Each input sample, represented in float64 precision, was passed through the quantum circuit using TensorFlow's map_fn, generating a scalar quantum output. This quantum feature was then reshaped into a column vector and fed into the subsequent classical layers. The QuantumLayer thus acts as a bridge between the quantum and classical components, serving as a high-dimensional feature extractor that enhances the model's ability to represent complex nonlinear patterns in dam inflow behavior. The classical part of the HQNN included two Dense layers with 92 and 7 neurons, respectively, both activated using the ReLU function, followed by a final Dense (1) layer with linear activation to predict inflow values. The model was compiled using the Adam optimizer with a learning rate of 0.0079, which was selected through a Random Search hyperparameter

tuning procedure. The model was trained to minimize the mean squared error (MSE) loss function, which quantifies the average squared difference between observed and predicted values. Training was conducted for 182 epochs with a batch size of 69, and an EarlyStopping callback with patience = 100 was employed to prevent overfitting and retain the best weights. The hybrid design allows the model to utilize quantum entanglement for improved feature encoding while maintaining the generalization strength of deep neural networks, resulting in enhanced prediction accuracy for dam inflow compared to purely classical approaches. A schematic representation of the quantum algorithm's workflow is provided in Figure 5.

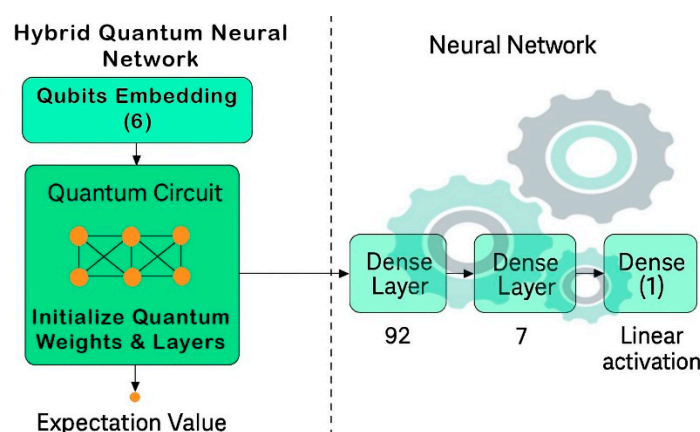


Figure 5. Structure and parameters of the HQNN model used in this study.

The HQNN model stands out in time series prediction for its combination of classical deep learning components (such as LSTM, CNN) with quantum neural networks. This hybrid design allows classical parts to extract temporal dependencies effectively, while the quantum component boosts feature representation through quantum properties like superposition and entanglement [30]. The quantum circuits expand the feature space nonlinearly, capturing complex patterns and subtle correlations in data that classical methods might miss. This results in improved accuracy and robustness, especially in handling nonlinear, noisy, or highly volatile time series. Another important feature of HQNN is the joint or sequential optimization of classical and quantum modules that enhances generalization and model efficiency. The quantum part typically requires fewer parameters while increasing model expressiveness, facilitating better pattern recognition with reduced extreme prediction errors [31]. By leveraging the strengths of both classical and quantum approaches, HQNN models yield more stable and reliable forecasts in complex time series tasks, such as financial markets, network traffic, or hydrological processes, where long-range temporal dependencies and intricate data relationships are crucial.

In this study, hyperparameter tuning was conducted using the Random Search algorithm. Unlike Grid Search, which systematically evaluates every possible parameter combination, Random Search selects configurations randomly from a predefined space, offering more efficient coverage—especially in high-dimensional settings where exhaustive search becomes computationally prohibitive. To improve result robustness, the training process was repeated across 10 independent runs. Model complexity reflects a model's capacity to approximate diverse functions. While architectures such as deep neural networks possess high representational power, they are prone to overfitting, often capturing noise instead of meaningful patterns. To balance underfitting and overfitting, the model architecture, depth, and parameter count were varied systematically. Each machine learning model underwent 50 Random Search iterations to explore the hyperparameter landscape while preserving computational tractability. For reproducibility, a fixed random seed was initial-

ized before each search and training cycle. During training, early stopping was employed based on validation loss: if no improvement occurred over 10 consecutive epochs, training was terminated to reduce computational overhead and promote generalization.

2.5. Model Evaluation Metrics

Assessing model performance is a fundamental aspect of disciplines such as machine learning, statistics, and data-driven analysis. Quantitative metrics offer a standardized means of evaluating how effectively a model predicts or classifies outcomes. Establishing consistent evaluation criteria enables researchers to systematically compare different modeling approaches and select the one that best aligns with the characteristics of their dataset and research objectives. This evaluative process enhances the reliability of model selection and supports informed decision-making. In the present study, five core metrics were employed to gauge the accuracy and robustness of the model: Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Percentage Error (MAPE), and Nash–Sutcliffe Efficiency (NSE). These indicators were instrumental in quantifying both the precision and consistency of the model's predictive outputs.

Root Mean Square Error (RMSE) quantifies the average magnitude of prediction errors by comparing forecasted values to actual observations, with a stronger penalty assigned to larger deviations [24]. Lower RMSE values indicate greater predictive accuracy and fewer substantial errors. The coefficient of determination (R^2) measures the proportion of variance in the observed data that is explained by the model, ranging from 0 to 1; higher values reflect a closer fit to the underlying data structure. Mean Absolute Percentage Error (MAPE) calculates the average percentage difference between predicted and actual values, offering an intuitive, scale-independent measure of accuracy—smaller MAPE values correspond to more precise forecasts [32]. Nash–Sutcliffe Efficiency (NSE) assesses the agreement between predicted and observed time series, particularly in hydrological modeling. An NSE of 1 denotes perfect predictive performance, values between 0 and 1 suggest acceptable accuracy, and negative values imply that the model performs worse than using the mean of the observed data as a predictor.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_{pi} - X_{oi})^2} \quad (8)$$

$$R^2 = \frac{\sum_{i=1}^N (X_{oi} - \bar{X}_o)(X_{pi} - \bar{X}_p)}{\sum_{i=1}^N (X_{oi} - \bar{X}_o)^2 \cdot \sum_{i=1}^N (X_{pi} - \bar{X}_p)^2} \quad (9)$$

$$MAPE = 100 \times \frac{1}{N} \sum_{i=1}^N \left| \frac{X_{oi} - X_{pi}}{X_{oi}} \right| \quad (10)$$

$$NSE = 1 - \left[\frac{\sum_{i=1}^N (X_{oi} - X_{pi})^2}{\sum_{i=1}^N (X_{oi} - \bar{X}_o)^2} \right] \quad (11)$$

where X_{pi} and X_{oi} are the predicted and observed values, $\bar{X}_o$ and $\bar{X}_p$ are the mean observed and predicted values, respectively, and N is the total number of data points.

3. Results and Discussion

Mean temperature, mean humidity, and precipitation are three of the most influential climatic factors for predicting dam inflow because they collectively determine the processes of evapotranspiration, snowmelt, and runoff generation within a watershed. Each plays a distinct hydrological role that affects total water availability and inflow variability to reservoirs. Mean temperature affects dam inflow primarily through its influence on evaporation

and snowmelt. Higher temperatures accelerate snow and glacier melt in mountainous basins, leading to short-term inflow increases, while simultaneously enhancing evapotranspiration, which reduces surface runoff and long-term inflow volumes. Studies, such as those by Han et al. [33], have shown that rising temperatures lead to overall declines in inflow, even when precipitation increases, due to stronger evaporation losses. Temperature also plays a role in modifying soil moisture, infiltration capacity, and antecedent flow conditions—all key predictors in machine learning inflow models.

Mean or relative humidity is closely linked to both evapotranspiration control and precipitation formation. Higher humidity reduces potential evapotranspiration, sustaining more effective runoff generation and maintaining baseflow contributions to inflows. Conversely, lower humidity enhances water loss from soil and vegetation surfaces, decreasing available inflow. Therefore, humidity serves as a significant balancing factor modulating evaporation–precipitation interactions in inflow regimes [34].

Precipitation is the dominant direct driver of dam inflow variability. It determines short-term runoff peaks and seasonal inflow rhythms, particularly during the wet season when catchment saturation is high. Machine learning and statistical studies consistently identify precipitation as the most sensitive and highly correlated variable with inflow magnitude across different temporal scales (daily to monthly). Inflow prediction models using only precipitation already capture significant inflow variability, but combining precipitation with temperature and humidity markedly improves accuracy for both dry and wet periods.

Also, three-month lags represent the time it typically takes for upstream precipitation in the Kura and Aras basins to translate into measurable discharge at Mughan and Mil collectors. Water that infiltrates or accumulates in irrigation zones and reservoirs (such as Takhtakorpu and Shamkirchay) traverses multiple conveyance systems—open canals, drainage channels, and groundwater channels—before reaching downstream storage, such as Mil Mughan. Thus, including a 3-month lag (1-, 2-, and 3-month lag) in models helps capture subsurface and delayed surface runoff patterns that significantly affect inflow predictability. Table 2 presents the categorization of scenarios assessed for forecasting dam inflow using three distinct models: SVR, CNN-BiLSTM, and HQNN.

Table 2. Division of the evaluated scenarios to predict the inflow of the dam with three models: SVR, CNN-BiLSTM, and HQNN.

Scenario	Input	Model	Output
1	T_{mean} , H_{mean} , PP, First Lag, Second Lag, Third Lag	SVR, CNN-BiLSTM, HQNN	Dam Inflow
2	First Lag, Second Lag, Third Lag	SVR, CNN-BiLSTM, HQNN	Dam Inflow

Inflow prediction for Mil Mughan Dam was carried out using two distinct input scenarios designed to assess the influence of meteorological variables and antecedent inflow data on forecasting accuracy. The first scenario integrated mean temperature, mean humidity, precipitation, and inflow values lagged by three months, representing a comprehensive dataset that reflects both climatic conditions and historical inflow trends. The second scenario, in contrast, utilized only three-month lagged inflow values, aiming to evaluate the models' capacity to predict dam inflow based strictly on observed antecedent hydrological patterns, which is vital when meteorological data are unavailable or uncertain.

The comparative results in Table 3 demonstrate that the HQNN model provided the highest predictive accuracy for the Mil Mughan Dam inflow across both input scenarios, notably with meteorological and lagged inflow variables included. In the testing step, it achieved an R^2 of 0.915, an RMSE of 37.318 MCM, an NSE of 0.908, and an MAPE of 8.343%

when leveraging all input features, indicating superior goodness-of-fit, low prediction error, and strong efficiency compared to observed inflows. When restricted to lagged inflow values alone, HQNN remained robust, only slightly declining in performance, highlighting its advanced capability to model nonlinear and temporal relationships in hydrological forecasting. The CNN-BiLSTM model demonstrated good predictive capabilities, particularly when meteorological variables were included. This can be attributed to its proficiency in modeling both temporal dynamics and spatial correlations. Performance metrics $R^2 = 0.867$, $RMSE = 46.506$ MCM, $NSE = 0.858$, and $MAPE = 10.795\%$ positioned it between HQNN and SVR, outperforming the latter but falling short of the former. Notably, its accuracy showed greater sensitivity to the absence of meteorological inputs. When the model relied solely on lagged inflow data, its effectiveness diminished, underscoring the value of incorporating external predictors in deep learning architectures. The SVR model delivered reliable, albeit consistently lower, predictive outcomes in both scenarios, with a maximum R^2 of 0.846 and the largest RMSE and MAPE values. SVR's kernel-based structure is competent for basic regression but less effective in modeling the highly nonlinear and complex dynamic patterns evident in dam inflow data. The comparison validates that hybrid and deep learning models—particularly those enhanced with quantum optimization—are favored for operational water inflow forecasting due to their capacity for processing multi-source data and capturing subtle relationships essential for effective reservoir management.

Table 3. Evaluation criteria for three SVR, CNN-BiLSTM, and HQNN models.

Dam	Scenario	Model	Testing				Training			
			R^2	RMSE (MCM)	NSE	MAPE (%)	R^2	RMSE (MCM)	NSE	MAPE (%)
Mil Mughan	1	HQNN	0.915	37.318	0.908	8.343	0.932	33.543	0.944	5.832
		CNN-BiLSTM	0.867	46.506	0.858	10.795	0.884	43.175	0.873	7.945
		SVR	0.846	52.372	0.821	12.772	0.861	50.056	0.853	10.034
	2	HQNN	0.855	48.561	0.845	9.979	0.879	46.463	0.867	7.634
		CNN-BiLSTM	0.810	56.126	0.793	11.456	0.841	53.946	0.814	9.435
		SVR	0.801	60.336	0.761	12.901	0.824	58.320	0.792	10.852

Figure 6 presents scatter plots with marginal histograms illustrating the correlation between actual and predicted dam inflow for the Mil Mughan Dam across two scenarios using three models: HQNN, CNN-BiLSTM, and SVR. Each scatter plot shows individual prediction points, with the red diagonal line representing perfect agreement (where predicted equals actual inflow). Points closer to this line indicate higher prediction accuracy. The distribution of points around the line reflects the model's predictive performance, with tighter clustering signifying better agreement. Marginal histograms on the top and right of each scatter plot display the frequency distribution of actual and predicted inflows, respectively. A similar shape and spread in these histograms imply that the model successfully captures the overall inflow distribution pattern.

The marginal histogram profiles for HQNN are highly aligned between actual and predicted values, supporting this model's ability to reproduce the observed inflow distribution, including peaks and variability. This visual evidence corroborates the model's outstanding R^2 of 0.915, outperforming the other methods. For CNN-BiLSTM, the scatter in Scenario 1 remains satisfactorily close to the diagonal, but there is a slightly broader dispersion, especially for higher inflow values. The corresponding marginal histograms are similar in shape, confirming that the model captures the general character of the inflow, but the increased spread and a few more outliers highlight that its predictions in some high- or low-inflow instances are less precise compared to HQNN. When the input is restricted to lagged

values only in Scenario 2, the performance of CNN-BiLSTM declines, with points drifting further from the diagonal and increased vertical scatter, revealing a greater sensitivity to the exclusion of meteorological features. SVR, as shown in both scenarios, consistently exhibits the widest spread of points from the diagonal and less overlap between actual and predicted marginal histograms. This indicates systematic under- or over-estimation in various ranges and reflects insufficient modeling of nonlinear dependencies or temporal patterns. Its scatter plots often underestimate at higher inflow ranges and overestimate at the lower ranges, contributing to its lower R^2 and higher error statistics. The contrast between models is especially pronounced in Scenario 2, where SVR's predictive limitations are magnified, underscoring the superior generalizability and resilience of HQNN and, to a lesser degree, CNN-BiLSTM for operational dam inflow forecasting, particularly when limited data inputs are available.

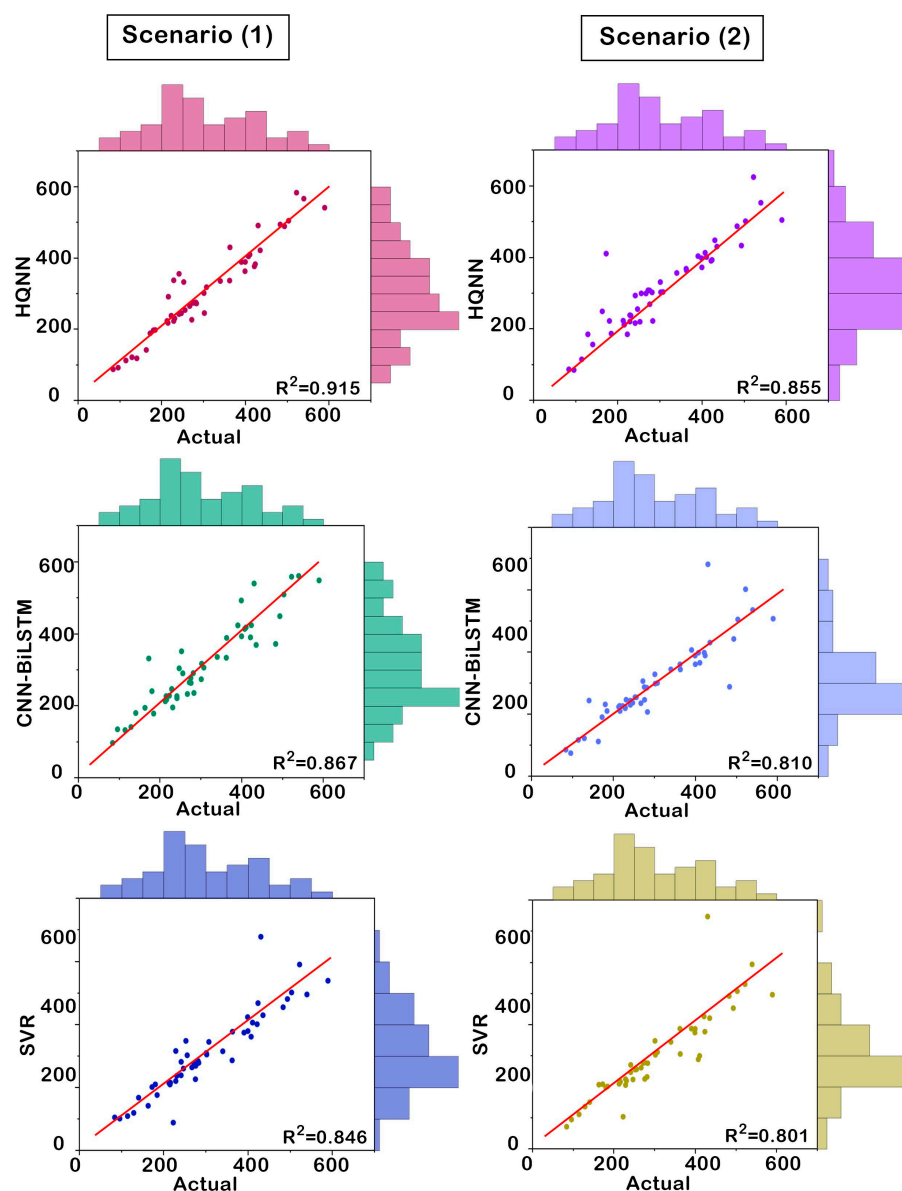


Figure 6. Scatter plots with marginal histograms for comparing observed and predicted dam inflow during testing using three models: SVR, CNN-BiLSTM, and HQNN.

The time series plots in Figure 7 compare the actual monthly inflow (in million cubic meters, MCM) with predicted inflow values from the HQNN, CNN-BiLSTM, and SVR models under two different scenarios for Mil Mughan Dam. In Scenario 1, which includes

mean temperature, humidity, and precipitation along with three-month lagged inflow values, the HQNN prediction closely traces the actual inflow pattern with minimal deviation, preserving both the peaks and low-flow periods effectively. This tight alignment indicates the model's excellent ability to capture complex nonlinear relationships and dynamic inflow fluctuations using rich meteorological and lagged data. CNN-BiLSTM's predictions also reflect the general trends well but show slightly more divergence at some peaks or dips, which manifests as fluctuating overlaps and minor lags behind actual values. The SVR model generally captures the inflow dynamics but visibly underestimates peak inflows and exhibits more variability in deviation, highlighting less precision and sensitivity to hydrological extremes. Under Scenario 2, which only relies on lagged inflow values, the prediction accuracy for all models generally decreases. HQNN, while still producing close approximations of the actual inflow, shows more lag and variation around peak flows, indicating its reliance on meteorological data for optimal precision. CNN-BiLSTM follows a similar declining trend, with increasing mismatches during sudden inflow changes. SVR's predictions become less aligned overall, with noticeable departures from actual values across several months, especially during high inflow events. This highlights its limitations when external climate features are unavailable.

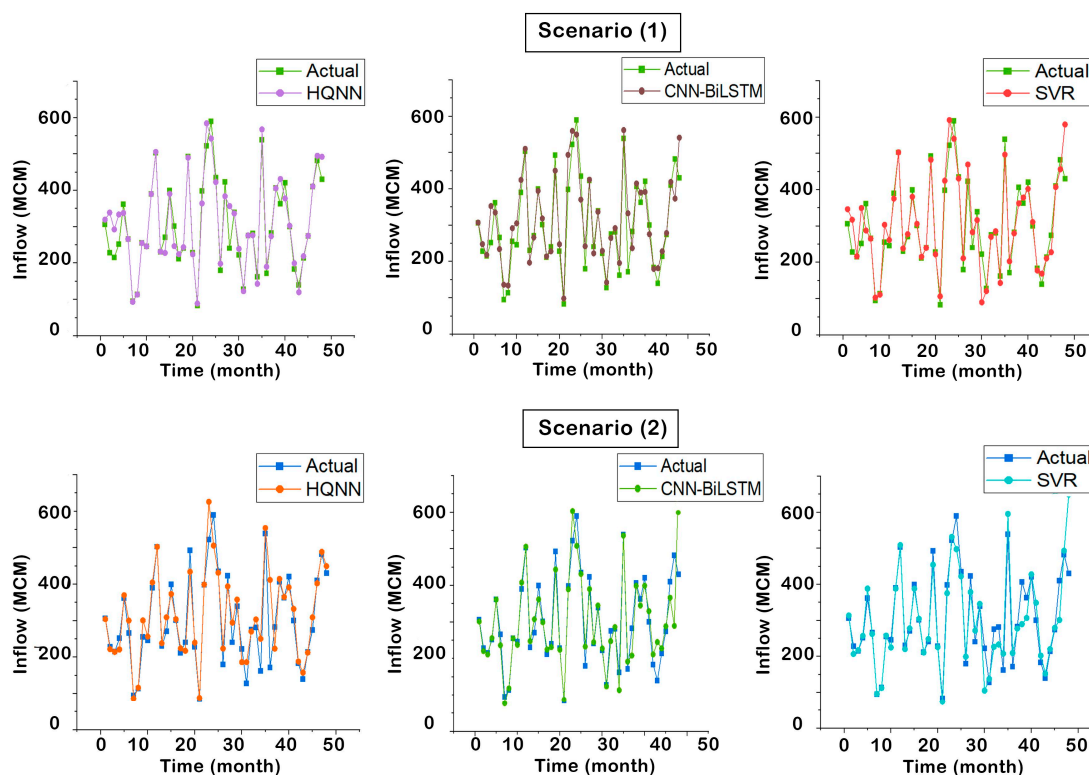


Figure 7. Time series plots for comparing observed and predicted dam inflow during testing using three models: SVR, CNN-BiLSTM, and HQNN.

Overall, these time series confirm that HQNN's hybrid and quantum-enhanced architecture lends it superior predictive fidelity, particularly when utilizing comprehensive and multi-source inputs. CNN-BiLSTM is a competent alternative but is more dependent on richer input data to maintain accuracy, while SVR serves as a baseline with consistent but comparatively coarser approximations. These visualizations complement numerical evaluations and substantiate HQNN as the most reliable model for dam inflow prediction in both data-rich and data-limited contexts.

The performance of each model was visualized using overlapping ridgeline plots in Figure 8, where the distribution of predicted inflow values was compared to the actual

inflow distribution. In Scenario 1, where meteorological variables are included alongside lagged inflow data, the HQNN and CNN-BiLSTM models demonstrate good performance for the actual inflow distribution. The HQNN's curve nearly mirrors the curve of actual inflow, indicating its superior ability to capture nonlinear relationships influenced by climate dynamics. CNN-BiLSTM, with its curve, also shows strong alignment, benefiting from its dual capacity to extract spatial and temporal features. In contrast, SVR under Scenario 1 exhibits a broader and less concentrated curve, deviating from the actual inflow distribution. This dispersion implies that SVR struggles to model the complex interactions between meteorological inputs and inflow behavior. While SVR is known for robustness in simpler regression tasks, its limitations become evident when tasked with high-dimensional, nonlinear hydrological forecasting.

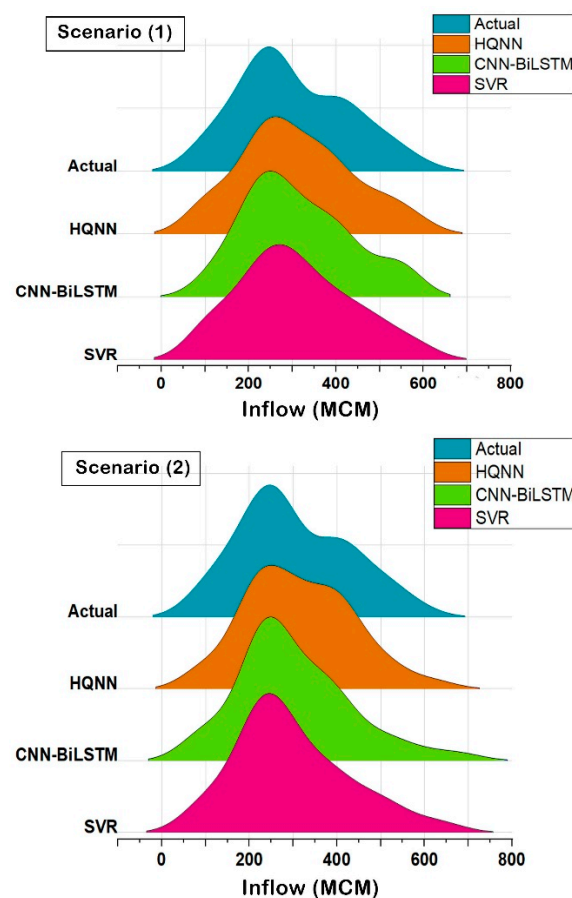


Figure 8. Ridgeline plots for comparing observed and predicted dam inflow using three models: SVR, CNN-BiLSTM, and HQNN.

Scenario 2, which relies solely on lagged inflow data, exposes the models to a more constrained learning environment. Here, HQNN maintains a relatively strong performance, indicating its resilience in capturing temporal dependencies even without external variables. However, CNN-BiLSTM shows a noticeable drop in precision, with its curve diverging from the actual inflow distribution. This suggests that HQNN's strength lies in integrating heterogeneous data sources—without them, its predictive edge diminishes. SVR, again, underperforms, with its distribution remaining diffuse and misaligned, reinforcing its limited adaptability to autoregressive-only inputs.

In addition to the visual comparison in the ridgeline plots (Figure 8), distribution similarity was evaluated using Jensen–Shannon distance (JS), and Wasserstein (Earth Mover's) distance between the predicted and observed inflow distributions for each model and scenario. Lower values of these metrics indicate closer agreement between the model's

distribution and the actual inflow, complementing the qualitative assessment from the ridgeline plots. Across both scenarios, the HQNN consistently yielded the smallest JS and Wasserstein distances to the actual inflow distribution, followed by CNN-BiLSTM, while SVR exhibited substantially larger distances, confirming the superior distributional fidelity of the HQNN observed visually in Figure 8.

Inclusion of climate and humidity predictors provides demonstrable improvements in model performance, especially in reservoirs subject to high seasonal variability or complex climate interactions. The clear separation in error metrics between HQNN and SVR further underscores the importance of advanced, data-driven approaches for accurate hydrological forecasting and sustainable dam operation. The adoption of deep learning and hybrid models has been widely recommended in the recent literature for their ability to handle nonlinearities and multivariable dependencies in reservoir inflow series. Thus, the presented approach and findings not only support operational decision-making for the Mil Mughan Dam but also offer a transferable framework relevant to other dam-regulated systems globally.

The quantum hybrid neural network (HQNN) outperformed classical machine learning counterparts largely due to its enhanced optimization capabilities and complex feature representation. Quantum-inspired algorithms used in HQNN improve the model's ability to find global optima during training, effectively navigating intricate, high-dimensional data landscapes such as those arising from meteorological and hydrological time series. This enables HQNN to avoid common pitfalls like parameter trapping and local minima, leading to more stable and generalizable predictions. Furthermore, the hybrid structure of HQNN integrates both quantum state encoding and deep learning, allowing for improved modeling of long-term dependencies and interactions among lag and climate variables, which are crucial for accurate dam inflow forecasting under varied environmental conditions.

The importance of QML in hydrology lies in its ability to handle nonlinear, high-dimensional problems and uncertain data—limitations that conventional models often face. Results confirm the high performance of Quantum Machine Learning (QML) models in hydrological prediction, consistent with recent advancements in the field. For instance, Khemapatapan and Thepsena [35] demonstrated the effectiveness of quantum classifiers on real weather data from the Pa Sak Jolasid Dam, where the Quantum Support Vector Machine (QSVM) achieved an accuracy of 85.3%, outperforming both the Variational Quantum Classifier (VQC) and Quantum Neural Network (QNN). Similarly, Fellner [36] highlighted the advantages of quantum reservoir computing and QNNs in time series prediction, noting their ability to accelerate training and reduce generalization errors compared to classical deep learning models. Also, findings resonate with those of Zhen and Bărbulescu [37], who reported that QNNs surpassed classical models such as LSTM, Backpropagation Neural Network (BPNN), and CNN-LSTM in predicting river discharge, particularly under extreme conditions. Their QNN model achieved an R^2 of 84.36%, with notably low MSE and MAE values. Furthermore, the application of Variational Quantum Regression (VQR) by Zhen and Bărbulescu [38] demonstrated enhanced predictive accuracy over both classical and hybrid AI models, reinforcing the practical viability of quantum approaches in hydrological modeling. Moreover, it demonstrated remarkable capability in predicting monthly discharge maxima, highlighting its potential as a robust and reliable approach for hydrological forecasting in complex, nonlinear, and highly variable data environments. Findings from these studies indicate that the quantum algorithm enhances the performance of classical machine learning models and boosts their reliability—aligning well with the outcomes observed in this investigation.

4. Conclusions

Accurate dam inflow prediction is a critical component of modern water resource management, essential for flood mitigation, hydropower optimization, and ensuring water security. This study underscores the significant potential of QML, particularly through its enhanced optimization capabilities and superior feature representation in high-dimensional spaces, to advance the state-of-the-art in hydrological forecasting. Quantum-inspired algorithms excel at navigating complex, nonlinear data landscapes, effectively avoiding local minima and capturing intricate temporal dependencies that often challenge classical models.

The research was conducted using a 14-year dataset from the transboundary Mil Mughan Dam on the Aras River, a vital hydroelectric and irrigation resource for Azerbaijan and Iran. To perform a comprehensive comparative analysis, three distinct machine learning architectures were implemented and evaluated under two forecasting scenarios: a multivariate framework incorporating meteorological variables (mean temperature, mean humidity, precipitation) alongside three-month lagged inflow data, and a univariate framework relying solely on the historical inflow lags. The models considered were SVR, a hybrid CNN-BiLSTM model, and a HQNN. The HQNN demonstrated superior performance, achieving a prediction accuracy of approximately 92%. It successfully explained over 91% of the variance in the dam's inflow data. The classical deep learning model (CNN-BiLSTM) also performed strongly with an accuracy of around 89–90%, while the Support Vector Regression (SVR) model served as a solid baseline with approximately 87% accuracy.

Despite the results, this study has certain limitations that improved upon for future research. A primary constraint is its focus on a single case study. The model was developed and validated specifically for the Mil Mughan Dam, which features a semi-arid climate. Consequently, the generalizability of the HQNN to dams in significantly different hydrological regimes—such as snowmelt-dominated alpine reservoirs or tropical watersheds with monsoon climates—remains unverified and requires further investigation. Furthermore, the current implementation operates within a simulated quantum environment. While this demonstrates the conceptual value of quantum-inspired algorithms, it does not leverage the computational advantages of actual quantum hardware, nor does it account for real-world quantum challenges like noise and decoherence. Finally, the model's input scope is limited to monthly data and a core set of meteorological variables, potentially overlooking the finer temporal dynamics of inflow events and the influence of other factors like snowpack or large-scale climate indices.

To address these limitations and advance the field, several key directions for future work are proposed. First, it is essential to validate and benchmark the HQNN framework across multiple dams in diverse geographical and climatic settings. This would rigorously test its transferability and robustness. Concurrently, as quantum hardware matures, a critical goal will be to transition the HQNN model from classical simulation to execution on real Noisy Intermediate-Scale Quantum (NISQ) processors, exploring tangible quantum advantage in hydrological forecasting. To enhance predictive power, future research should also focus on integrating multi-scale and novel data sources, such as high-resolution remote sensing data for soil moisture and snow cover, as well as climate teleconnection indices like ENSO. Also, time series cross-validation techniques, such as TimeSeriesSplit or blocked cross-validation, tailored to hydrological data can be explored to verify model robustness on limited samples and reduce the risks of overfitting.

Beyond model performance, future efforts will aim to improve the operational utility and transparency of the HQNN. Developing Explainable AI (XAI) techniques tailored for quantum-classical hybrids is crucial for building trust with water managers by clarifying

the model's decision-making process. The ultimate objective is to evolve this research into a robust operational forecasting system. This involves creating automated pipelines for real-time data ingestion and, importantly, enhancing the model to provide probabilistic forecasts with uncertainty quantification. Such advancements will empower dam operators with reliable, risk-informed insights, solidifying the role of quantum machine learning in building resilient water management systems for the future.

The computational complexity of the proposed models, particularly the HQNN, poses substantial challenges for practical implementation. The sophisticated nature of HQNN, alongside CNN-BiLSTM and SVR, requires a high level of expertise in advanced artificial intelligence (AI) techniques, including quantum machine learning and deep learning architectures. However, personnel at dam facilities, such as those operating the Mil Mughan Dam, may lack familiarity with these cutting-edge methodologies, limiting their ability to effectively deploy and maintain such models in operational settings. This complexity could hinder real-time applications, where rapid and reliable inflow predictions are critical for public safety, water resource management, and hydropower generation. Furthermore, the computational demands of these models may necessitate advanced infrastructure, which may not be readily available in resource-constrained environments, further exacerbating implementation challenges.

Conversely, a significant advantage of this study lies in its regional and international relevance, stemming from the location of the Mil Mughan Dam on the Aras River, a transboundary watercourse forming part of the border between Azerbaijan and Iran in the South Caucasus. This geopolitical context underscores the potential for this work to foster cross-border scientific collaboration and support joint management of shared water resources. The robust performance of the HQNN model in both data-rich and data-limited scenarios ($R^2 = 0.855$, RMSE = 48.56 MCM in Scenario 2) provides a reliable framework for inflow forecasting that can enhance cooperative decision-making, ensuring sustainable water allocation and hydropower operations across national boundaries. This dual advantage—high predictive accuracy and the facilitation of international collaboration—positions the study as a valuable contribution to both hydrological modeling and transboundary water resource management.

Author Contributions: Conceptualization, E.A. and M.T.S.; methodology, resources, and software, E.A.; validation, and data curation, M.T.S.; formal analysis, investigation, S.S.; writing—original draft preparation, E.A. and S.S.; writing—review, editing, visualization, supervision, and project administration, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy. The data used in this study were obtained from the Regional Water Company of East Azerbaijan Province, Iran. Data can be sent upon request and with official permission from this company.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Farzad, R.; Lotfi, V. Modal analysis of concrete arch dam–reservoir–massed foundation system in frequency domain. *Proc. Inst. Civ. Eng.-Struct. Build.* **2023**, *176*, 244–256. [\[CrossRef\]](#)
2. Abdi, E.; Taghi Sattari, M.; Milewski, A.; Ibrahim, O.R. Advancements in Hydrological Modeling: The Role of bRNN-CNN-GRU in Predicting Dam Reservoir Inflow Patterns. *Water* **2025**, *17*, 1660. [\[CrossRef\]](#)
3. De, S.; Farzad, R.; Brewick, P.T.; Johnson, E.A.; Wojtkiewicz, S.F. Likelihood level adapted estimation of marginal likelihood for Bayesian model selection. *Comput. Methods Appl. Mech. Eng.* **2025**, *445*, 118141. [\[CrossRef\]](#)

4. Brewick, P.T.; Farzad, R. Hierarchical Bayesian calibration of Bouc–Wen hysteretic models with applications to seismic isolators. *Mech. Syst. Signal Process.* **2025**, *237*, 113021. [\[CrossRef\]](#)
5. Soltani, S.; Valizadeh, J.; Aghdamigargari, M.; Mirzaei Varzeghani, N.; Mozafari, P. A cooperative game model for emergency transportation planning using the internet of things. *J. Intell. Transp. Syst.* **2025**, 1–20. [\[CrossRef\]](#)
6. Noorazar, H.; Brady, M.P.; Savalkar, S.; Kandelati, A.N.; Liu, M.; Beale, P.; McGuire, A.M.; Waters, T.; Rajagopalan, K. Monitoring Double-Cropped Extent with Remote Sensing in Areas with High Crop Diversity. *Plants* **2025**, *14*, 1362. [\[CrossRef\]](#)
7. Hong, J.; Lee, S.; Bae, J.H.; Lee, J.; Park, W.J.; Lee, D.; Kim, J.; Lim, K.J. Development and evaluation of the combined machine learning models for the prediction of dam inflow. *Water* **2020**, *12*, 2927. [\[CrossRef\]](#)
8. Enayati, S.M.; Najarchi, M.; Mohammadpour, O.; Mirhosseini, S.M. Evaluating machine learning models in predicting dam inflow and hydroelectric power production in multi-purpose dams (case study: Mahabad Dam, Iran). *Appl. Water Sci.* **2024**, *14*, 206. [\[CrossRef\]](#)
9. Zhang, S.; Xiang, Z.; Wang, Y.; Xie, S. Short-Term Downstream Water Level Prediction Model for Three Gorges–Gezhouba Cascade Reservoir Operation Based on LSTM Algorithm. In *Proceedings of the 8th International Conference on Water Resource and Environment*; Springer Nature: Singapore, 2022; pp. 223–234.
10. Lee, S.; Kim, J.; Bae, J.H.; Lee, G.; Yang, D.; Hong, J.; Lim, K.J. Development of multi-inflow prediction ensemble model based on auto-sklearn using combined approach: Case study of soyang river dam. *Hydrology* **2023**, *10*, 90. [\[CrossRef\]](#)
11. Latif, S.D.; Ahmed, A.N. Ensuring a generalizable machine learning model for forecasting reservoir inflow in Kurdistan region of Iraq and Australia. *Environ. Dev. Sustain.* **2024**, *26*, 12513–12544. [\[CrossRef\]](#)
12. Deb, D.; Arunachalam, V.; Raju, K.S. Daily reservoir inflow prediction using stacking ensemble of machine learning algorithms. *J. Hydroinform.* **2024**, *26*, 972–997. [\[CrossRef\]](#)
13. Ghosh, A.; Ghosh, S. AI-driven Reverse Engineering of QML Models. In *Proceedings of the 2025 26th International Symposium on Quality Electronic Design (ISQED)*, San Francisco, CA, USA, 23–25 April 2025; pp. 1–7.
14. Niu, W.J.; Feng, Z.K.; Cheng, C.T.; Zhou, J.Z. Forecasting daily runoff by extreme learning machine based on quantum-behaved particle swarm optimization. *J. Hydrol. Eng.* **2018**, *23*, 04018002. [\[CrossRef\]](#)
15. Grzesiak, M.; Thakkar, P. Flood Prediction using Classical and Quantum Machine Learning Models. *Int. J. Comp. Sci. Mob. Appl.* **2024**, *12*, 84–98.
16. Vajpayee, K.; Singh, A.; Shukla, S.; Prakash, S. Optimization of Flood Mitigation Strategies Using Quantum Learning. In *Proceedings of the 2025 11th International Conference on Communication and Signal Processing (ICCSP)*, Osaka, Japan, 28–30 November 2025; pp. 1626–1630.
17. Pasha, N.H.; Zengin, E.; Baylarli, A.T. Sustainability of Water Resources of Azerbaijan, Future Prospects and Problems. *J. Manag. Econ. Ind. Organ.* **2023**, *7*, 24–37.
18. Wolfram Research. Weather Data, Wolfram Language Function. 2008. (Updated 2014). Available online: <https://reference.wolfram.com/language/ref/WeatherData.html> (accessed on 1 December 2024).
19. Awad, M.; Khanna, R.; Awad, M.; Khanna, R. Support vector regression. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*; Apress: Berkeley, CA, USA, 2015; pp. 67–80. [\[CrossRef\]](#)
20. Zhang, F.; O'Donnell, L.J. Support vector regression. In *Machine Learning*; Academic Press: Cambridge, MA, USA, 2020; pp. 123–140. [\[CrossRef\]](#)
21. Swiderski, B.; Osowski, S.; Gwardys, G.; Kurek, J.; Slowinska, M.; Lugowska, I. Random CNN structure: Tool to increase generalization ability in deep learning. *Eurasip J. Image Video Process.* **2022**, *2022*, 3. [\[CrossRef\]](#)
22. Wang, H.; Guo, M.; Tian, L. A deep learning model with signal decomposition and informer network for equipment vibration trend prediction. *Sensors* **2023**, *23*, 5819. [\[CrossRef\]](#)
23. Li, X.; Zhou, S.; Wang, F.; Fu, L. An improved sparrow search algorithm and CNN-BiLSTM neural network for predicting sea level height. *Sci. Rep.* **2024**, *14*, 4560. [\[CrossRef\]](#)
24. Ibrahim, O.R.; Vafaei, A.; Ansari, S.; Abdi, E.; Sifaei, M.; Jafari Mohammadi, S.M. Comparative insights into independent and hybrid modeling strategies for effective river water level prediction and management. *Model. Earth Syst. Environ.* **2025**, *11*, 415. [\[CrossRef\]](#)
25. Staffini, A. A CNN–BiLSTM Architecture for Macroeconomic Time Series Forecasting. *Eng. Proc.* **2023**, *39*, 33.
26. Bischof, L.; Teodoropol, S.; Füchslin, R.M.; Stockinger, K. Hybrid quantum neural networks show strongly reduced need for free parameters in entity matching. *Sci. Rep.* **2025**, *15*, 4318. [\[CrossRef\]](#)
27. Abbas, A.H.; Abdel-Ghani, H.; Maksymov, I.S. Resonant-Tunnelling Diode Reservoir Computing System for Image Recognition. *Electronics* **2025**, *14*, 4471. [\[CrossRef\]](#)
28. Li, S.; Cui, J.; Ren, J. Hybrid classical–quantum neural networks enhanced by quantum architecture search for coronary artery stenosis detection. *Neurocomputing* **2025**, *618*, 129111. [\[CrossRef\]](#)
29. Cowlessur, H.; Thapa, C.; Alpcan, T.; Camtepe, S. A hybrid quantum neural network for split learning. *Quantum Mach. Intell.* **2025**, *7*, 76. [\[CrossRef\]](#)

30. Su, L.; Li, D.; Qiu, D. BLS-QLSTM: A novel hybrid quantum neural network for stock index forecasting. *Humanit. Soc. Sci. Commun.* **2025**, *12*, 1011. [\[CrossRef\]](#)
31. Ceschini, A.; Rosato, A.; Panella, M. Hybrid quantum-classical recurrent neural networks for time series prediction. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
32. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [\[CrossRef\]](#)
33. Han, H.; Kim, D.; Wang, W.; Kim, H.S. Dam inflow prediction using large-scale climate variability and deep learning approach: A case study in South Korea. *Water Supply* **2023**, *23*, 934–947. [\[CrossRef\]](#)
34. Fan, M.; Liu, S.; Lu, D. Advancing subseasonal reservoir inflow forecasts using an explainable machine learning method. *J. Hydrol. Reg. Stud.* **2023**, *50*, 101584. [\[CrossRef\]](#)
35. Khemapatapan, C.; Thepsena, T. Forecasting the Weather behind Pa Sak Jolasid Dam using QuantumMachine Learning. *Adv. Sci. Technol. Eng. Syst. J.* **2023**, *8*, 54–62. [\[CrossRef\]](#)
36. Fellner, T.; Kreplin, D.; Tovey, S.; Holm, C.; Main, J. Quantum Machine Learning for Time Series Prediction. Master's Thesis, University of Stuttgart, Stuttgart, Germany, 2024. [\[CrossRef\]](#)
37. Zhen, L.; Bărbulescu, A. Quantum Neural Networks Approach for Water Discharge Forecast. *Appl. Sci.* **2025**, *15*, 4119. [\[CrossRef\]](#)
38. Zhen, L.; Bărbulescu, A. Variational Quantum Regression Application in Modeling Monthly River Discharge. *Water* **2025**, *17*, 1836. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.