SCOPUA Journal of Applied Statistical Research (JASR)

Vol.2, Issue 3, September 2026
DOI: https://doi.org/10.64060/JASRv2i32

A Multi-Tier Hybrid Machine and Deep Learning Ensemble Framework for Water-Level Prediction: A Case Study of the Indus Basin River

Aqsa Saleem [1], Hafiza Mamona Nazir [1*], Muhammad Zubair [1]

[1]Department of Statistics, University of Sargodha, Sargodha 40100, Pakistan

* Corresponding Email: mamona.nazir@uos.edu.pk

Received: 22 May 2026 / Revised: 28 June 2026 / Accepted: 08 July 2026 / Published online: 24 July 2026

ABSTRACT

Accurate river water-level forecasting supports flood risk mitigation and water-resource management in the Indus Basin System, despite increased hydrological variability. This study proposes a hybrid multi-tier ensemble framework for daily river-level forecasting that incorporates 13 heterogeneous machine learning (ML) and deep learning (DL) models from the tree-based, kernel-based, neural network, and deep learning families, which are combined using Simple Averaging, Inverse-RMSE, Adaptive Inverse-RMSE weighting, and stacking-based strategies. Lag-based feature engineering and chronological TimeSeriesSplit validation maintained temporal relationships while preventing information leakage. Support Vector Regression (SVR) and Extra Trees (ET) outperformed ML models, while GRU and LSTM dominated DL architectures. The proposed hybrid ensemble outperformed the persistence benchmark, with an RMSE of 1.7621 m and R^2 of 0.3012. Diebold-Mariano statistical testing verified the comparative forecasting ability with a rigorous statistical significance assessment. Furthermore, the findings show that predicting accuracy increased only up to an ideal ensemble size, after which additional models yielded declining benefits. Overall, the findings show that model diversity and complementary learning characteristics have a greater impact on forecasting performance than increasing ensemble size or using increasingly complex weighting strategies, resulting in a strong framework for operational river water-level forecasting in data-scarce hydrological environments.

Keywords:Flood forecasting; River level prediction; Indus Basin; Hybrid machine learning; Deep learning; Ensemble learning; Hydrological modeling; Water resources management

1. Introduction

Accurate river water-level forecasting is critical for flood risk mitigation, reservoir management, and long-term water resource planning. The Indus Basin System (IBS) in Pakistan supports almost 18 million hectares of irrigated land, making it the primary source of agricultural productivity and water supply (Ahmad 1994; Basharat and Rizvi 2016). The Himalaya-Karakoram-Hindukush region's glaciers and snowmelt contribute to the basin's flow regime (Lutz et al. 2016). Recent hydroclimatic changes, such as altered precipitation patterns, glacier retreat, and increasing flood frequency, have increased the uncertainty and nonstationarity of river systems. This highlights the need for reliable forecasting approaches to support timely water-management decisions (Khan et al. 2025).

Traditional hydrodynamic models are computationally intensive and require extensive calibration (Aricò et al. 2016; Elhanafy and Copeland 2007). Statistical models such as ARIMA and SARIMA are constrained by linear assumptions (Kader et al. 2020; Al-Mansori et al. 2020; Tadesse and Dinka 2017; Agaj et al. 2024). River water-levels frequently demonstrate considerable temporal persistence, making persistence forecasting a difficult standard for predictive models. Previous research indicates that simple persistence forecasts can give extremely competitive short-term hydrological predictions, especially in systems with considerable temporal autocorrelation (Ghimire and Krajewski 2020). As a result, this work focuses on comparing ML, DL, and ensemble techniques to a persistent baseline.

Machine learning (ML) and deep learning (DL) algorithms have emerged as viable hydrological forecasting alternatives due to their capacity to discover nonlinear relationships directly from data. Early applications of Artificial Neural Networks

demonstrated good performance in rainfall-runoff modeling (Shamseldin 1997). Support Vector Regression enhanced multi-scale streamflow prediction (Asefa et al. 2006). Random Forest (RF), Gradient Boosting (GB), Extra Trees (ET), and Extreme Gradient Boosting (XGBoost) ensemble tree-based models have demonstrated robust predictive ability across varied hydroclimatic situations (Abdoulhalik and Ahmed 2024; Khozani et al. 2025; Kumar et al. 2023). Deep learning architectures including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) improve forecasting performance by incorporating temporal relationships in hydrological time series (Kratzert et al. 2019). TCNs have been studied for their capacity to simulate complex temporal patterns using causal dilated convolutions (Xu et al. 2021). Despite progress, individual models are nevertheless susceptible to noise, prone to overfitting, and typically display reduced resilience under extreme hydrological circumstances (Khozani et al. 2025).

To address the limitations of individual forecasting models, hybrid and ensemble learning approaches have garnered growing interest in hydrological forecasting (Ebtehaj and Bonakdari 2022). Ensemble frameworks, which include numerous predictive models, can improve generalization capability and forecasting robustness under complicated hydroclimatic conditions (Al-Sulttani et al. 2021). Because of their simplicity and interpretability, rule-based weighting methods such as simple averaging and performance-based weighting have gained popularity. Recently, stacking-based meta-learning approaches have shown promising results by learning optimal combinations of base learners through a secondary meta-model (Chowdhury et al. 2025; Li et al. 2020). Hybrid machine learning and deep learning architectures have been successfully employed to capture nonlinear and temporal correlations in hydrological processes (Workneh and Jha 2025; Sun et al. 2022). Despite the growing popularity of hybrid and ensemble learning methods in hydrological forecasting, some significant research gaps remain. Most studies prioritize predictive accuracy through increasingly complex ensemble architectures, with limited attention to how ensemble size affects forecasting performance (Kumar et al. 2023; Workneh and Jha 2025). It is unclear whether larger ensembles consistently improve forecasts or whether gains saturate beyond an optimal configuration. Second, the role of model diversity in ensemble construction remains comparatively underexplored, particularly for river water-level forecasting. Heterogeneous learning paradigms may capture complementary hydrological behaviors more effectively than larger but redundant ensembles (Kumar et al. 2023; Khozani et al. 2025). Third, most research analyze only one integration strategy—rule-based weighting, optimization-based aggregation, or stacking—making it difficult to compare ensemble processes under a unified framework (Parasar et al. 2025; Chowdhury et al. 2025). While stacking-based meta-learning has shown encouraging results (Chowdhury et al. 2025; Li et al. 2020), few research systematically compare it to simpler, more interpretable rule-based weighing under identical validation conditions. These gap require a unifying paradigm for assessing ensemble size, model diversity, and integration method in a consistent experimental setting.

To overcome these gaps, this study investigates at hybrid ML-DL ensembles for daily river water-level forecasting in Pakistan's Jhelum River Basin. The study compares several forecasting models and ensemble integration methodologies to see how ensemble size, model diversity, and aggregation mechanisms influence predictive performance. The study aims to identify robust and economical forecasting algorithms capable of improving water resource management and flood risk mitigation in complicated hydrological situations through a comprehensive comparative examination. The major contributions of this study are summarized as follows:

1. A comprehensive benchmark of heterogeneous ML and DL models is established for daily river water-level forecasting in the Jhelum River Basin.
2. Multiple ensemble integration strategies are systematically compared to identify effective approaches for improving forecasting accuracy and robustness.
3. The influence of ensemble size on predictive performance is investigated to determine whether larger ensembles consistently provide forecasting benefits.
4. The relationship between model diversity and ensemble effectiveness is examined, highlighting the importance of complementary learning paradigms in hydrological forecasting.
5. In order to provide rigorous pairwise comparisons of predicted accuracy, the Diebold–Mariano (DM) test with Newey–West HAC variance estimation and Holm–Bonferroni correction is used to evaluate the statistical significance of forecasting improvements.

The remaining sections of this study are organized as follows. Section 2 outlines the study field, dataset, data preprocessing, and proposed multi-tier ensemble methodology. Section 3 presents the findings of baseline model evaluation, hybrid ensemble comparison, and statistical significance analysis. Section 4 examines the findings in the context of existing literature, and Section 5 summarizes the work with recommendations for further research.

2. Methodology

The overall methodology adopted in this study is summarized in Figure 1, which outlines the end-to-end workflow from data preprocessing through individual model training, ensemble construction, benchmarking, and final performance evaluation. Each stage of this pipeline is described in detail in the following subsections.

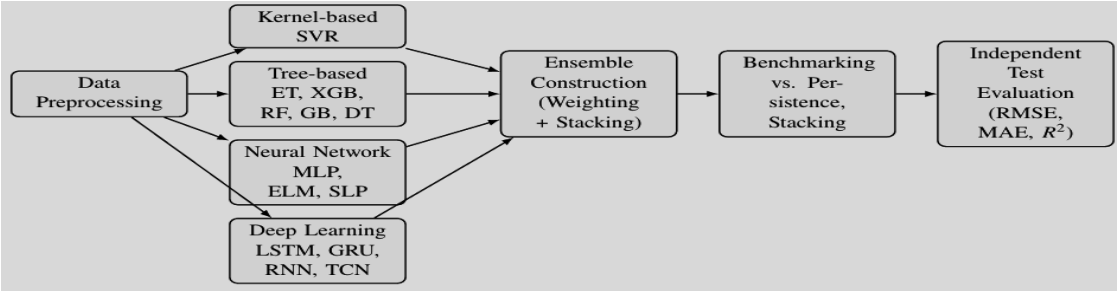


Figure 1: Detailed pipeline showing the four model families feeding into ensemble construction, benchmarking, and final evaluation.

2.1 Study Area and Data Source

The dataset for this study comprises 3,896 daily river hydrological records obtained from the Jhelum River basin, as maintained by WAPDA, Pakistan, and illustrated in Figure 2. The dataset includes daily water level, inflow, and outflow parameters, continuously recorded from 2015 to 2025.

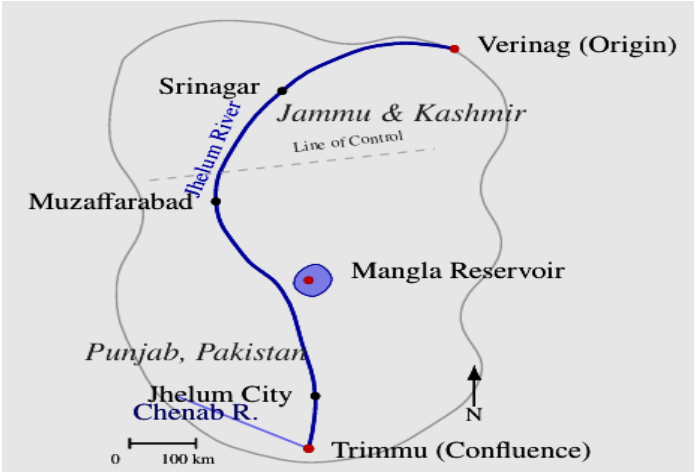


Figure 2: Schematic map of the Jhelum River basin showing key locations from origin to confluence with the Chenab River.

The study focuses on the Jhelum River basin, a crucial hydrological resource, situated in the north-eastern region of Punjab, Pakistan, within the geographical coordinates of (32.50°–33.30° N, 72.50°–73.90° E). The Jhelum River, a primary source of snow and glacial melt for the Indus river system, originates at the Verinag spring and extends approximately 735 km before merging with the Chenab River at Trimmu. The basin exhibits a complex hydrological climate, encompassing mountain ranges reaching 6000 m, transitioning to a semi-arid zone below 200 m in elevation. It also features a combination of western disturbance snowfalls, glacial runoff, and monsoon activity, with the hydrological point of divergence established at Mangla Reservoir (33.13° N, 73.65° E), a key basin for hydropower and irrigation within Pakistan's hydrops scheme.

2.2 Data Preprocessing

Prior to model formation, the hydrological dataset was subject to a comprehensive preprocessing approach to assure data quality, create informative predictors, reduce temporal dependence, and establish a rigorous validation framework appropriate for time-series forecasting. The preparation steps included data quality management, temporal dependence analysis, first-order differencing, lag-based feature engineering, predictor selection, and chronological validation design.

2.3 Data Quality Control and Descriptive Analysis

The hydrological observations were examined for missing values, duplicate records, physically implausible data, sudden sensor spikes, and temporal discrepancies. No missing observations were found in the dataset. However, during the exploratory study, a small number of aberrant data related to measurement noise and recording inconsistencies were discovered. These observations were rectified using local temporal interpolation based on surrounding observations in order to retain hydrological continuity and physically realistic transitions within the time series.

Following quality control, the final dataset included 3,896 daily observations from January 2015 to August 2025. This study's Jhelum River water-level series is summarized in Table 1, which includes descriptive statistics.

Table 1: Descriptive statistics of the Jhelum River water-level series

Statistic	Value
Observations	3,896
Period	Jan 2015–Aug 2025
Missing values	0
Mean (m)	1161.5
Standard deviation (m)	47.0
Minimum (m)	1046.5

Statistic	Value
25th percentile (m)	1123.6
Median (m)	1166.0
75th percentile (m)	1199.4
Maximum (m)	1242.0

To prevent information leakage, all preprocessing processes were carried out using data accessible prior to prediction. Feature-scaling parameters were computed solely from training data and then applied to validation and testing subsets.

2.4 Temporal Dependence Analysis and First-Order Differencing

Exploratory analysis demonstrated extremely high persistence and autocorrelation within the initial water-level series, which is typical of regulated river systems, influenced by basin storage effects and delayed hydrological response processes. To assess temporal dependence, the Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), and Spearman rank correlation analyses were used. The findings showed extraordinarily great persistence over both short- and medium-term lag periods. Table 2 shows that correlation coefficients exceeded 0.99 for short lags and remained over 0.90 even after a 21-day lag, showing significant hydrological memory within the Jhelum River system.

Table 2: Spearman rank correlation coefficients between river water-level and lagged observations

Lag (Days)	Spearman Correlation (ρ)	p-value
1	0.9995	< 0.001
2	0.9986	< 0.001
3	0.9970	< 0.001
5	0.9925	< 0.001
7	0.9861	< 0.001
14	0.9526	< 0.001
21	0.9058	< 0.001

Such strong persistence can boost forecasting performance by allowing models to rely on serial dependence rather than learning hydrological dynamics. Thus, first-order differencing was applied.

$$\Delta y_t = y_t - y_{t-1} \tag{1}$$

where y_t denotes the observed water-level at time step t. Moreover, transforming the forecasting task from predicting absolute water levels to predicting daily water-level changes. Consequently, all ML, DL, and ensemble models were trained and evaluated in first-difference space. Under this formulation, the persistence benchmark assumes no daily change, such as:

$$\Delta \hat{y}_t = 0 \tag{2}$$

Persistence has been chosen as the primary benchmark, with models required to surpass the naive assumption of zero percent daily change. Evaluation in the first-difference space decreases performance inflation induced by great hydrological persistence, allowing for a more accurate assessment of model generalization. Figure 3 shows the appropriate ACF and PACF plots.

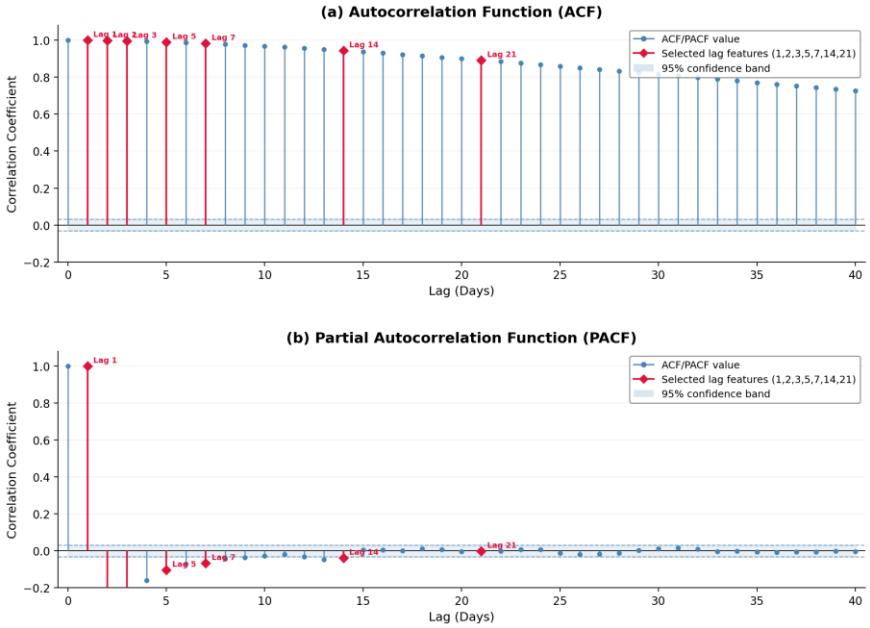


Figure 3: Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the Jhelum River water-level series showing strong temporal dependence and persistence across multiple lag horizons

2.5 Lag-Based Feature Engineering and Predictor Selection

A multi-scale lag embedding framework was developed using the persistence structure discovered by the ACF, PACF, and Spearman correlation analyses. Lag periods of 1, 2, 3, 5, 7, 14, and 21 days were chosen to capture both short-term hydrological response, weekly persistence patterns, and long-term basin memory effects. The resulting predictor vector is defined as

$$\boldsymbol{X_t} = [\boldsymbol{\Delta y_{t-1}}, \boldsymbol{\Delta y_{t-2}}, \boldsymbol{\Delta y_{t-3}}, \boldsymbol{\Delta y_{t-5}}, \boldsymbol{\Delta y_{t-7}}, \boldsymbol{\Delta y_{t-14}}, \boldsymbol{\Delta y_{t-21}}] \quad (3)$$

where Δy_t represents the first-differenced water-level series.

In addition to lagged predictors, the available hydrological variables, including Jhelum inflow, Jhelum outflow, and Chenab inflow, were evaluated using Spearman rank correlation analysis. The results are presented in Table 3**.**

Table 3: Spearman rank correlation between river water level and available exogenous variables

Variable	**Spearman Correlation (ρ)**	p**-value**
Jhelum Inflow	0.1245	< 0.001
Jhelum Outflow	0.0754	< 0.001
Chenab Inflow	0.4661	< 0.001

The findings show that the temporal information included in lagged water-level data is significantly stronger than that provided by the available exogenous variables. Even the 21-day lag maintained a correlation coefficient greater than 0.90, while the best external predictor (Chenab inflow) had a correlation coefficient of only 0.466. Consequently, lagged water-level measurements were chosen as the major predictor set for model construction. This finding implies that the predictive information included in the historical water-level series is significantly stronger than that provided by the exogenous factors. As a result, integrating weakly linked external factors was unlikely to increase forecasting accuracy and may contribute more noise into the learning process.

This univariate lag-based framework allows for a focused examination of predicting effectiveness while minimizing any discrepancies caused by meteorological and hydroclimatic variables that were not available at consistent daily resolution throughout the study period.

Chronological Validation Framework

To maintain temporal causality and prevent information leaking, the dataset was divided chronologically into training (80%; 3,117 observations) and independent testing (20%; 779 observations) subsets, with the test set containing the latest observations.

Random K-fold cross-validation is not suitable for time-series forecasting because it violates temporal ordering and yields overly optimistic performance estimates. A five-fold expanding-window TimeSeriesSplit framework was used for training data. In each fold, training observations came before validation observations, maintaining chronological consistency.

The resulting Out-of-Fold (OOF) predictions were then used for ensemble creation and model comparison, with the independent holdout test set reserved solely for final evaluation. This historical validation approach allows for an unbiased assessment of predicting performance under realistic operational situations while preventing information leakage. Figure 4 shows the adopted train-test partitioning and TimeSeriesSplit technique. This study used a chronological train-test partitioning and TimeSeriesSplit-based validation technique, as illustrated in Figure 4.

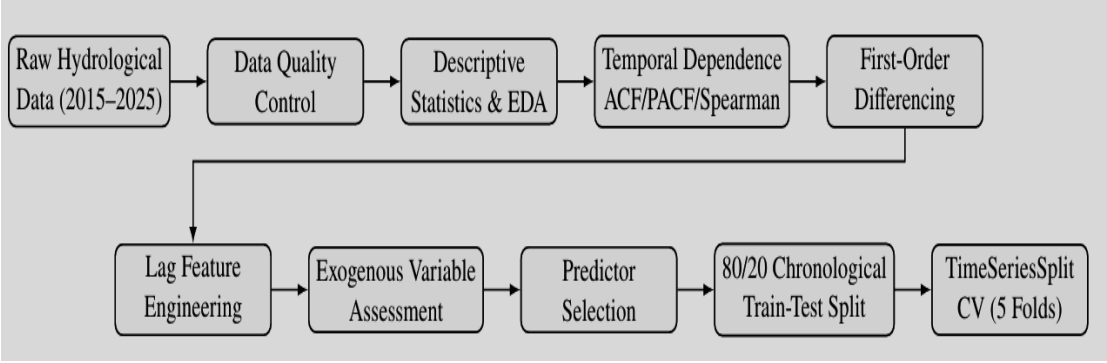


Figure 4: Overview of the data preprocessing framework.

2.6 Base Learners Architectures (Tier-I)

Tier-I of the proposed framework encompasses a wide range of ML and DL models that are used as base learners to effectively model the nonlinear, nonstationary, and multi-scale nature of river water-level processes. Each base learner is trained on the same input feature vector X_t and generates a separate prediction $\hat{y}_{t,j}$ at time step t, where $j = 1,2,\ldots,M$ represents the base learner index and M denotes the total number of base models. The incorporation of different model architectures promotes complementary learning of hydrological processes, thus improving the diversity of the base learner predictions and, in turn, the efficacy of the ensemble integration process.

2.6.1 Tree Based and Boosting Models

The tree-based learning category includes DT, RF, ET, GB, and XGBoost. Although these algorithms internally rely on ensemble of decision trees. Classifying them into individual learners within the context of this study's multi-tier framework is necessary for diversity(Kedam et al. 2024). Boosting-based models, such as GB and XGBoost, learn predictive functions in an additive fashion by iteratively solving for a weak learner to fit the residual of the previous iteration. At the m^{th} iteration, the loss function is optimized by solving the following regularized loss function:

$$\mathcal{L}^{(m)} = \sum_{t=1}^{N} \ell\left(y_t, \hat{y}_t^{(m-1)} + f_m(X_t)\right) + \Omega(f_m), \tag{4}$$

where y_t is the observed river water level at time step t, $\hat{y}_t^{(m-1)}$ is the aggregated prediction from the previous boosting iteration, $f_m(\cdot)$ is the newly introduced weak learner, $\ell(\cdot)$ is the loss function, and $\Omega(\cdot)$ is the regularization term that regulates model complexity and prevents overfitting(Kedam et al. 2024). The additive learning process allows boosting models to improve their prediction accuracy through the correction of residual errors at each iteration.

2.6.2 Kernel-Based and Shallow Neural Networks

Support Vector Regression (SVR) and simple neural network models were employed as the complementary base models to address the nonlinear input-output relationships in river water-level dynamics(Asefa et al. 2006). The SVR maps the input feature vector of the example, for instance, $\boldsymbol{X_t}$, a high-dimensional feature space with the help of kernel functions and determines an optimal regression hyperplane by minimizing a regularized ε-insensitive loss function. The problem of the SVR optimization is stated as:

$$\min_{w,b,\xi_t,\xi_t^*} \frac{1}{2} \| w \|^2 + C \sum_{t=1}^{N} (\xi_t + \xi_t^*), \tag{5}$$

$$y_t - (w^\top \phi(X_t) + b) \leq \varepsilon + \xi_t, \tag{6}$$

$$(w^\top \phi(X_t) + b) - y_t \leq \varepsilon + \xi_t^*, \tag{7}$$

where W is weight vector, $\boldsymbol{b}$ is bias term ,$\phi(\cdot)$ nonlinear mapping feature, C regularization parameter.

A Multi-Layer Perceptron (MLP) model was also utilized to approximate nonlinear hydrological relationships through layered nonlinear transformations (Shamseldin 1997). The hidden layer activation for a single-layer network is defined as:

$$\boldsymbol{H_t} = \boldsymbol{\sigma}(\boldsymbol{W X_t} + \boldsymbol{b}) \tag{8}$$

in which W weight vector, $\boldsymbol{b}$ is bias term and $\sigma(\cdot)$ nonlinear activation function.

ELM is a fast learning neural baseline where the output weights are computed analytically (Kedam et al. 2024). The output matrix is as:

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger} \boldsymbol{y} \tag{9}$$

where H representing the hidden-layer output matrix, $H^{\dagger}$ is Moore–Penrose pseudoinverse.

2.6.3 Recurrent and Temporal Deep Architectures

The forecasting of hydrological time series relies on prior values, contingent on the degree of persistence (memory). Recurrent Neural Networks (RNNs) facilitate this through feedback loops. Generalized RNNs can transmit input in either direction, to and from all layers. Consequently, the network's output depends not only on external input but also on its state from the previous time step. However, RNNs were employed to capture short-term autoregressive relationships as:

$$\boldsymbol{h_t} = \boldsymbol{\sigma}(\boldsymbol{W_h h_{t-1}} + \boldsymbol{U_h X_t} + \boldsymbol{b_h}) \tag{10}$$

where h_t is the hidden state at time t,X_t is input vector and W_h, U_h is weight matrix.

The Long Short-Term memory (LSTM) network model is a kind of recurrent neural network (RNN) where the standard hidden layer is replaced by a memory component consisting of an input gate, an output gate, a memory cell, and a forget gate. The input gate is used to decide which information should enter the memory cell state, and the forget gate is used to decide which information should be erased from the memory cell state (Kratzert et al. 2018). LSTMs build on the above formulation by adding a series of gates that aim to keep long-term hydrological memory and avoid the problem of vanishing gradient effects (Bengio et al. 1994). The update of the memory cell is given by:

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_f h_{t-1}} + \boldsymbol{U_f X_t} + \boldsymbol{b_f}) \tag{11}$$

$$i_t = \sigma(W_i h_{t-1} + U_i X_t + b_i) \tag{12}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c h_{t-1} + U_c X_t + b_c) \tag{13}$$

$$o_t = \sigma(W_o h_{t-1} + U_o X_t + b_o) \tag{14}$$

$$h_t = o_t \odot \tanh(C_t) \tag{15}$$

where f_t, i_t, o_t denotes forget, input and output gate respectively, and $\boldsymbol{C}_t$ the memory cell state.

The Gated Recurrent Unit (GRU) was introduced by Cho et al. [2014] to enable each recurrent unit to adaptively learn dependencies on different time scales. Like the LSTM unit, the GRU also has gating units controlling the flow of information within the unit, but without the need for memory cells (Chung et al. 2014).

$$\boldsymbol{z_t} = \boldsymbol{\sigma}(\boldsymbol{W_z h_{t-1}} + \boldsymbol{U_z X_t} + \boldsymbol{b_z}) \tag{16}$$

$$r_t = \sigma(W_r h_{t-1} + U_r X_t + b_r) \tag{17}$$

$$\tilde{h}_t = \tanh(W_h(r_t \odot h_{t-1}) + U_h X_t + b_h) \quad (18)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t) \quad (19)$$

where z_t and r_t denote the update and reset gates, respectively.

Temporal Convolutional Networks (TCN) were used in order to learn temporal patterns by means of causal dilated one-dimensional convolutions with no recurrent feedback (Xu et al. 2021). The convolutional operation can be defined as: what allows the processing sequence in parallel, with time causality.

$$\boldsymbol{h_s = \sum_{i=0}^{k-1} f_i \cdot x_{s-d_i}} \quad (20)$$

where k denotes the kernel size, f_i represents convolutional filter weights, and d_i denotes the dilation factor that controls the receptive field size.

2.7 Hybrid Ensemble Framework (Tier-II)

Tier-II of the proposed framework is to improve forecasting performance and investigate the influence of ensemble size and model diversity, a hybrid ensemble framework was developed by combining predictions from the heterogeneous ML and DL base learners described in Tier-I. Let $\hat{y}_{t,j}$ denote the prediction of the j-th base learner at time step t, where $j = 1,2, \ldots, M$ and M represents the number of ensemble members. The ensemble prediction is obtained as

$$\boldsymbol{\hat{y} * t = \sum * j = 1^M \omega_j \hat{y}_{t,j}} \quad (21)$$

subject to:

$$\boldsymbol{\sum_{j=1}^{M} \omega_j = 1, \qquad \omega_j \geq 0.} \quad (22)$$

Three deterministic weighting techniques were investigated to aggregate the base learners' predictions: Simple Averaging, Inverse-RMSE Weighting, and Adaptive Inverse-RMSE Weighting. These methods were chosen because they are computationally efficient, interpretable, and appropriate for the systematic study of ensemble saturation behavior.

2.7.1 Simple Averaging

Simple averaging assigns equal importance to all ensemble members and is defined as

$$\boldsymbol{\hat{y} * t = \frac{1}{M} \sum * j = 1^M \hat{y}_{t,j}.} \quad (23)$$

2.7.2 Inverse-RMSE Weighting

Inverse-RMSE weighting assigns larger weights to models with lower validation errors. The weights are computed as

$$\boldsymbol{\omega_j = \frac{1/RMSE_j}{\sum_{k=1}^{M}(1/RMSE_k)}} \quad (24)$$

and the final prediction is

$$\boldsymbol{\hat{y} * t = \sum * j = 1^M \omega_j \hat{y}_{t,j}.} \quad (25)$$

2.7.3 Adaptive Inverse-RMSE Weighting

To further emphasize high-performing models while penalizing weaker learners, an adaptive weighting strategy based on exponential error decay was adopted. The ensemble weights are calculated as

$$\boldsymbol{\omega_j = \frac{\exp(-RMSE_j)}{\sum_{k=1}^{M} \exp(-RMSE_k)}} \quad (26)$$

and the final ensemble prediction is

$$\boldsymbol{\hat{y} * t = \sum * j = 1^M \omega_j \hat{y}_{t,j}.} \quad (27)$$

where $RMSE_j$ denotes the validation RMSE of the j-th base learner. This adaptive scheme increases the influence of highly accurate models while reducing the contribution of poorly performing ensemble members.

To investigate ensemble saturation effects, ensemble sizes ranging from two-model compact ensembles to full heterogeneous ensembles were systematically evaluated under all three weighting strategies.

2.8 Benchmark Models for Comparative Evaluation (Tier-III)

To provide a rigorous assessment of the proposed hybrid ensemble framework, two benchmark approaches were considered: a persistence forecasting model and a stacking-based ensemble framework. These benchmarks represent widely used reference approaches in hydrological forecasting and ensemble learning.

2.8.1 Persistence Baseline

Persistence forecasting was chosen as the primary baseline model since all machine learning and deep learning models were trained on the first-differenced river water-level dataset. Persistence assumes no future change in water-level, resulting in the equation:

$$\boldsymbol{\Delta \hat{y}_t = 0} \quad (28)$$

where $\Delta \hat{y}_t$ represents the expected change in water level at time step t. Persistence forecasting is a useful benchmark since any improvement above this baseline suggests that the forecasting model has retrieved predictive information beyond simple temporal dependency and autocorrelation (Ghimire and Krajewski 2020).

2.8.2 Stacking-Based Ensemble Benchmark

In addition to persistence forecasting, stacking-based ensemble architecture was used as a comparison tool. Stacking combines the predictions of numerous base learners using a secondary learning process known as a meta-learner. To prevent information leaking, Out-of-Fold (OOF) predictions provided by the TimeSeriesSplit validation technique served as meta-features.

Let $\widehat{\boldsymbol{y}}_{\boldsymbol{t,j}}$ represent the prediction of the $\boldsymbol{jth}$ base learner at time step $\boldsymbol{t}$, with $\boldsymbol{j} = \boldsymbol{1, 2, \dots, M}$. The meta-feature vector is defined as:

$$\boldsymbol{Z} * \boldsymbol{t} = [\widehat{\boldsymbol{y}} * \boldsymbol{t}, \boldsymbol{1}, \widehat{\boldsymbol{y}} * \boldsymbol{t}, \boldsymbol{2}, \dots, \widehat{\boldsymbol{y}} * \boldsymbol{t}, \boldsymbol{M}] \tag{29}$$

The final stacked prediction is obtained as

$$\widehat{\boldsymbol{y}} * \boldsymbol{t} = \boldsymbol{f} * \text{meta}(\boldsymbol{Z_t}), \tag{30}$$

where $f_{\text{meta}}(\cdot)$ denotes the meta-learning algorithm.

2.8.2.1 Ridge Regression Meta-Learner

Ridge Regression was used as a linear meta-learner because it can reduce multicollinearity between base-model predictions. The optimization problem is represented as:

$$\min_{\boldsymbol{\beta}} |\boldsymbol{y} - \boldsymbol{Z\beta}|_{\boldsymbol{2}}^{\boldsymbol{2}} + \boldsymbol{\lambda}|\boldsymbol{\beta}|_{\boldsymbol{2}}^{\boldsymbol{2}}, \tag{31}$$

where Z denotes the OOF prediction matrix, $\boldsymbol{y}$ represents the observed target values, β is the coefficient vector, and λ is the regularization parameter.

2.8.2.2 Gradient Boosting Regressor Meta-Learner

To describe nonlinear interactions between base-model predictions, a Gradient Boosting Regressor (GBR) was used as a nonlinear meta-learner. The boosting procedure is described as:

$$\boldsymbol{F_m(Z)} = \boldsymbol{F_{m-1}(Z)} + \boldsymbol{\gamma_m h_m(Z)} \tag{32}$$

where $h_m(\cdot)$ represents the weak learner fitted at iteration m, and γ_m is the learning rate. GBR can capture higher-order nonlinear interactions among base learner predictions through iterative residual correction (Friedman 2001).

The stacking framework serves as a reliable ensemble-learning baseline against which the suggested rule-based hybrid ensemble techniques are compared.

2.9 Performance Evaluation

In this study, MAE, RMSE, and R^2 are used as statistical tools to judge the predictive accuracy of models. MAE denotes the average absolute difference between observed and predicted values, RMSE denotes the spread of prediction error, and R^2 denotes the goodness of fit of the simulated to observed series values. If MAE and RMSE are low suggesting the better performance where R^2 has a value between 0 and 1 and close to 1, indicate good performance (Parasar et al. 2025). The evaluation metrics are as followed:

$$\boldsymbol{RMSE} = \sqrt{\frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{t=1}}^{\boldsymbol{N}}(\boldsymbol{y_t} - \widehat{\boldsymbol{y}}_{\boldsymbol{t}})^{\boldsymbol{2}}} \tag{33}$$

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t| \tag{34}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{N}(y_t - \bar{y})^2} \tag{35}$$

where y_t and $\hat{y}_t$ denote the observed and predicted river water levels at time step t, respectively; N represents the total number of test samples, and $\bar{y}$ denotes the mean of observed values. In addition to this, the predictive accuracy of different models was checked for statistical significance by using the Diebold-Mariano test.

$$\boldsymbol{DM} = \frac{\bar{\boldsymbol{d}}}{\sqrt{\hat{\boldsymbol{\sigma}}_{\boldsymbol{d}}^{\boldsymbol{2}}/\boldsymbol{T}}} \tag{36}$$

where $\bar{d}$ represents the mean loss differential between two forecasting models, $\hat{\sigma}_d^2$ denotes the estimated long-run variance of the loss differential series, and T is the number of forecast observations. The DM test evaluates the null hypothesis of equal predictive accuracy between two competing models and provides a rigorous statistical basis for model comparison. The DM test has its theoretical footing by Diebold and Mariano (1995).

3. Results

3.1 Hyperparameter Configuration

To ensure a fair and reproducible comparison of all forecasting methods, fixed hyperparameter values were used throughout the study. All ML, DL, and ensemble models were trained with the same preprocessing framework, chronological data segmentation technique, and TimeSeriesSplit cross-validation method. Hyperparameters were chosen through exploratory testing and then held constant throughout all model assessments. Tables 4, 5, and 6 provide a summary of the final settings used in this study.

Table 4: Hyperparameter configuration of ML models

Model	Parameter	Value
RF	Number of Trees, Max Depth, Min Samples Split	100, 10, 10
ET	Number of Trees, Max Depth, Min Samples Split	100, 10, 10
GB	Trees, Learning Rate, Max Depth, Subsample	100, 0.05, 5, 0.70
XGBoost	Trees, Max Depth, LR, Subsample, Col Sample, L1, L2	100, 6, 0.05, 0.70, 0.70, 0.10, 1.00
SVR	Kernel, C, ϵ, Gamma	RBF, 1.0, 0.10, Scale
MLP	Hidden, Activation, Optimizer, α, Iterations	(64,32), ReLU, Adam, 0.01, 300
ELM	Hidden Neurons, α	30, 0.10
SLP	Regression Type, α	Ridge, 5.0
DT	Max Depth, Min Samples Split	5, 15

Table 5: Hyperparameter configuration of DL models

Model	Parameter	Value
LSTM	Hidden, Dropout, Recurrent Dropout, L2, Learning Rate	(128, 64), 0.30, 0.20, 10^{-4}, 0.001
GRU	Hidden, Dropout, Recurrent Dropout, L2, Learning Rate	(96, 48), 0.30, 0.20, 10^{-4}, 0.001
RNN	Hidden, Dropout, Recurrent Dropout, L2, Learning Rate	(64, 32), 0.30, 0.20, 10^{-4}, 0.0005
TCN	Filters, Kernel Size, Dilation, Dropout, L2, Learning Rate	64, 3, (1, 2, 4), 0.20, 10^{-4}, 0.0005
Training	Batch Size, Epochs, Early Stopping Patience, LR Reduction Patience	32, 300, 20, 10

Table 6: Hyperparameter configuration of ML models

Method	Parameter	Value
Simple Average	Weighting	Equal
Inverse-RMSE	Weighting	RMSE-Based
Adaptive Inverse-RMSE	Weighting	Exponential RMSE
Ridge Stacking	Meta-Learner, α, Training Data	Ridge, 5.0, OOF
GBR Stacking	Meta-Learner, Trees, Learning Rate, Max Depth, Subsample	GBR, 100, 0.05, 3, 0.80
Cross-Validation	Validation Strategy	TimeSeriesSplit, 5 Folds, Chronological

The ensemble and stacking approaches were trained using OOF predictions generated through five-fold TimeSeriesSplit validation. This strategy preserves temporal ordering and prevents information leakage during model training. The resulting OOF predictions were subsequently used as validation predictions for the weighting-based ensembles and as meta-features for the stacking models. Consequently, all forecasting approaches were evaluated under identical experimental conditions, enabling a fair assessment of the influence of weighting strategy, model diversity, and ensemble size on forecasting performance.

3.2 Performance of ML Models

Table 7 summarizes the OOF validation and independent test performance of the ML models. The OOF measures were acquired using the five-fold TimeSeriesSplit framework, whereas the test metrics were computed using the independent holdout dataset. All models were trained and tested in the first-difference space.

Table 7: Performance comparison of ML models of different families

Model	OOF Validation				Independent Test Set			
	RMSE	MAE	R^2	Corr	RMSE	MAE	R^2	Corr
ET	1.197	0.619	0.269	0.627	1.797	0.44	0.271	0.523
XGB	1.212	0.647	0.25	0.613	1.806	0.454	0.264	0.514
RF	1.209	0.638	0.254	0.629	1.877	0.462	0.205	0.47
GB	1.229	0.639	0.229	0.62	1.924	0.466	0.164	0.437
DT	1.232	0.668	0.226	0.616	2.016	0.512	0.083	0.382
SVR	1.019	0.56	0.311	0.683	1.788	0.361	0.278	0.546
MLP	1.209	0.635	0.253	0.631	2.177	0.495	-0.07	0.299
ELM	1.199	0.633	0.266	0.62	2.48	0.522	-0.388	0.166
SLP	1.223	0.68	0.236	0.58	2.238	0.516	-0.131	0.232

SVR produced the lowest RMSE (1.788 m), highest correlation coefficient (0.546), and highest coefficient of determination ($\boldsymbol{R^2 = 0.278}$) among all ML approaches, demonstrating the best generalization performance on the independent test dataset. SVR's better performance suggests that nonlinear interactions in the differenced hydrological series were well captured by kernel-based learning.

Strong forecasting ability was also shown by the ensemble tree-based systems. ET scored second on the test dataset ($\boldsymbol{R^2 = 0.271}$) and had the greatest OOF validation performance ($\boldsymbol{R^2 = 0.269}$), closely followed by XGBoost ($\boldsymbol{R^2 = 0.264}$). The steady performance of ET, XGBoost, and RF indicates that ensemble tree-learning techniques can efficiently take advantage of nonlinear interactions between lagged river-level observations while retaining strong generalization capabilities. The advantages of variance reduction and enhanced resilience attained through ensemble tree-learning

techniques were highlighted by the single-tree DT model's significantly worse predictive performance compared to its ensemble equivalents. On the independent test dataset, the shallow neural models (MLP, ELM, and SLP) also showed a significant decline in performance. These models showed negative test-set R^2 values, indicating weak generalization power and possible overfitting to the training period, even though they achieved competitive OOF validation scores.

Overall, the findings indicate that the best individual ML techniques for predicting daily river water-level changes in first-difference space are kernel-based and ensemble tree-based approaches.

3.3 Performance of DL Models

Table 8 presents the OOF validation and independent test performance of the deep learning architectures.

Table 8: Performance comparison of DL models

Model	OOF Validation				Independent Test Set			
	RMSE	MAE	R^2	Corr	RMSE	MAE	R^2	Corr
LSTM	1.2897	0.8152	0.1544	0.5093	1.9319	0.7216	0.1594	0.4008
GRU	1.3043	0.8226	0.1352	0.4738	1.9166	0.7068	0.1727	0.4189
RNN	1.4663	1.0245	-0.0929	0.3287	2.0507	0.9840	0.0529	0.2586
TCN	1.3362	0.8719	0.0923	0.5154	1.9972	0.8460	0.1016	0.3675

With the lowest RMSE (1.917 m), greatest R^2 value (0.173), and strongest correlation coefficient (0.419) among the DL architectures, GRU demonstrated the best overall prediction performance on the independent test dataset. The better performance of GRU indicates that, while keeping a relatively compact design, its gating mechanism successfully captured the short- and medium-term temporal dependencies found in the differenced river-level series.

With an RMSE of 1.932 m and R^2 of 0.159, LSTM demonstrated similar performance. First-order differencing significantly diminishes long-range persistence within the original water-level series, so even though LSTM is especially made to maintain long-term temporal memory, its benefit seems limited in the current forecasting challenge. As a result, LSTM's extra memory capacity does not result in appreciable speed improvements over GRU. The TCN model demonstrated a respectable level of predictive ability, surpassing the traditional RNN but falling short of both GRU and LSTM. Although TCN was able to effectively capture local temporal patterns through the use of causal dilated convolutions, its capacity to describe intricate hydrological connections was still rather restricted.

In both the validation and testing stages, the traditional RNN consistently yielded the worst results. Its decreased explanatory power and poorer correlation coefficients suggest that developing stable temporal representations is challenging. This behavior is in line with the established drawbacks of conventional recurrent architectures, such as diminished capacity to capture longer temporal relationships and vanishing-gradient effects.

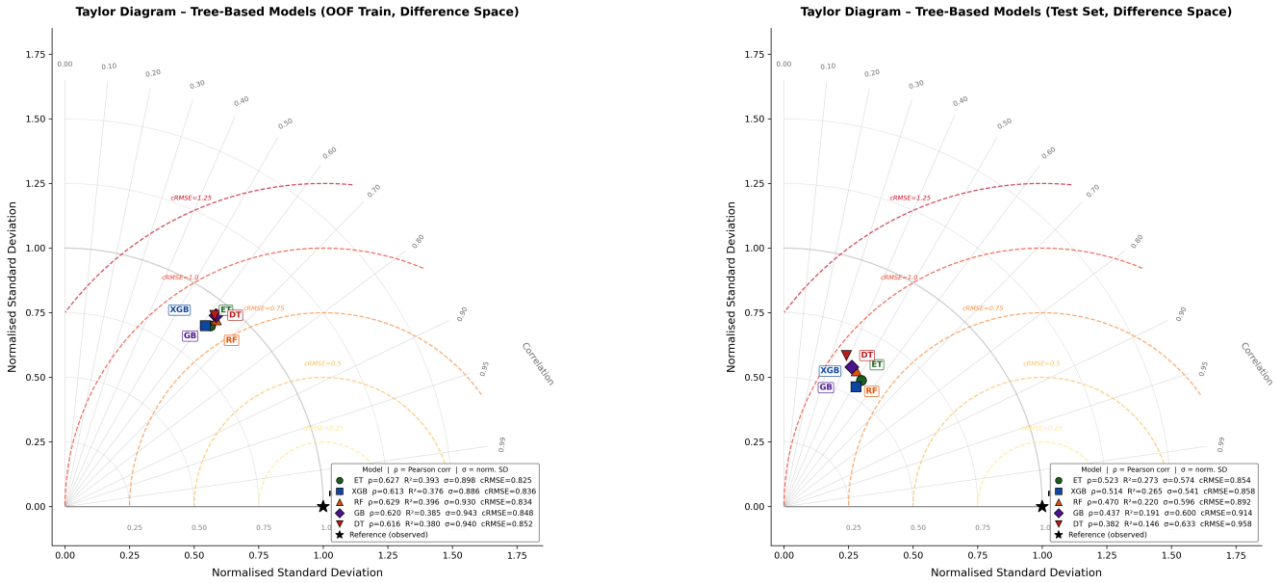
The best ML models (SVR, ET, and XGBoost) beat all deep learning architectures on the independent test dataset, according to a comparison between Tables 7 and 8. This result implies that well-designed machine learning algorithms offer better representations of daily river-level changes than standalone deep learning models for the available dataset size and forecasting horizon.

3.4 Graphical Performance Assessment Using Taylor Diagrams

Taylor diagrams give a thorough graphical evaluation by concurrently summarizing correlation, normalized standard deviation, and centred Root Mean Square Error (cRMSE), whilst traditional performance metrics offer quantitative measures of forecasting accuracy. Better predicting ability is indicated by models that are closer to the reference observation because they show higher agreement with the observed series.

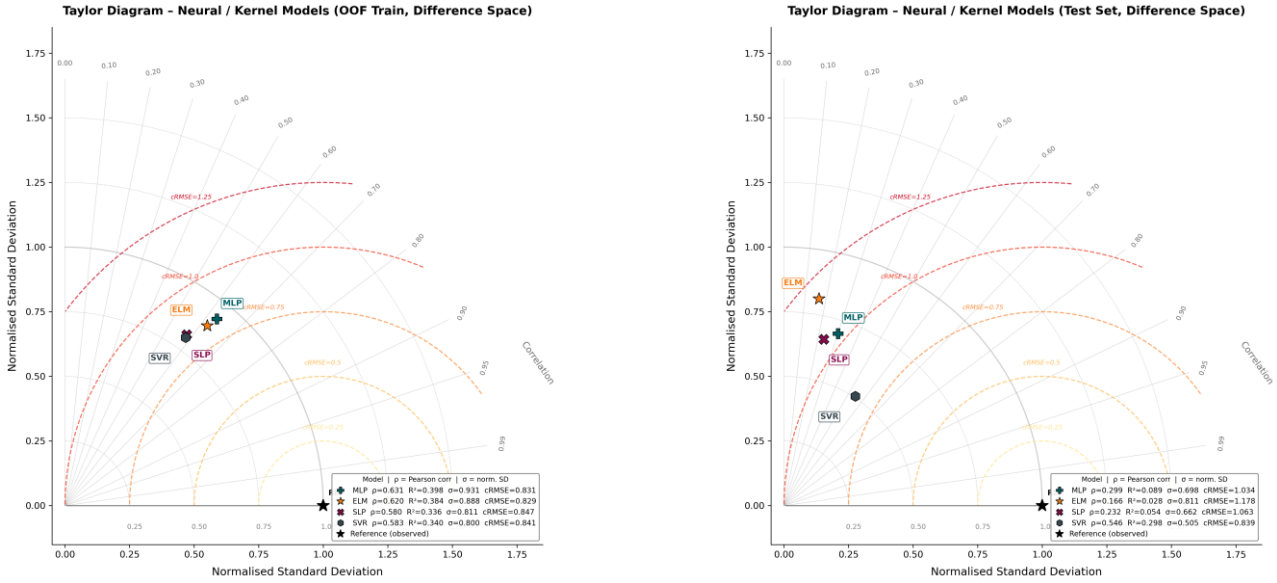
3.4.1 ML Models

Figures 5 and 6 present the Taylor diagrams for the machine learning models during the OOF validation and independent testing phases, respectively.



OOF validation Independent test

Figure 5: Taylor diagram comparison of tree-based ML models in first-difference space



OOF validation Independent test

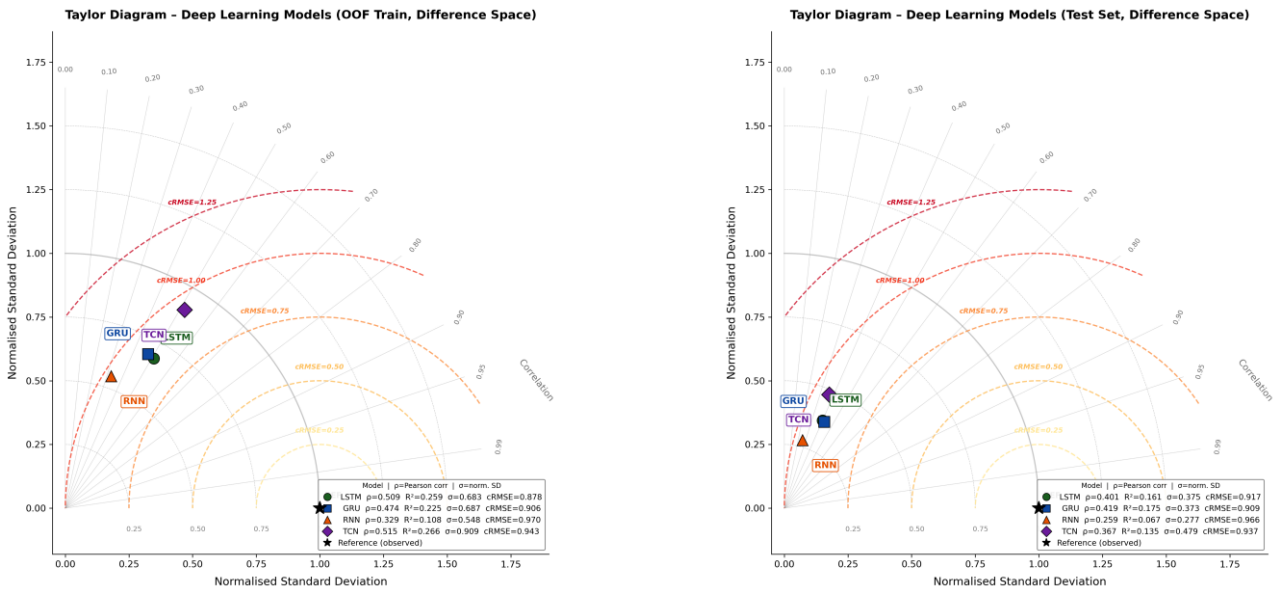
Figure 6: Taylor diagram comparison of nn-based ML models in first-difference space

The OOF and test Taylor diagrams show that ET and XGBoost regularly occupy the spots closest to the reference observation, implying better representation of observed variability and reduced centered prediction errors. ET had the lowest cRMSE and one of the greatest correlation coefficients among the tree-based techniques, demonstrating its strong generalizability. RF showed slightly weaker agreement with the observed series, but GB and DT had higher cRMSE values and smaller correlations.

According to Fig 6, SVR is the strongest individual ML NN model based on test-set RMSE and $\boldsymbol{R^2}$. Overall, the Taylor-diagram study indicates that ET and XGBoost are the most resilient tree-based learners, with SVR remaining the greatest overall ML benchmark.

3.4.2 DL Models

Figures 7 and 8 illustrate the Taylor diagrams for the deep learning architectures during OOF validation and independent testing.



OOF validation Independent test

Figure 7: Taylor diagram comparison of DL models in first-difference space

The DL Taylor diagrams show that GRU and LSTM remain closest to the reference observation throughout the validation and testing phases. Their substantially greater correlations and lower cRMSE values suggest that they are better at capturing short- and medium-term temporal dependencies than the typical RNN design. Table 5 shows that GRU outperforms LSTM on the independent test dataset by a small margin.

The TCN model obtained intermediate performance, demonstrating that causal convolutional structures can effectively learn local temporal patterns but are less competitive than gated recurrent architectures in the current forecasting challenge. In contrast, the standard RNN continuously occupies the most distant position from the reference observation, indicating a worse correlation, less variability representation, and bigger prediction errors.

Overall, the Taylor-diagram analysis matches the numerical findings, indicating that GRU and LSTM are the strongest individual DL models, whereas RNN has the lowest predictive performance of the architectures tested.

3.5 Ensemble Saturation and Weighting Strategy Analysis

This section looks at the impact of ensemble size and weighting approach on forecasting performance. Three ensemble types were tested: ML, DL, and hybrid ML-DL ensembles. Simple averaging, inverse-RMSE, and adaptive inverse-RMSE weighting schemes were used for each category to determine whether increasing model diversity continuously improves forecasting accuracy or if there is a saturation point beyond which adding more models is deleterious.

3.6 ML Ensemble Analysis

Table 9: Performance comparison of hybrid ML ensemble configurations under different ensemble sizes and weighting strategies on the independent test dataset.

Size	Models	Weighting	RMSE	MAE	R^2	Corr
Top-2	SVR+ET	Simple	1.7797	0.4329	0.2963	0.5363
		Inverse	1.7780	0.4322	0.2965	0.5364
		Adaptive	**1.7696**	**0.4320**	**0.2983**	**0.5464**
Top-3	SVR+ET+XGB	Simple	1.7962	0.4481	0.2877	0.5360
		Inverse	1.7862	0.4478	0.2878	0.5387
		Adaptive	**1.7761**	**0.4468**	**0.2879**	**0.5391**
Top-4	SVR+ET+XGB+RF	Simple	1.8044	0.4639	0.2731	0.5242
		Inverse	1.7937	0.4626	0.2737	0.5248
		Adaptive	**1.7829**	**0.4613**	**0.2743**	**0.5254**
Top-5	SVR+ET+XGB+RF+GB	Simple	1.8272	0.4756	0.2545	0.5073
		Inverse	1.8155	0.4763	0.2560	0.5086
		Adaptive	**1.8038**	**0.4771**	**0.2573**	**0.5098**
Top-6	SVR+ET+XGB+RF+GB+DT	Simple	1.8433	0.4747	0.2330	0.4885
		Inverse	1.8397	0.4743	0.2359	0.4910
		Adaptive	**1.8302**	**0.4739**	**0.2389**	**0.4935**
Top-7	SVR+ET+XGB+RF+GB+DT+MLP	Simple	1.8811	0.4751	0.2012	0.4621
		Inverse	1.8730	0.4747	0.2081	0.4677

		Adaptive	**1.8651**	**0.4744**	**0.2147**	**0.4731**
Top-8	SVR+ET+XGB+RF+GB+DT+MLP+SLP	Simple	1.9178	0.4751	0.1697	0.4353
		Inverse	1.9053	0.4752	0.1805	0.4439
		Adaptive	**1.8932**	**0.4714**	**0.1909**	**0.4524**
Full	SVR+ET+XGB+RF+GB+DT+MLP+SLP+ELM	Simple	1.9658	0.4765	0.1276	0.4034
		Inverse	1.9445	0.4761	0.1464	0.4174
		Adaptive	**1.9244**	**0.4760**	**0.1640**	**0.4309**

Table 9 summarizes the performance of the ML ensembles, while Figures 8 and 9 illustrate the corresponding saturation behavior and bootstrap performance distributions.

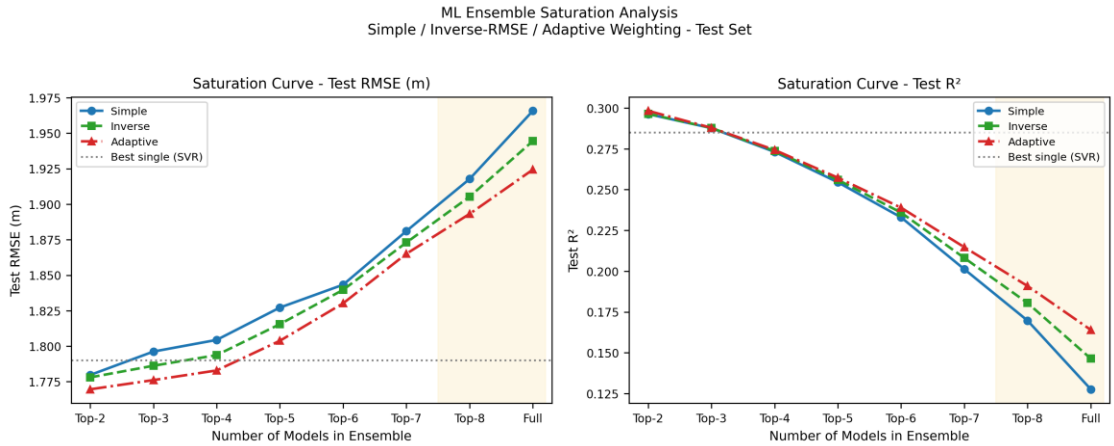


Figure 8: Machine learning ensemble saturation analysis showing the influence of ensemble size on RMSE and R^2 under different weighting strategies. The horizontal dashed line represents the best individual model (SVR)

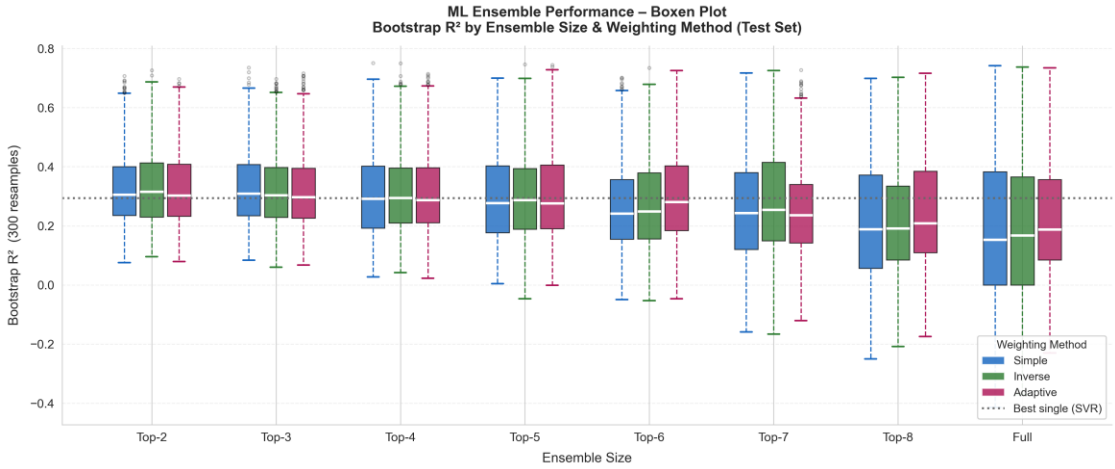


Figure 9: Bootstrap distribution of machine learning ensemble performance across different ensemble sizes and weighting strategies

The data show a definite saturation effect. The Adaptive Top-2 Ensemble (SVR+ET) had the best performance, with RMSE = 1.7696 m and $\boldsymbol{R^2}$ = 0.2983. As more low-performing models were added, performance declined, demonstrating that model selection is more important than ensemble size in determining ensemble quality. Although adaptive weighting generally delivered the best results, the gains over basic and inverse-RMSE weighing were modest.

- In the ideal ML ensemble, just two models (SVR+ET) were used. However, forecasting accuracy gradually decreased beyond the top-2 configuration.
- Ensemble composition had a significantly greater influence than weighting strategy.
- Adaptive Inverse-RMSE weighting was the most successful deterministic aggregation method.

3.7 DL Ensemble Analysis

Table 10 and Figures 10-11 present the performance of the DL ensemble configurations.

Table 10: Performance comparison of hybrid DL ensemble configurations under different ensemble sizes and weighting strategies on the independent test dataset.

Size	**Combination**	**Weighting**	**RMSE**	**MAE**	R^2	**Corr**
Size-2	GRU+TCN	Simple	1.9015	0.6544	0.1856	0.4450
		Inverse	1.9007	0.6524	0.1864	0.4459
		Adaptive	**1.8999**	**0.6506**	**0.1870**	**0.4468**
Size-3	LSTM+GRU+TCN	Simple	1.8962	**0.6357**	0.1902	0.4475

Size	Combination	Weighting	RMSE	MAE	R^2	Corr
		Inverse	1.8962	0.6351	0.1902	0.4475
		Adaptive	**1.8961**	0.6345	**0.1903**	**0.4476**
Full	LSTM+GRU+RNN+TCN	Simple	1.9089	0.7015	0.1793	0.4430
		Inverse	1.9082	0.6996	0.1800	0.4436
		Adaptive	**1.9074**	**0.6977**	**0.1806**	**0.4441**

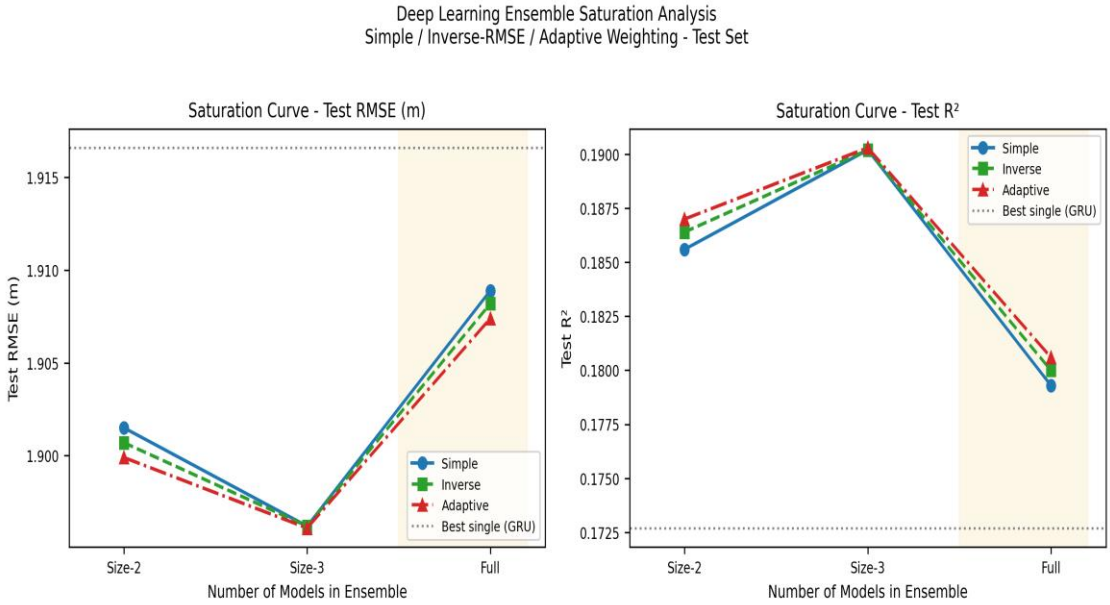


Figure 10: DL ensemble saturation analysis showing the influence of ensemble size on RMSE and R^2 under different weighting strategies. The horizontal dashed line represents the best individual model (SVR)

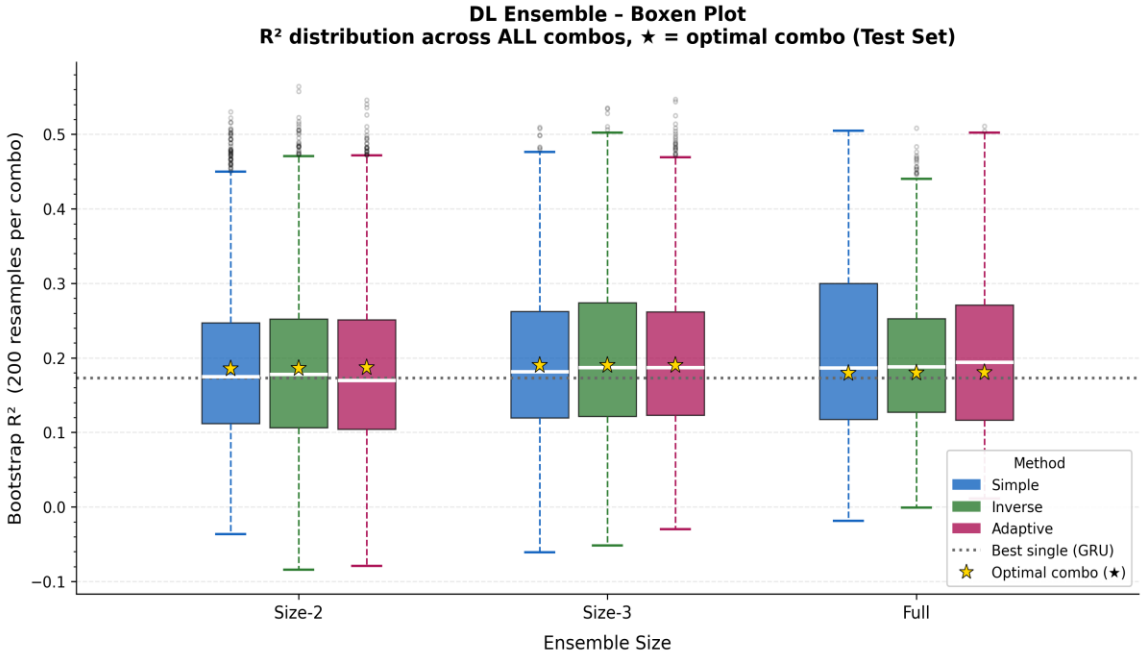


Figure 11: Bootstrap distribution of DL ensemble performance across different ensemble sizes and weighting strategies

The DL ensembles had a weaker saturation pattern. The Adaptive LSTM+GRU+TCN ensemble outperformed other ensembles (RMSE = 1.8961 m; $\boldsymbol{R^2}$ = 0.1903) as the size increased. However, using all four DL models resulted in lower forecasting accuracy, indicating that excessive architectural redundancy limits ensemble efficacy.

- The optimal DL ensemble included three models: LSTM, GRU, and TCN. Moderate diversity across recurrent and convolutional architectures increased predicting performance.
- Larger DL ensembles provide minimal extra predictive information.
- Similar to ML ensembles, adaptive weighting produced the best results.

3.8 Hybrid ML-DL Ensemble Analysis

Table 11 and Figures 12-13 present the performance of the DL ensemble configurations.

Table.11. Performance comparison of hybrid ML-DL ensemble configurations under different ensemble sizes and weighting strategies on the independent test dataset.

Size	Combination	Weighting	RMSE	MAE	R^2	Corr
Size-2	ET+LSTM	Simple	1.7814	0.5190	0.2858	0.5477
		Inverse	1.7805	0.5159	0.2866	0.5479
		Adaptive	**1.7796**	**0.5128**	**0.2873**	**0.5480**
Size-3	ET+SVR+LSTM	Simple	1.7629	0.5014	0.3006	0.5650
		Inverse	1.7625	0.5004	0.3009	0.5648
		Adaptive	**1.7601**	**0.4994**	**0.3012**	**0.5646**
Size-4	ET+XGB+SVR+LSTM	Simple	1.7768	0.5296	0.2975	0.5553
		Inverse	1.7767	0.5290	0.2976	0.5550
		Adaptive	**1.7767**	**0.5284**	**0.2976**	**0.5547**
Size-5	ET+XGB+SVR+LSTM+TCN	Simple	1.7722	0.5369	0.2932	0.5529
		Inverse	1.7713	0.5362	0.2940	0.5532
		Adaptive	**1.7704**	**0.5357**	**0.2946**	**0.5534**
Full	ET+XGB+RF+GB+DT+MLP+ELM+SLP+SVR+LSTM+ GRU+RNN+TCN	Simple	1.8930	0.5576	0.1936	0.4417
		Inverse	1.8786	0.5576	0.2058	0.4541
		Adaptive	**1.8651**	**0.5576**	**0.2172**	**0.4662**

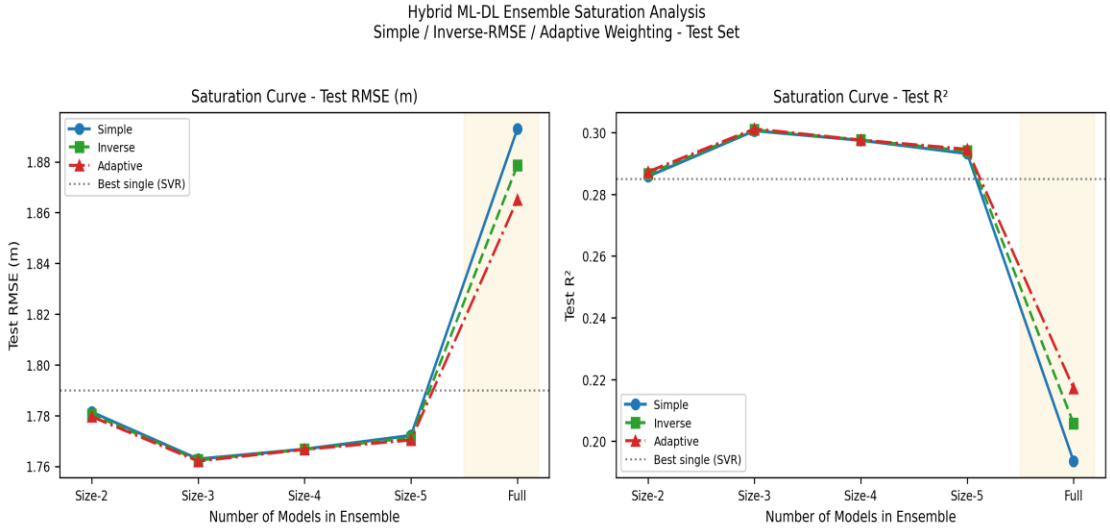


Figure 12: Hybrid ML-DL ensemble saturation analysis showing the influence of ensemble size on RMSE and R^2 under different weighting strategies. The horizontal dashed line represents the best individual model (SVR)

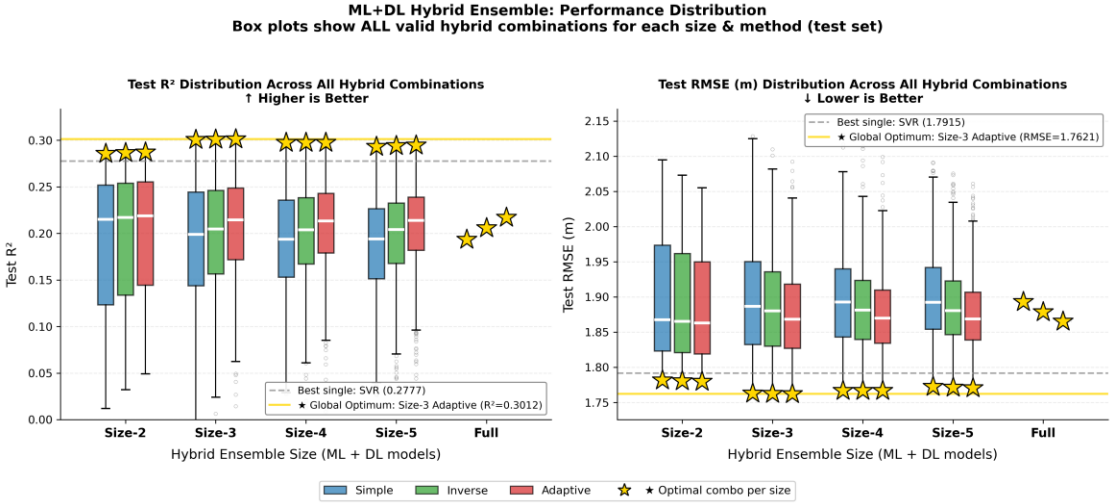


Figure 13: Bootstrap distribution of ML-DL ensemble performance across different ensemble sizes and weighting strategies

The hybrid ensembles had the best overall forecasting performance. The Adaptive Size-3 ensemble (ET+SVR+LSTM) had the lowest RMSE (1.7621 m) and the highest $\boldsymbol{R^2}$ (0.3012), exceeding the top standalone ML and DL ensembles.

Performance improved from Size-2 to Size-3 before gradually declining when more models were added, demonstrating the presence of a hybrid saturation point.

Across all ensemble categories, adaptive inverse-RMSE weighting consistently outperformed Simple Averaging and conventional Inverse-RMSE weighting; nevertheless, the gains were modest. In contrast, ensemble composition had a much greater impact on forecasting accuracy than the weighting approach itself.

The findings also show the presence of an ensemble saturation point, beyond which increasing ensemble size leads to performance deterioration rather than improvement. More importantly, the results show that ensemble diversity is more essential than ensemble size. The largest ensembles did not perform as well as compact ensembles made up of complimentary learners with fundamentally distinct learning mechanisms. The best hybrid ensemble, in particular, integrated a tree-based learner (ET), a kernel-based learner (SVR), and a deep sequential model (LSTM), implying that cross-paradigm diversity is more successful at improving forecasting accuracy than just increasing the number of constituent models.

The Adaptive Size-3 hybrid ensemble (ET+SVR+LSTM) outperformed all other deterministic ensemble configurations, with an RMSE of 1.7601 m and $\boldsymbol{R^2}$ of 0.3012. These findings imply that well chosen diverse models outperform large homogenous ensembles in terms of predictive benefit, emphasizing diversity over ensemble size as the major driver of good ensemble forecasting.

3.9 Benchmark Comparison

To further assess the effectiveness of the proposed ensemble framework, the best hybrid ensemble was compared against the Persistence benchmark, the best standalone model (SVR), and two stacking approaches (Ridge and GBR) trained using OOF predictions from all ML and DL base models.

Table 12 shows that the suggested Adaptive Hybrid ensemble (ET+SVR+LSTM) has the best overall forecasting performance, with the lowest RMSE (1.7621 m), greatest $\boldsymbol{R^2}$ (0.3012), and strongest correlation (0.5646). Compared to the Persistence benchmark, the suggested ensemble lowered RMSE by 16.3%, demonstrating greater predictive capability.

GBR surpassed Ridge Regression among stacking techniques, implying that nonlinear meta-learning is more successful than linear aggregation. However, both stacking strategies were still inferior to the best standalone model (SVR) and the proposed hybrid ensemble. This research implies that incorporating all available predictors using meta-learning does not always result in the best projections.

Table 12: Performance comparison of benchmark, stacking, individual, and hybrid ensemble models on the independent test dataset

Model	**RMSE**	**MAE**	R^2	**Corr**
Persistence	2.1041	1.1066	-0.0002	–
Ridge Stacking	2.0149	0.6341	0.0856	0.3389
GBR Stacking	1.9211	0.5831	0.1688	0.4492
SVR (best individual)	1.7880	0.6610	0.2780	0.5460
Adaptive Hybrid (ET+SVR+LSTM)	**1.7601**	**0.4994**	**0.3012**	**0.5646**

- The Adaptive Hybrid ensemble (ET+SVR+LSTM) achieved the best overall performance.
- The individual SVR model outperformed both Ridge and GBR stacking.
- The proposed hybrid further reduced RMSE by 12.64% and 8.38% relative to Ridge and GBR stacking, respectively, demonstrating the benefit of heterogeneous ensemble integration.
- GBR stacking performed better than Ridge stacking but remained inferior to the proposed hybrid ensemble.
- Carefully selected complementary models from different learning paradigms (tree-based, kernel-based, and deep learning) were more effective than aggregating all available models.
- Model diversity and complementarity contributed more to forecasting accuracy than ensemble size or stacking complexity.

3.10 Diebold-Mariano Test for Predictive Accuracy

The DM test was conducted using squared forecast errors to determine the statistical significance of differences in predictive performance among forecasting models. To account for serial correlation in forecast error differentials, Newey-West heteroskedasticity and autocorrelation consistent (HAC) variance estimation were used, with Holm-Bonferroni correction controlling for repeated pairwise comparisons. Table 8 shows that the suggested ET–SVR–LSTM hybrid exceeded from the persistence benchmark, the best individual model (SVR) ($p < 0.001$) and Ridge stacking ensemble. Despite attaining the lowest RMSE, there was no significant difference in predictive performance between the suggested hybrid and GBR stacking ensembles after multiple-comparison correction ($p > 0.05$).

Table.13: DM test comparing the proposed ET–SVR–LSTM hybrid ensemble with benchmark models on the independent test set.

Comparison	**DM Statistic**	**Raw p-value**	**Adjusted p-value[a]**	**Decision**
Persistence vs. Hybrid	5.631	< 0.001	< 0.001	Significant
SVR vs. Hybrid	3.676	< 0.001	< 0.001	Significant
Ridge Stacking vs. Hybrid	2.644	0.014	0.029	Significant

Comparison	DM Statistic	Raw p-value	Adjusted p-value[a]	Decision
GBR Stacking vs. Hybrid	1.229	0.219	0.219	Not Significant

Note: DM tests were computed using squared forecast errors with Newey–West heteroskedasticity and autocorrelation consistent (HAC) variance estimation. Adjusted ***p***-values were obtained using the Holm–Bonferroni procedure to account for multiple comparisons.

4. Discussion

This study evaluated 13 heterogeneous models from the tree-based, kernel-based, neural network, and DL families, as well as hybrid ensemble techniques, for daily Jhelum River water-level prediction, with a focus on ensemble size and model diversity impacts. GRU and LSTM surpassed RNN and TCN in deep learning architectures, while SVR and Extra Trees (ET) dominated ML models; the hybrid ensemble integrating ET, SVR, and LSTM had the best overall performance.

GRU and LSTM outperform RNN and TCN models in capturing long-term temporal dependencies and delayed basin response (Kratzert et al. 2018; Chung et al. 2014), indicating the importance of temporal memory in water-level prediction. SVR's kernel-based ability to describe nonlinear relationships and ET's variance-reducing randomized construction are likely to explain their good findings, as reported in previous studies of nonlinear flow processes (Abdoulhalik and Ahmed 2024; Kumar et al. 2023). The ET+SVR+LSTM hybrid's superior performance suggests that tree-based, kernel-based, and sequence-learning models capture complementary aspects of river dynamics, consistent with recent work on hybrid ML–DL frameworks (Agaj et al. 2024; Workneh and Jha 2025). Combining heterogeneous learning paradigms improves both accuracy and generalization.

A key finding was an ensemble saturation effect: performance improved only up to an ideal size before worsening with more models (two models for the best ML ensemble, three for the best hybrid ensemble). This suggests that variety and complementarity are more important than ensemble size; if dominant nonlinear and temporal patterns are identified; additional learners offer redundancy rather than new information. Supporting this, the weighting technique had only a little impact on performance as compared to model selection, and the hybrid ensemble beat both Ridge and GBR stacking, implying that a small number of complimentary learners can outperform bigger, more sophisticated ensemble structures. The DM test confirmed statistically significant improvements over the persistence benchmark, the best individual model (SVR) and Ridge stacking ensemble, but differences with GBR stacking ensemble was not statistically significant, indicating comparable predictive accuracy among the strongest ensemble approaches. A limitation of this study is its reliance on a single river basin and a univariate lag-based feature set, which, while effective here, may not generalize to basins with more limited hydrological memory or stronger exogenous influences. Strengths include the leakage-free, rolling-origin validation design and the systematic comparison of ensemble size, diversity, and weighting strategies, which together provide a more rigorous basis for the saturation finding than ensemble size alone would suggest.

These results have practical implications for operational forecasting: the approach is appealing for real-time or resource-constrained deployment since a small, varied ensemble can match or surpass the performance of bigger, more computationally expensive ensembles. Future research should investigate adaptive mechanisms for dynamically choosing complimentary models instead of using predefined ensemble configurations and evaluate whether this saturation effect applies to other regulated river systems.

5. Conclusion

This study proposed a hybrid multi-tier ensemble framework for daily river water-level forecasting that incorporated 13 heterogeneous models from the tree-based, kernel-based, neural network, and deep learning families using deterministic weighting and stacking algorithms. Individual ML and DL models were regularly surpassed by hybrid ensembles, with the Adaptive ET+SVR+LSTM ensemble outperforming all others in terms of forecasting. The investigation also revealed an ensemble saturation effect, in which accuracy improved only up to an optimal ensemble size and then decreased as more models were added. Although adaptive inverse-RMSE weighting regularly outperformed traditional weighting techniques, the improvements were small, demonstrating that model variety and complementarity have a greater influence on predicting performance than weighting strategy alone. The proposed compact hybrid ensemble also outperformed traditional stacking methods, demonstrating that carefully chosen heterogeneous learners are more effective than aggregating a huge number of predictors. However, DM statistical analysis also verified the reliability of the comparative results in that the proposed hybrid significantly outperformed conventional benchmark models and showed predictive accuracy that was comparable to the leading stacking ensembles. These findings provide useful information for developing efficient, robust ensemble forecasting systems for river water level prediction and flood early warning applications. Future study may look into transformer-based designs, uncertainty quantification, and physics-informed hybrid models to increase forecasting accuracy under extreme hydrological circumstances.

References

Abdoulhalik, A., & Ahmed, A. A. (2024). A comparative analysis of advanced machine learning techniques for river streamflow time-series forecasting. *Sustainability, 16*(10), 4005. https://doi.org/10.3390/su16104005

Agaj, T., Budka, A., Janicka, E., & Bytyqi, V. (2024). Using ARIMA and ETS models for forecasting water level changes for sustainable environmental management. *Scientific Reports, 14*(1), 22444. https://doi.org/10.1038/s41598-024-73405-9

Ahmad, S. (1994). Irrigation and water management in the Indus Basin of Pakistan: A country report. In *Irrigation performance and evaluation for sustainable agricultural development* (pp. 190–211).

Al-Mansori, N. J. H., Al-Zubaidi, L. S. A., & Ashoor, A. Z. L. S. (2020). One-dimensional hydrodynamic modeling of the Euphrates River and prediction of hydraulic parameters. *Civil Engineering Journal, 6*(6), 1074–1090. https://doi.org/10.28991/cej-2020-03091532

Al-Sulttani, A. O., Al-Mukhtar, M., Roomi, A. B., Farooque, A. A., Khedher, K. M., & Yaseen, Z. M. (2021). Proposition of new ensemble data-intelligence models for surface water quality prediction. *IEEE Access, 9*, 108527–108541. https://doi.org/10.1109/ACCESS.2021.3102118

Aricò, C., Filianoti, P., Sinagra, M., & Tucciarelli, T. (2016). The FLO diffusive 1D–2D model for simulation of river flooding. *Water, 8*(5), 200. https://doi.org/10.3390/w8050200

Asefa, T., Kemblowski, M., McKee, M., & Khalil, A. (2006). Multi-time scale stream flow predictions: The support vector machines approach. *Journal of Hydrology, 318*(1–4), 7–16. https://doi.org/10.1016/j.jhydrol.2005.06.001

Basharat, M., & Rizvi, S. (2016). *Irrigation and drainage efforts in Indus Basin: A review of past, present and future requirements.*

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

Chowdhury, M. A. H., Goni, M. O., Gazi, M. U., Ahmed, M. S., Jubayer, M. F., Debnath, P., & Sarker, M. A. R. (2025). Meta-learning Model for Low-Data River Water Quality Analysis in The Northeastern Region of Bangladesh.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*. https://doi.org/10.48550/arXiv.1412.3555

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics, 13*(3), 253–263. https://doi.org/10.1080/07350015.1995.10524599

Ebtehaj, I., & Bonakdari, H. (2022). A reliable hybrid outlier robust non-tuned rapid machine learning model for multi-step ahead flood forecasting in Quebec, Canada. *Journal of Hydrology, 614*, 128592. https://doi.org/10.1016/j.jhydrol.2022.128592

Elhanafy, H., & Copeland, G. J. M. (2007). Flash floods simulation using Saint Venant equations. *Proceedings of the International Conference on Aerospace Sciences and Aviation Technology, 12*, 1–14.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Ghimire, G. R., & Krajewski, W. F. (2020). Exploring persistence in streamflow forecasting. *JAWRA Journal of the American Water Resources Association, 56*(3), 542–550. https://doi.org/10.1111/1752-1688.12821

Kader, M. Y. A., Badé, R., & Saley, B. (2020). Study of the 1D Saint-Venant equations and application to the simulation of a flood problem. *Journal of Applied Mathematics and Physics, 8*(7), 1193–1206. https://doi.org/10.4236/jamp.2020.87100

Kedam, N., Tiwari, D. K., Kumar, V., Khedher, K. M., & Salem, M. A. (2024). River stream flow prediction through advanced machine learning models for enhanced accuracy. *Results in Engineering, 22*, 102215. https://doi.org/10.1016/j.rineng.2024.102215

Khan, U., Jamshed, R., Tahir, A. A., *et al.* (2025). Anticipating future hydrological changes in the northern river basins of Pakistan: Insights from the snowmelt runoff model and an improved snow cover data. *Water, 17*(14), 2104. https://doi.org/10.3390/w17142104

Khozani, Z. S., Precht, E., & Ionita, M. (2025). Weekly streamflow forecasting of the Rhine River based on machine learning approaches. *Natural Hazards, 121*(4), 4135–4153. https://doi.org/10.1007/s11069-024-06962-x

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences, 22*(11), 6005–6022. https://doi.org/10.5194/hess-22-6005-2018

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. S. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences, 23*(12), 5089–5110. https://doi.org/10.5194/hess-23-5089-2019

Kumar, V., Kedam, N., Sharma, K. V., Mehta, D. J., & Caloiero, T. (2023). Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models. *Water, 15*(14), 2572. https://doi.org/10.3390/w15142572

Li, Y., Liang, Z., Hu, Y., Li, B., Xu, B., & Wang, D. (2020). A multi-model integration method for monthly streamflow prediction: Modified stacking ensemble strategy. *Journal of Hydroinformatics, 22*(2), 310–326. https://doi.org/10.2166/hydro.2019.092

Lutz, A. F., Immerzeel, W. W., Kraaijenbrink, P. D. A., Shrestha, A. B., & Bierkens, M. F. P. (2016). Climate change impacts on the upper Indus hydrology: Sources, shifts, and extremes. *PLOS ONE, 11*(11), e0165630. https://doi.org/10.1371/journal.pone.0165630

Parasar, P., Moral, P., Srivastava, A., Krishna, A. P., Majumdar, S., Bhattacharjee, R., Mishra, A. P., Mustafi, D., Rathore, V. S., Sharma, R., & Mustafi, A. (2025). Integrating genetic algorithms and machine learning for spatiotemporal groundwater potential zoning in fractured aquifers. *Journal of Hydrology: Regional Studies, 62*, 102800. https://doi.org/10.1016/j.ejrh.2025.102800

Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall–runoff modelling. *Journal of Hydrology, 199*(3–4), 272–294. https://doi.org/10.1016/S0022-1694(96)03330-6

Sun, X., Zhang, H., Wang, J., Shi, C., Hua, D., & Li, J. (2022). Ensemble streamflow forecasting based on variational mode decomposition and long short-term memory. *Scientific Reports, 12*(1), 518. https://doi.org/10.1038/s41598-021-03725-7

Tadesse, K. B., & Dinka, M. O. (2017). Application of SARIMA model to forecasting monthly flows in Waterval River, South Africa. *Journal of Water and Land Development, 35*(1), 229–236. https://doi.org/10.1515/jwld-2017-0088

Workneh, H., & Jha, M. (2025). Utilizing hybrid deep learning models for streamflow prediction. *Water, 17*(13), 1913. https://doi.org/10.3390/w17131913

Xu, Y., Hu, C., Wu, Q., Li, Z., Jian, S., & Chen, Y. (2021). Application of temporal convolutional network for flood forecasting. *Hydrology Research, 52*(6), 1455–1468. https://doi.org/10.2166/nh.2021.050

Declaration

Competing Interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of AI Use: The authors used artificial intelligence tools to improve language, grammar, readability, and manuscript organization. The AI tools were not used to generate research ideas, research data, analyses, experimental results, or conclusions. All content was critically reviewed, verified, and approved by the authors, who take full responsibility for the accuracy and integrity of the manuscript.

Funding Statement: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethical Approval & Consent to Participate: Not Applicable. This study used secondary hydrological data and did not involve human participants or animals.

Author Contribution Statement: Aqsa Saleem developed the methodology, performed data analysis, implemented the machine learning and deep learning models, and wrote the manuscript. Hafiza Mamona Nazir supervised the research, validated the methodology, reviewed the manuscript, and provided academic guidance. Muhammad Zubair contributed to data interpretation, manuscript review, and technical improvements.

Acknowledgments: The authors are thankful to the Department of Statistics and the Interdisciplinary Research Center for One Health, University of Sargodha, for providing academic support during this research.

Author(s) Bio / Authors' Note

Aqsa Saleem *is an MPhil Statistics scholar in the Department of Statistics, University of Sargodha, Pakistan. Her research interests include statistical learning, machine learning, deep learning, hybrid ensemble modeling, time-series forecasting, hydrological prediction, and environmental data analytics. Her current research focuses on developing intelligent predictive frameworks for water-resource management using machine learning and deep learning techniques. Email:* fatima.zahra4446@gmail.com

Hafiza Mamona Nazir *is an Assistant Professor at the Research Center, University of Sargodha, Pakistan. Her research interests include geostatistics, spatial data analysis, environmental statistics, applied machine learning, and interdisciplinary data-driven research. She has supervised numerous research projects in statistical modeling, spatial analytics, and predictive data analysis. Email:* mamona.nazir@uos.edu.pk

Muhammad Zubair *is an Associate Professor in the Department of Statistics, University of Sargodha, Pakistan. His research interests include sampling theory, probability theory, machine learning, deep learning, quality and reliability methods, survival analysis, and applied statistical modeling. His research focuses on developing advanced statistical methodologies and intelligent data-driven solutions for real-world applications. Email:* zubair.bakhsh@uos.edu.pk