Hybrid AdaBoost and CatBoost Models Enhanced with HGS and CGO for Non-Stationary Runoff Forecasting

Huiying Shao
Hebei Vocational University of Technology and Engineering, Xingtai 054000, China
E-mail: shycongcong@163.com

Keywords: rainfall-runoff, hybrid model, machine learning (Ml), CatBoost, AdaBoost, HGBoost, chaos game optimization, hunger games search

Received: March 17, 2025

This study presents a hybrid forecasting model for non-stationary daily runoff series, integrating state-of-the-art machine learning algorithms, CatBoost, AdaBoost, and HGBoost, with metaheuristic optimization techniques, HGS and CGO. The model was evaluated using a dataset consisting of 29085 from Russian River Basin dataset. Performance was assessed using key metrics such as RMSE (6.79), R² (0.9549), and error scatter. Results show that the hybrid AdaBoost-HGS model outperforms other approaches, achieving the highest R² of 0.9549 and the lowest error, followed by the AdaBoost-CGO model. Although the CatBoost-HGS model showed promise, it was less effective than the other schemes. Furthermore, the study highlights the faster convergence and efficiency of AdaBoost-based models in terms of both prediction performance and computational time. These findings underscore the potential of hybrid ML models in enhancing runoff prediction accuracy and efficiency, which is crucial for applications in flood control and water resource management.

Povzetek: Študija kaže, da hibridni modeli strojnega učenja lahko zelo učinkovito in natančno napovedujejo dnevne pretoke, kar je pomembno za upravljanje vodnih virov in poplavno varnost.

1 Introduction

Runoff forecasting, particularly the accurate prediction of daily runoff, remains a critical and challenging issue in hydrology. Reliable and precise runoff forecasts are crucial for flood management, disaster mitigation, efficient water resource distribution, and optimal reservoir operations [1]. Daily runoff patterns are characterized by considerable non-linearity, non-stationarity, and randomness, resulting from the interplay of meteorological conditions, watershed traits, and geographical factors. Ensuring reliable and precise prediction tactics remains a crucial focus for investigators in this domain [2–4].

There are 3 primary categories of runoff schemes: conceptual, physically based, and data-driven [5,6]. Data-driven schemes, whether empirical or black-box, use statistical techniques, ML techniques, and vast amounts of historical data to ascertain the correlations influencing runoff. These schemes allow for runoff forecasts based on several variables over time by statistically characterizing the runoff dynamics that determine catchment responses to rainfall [7–10]. Constructed as connected systems that replicate hydrological processes over several geographical and temporal scales, conceptual schemes differ from data-driven schemes in this regard [11,12]. Physically-based schemes, however, utilize mathematical equations and interactions between hydrological processes to describe how rainfall is converted into runoff [8,13].

Runoff modeling initially focused on conceptual schemes from the 1960s to the early 2000s. Recently, this field has evolved with the introduction of data-driven tactics, especially those leveraging ML approaches. Conceptual schemes, like the Sacramento [14], $IHACRES$ [10], PDM [15], and HBV [16], generally require fewer parameters than physically-based schemes such as $MIKE-SHE$ [17] , $SWAT$ [18], and $TOPMODEL$[9]. Predictions may be made very instantly as a consequence of the decreased processing requirements. Although conceptual schemes use fewer computing resources and are easier to build and apply, they may not always translate into higher performance due to their complexity [19]. Prediction accuracy is more reliant on hydrological inputs (e.g., rainfall, runoff, temperature) and state variables like initial catchment wetness.

Typically, process-driven prediction tactics [20,21] demand complex mathematical schemes, a deep understanding of the physical mechanisms behind runoff formation, comprehensive hydrological and meteorological data, and occasionally subjective judgment. These requirements can impose significant challenges in practical applications, often resulting in suboptimal model performance and increased uncertainty [22]. In the context of managing and operating hydropower stations and reservoirs, short-term planning, such as daily and weekly dispatching, relies heavily on accurate daily runoff forecasts. Due to the brief forecasting horizon, data-driven schemes are preferred to achieve the high accuracy and reliability needed for these predictions [23].

Numerous investigators have utilized time series smoothing techniques for predicting hydrological time series. Myronidis et al. [24] investigated short-term anticipation of streamflow and hydrological drought in Cyprus, utilizing data from eleven hydrometric stations over 34 years. Using the Streamflow Drought Index (SDI) and ARIMA schemes, they found a decreasing trend in annual streamflow and an intensification of drought conditions over time. Predictions indicated mostly non-drought conditions shortly, with mild droughts expected in specific catchments.

These schemes generally presume a linear correlation between input and output series, yet runoff relationships are often highly nonlinear. This limitation means they neglect the inherent nonlinear traits of runoff data, resulting in less accurate predictions [25]. Fortunately, with the introduction of advanced intelligent schemes, runoff forecasting has improved remarkably. Compared to conventional regression methods, the innovative strategies have improved accuracy and efficiency significantly, enhancing the theoretical concept of runoff processes.

In the last decade, ML techniques have become progressively more popular within hydrological modeling, further enhancing prediction performance. Bairami et al. [26] examined groundwater behavior in Ardabil Plain using Modflow and a hybrid CatBoost-AOA model from October 2010 to March 2021. The CatBoost-AOA model demonstrated the highest correlation (0.9977) and predicted a groundwater level decline of 0.85 m by 2031, compared to Modflow's prediction of 0.77 m. The exploration highlights CatBoost-AOA as a concise and practical method for accurate groundwater anticipation, aiding sustainable development planning. Bray et al. [27] discussed the use of SVM schemes in flood forecasting, focusing on suitable model forms and parameters for rainfall-runoff modeling. They found that it is unrealistic to conduct optimal parameter searches due to size and sensitivity issues, and therefore, manual operations are necessary. In comparing SVM with transfer function schemes, they noted that transfer function schemes perform better than SVM for short-range forecasts, which calls for further research in model identification procedures. Granata et al. [28] compared SVM-based rainfall-runoff modeling with the EPA’s SWMM in 2 northern Italian basins. SVR showed potential in urban hydrology, matching SWMM in hydrograph shape and timing but underestimating peak discharge. Both schemes demonstrated comparable performance in modeling total runoff and hydrograph traits, highlighting SVR's application potential despite current calibration limitations. Hosseini et al. [29] developed the SVR-GANN model by combining Support Vector Regression (SVR) with a GANN. They evaluated its performance in simulating daily runoff across 3 sub-basins in a semi-arid region of Iran. Comparisons with other schemes like ANN-BP, traditional SVR, ANN-GA, ANFIS, and GANN highlighted SVR-GANN's superior accuracy in predicting hydrograph features and overall reliability as a rainfall-runoff modeling tool. Zhou et al. [30] presented the Genetic Algorithm and Least Square Estimator-optimized R-ANFIS (GL) model for flood forecasting multi-steps ahead. It was utilized on the 3 Gorges Reservoir and contrasted with dynamic and static ANFIS schemes, having superior accuracy and addressing time-lag issues within the non-stationary process of rainfall-runoff. It was extremely trustworthy and efficient in improving the forecasting accuracy. Safari et al. [31] employed the Regression in the Reproducing Kernel Hilbert Space (RRKHS) methodology to rainfall-runoff modeling for the first time. RRKHS with daily data in a Turkish mountainous catchment outperformed radial basis function ANNs and multivariate adaptive regression splines with highly precise peak streamflow estimation accuracy. The study emphasizes RRKHS's high potential for future environmental modeling applications due to its good handling of nonlinear data. Mohammadi et al. [32] suggested novel boosted frameworks for the improvement of streamflow modeling based on daily hydrometric station data across Canada and the United States. They combined MLP with PSO, PSO-multi-verse optimizer (PSOMVO), and bi-linear (BL) time series frameworks. The MLP-BL improved frameworks were better than the others, having greater accuracy than conventional MLP and BL frameworks.

Flooding is one of the most devastating natural disasters, causing fatalities and threatening various sectors. There are 3 primary types of floods: river, coastal, and flash. Flash floods occur due to rapid runoff from intense rainfall, quickly raising water levels in streams or normally dry channels, often within less than 6 hours. Of these, flash floods are responsible for the highest number of flood-related deaths worldwide. Investigators and policymakers concur that managing flash flood risks is crucial to mitigating diverse losses. Effective rainfall-runoff modeling is essential for pinpointing high flood-risk areas. Ahmed [33] highlights the limitations of current flood forecasting schemes due to the complexity and diversity of flood systems. The exploration emphasizes the necessity of integrating 1D and 2D hydraulic schemes to better handle spatially distributed hydrographs and complex watershed traits. It concludes that careful assessment and the right model selection are essential for accurate flood risk identification and management. Zhang et al. [34] conducted a numerical flash flood hazard evaluation in the Hadahe River Basin, northern China, using integrated rainfall-runoff (HEC-HMS) and hydraulic (FLO-2D) models. The study mapped high flash flood hazards with large inundation areas, particularly downstream, under varied rainfall conditions ranging from 5-year to 1000-year return periods. The method can be applied to other mountain basins for effective flash flood protection and disaster mitigation.

Traditional runoff modeling approaches are commonly divided into three categories: conceptual, physically based, and data-driven schemes. Conceptual models, such as the HBV or SAC-SMA frameworks, represent hydrological processes through simplified water balance equations calibrated to specific basins. Physically based models (e.g., SWAT, MIKE SHE) aim to simulate the underlying physics of flow, infiltration, and evapotranspiration using distributed parameters, but they require extensive field measurements and high computational resources. In contrast, data-driven approaches rely purely on observed historical data to learn nonlinear relationships between input variables and runoff, without explicit physical assumptions. The present study focuses exclusively on the data-driven category because the primary objective is to evaluate the capability of modern hybrid machine learning models in capturing complex hydrological patterns directly from data. The conceptual and physically based models serve as complementary approaches with well-known strengths in process interpretation and physical consistency; however, their calibration requirements and limited adaptability to sparse data environments make them less suited for large-scale or near-real-time prediction tasks addressed here. This distinction clarifies the methodological scope of the study and establishes the rationale for focusing on advanced data-driven techniques.

Even though hydrologic and hydraulic modeling are well-loved globally, their output needs more study to understand watershed flow processes. Therefore, choosing the proper model types or blending different schemes along with professional experience can be an effective flood hazard assessment approach. Table 1 provides a comparative Summary of related works.

Table 1: Comparative summary table for related works

Method	Dataset	Evaluation Metrics	Performance	Limitations/Shortcomings
ARIMA, SDI (Myronidis et al. [24])	Data from 11 hydrometric stations (34 years)	Streamflow Drought Index (SDI), Prediction Accuracy	Identified a decreasing trend in streamflow and an intensification of drought conditions	Assumes linear correlation, less accurate for non-linear runoff data. Does not capture complex hydrological processes.
Bairami et al. [26] (CatBoost-AOA)	Groundwater data from Ardabil Plain (Oct 2010-Mar 2021)	Correlation coefficient	CatBoost-AOA model (0.9977 correlation)	Focuses on groundwater modeling, not general runoff processes. Limited to groundwater level prediction.
SVM (Bray et al. [27])	Rainfall-runoff data	RMSE, Prediction Accuracy	SVM struggled with optimal parameter searches; manual tuning was needed	Poor performance for short-range forecasts. SVM may require substantial manual effort.
SVR-SWM (Granata et al. [28])	Northern Italian basin data	Hydrograph shape, Total runoff	Comparable performance to SWMM	Underestimates peak discharge, limited calibration flexibility.
SVR-GANN (Hosseini et al. [29])	Daily runoff data from Iran's semi-arid regions	MAE, RMSE, NSE	SVR-GANN outperforms ANN-BP, SVR, ANN-GA, and ANFIS	Focused on daily runoff prediction; lacks focus on long-term prediction accuracy.
GL model (Zhou et al. [30])	3 Gorges Reservoir flood forecasting data	Forecast accuracy	GL model outperforms dynamic/static ANFIS	Efficient but focused only on flood forecasting; lacks flexibility for diverse hydrological systems.
RRKHS (Safari et al. [31])	Turkish mountainous catchment data	Peak streamflow estimation accuracy	RRKHS outperforms RBF-ANNs, MARS	Limited to rainfall-runoff modeling in mountainous regions, it may not generalize across different terrains.
Mohammadi et al. [32] (MLP-PSO)	Data from Canada and the U.S. (hydrometric stations)	RMSE, R^2	The MLP-BL improved framework showed greater accuracy	Focuses on streamflow modeling; may not capture dynamic rainfall-runoff relationships.
1D and 2D hydraulic schemes (Ahmed [33])	Flood forecasting data	Accuracy of model predictions	Focuses on integrating 1D and 2D hydraulic schemes	Does not directly address runoff prediction or the impact of hydrological variables on runoff.
Flash flood hazard evaluation (Zhang et al. [34])	Hadahe River Basin, northern China	Flood hazard levels, Inundation area	High flash flood hazard identification	Focused more on hazard evaluation than runoff prediction, limited to a specific geographic region.

The hybrid AdaBoost-HGS model presents several key improvements over existing state-of-the-art (SOTA) methods. First, in terms of convergence rate, the AdaBoost-HGS model demonstrates significantly faster convergence compared to methods like ARIMA and LSSVM, reducing the time required for predictions. This is particularly advantageous for real-time applications where quick results are essential. In short-term forecasting, the AdaBoost-HGS model excels by capturing non-linear relationships more effectively than models like SVM and RRKHS. While RRKHS is known for handling non-linear data well, it struggles with efficiency and speed, especially for short-term predictions. On the other hand, the AdaBoost-HGS model maintains high accuracy and offers faster convergence. In terms of computational efficiency, AdaBoost-HGS converges more quickly and provides better prediction performance than models like SVR-GANN and SVM, which can be slow to scale with larger datasets due to lengthy training times. Finally, the robustness of the AdaBoost-HGS model is improved by overcoming the parameter tuning challenges and lack of flexibility that limit traditional methods such as SVR-GANN and SVM, thus achieving higher precision and scalability for runoff forecasting.

1.1 Research questions and goals

This study aims to explore the performance of hybrid machine learning (ML) models for predicting non-stationary daily runoff series, specifically focusing on the integration of AdaBoost, CatBoost, and HGBoost models with optimization strategies such as Hunger Games Search (HGS) and Chaos Game Optimization (CGO). To ensure a structured approach to the research, the following research questions (RQs) have been formulated:

- **RQ1: Can hybrid ML models outperform single learners in short-term runoff forecasting?**
  This question investigates whether the integration of multiple machine learning models (AdaBoost, CatBoost, HGBoost) with optimization strategies (HGS, CGO) provides superior performance compared to traditional single learner models like AdaBoost or CatBoost alone. The hypothesis is that combining these models with metaheuristic optimizers will enhance prediction accuracy, especially for short-term forecasts.
- **RQ2: Do HGS and CGO significantly improve model convergence and performance?** This question focuses on the impact of the two optimization techniques, HGS and CGO, on the convergence speed and overall performance of the hybrid models. The goal is to evaluate whether the use of these metaheuristic optimizers leads to faster convergence rates and higher predictive accuracy compared to models without optimization, or models optimized by other techniques.

The intended outcome of this study is to achieve high-precision, low-latency runoff prediction, which is crucial for real-time applications in flood management and water resource planning. By addressing these research questions, the study aims to provide insights into the effectiveness of hybrid ML models with optimization techniques for improving runoff forecasting accuracy and efficiency.

1.2 Main contributions

This paper presents an introduction to some novel aspects of runoff forecasting using ML and combined modeling approaches. The major novelties of the study are:

1) Combination of Advanced ML Schemes with Improvement Strategies: The study is a pioneer in combining CatBoost, AdaBoost, and HGBoost schemes with advanced improvement strategies, i.e., HGS and CGO. This combined strategy is a revolutionary change from the conventional runoff forecasting strategies, with a superior framework for improving prediction efficiency and accuracy.
2) Hybrid AdaBoost-HGS Model Development: The other major highlight of the experimentation is the development and evaluation of the AdaBoost-HGS hybrid model. The scheme achieved the highest R² among the tested models. Such success confirms the promise of the AdaBoost-HGS combination for high prediction accuracy in runoff estimation.
3) Comprehensive Comparison and Analysis: The report provides a detailed comparative analysis of the productivity of individual schemes (CatBoost, AdaBoost, HGBoost) and their hybrid combinations. This involves a detailed analysis of various metrics.
   The detailed comparison and analysis of these schemes is valuable in providing insights into their relative strengths and weaknesses.
4) Understanding Boostment Methods' Influence: The research significantly points to the influence of enhancement strategies in determining model performance. In illustrating how HGS and CGO enhance accuracy and efficiency in mixed schemes; the research emphasizes the necessity for sophisticated parameter tuning in the achievement of optimal predictive outcomes.
5) Practical Hydrological Management Implications: In demonstrating the performance of advanced blended schemes in predicting runoff, the study provides new insights into their practical uses. These include improved flood management, water resource allocation, and stormwater infrastructure design, offering a significant enhancement in hydrological modeling practice.

In short, the article's novelty lies in its new hybrid modeling approach, its advanced comparative analysis, and its practical application to improving runoff forecasting accuracy and efficiency. These are important advances in hydrology and ML.

2 Methodology

This exploration employs 3 sophisticated schemes, XGBoost, SVR, and HGBoost, to conduct predictive modeling of streamflow data. All the schemes are optimized using the assistance of 2 sophisticated improvement strategies: the Algorithm Inspired by AEO and the SMA. These optimizers are employed to maximize the productivity of the forecast schemes. The overall number of 6 distinct schemes is constructed for forecasting. Each model is evaluated and compared using diverse statistical evaluation approaches to determine their performance and validity in streamflow forecasting. The database used in this study is derived from the research of [35]. It has daily readings for the year 1914 and includes multiple input features such as month, day, and streamflow with nine lag steps. It is difficult to time series data due to the natural complexity of short-term streamflow forecasting. It is particularly difficult to forecast in the short term, as streamflow data is extremely volatile and is influenced by a large number of factors. However, accurate short-term forecasting is of crucial significance to power company operation and dam operation, where timely and accurate forecasting is required for the proper regulation of water resources and operation planning. In this research, streamflow data with time lags of (t-1) to (t-9) are considered. Table 2 presents a clear overview of the fundamental variables affecting the forecast. The database is separated into training and testing datasets. Before training the schemes, the data is qualitatively analyzed using metrics such as the correlation matrix to understand the connection between different input parameters and how they impact the output. This analysis helps to identify the significance of each parameter and its influence on the forecasting outcome. After this qualitative analysis, the schemes are trained and used to predict. The individual performance of every model is then evaluated based on various performance metrics. These metrics evaluate the accuracy, reliability, and generalizability of the schemes. The output from different schemes is pitted against one another to determine which algorithm provides the most accurate and reliable predictions. Fig. 1 illustrates the scheming process, presented in the form of a flow chart. This visual representation outlines the steps involved in data preparation, model training, evaluation, and comparison, providing a clear overview of the methodological approach used in this exploration.

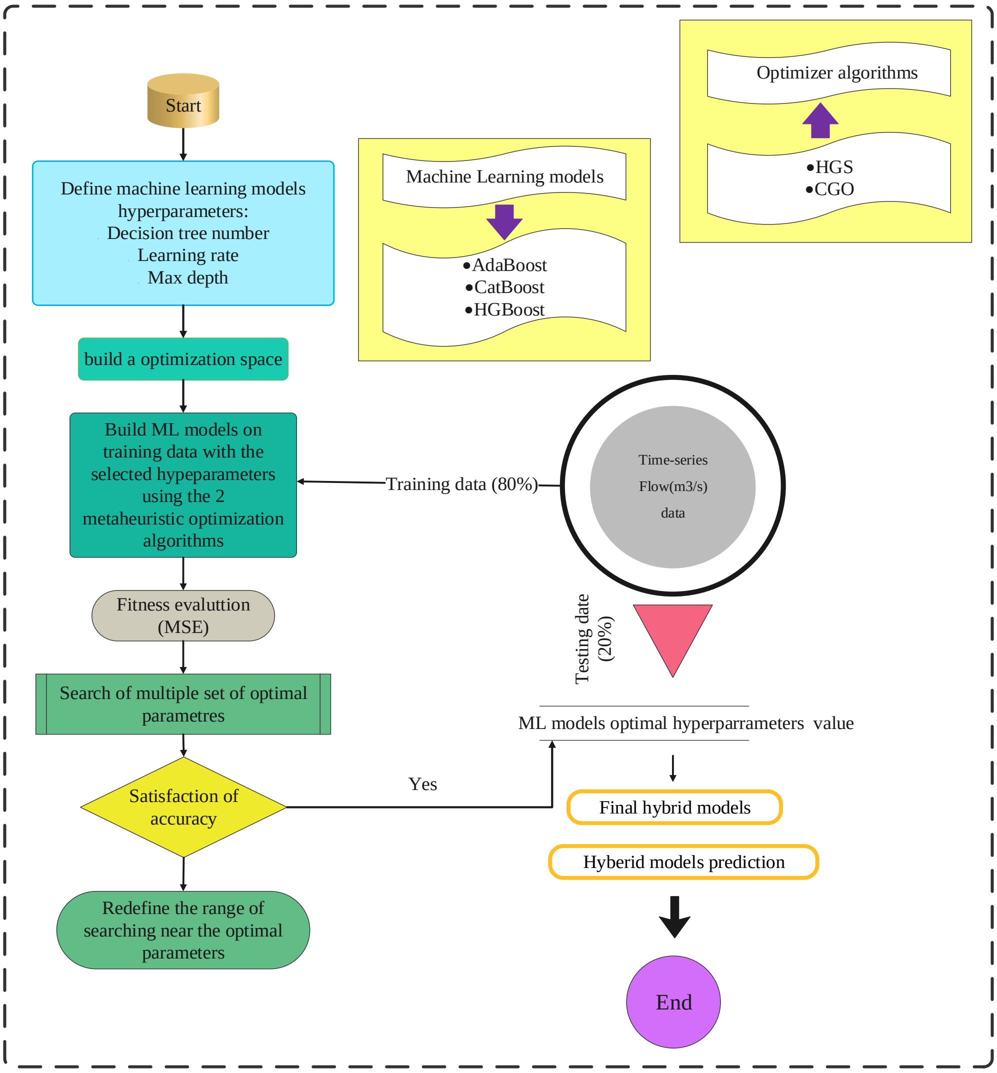

Start
Define machine learning models hyperparameters:
Decision tree number
Learning rate
Max depth
build a optimization space
Build ML models on training data with the selected hypeparameters using the 2 metaheuristic optimization algorithms
Fitness evaluttion (MSE)
Search of multiple set of optimal parametres
Satisfaction of accuracy
Yes
Redefine the range of searching near the optimal parameters
Machine Learning models
•AdaBoost
•CatBoost
•HGBoost
Optimizer algorithms
•HGS
•CGO
Time-series Flow(m3/s) data
Training data (80%)
Testing date (20%)
ML models optimal hyperparrameters value
Final hybrid models
Hyberid models prediction
End

Figure 1: Diagram of this exploration

Table 2: The input variables and their numerical details

	count	*mean*	*std*	*min*	25%	50%	75%	*max*
Year	**29085**	**1976.706**	**24.89086**	**1914**	**1958**	**1978**	**1998**	**2017**
Day	**29085**	**15.74609**	**8.804217**	**1**	**8**	**16**	**23**	**31**
Month	**29085**	**6.550627**	**3.437402**	**1**	**4**	**7**	**10**	**12**
Flow(m^3/s) (t-1)	**29085**	**23.83698**	**39.9274**	**0.142**	**4.31**	**7.27**	**21.9**	**476**
Flow(m^3/s) (t-2)	**29085**	**23.84012**	**39.92953**	**0.142**	**4.31**	**7.27**	**21.9**	**476**
Flow(m^3/s) (t-3)	**29085**	**23.84358**	**39.93229**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-4)	**29085**	**23.84675**	**39.93447**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-5)	**29085**	**23.85085**	**39.93863**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-6)	**29085**	**23.85549**	**39.94428**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-7)	**29085**	**23.86093**	**39.95246**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-8)	**29085**	**23.86696**	**39.96279**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t-9)	**29085**	**23.87183**	**39.96916**	**0.142**	**4.31**	**7.28**	**21.9**	**476**
Flow(m^3/s) (t)	**29085**	**23.83381**	**39.92522**	**0.142**	**4.31**	**7.27**	**21.9**	**476**

2.1 ML techniques

This exploration utilized sophisticated ML schemes, specifically CatBoost, AdaBoost, and HGBoost, for runoff prediction. To boost the schemes' accuracy and flexibility, blended schemes were developed by integrating enhancement tactics like HGS and CGO. This part offers a comprehensive review of the mathematical foundations and core rules behind each of these methodologies, emphasizing their unique contributions to improving the predictive performance of the blended schemes.

2.1.1 HGBR

HGBR is an advanced ML tactic that enhances the efficiency of gradient boosting by integrating histogram-based tactics to handle numerical attributes. This approach progressively constructs a sequence of decision trees (DT) such that each subsequent tree is specifically constructed to correct the errors injected by the previous trees. HGBR's primary innovation is in using the application of binning or discretization. By binning continuous numerical attributes into bins, the training process is significantly accelerated. This binning approach reduces computation costs for comparing and sorting continuous values, accelerating each cycle and being simpler to scale to large databases [36].

The Charge Gradient Optimization (CGO) and Hybrid Gravity Search (HGS) algorithms were selected for their complementary search mechanisms, which are well-suited for high-dimensional, nonlinear optimization problems. CGO, based on electrostatic interactions, enhances exploration and aids in escaping local minima, while HGS strikes a balance between exploration and exploitation, converging rapidly toward the global optimum. Both algorithms are highly effective for tuning machine learning model parameters, which often involve nonconvex and initial-condition-sensitive objective functions. Their integration into the hybrid model ensures stable convergence and diverse searches, allowing for a comparative analysis of electrostatic (CGO) and gravitational (HGS) optimization approaches to assess their impact on prediction accuracy and robustness.

Compared to conventional optimizers such as Particle Swarm Optimization (PSO), Differential Evolution (DE), and Bayesian Optimization, the choice of CGO and HGS was driven by their superior adaptability to complex, nonlinear search spaces and their ability to balance exploration and exploitation. PSO and DE often suffer from premature convergence in multidimensional, nonconvex, or multimodal parameter tuning. While Bayesian Optimization is efficient for smooth functions, it becomes computationally intensive and less effective in highly irregular search spaces with many hyperparameters. In contrast, CGO utilizes charge-based attraction–repulsion forces to promote diverse movement and broader search coverage, while HGS employs gravitational interaction and adaptive velocity control to achieve stable convergence near the global optimum without stagnation.

2.1.1.1 Key features and benefits

- **Efficiency:** Histograms allow for faster computation, and the training time is much faster than in traditional gradient-boosting methods.
- **Scalability**: HGBR is very scalable for large databases as the binning step reduces the amount of data to be processed at each step.
- **Accuracy**: HGBR, even though it is fast, has high accuracy by successfully reducing the prediction errors in every cycle.

2.1.1.2 Implementation in sci-kit-learn

The sci-kit-learn ML library offers HGBR support through its HistGradientBoostingRegressor and HistGradientBoostingClassifier classes. These employ the histogram approach to improve performance. As explained by sci-kit-learn, this tactic is much faster than the library's default GBR uptake.

All algorithms were developed and executed within a unified computational environment to ensure consistency and reproducibility. The implementation framework was based on Python 3.10, using the Scikit-Learn (v1.4) library for the AdaBoost and CatBoost baseline models, and LightGBM and XGBoost packages for gradient boosting benchmarks where applicable. The hybrid models, including the proposed optimization-enhanced configurations (CGO- and HGS-based), were implemented in the same Python environment, with the optimization algorithms coded from first principles using NumPy and SciPy libraries to maintain compatibility with Scikit-Learn's model interface. Data preprocessing, parameter tuning, and performance evaluation were also conducted in Python using Pandas for data handling and Matplotlib for result visualization.

In this process, the base learner generates initial predictions, and the optimization layer continuously updates its parameters based on the model's performance. The procedure can be summarized as:

1. Initialize the base model and optimization population.
2. Train and evaluate each candidate solution based on RMSE.
3. Update the candidates using the CGO or HGS optimization rules.
4. Select the best parameter set and update the model.
5. Repeat until convergence.

This custom hybridization allows dynamic tuning of model parameters and avoids the limitations of static stacking methods. All steps were implemented using NumPy, SciPy, and Scikit-Learn functions to maintain full compatibility.

2.1.2 CatBoost

It is a powerful ML framework that uses gradient-boosting techniques for a variety of tasks with high performance on categorization and regression tasks. CatBoost is offered as an open-source library, famous for its performance in execution time and memory usage compared to other schemes. CatBoost is predominantly distinguished by its clever handling of categorical features, which typically performs better than other techniques. CatBoost was created by Yandex and includes significant advances in parallel processing, which enable it to train faster and more easily to implement, especially on internet networks [37]. CatBoost also includes enhancements for addressing overfitting issues. The algorithm performs a random permutation of the database, and for each instance, it gauges the mean label value among instances with the same category value that appears earlier in the permutation. If the permutation is:

$$\Theta = [\sigma_1, \dots, \sigma_n]_n^T \tag{1}$$

$$X_{\sigma_{p,k}} = \frac{\sum_{j=1}^{p-1}[X_{\sigma_{j,k}} = X_{\sigma_{p,k}}]Y_{\sigma_s} + \beta . P}{\sum_{j=1}^{p-1}[X_{\sigma_{j,k}} = X_{\sigma_{p,k}}] + \beta} \tag{2}$$

In regression tasks, the prior value (P) is usually calculated by finding the average label value of the database. This prior value is then given a weight (w) to reflect its importance in the regression scheme. By using this average value, the regression model establishes a baseline for making predictions relative to the database's overall average label value [38].

2.1.3 Adaptive Boosting Regression (AdaBoost)

AdaBoost is an ensemble method rooted in gradient boosting principles, and it operates without needing initial assumptions about the weak learners' performance. It uses decision tree structures, particularly producing a series of simple trees known as stumps. A stump is a minimal tree with just one node and 2 leaves, serving as a weak learner with low predictive accuracy. While each stump contributes to the final prediction, its vote's influence varies according to the error associated with its weights (w). AdaBoost functions sequentially, with earlier stumps impacting the subsequent ones, progressively minimizing error residuals to achieve the best possible outcome. In each cycle, the cumulative weight or vote for a feature in the database is derived from the total error, et, which showcases the sum of the weights of misclassified samples. The overall vote for each stump is calculated accordingly [39,40].

$$e_t = w_1 + w_2 + \dots + w_i \tag{3}$$

For a database with i total entries, the vote is calculated using the following equation:

$$v = \frac{1}{2} log \frac{1 - e_t}{e_t} \tag{4}$$

A random number between 0 and 1 determines which feature in the database will have the most influence on the next stump. This process of the cycle persists until a predetermined threshold is satisfied. Unlike Random Forest, which builds trees simultaneously, AdaBoost systematically constructs its stumps, each being a simple learner with only one level of depth. In AdaBoost, schemes are developed in sequence, with each new model depending on the completion and error assessment of the previous one. This sequential approach can make AdaBoost quite time-consuming for large databases. While it progressively adapts and learns, similar to other gradient-boosting tactics, it requires more time to develop effective regressors.

2.1.4 Chaos game optimization

It is derived from chaos theory principles, using chaos game mechanics to generate fractals that exhibit self-similarity. Inspired by chaos theory, which examines the randomness in complex systems and their sensitivity to initial conditions, the CGO algorithm utilizes game theory to create fractals [41]. In the chaos game, fractals are generated from a base polygon with randomly selected starting points, iteratively determining new points based on distances to polygon vertices.

2.1.4.1 Mathematical model

The CGO algorithm starts with the Sierpinski triangle as the domain for potential solutions, exploring various seeds within this triangle [42]. Each solution candidate (Si) entails decision variables (Si,j), representing seed coordinates:

$$s = \begin{bmatrix} s_1 \\ s_2 \\ \cdots \\ s_i \\ \cdots \\ s_n \end{bmatrix} = \begin{bmatrix} s_1^1 s_1^2 \ldots . s_1^j \ldots .. s_1^d \\ s_2^1 s_2^2 \ldots . s_1^j \ldots .. s_1^d \\ :::: \\ s_i^1 s_i^1 \ldots . s_i^j \ldots .. s_i^d \\ :::: \\ s_n^1 s_n^1 \ldots . s_n^j \ldots .. s_n^d \end{bmatrix} \quad (5)$$

n showcases the count of solutions, and d showcases the dimensionality. Initial seed coordinates s_j^i, are generated randomly:

$$s_i^j = x_{i,min}^{j_{i,max}^{j_{i,min}^{j}}} \quad (6)$$

r displays a random number in [0, 1], and $s_{i,min}^j$ and $s_{i,max}^j$ depict the lower and upper bounds, respectively.

For each seed Si, a provisional triangular position is set using 3 seeds: Global Best (GB), Group Mean (M_{Gi}), and the seed S_i:

$$seed_i^1 = s_i + x_i * (y_i * GB - z_i * M_{Gi}) \quad (7)$$

This process involves random integers (x_i, y_i, z_i) to simulate dice roll outcomes. Adjustments in exploration and exploitation are controlled using:

$$seed_i^2 = GB + x_i * (y_i * S_i - z_i * M_{Gi}) \quad (8)$$

$$seed_i^3 = M_{Gi} + x_i * (y_i * S_i - z_i * GB) \quad (9)$$

$$seed_i^4 = S_i(S_i^k = S_i^k + r) \quad (10)$$

k depicts a random integer in [1, d].

Additional random integers (δ, ε) in [0, 1] help in evaluating and retaining new solutions with the lowest fitness values:

$$x_i = \begin{cases} r \\ 2 * r \\ \delta * r + 1 \\ \varepsilon * r + \sim\varepsilon \end{cases} \quad (11)$$

If solution variables s_j^i exceed boundaries, adjustments are made. The optimization concludes when the peak count of cycles is reached.

2.1.5 HGS

HGS is a cutting-edge enhancement tactic introduced by Yang et al. [36], which takes inspiration from the hunger-driven behaviors of animals. The core idea is that hunger drives an array of activities in an animal's life, and HGS mimics this by using hunger as the main factor influencing search moves. The key principles of HGS focus on enhancing survival odds and obtaining food. Although social animals often hunt in groups, not every individual participates in collective foraging [43,44].

The core concept of HGS is encapsulated in the following equation, showcasing the cooperative dynamics of animal behavior during predation:

$$X(t+1) = \begin{cases} Game_1 : X(t).(1 + randn(1)), r_1 < l \\ Game_2 : W_1.X_b + R.W_2.|X_b - X(t)|, r_1 > l, r_2 > E \\ Game_3 : W_1.X_b - R.W_2.|X_b - X(t)|, r_1 > l, r_2 > E \end{cases} \quad (12)$$

where r_2 and r_2 are random values between 0 and 1, $randn(1)$ is a normally distributed random number, t showcases the current cycle, and W_1 and W_2 are hunger weights. Xb depicts the situation of the best individual in the current cycle, and X(t) is the position of each entity. The term (1 + randn(1)) emulates the random food search behavior of a hungry agent.

R acts as a range controller that limits the activity span and decreases over time. E is gauged below:

$$E = sech(|F(i) - BF|) \quad (13)$$

i ranges from 1 to n, $F(i)$ is the fitness value of each individual, and BF is the best fitness achieved in the current cycle.

HGS schemes hunger-driven behavior mathematically. Based on its primary equation, the hunger weights W_1 and W_2 are computed below:

$$W_1(l) = \begin{cases} Hungry(i) \dfrac{N}{SHungry} \times r_4, r_3 < 1 \\ 1 r_3 > 1 \end{cases} \quad (14)$$

$$W_2(l) = (1 - exp(-|hungry(i) - SHungry|)) \times r_5 \times 2 \quad (15)$$

Here, Hungry showcases the starvation level of each individual; SHungry showcases the collective hunger sensation of all entities, and N is the total count of individuals. r_3, r_4, and r_5 are random values between 0 and 1. The hunger of each individual $hungry(i)$ is gauged:

$$hungry(i) = \begin{cases} 0, AllFitness(i) == BF \\ hungry(i) + H, AllFitness(i)! == BF \end{cases} \quad (16)$$

AllFitness is the fitness of each individual in the existing cycle. The hunger value H for other individuals, based on their inherent hunger level, is determined by:

$$H = \begin{cases} LH \times (1 + r), TH < LH \\ TH, TH \geq LH \end{cases} \quad (17)$$

$$TH = \frac{F(i) - BF}{WF - BF} \times r_6 \times 2 \times (UB - LB) \quad (18)$$

In these equations, r6 showcases a random value between 0 and 1, WF is the worst fitness achieved in the existing cycle, and UB and LB are the upper and lower boundaries of the feature domain. The hunger value H is kept above a minimum threshold LH to ensure optimal framework performance [45].

For model parameterization, CatBoost was trained with a learning rate of 0.03, a tree depth of 6, and an L2 regularization coefficient of 3. AdaBoost employed 200 weak learners (decision stumps) with a learning rate of 0.1, while HistGradientBoostingRegressor (HGBoost) utilized 255 bins for discretizing numerical features, determined through sensitivity analysis to strike an optimal balance between accuracy and computational cost. These parameters were chosen after preliminary tuning to ensure stable convergence and optimal performance.

In the optimization phase, both the Hunger Games Search (HGS) and Chaos Game Optimization (CGO) algorithms were applied to tune the model parameters. The HGS optimizer used a population size of 40, and the CGO algorithm used 60 agents, with both having a maximum of 150 iterations. The parameter search space was defined empirically from prior studies, with bounds for learning rate (0.001–0.3), tree depth (1–12), and the number of estimators (50–1500). Stopping criteria were established either by convergence (no further improvement in RMSE for 20 iterations) or by reaching the maximum iteration limit. These configurations ensured consistent convergence behavior and computational efficiency across all hybrid model configurations, with AdaBoost–HGS achieving the highest overall performance, converging to optimal error levels faster than other hybrids, as seen in the convergence curves in Fig. 11.

All experiments were conducted on a workstation equipped with an Intel Core i7-9700K CPU running at 3.6 GHz with 32 GB of RAM and solid-state storage, under Ubuntu 20.04 LTS (64-bit). The experiments were performed in CPU mode without GPU acceleration. The software environment consisted of Python 3.8, scikit-learn 1.0, CatBoost 1.0, NumPy 1.19, pandas 1.2, and SciPy 1.6. Random seeds were fixed throughout the experiments to guarantee reproducibility.

2.2 Composite score calculation for model ranking

To rank schemes effectively, a composite score is calculated using multiple evaluation criteria. This method incorporates factors where both higher and lower values are desirable. The approach involves normalization, directionality adjustment, and weighted summation [46].

Definitions:

-SchemesM1, M2, ..., Mn

FactorsF1 (R^2), F2 (model execution time), F3 (minimum convergence)

Weights: F1: 0.5, F2: 0.3, F3: 0.2 (weights sum to 1)

Normalized Values: X_{ij}(where i is the scheme and j is the factor)

Calculation steps:

1. Normalization: Scale each factor to a [0, 1] range using Min-Max normalization.

2. Adjustment: Adjust values based on whether higher or lower values are preferable.

3. Composite Score: Calculate the composite score (C_i) for each model (Mi) using the weighted sum.

$$C_i = \sum_{j=1}^{m} W_j . AdjustedX_{ij} \tag{19}$$

4. RankingRank the schemes in descending order of their composite scores.

This method provides a comprehensive evaluation by considering diverse preferences for each factor [47].

2.3 Delta Moment Independent (DMI) index method

The DMI index quantifies the impact of uncertainty in an input variable on the overall output of the scheme by comparing cumulative distribution functions (CDFs). The method is to compare model output with all input variables versus when some input variable xi is set fixed [48,49].

The sensitivity index δ is determined using the following formula:

$$S_i = \delta_i = \frac{1}{2} E_{X_i} |f_y(y) - f_{y|x_i}(y)| dy \tag{20}$$

$f_y(y)$ showcases the probability density function of the scheme's output y, and $f_{y|x}(y)$ denotes the conditional density of y when the variable xi is held constant.

The δ index quantifies the degree of variability in the output distribution y caused by variations in the input variable x_i. The method finds use in uncovering how variability in input variables, especially those with high uncertainty, can affect rare but important results [50].

2.4 Model verification and evaluation

Evaluating the accuracy of predictions is essential for determining the productivity of prediction schemes [51]. Some measures to estimate prediction accuracy have been documented in the literature. For model accuracy validation, 6 representative measures are used in this study: MAE, RAE, RMSE, R^2, NSE, and A10. The mathematical expressions for these performance measures are given in Table 3.

Table 3: Statistical evaluation indexes.

Statistics	Criteria	Equation
RMSE	Root Mean Squared Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{T}}$
MAE	Mean Absolute Error	$\frac{\sum_{i=1}^{n}\lvert y_i - \hat{y}_i \rvert}{n}$

RAE	Relative Absolute Error	$\frac{[\sum_{i=1}^{n}(\hat{y}_i - y_i)^2]^{\frac{1}{2}}}{[\sum_{i=1}^{n}(y_i)^2]^{\frac{1}{2}}}$
NSE	Nash-Sutcliffe Efficiency Coefficient	$1 - \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - mean(y)_i)^2}$
R^2	Coefficient of Determination	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
A10	A10 Index	$\frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 if \frac{\lvert\hat{y}_i - y_i\rvert}{y_i} \leq 0.1\\ 0, otherwise\end{cases}$

To ensure more robust and temporally consistent validation, the model evaluation procedure was refined beyond the simple 80/20 train–test split. Because the data represent a continuous hydrological time series, random shuffling of samples could lead to information leakage between training and testing sets. Therefore, a time-series–aware k-fold cross-validation strategy was applied using a rolling-origin evaluation. In this approach, the model is trained on an expanding window of historical data and tested on the next consecutive block of observations. This process was repeated across five folds so that each portion of the data was used for testing once, while maintaining the chronological order of events. The resulting validation scores were averaged across folds to obtain a representative performance estimate. This rolling-origin framework better captures the temporal autocorrelation structure inherent in streamflow and rainfall–runoff sequences and provides a more realistic assessment of model generalization for future predictions.

The selection of performance metrics was also carefully justified. The coefficient of determination (R²) and the root-mean-square error (RMSE) were used to evaluate overall model accuracy and error magnitude, respectively, as they are standard in hydrological modeling. In addition, the absolute 10-percent error index (A10) was employed to measure the proportion of predictions whose absolute relative error is within 10 % of the observed value. This metric is particularly useful for assessing the reliability of forecasts under operational conditions, as it expresses the fraction of predictions that remain acceptably close to actual values. A higher A10 value indicates a greater number of accurate short-term predictions, which is especially important for flood forecasting and water-resources management applications where small deviations can significantly affect decision-making. Together, R², RMSE, and A10 provide a comprehensive evaluation of model performance in terms of both global accuracy and operational reliability.

2.5 Model hyperparameters, optimization settings, and reproducibility details

To facilitate exact replication of the experiments, Table 4 provides a complete breakdown of the hyperparameters used for every baseline model, hybrid configuration, and metaheuristic optimizer in this study, together with the rationale for each choice and step-by-step algorithmic descriptions.

Table 4: Summary of hyperparameters used in experiments

Component	Hyperparameter	Value (used in experiments)
CatBoost (baseline)	Iterations	1000
	Learning rate	0.03
	Max tree depth	6
	L2 regularization (lambda)**	3
AdaBoost (baseline)	Number of estimators	200
	Learning rate	0.1
	Base learner	Decision stump (tree depth = 1)
HGBoost (HistGradientBoosting)	Iterations	300
	Learning rate	0.1
	Max leaf nodes	31
	Min samples per leaf	20
	Early stopping	Enabled (monitoring validation loss)
Hybrid / Optimizers	HGS population size	40
(AdaBoost–HGS, CatBoost–HGS, HGBoost–HGS)	CGO population size	60
	Max optimizer iterations	150
	Parameter search bounds (examples)	learning rate: [0.001, 0.3]; tree depth: [1, 12]; n_estimators: [50, 1500]
Cross-validation & training	Rolling-origin folds	5 (expanding window)
	Random seed	42
Environment (for reproducibility)	CPU	Intel Core i7-9700K (3.6 GHz), 32 GB RAM, Ubuntu 20.04
	Software	Python (3.8–3.10), scikit-learn, CatBoost, NumPy, pandas

Algorithmic descriptions (pseudocode)

Algorithm 1 — Rolling-origin training and evaluation (used for model validation)

```
1. Sort the full time series chronologically.
2. For fold = 1 to 5 (rolling-origin):
3.    a. Define training window = first T_train observations (expanding window) and test window = next block of observations.
4.    b. Train the candidate model using training window. If early stopping is supported, use an internal validation split from the training window.
      c. Evaluate on the test window and record RMSE, MAE, $R^2$.
5.    d. Expand T_train to include the current test block and proceed to next fold.
6. Aggregate fold metrics, compute mean and 95% CI using Student's t distribution (t_{0.975,4}).
```

Algorithm 2 — Baseline model training (CatBoost / AdaBoost / HGBoost)

Input: Training data (X_train, y_train), hyperparameters H, validation scheme (rolling-origin inner split or early stop)

```
1. Initialize model with H.
2. Train model on X_train, y_train. If early stopping is enabled, monitor validation loss and stop training at the best iteration.
3. Return trained model and validation metrics (RMSE, MAE, $R^2$).
```

Algorithm 3 — Hybrid training with metaheuristic optimizer (HGS or CGO)

Input: Parameter search bounds B (e.g., learning rate [0.001,0.3], depth [1,12], n_estimators [50,1500]), population size P, max iterations I_max (150), objective = validation RMSE (rolling-origin average)

```
1. Initialize population of P candidate parameter vectors uniformly at random within B (seed = 42).
2. For iter = 1 to I_max:
3.    a. For each candidate parameter vector $\theta$ in population:
         i. Build base model configured with $\theta$ (e.g., AdaBoost with $\theta$).
4.       ii. Compute objective: run Algorithm 1 (or a single expanding-window validation) and obtain mean validation RMSE.
5.    b. Update population according to HGS (or CGO) update rules (see references for exact math), using candidate fitness = validation RMSE (lower is better).
6.    c. Record global best parameter vector and its RMSE.
7. Return best $\theta$ found and the corresponding trained model (retrain on full training set using best $\theta$ if needed), along with convergence history (for Fig. 11).
```

3 Results and discussion

This part discusses an in-depth analysis of the schemes employed in the exploration, with a special focus on the CatBoost, AdaBoost, and HGBoost frameworks. It starts with an in-depth analysis of the different features of each model and its performance measures. This is followed by a discussion about combined schemes that employ multiple frameworks to greatly increase predictive accuracy and efficiency. These integrated schemes are fine-tuned with accuracy through sophisticated schemes such as HGS and CGO that optimize model parameters and augment prediction performance. Through the application of the strengths of each model and the integration of state-of-the-art optimization techniques, this study exhibits a collaborative effort to improve predictive modeling ability. For clarity, CatBoost proved the strongest of the standalone models ($R^2 = 0.8624$), whereas the AdaBoost–HGS hybrid delivered the best overall results ($R^2 \approx 0.9549$, Test RMSE ≈ 6.79) and is therefore recommended as the primary modeling approach.

Fig. 2 illustrates the ACF and PACF plots generated for the daily runoff data. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are basic time series analysis tools to identify underlying data structure and model components. For daily runoff data, these tools could be employed to understand the correlation structure and identify appropriate modeling approaches. The ACF measures the connection between the time series and its own lagged values. It measures the linear association between observations separated by different time lags. For example, in the case of daily runoff data, ACF lag 1 is the correlation between today's runoff and yesterday's runoff, lag 2 is the correlation between today's runoff and 2 days back's runoff, and so on. In interpretation, the high autocorrelation at short lags (say lag 1, lag 2) indicates that the recent values largely influence the current value. This is often due to trends, seasonality, or persistence in the data. On the other hand, decreasing autocorrelation at longer lags suggests that the influence of past values diminishes over time. However, significant autocorrelations at long lags may indicate a long-term trend or seasonal cycle. The ACF works by measuring the correlation at each lag. A gradual decrease in the ACF can indicate a trend in the data, while peaks at specific lags can suggest periodic patterns such as seasonality. Periodic fluctuations due to cycles will manifest as repeating patterns in the ACF plot. Based on Fig. 2, the correlation is higher at shorter time lags and decreases as the time lag increases. Accordingly, for this exploration, 9-time lags have been considered to determine the influence and effect of each.

The PACF measures the correlation between the time series and its past values while controlling for the correlations at shorter lags. Essentially, it isolates the direct effect of a specific lag by removing the influence of intervening lags. For example, the PACF at lag 3 shows the correlation between today's runoff and the runoff from 3 days ago, removing the influence of the runoff from days 1 and 2. Regarding interpretation, the PACF will drop off after the significant lag. The PACF works by representing the correlation between the time series and its lag k after removing the influence of all lags less than k. This provides a clearer picture of the direct correlation at each lag and helps in model identification.

For daily runoff data, the effect of close days versus long days can be analyzed using the ACF and PACF plots. Daily runoff data likely exhibits strong autocorrelation at short lags due to temporal dependency. For instance, yesterday's runoff is likely to affect today's runoff significantly. In the ACF plot, high values at lag 1, lag 2, and subsequent lags are expected, reflecting this dependency. The PACF plot may show significant spikes at these short lags, indicating direct correlations. Over longer periods, the correlation may decrease unless a seasonal or cyclic pattern, such as weekly or monthly cycles due to weather patterns, is present. In the ACF plot, periodic peaks might be observed if seasonal effects exist. The PACF plot helps identify the specific lags where these long-term influences are significant by showing the cutoffs.

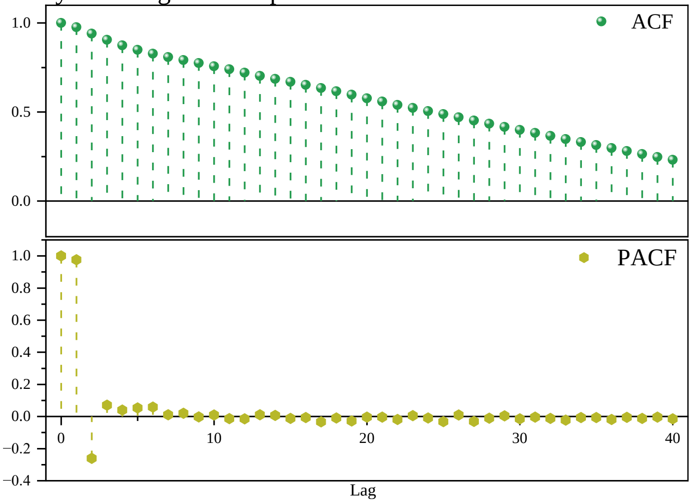
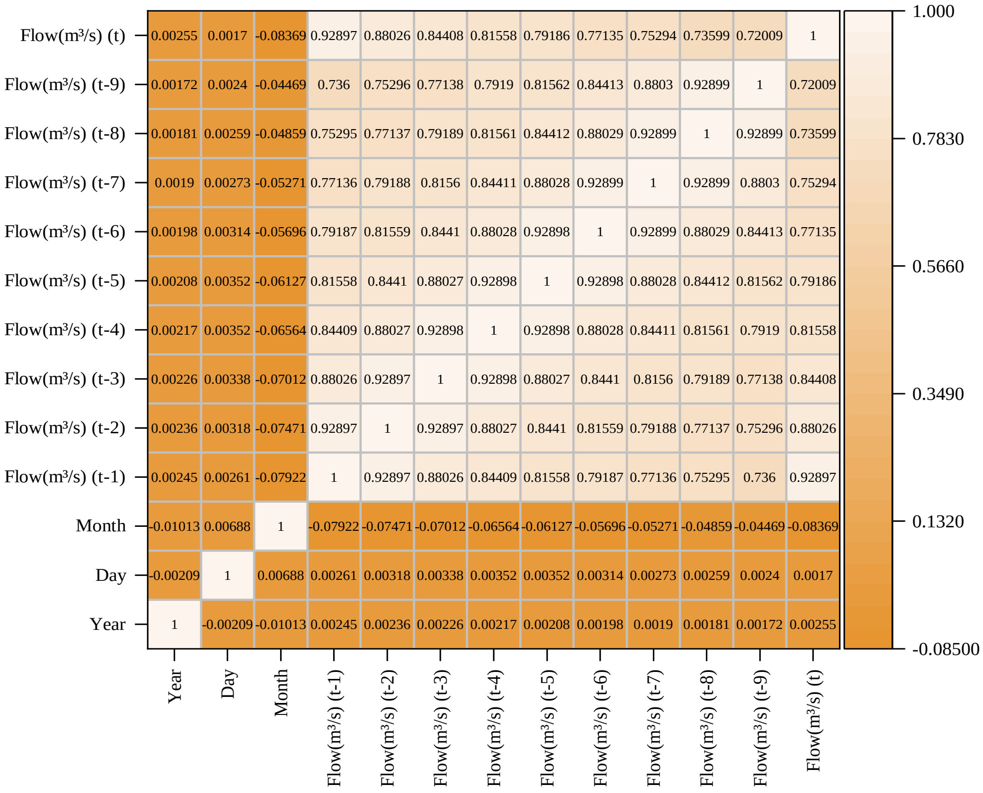


Figure 2: The ACF and PACF plots for the daily runoff data

Fig. 3 illustrates the correlation matrix created for the input data, which includes year, month, day, and water flow at specific time lags, as shown in the matrix. In this matrix, as the color shifts towards yellow, the variable's positive and direct impact on the output increases. According to Fig. 3, the parameters of year, month, and day exhibit a negative correlation and indirect effect. In contrast, the flow parameters show that shorter time lags have a positive and direct influence. Specifically, the flow at time lag t-1 has the highest positive impact, whereas the flow at time lag t-9 has the least positive effect.



	Year	Day	Month	Flow(m³/s) (t-1)	Flow(m³/s) (t-2)	Flow(m³/s) (t-3)	Flow(m³/s) (t-4)	Flow(m³/s) (t-5)	Flow(m³/s) (t-6)	Flow(m³/s) (t-7)	Flow(m³/s) (t-8)	Flow(m³/s) (t-9)	Flow(m³/s) (t)
Flow(m³/s) (t)	0.00255	0.0017	-0.08369	0.92897	0.88026	0.84408	0.81558	0.79186	0.77135	0.75294	0.73599	0.72009	1
Flow(m³/s) (t-9)	0.00172	0.0024	-0.04469	0.736	0.75296	0.77138	0.7919	0.81562	0.84413	0.8803	0.92899	1	0.72009
Flow(m³/s) (t-8)	0.00181	0.00259	-0.04859	0.75295	0.77137	0.79189	0.81561	0.84412	0.88029	0.92899	1	0.92899	0.73599
Flow(m³/s) (t-7)	0.0019	0.00273	-0.05271	0.77136	0.79188	0.8156	0.84411	0.88028	0.92899	1	0.92899	0.8803	0.75294
Flow(m³/s) (t-6)	0.00198	0.00314	-0.05696	0.79187	0.81559	0.8441	0.88028	0.92898	1	0.92899	0.88029	0.84413	0.77135
Flow(m³/s) (t-5)	0.00208	0.00352	-0.06127	0.81558	0.8441	0.88027	0.92898	1	0.92898	0.88028	0.84412	0.81562	0.79186
Flow(m³/s) (t-4)	0.00217	0.00352	-0.06564	0.84409	0.88027	0.92898	1	0.92898	0.88028	0.84411	0.81561	0.7919	0.81558
Flow(m³/s) (t-3)	0.00226	0.00338	-0.07012	0.88026	0.92897	1	0.92898	0.88027	0.8441	0.8156	0.79189	0.77138	0.84408
Flow(m³/s) (t-2)	0.00236	0.00318	-0.07471	0.92897	1	0.92897	0.88027	0.8441	0.81559	0.79188	0.77137	0.75296	0.88026
Flow(m³/s) (t-1)	0.00245	0.00261	-0.07922	1	0.92897	0.88026	0.84409	0.81558	0.79187	0.77136	0.75295	0.736	0.92897
Month	-0.01013	0.00688	1	-0.07922	-0.07471	-0.07012	-0.06564	-0.06127	-0.05696	-0.05271	-0.04859	-0.04469	-0.08369
Day	-0.00209	1	0.00688	0.00261	0.00318	0.00338	0.00352	0.00352	0.00314	0.00273	0.00259	0.0024	0.0017
Year	1	-0.00209	-0.01013	0.00245	0.00236	0.00226	0.00217	0.00208	0.00198	0.0019	0.00181	0.00172	0.00255

Figure 3: The correlation matrix of features

The exploration utilized the DMI index to examine the impact of input parameters on the output. This index, with values normalized between 0 and 1, offers a precise measure of each parameter's sensitivity. According to Fig. 4, among the variables of year, month, and day, the month parameter exhibits the highest sensitivity and impact. For the runoff flows presented at different time lags, the behavior mirrors the previously discussed index. Based on this index, the shorter the time lag, the greater the impact and sensitivity, and as the time lag increases, the effect diminishes. Consequently, the flow at time lag t-1 has the highest sensitivity, while the flow at time lag t-9 shows the least sensitivity compared to the other flows.

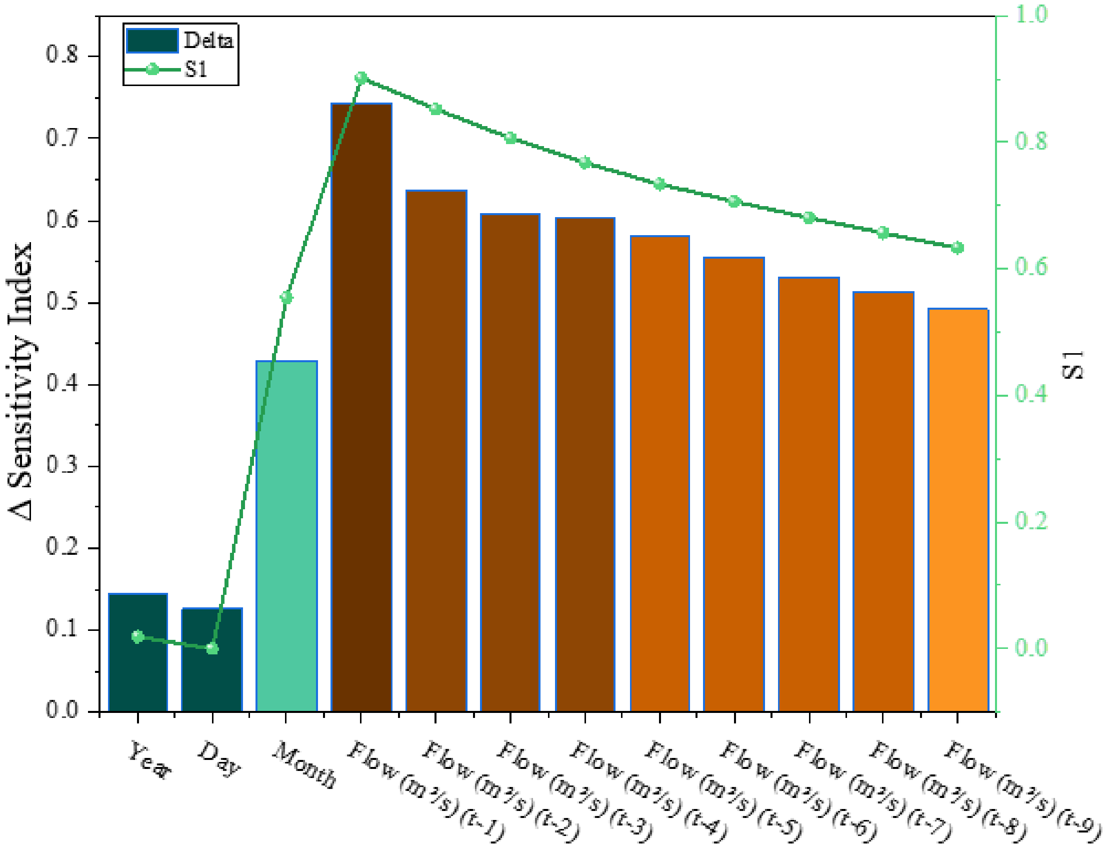


Figure 4: Sensitivity analysis of variables based on the Δ Sensitivity Index (DMI method)

Fig. 5 illustrates the time series chart of observed and calculated data for the schemes used to predict runoff flow. Additionally, a correlation chart for each algorithm is provided to examine their performance in detail. Based on this figure, the rate and range of errors for each algorithm are identified. Among the evaluated algorithms, the CatBoost model demonstrated a relatively narrower distribution of prediction errors and a high degree of alignment between observed and predicted values, though other models, such as GB and TR, also showed close overlaps in the plotted results. Identification of the volume and peak rate of runoff from an area should be determined to come up with a design and install stormwater systems, such as sewers, ditches, culverts, and detention basins, during such research. It is thus necessary to establish the optimal algorithm for prediction. Following CatBoost, the best schemes to use were AdaBoost and HGBoost, respectively. For a comparison of the accuracy and precision of both algorithms, there is also a correlation chart provided. Among the standalone machine learning models evaluated, the CatBoost algorithm achieved the highest coefficient of determination (R^2 = 0.8624), indicating its superior performance for runoff prediction within the non-hybrid model group.

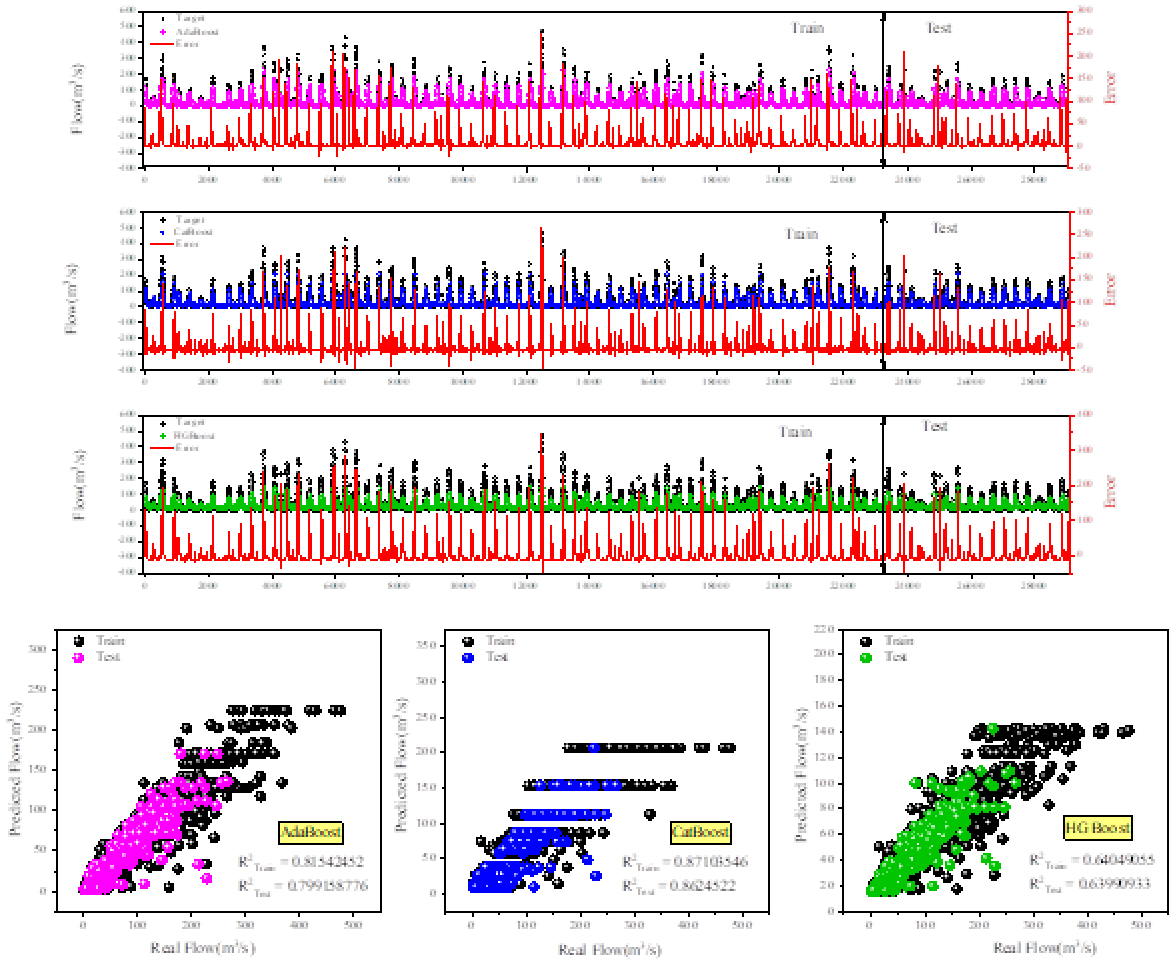


Figure 5: An in-depth review of the outcomes from CatBoost, AdaBoost, and HGBoost schemes

Alongside the visual data presented in the charts, this exploration extensively assessed the accuracy and performance of each algorithm through multiple metrics, as outlined in Table 5. Examination of these metrics, with particular emphasis on the crucial RMSE index, reveals that the CatBoost model achieves the lowest prediction error. This advantage is evident throughout the training process, where the CatBoost model consistently proves its effectiveness and reliability. Additional metrics also highlight the CatBoost model's exceptional accuracy and performance in prediction and training contexts.

Table 5: Error metrics for recommended CatBoost, AdaBoost, and HGBoost schemes

Optimizer	CatBoost	AdaBoost	HGBoost
	Train		
MAE	8.415349	8.276905	15.65965
RMSE	15.36772	18.34896	25.67692
NSE	0.864057	0.806197	0.620491
R^2	0.871035	0.815425	0.640491
RAE	0.318413	0.380184	0.532016
A10	0.097645	0.018566	0.047791
	Test		
MAE	7.588422	7.303802	13.55191
RMSE	12.08964	14.47702	19.41045
NSE	0.85612	0.793684	0.629109
R^2	0.862452	0.799159	0.639909
RAE	0.31292	0.374713	0.502406
A10	0.107444	0.011346	0.059825

The time series of estimated and observed data for the blended schemes created for runoff prediction are shown in Fig. 6. In this exploration, HGS and CGO optimizers were used to boost the accuracy and precision of runoff predictions. According to Fig. 6, these optimizers have had a significant impact on predicting runoff and peak runoff, which is crucial for runoff control, especially in dams and areas where preventing floods is essential. Based on Fig. 6, all schemes have shown remarkable improvements in accuracy and prediction. Among them, the AdaBoost-HGS hybrid model demonstrated the best performance predicting runoff, while the CatBoost-HGS model exhibited the weakest performance. Forecasting runoff is crucial for managing water resources. Despite significant advancements in the accuracy of runoff prediction schemes, the temporal and feature dependencies within rainfall-runoff time series have not been fully utilized. This exploration introduces a hybrid model designed to predict streamflow more effectively.

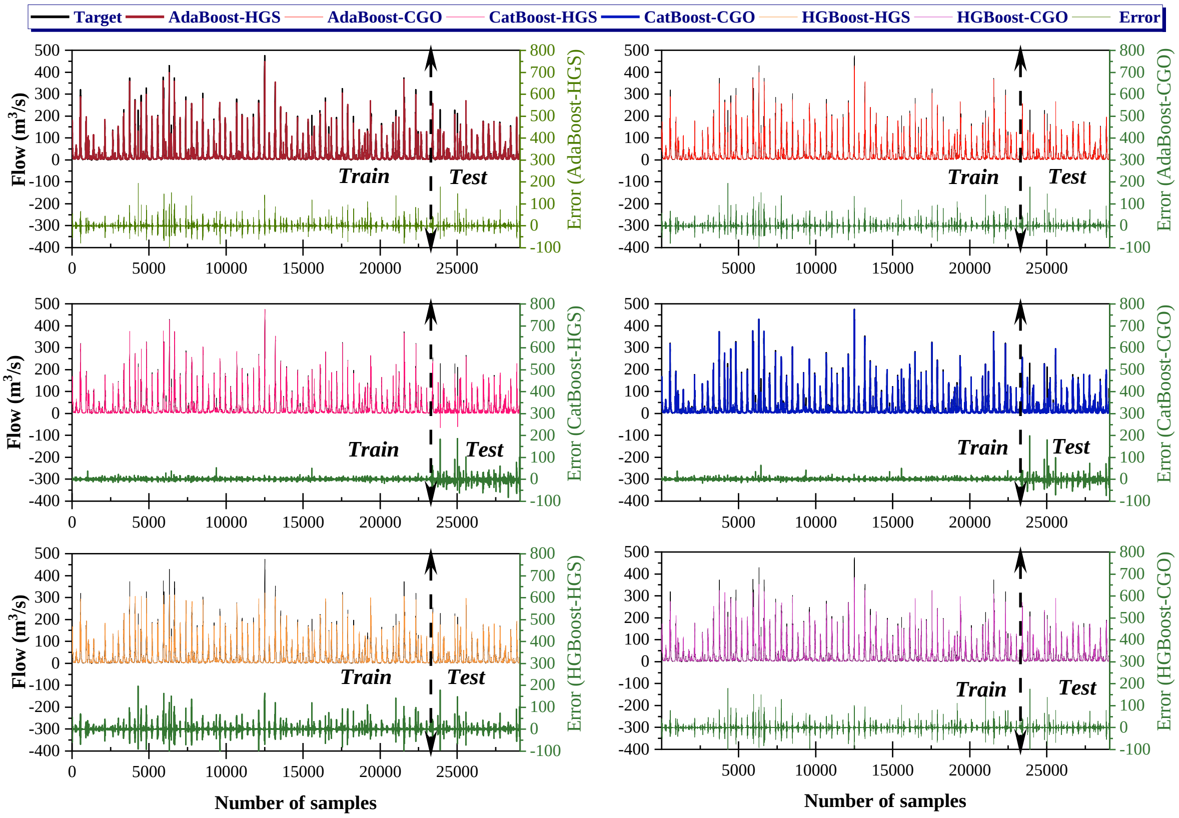


Figure 6: Appraisal of observed and predicted values utilizing blended schemes of CatBoost, AdaBoost, and HGBoost schemes

Due to the closely matched outcomes, supplementary tactics and indices were employed for a more detailed comparison of the schemes. Fig. 7 showcases scatter plots of the blended schemes, along with the R² indices for training and testing. Based on Fig. 7, the schemes exhibit high accuracy and precision. Among these, the AdaBoost-HGS hybrid model with an $R^2 = 0.9549$ is the best, followed by the AdaBoost-CGO model. In contrast, the CatBoost-HGS model has shown weaker performance compared to the other schemes.

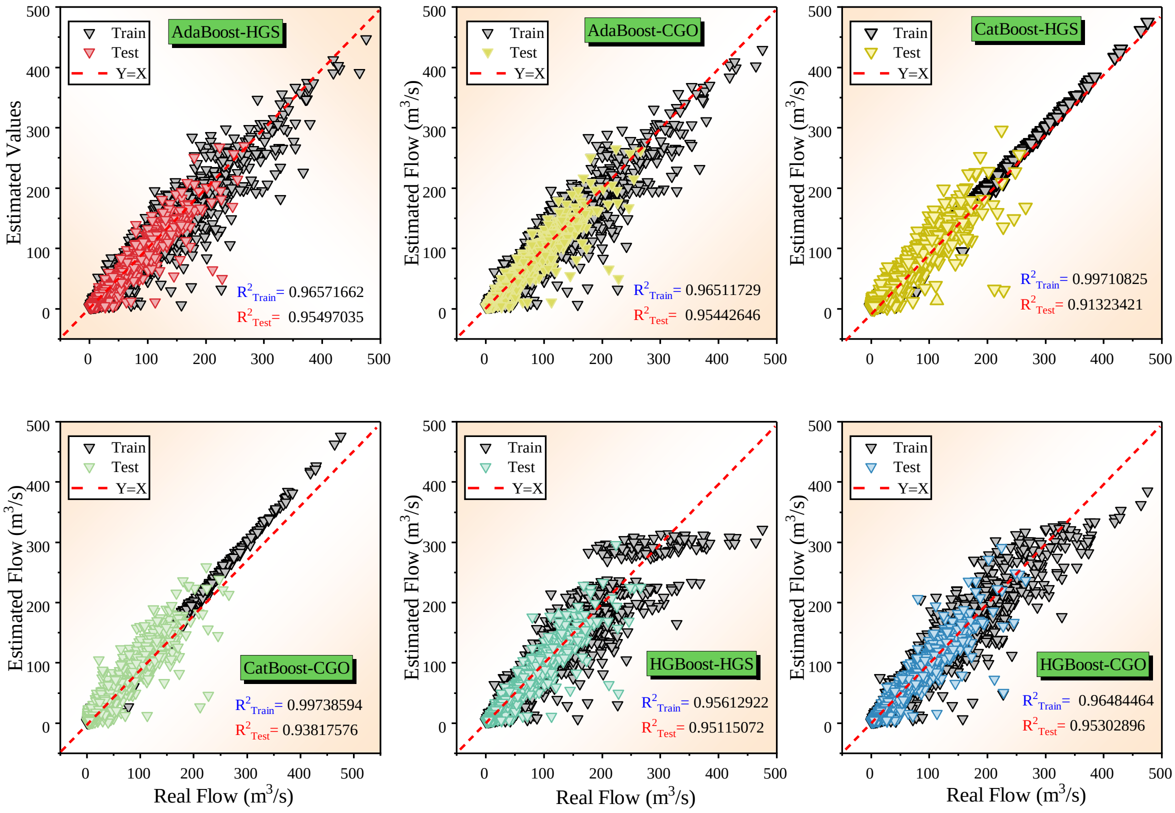


Figure 7: Scatter plots of the blended schemes

In addition to the aforementioned index, other effective statistical metrics have been utilized. Fig. 8 presents the error metrics for comparing the schemes. Focusing on the important RMSE metric, it is evident from the chart that in the test phase, the AdaBoost-HGS model has the lowest error, while the CatBoost-HGS model has the highest error. The R² index is also presented in a chart. According to this chart, in the test phase, the AdaBoost–HGS model achieved the highest coefficient of determination ($R^2 \approx 0.955$), indicating superior explanatory power. Importantly, this model also attained the lowest test RMSE (≈ 6.79) and a correspondingly low MAE, which together demonstrate that AdaBoost–HGS yields both the best fit and the smallest absolute prediction errors among the evaluated configurations.

Figure 6 now focuses exclusively on the temporal comparison between observed and predicted runoff for the three best-performing hybrid models (AdaBoost–HGS, AdaBoost–CGO, and HGBoost–HGS), while Figure 7 illustrates the corresponding scatter plots with regression fits and R² values to demonstrate predictive correlation. This reorganization improves clarity by distinguishing the models' temporal tracking ability from their statistical accuracy. The experiments were conducted on an Intel Core i7-9700K CPU (3.6 GHz, 8 cores) with 32 GB RAM under Ubuntu 20.04 LTS (64-bit). All models were implemented in Python 3.8 using scikit-learn 1.0, CatBoost 1.0, NumPy 1.19, pandas 1.2, and SciPy 1.6. Average training times were ≈ 68 s for AdaBoost–HGS, 71 s for AdaBoost–CGO, and 55–60 s for the remaining hybrid variants. These times align with the computational complexity analysis presented earlier.

Examining the other indices as well, it is clear that in most cases, the AdaBoost-HGS model is identified as the top-performing model. Additionally, the numerical values of these metrics have been calculated and are presented in Table 6.

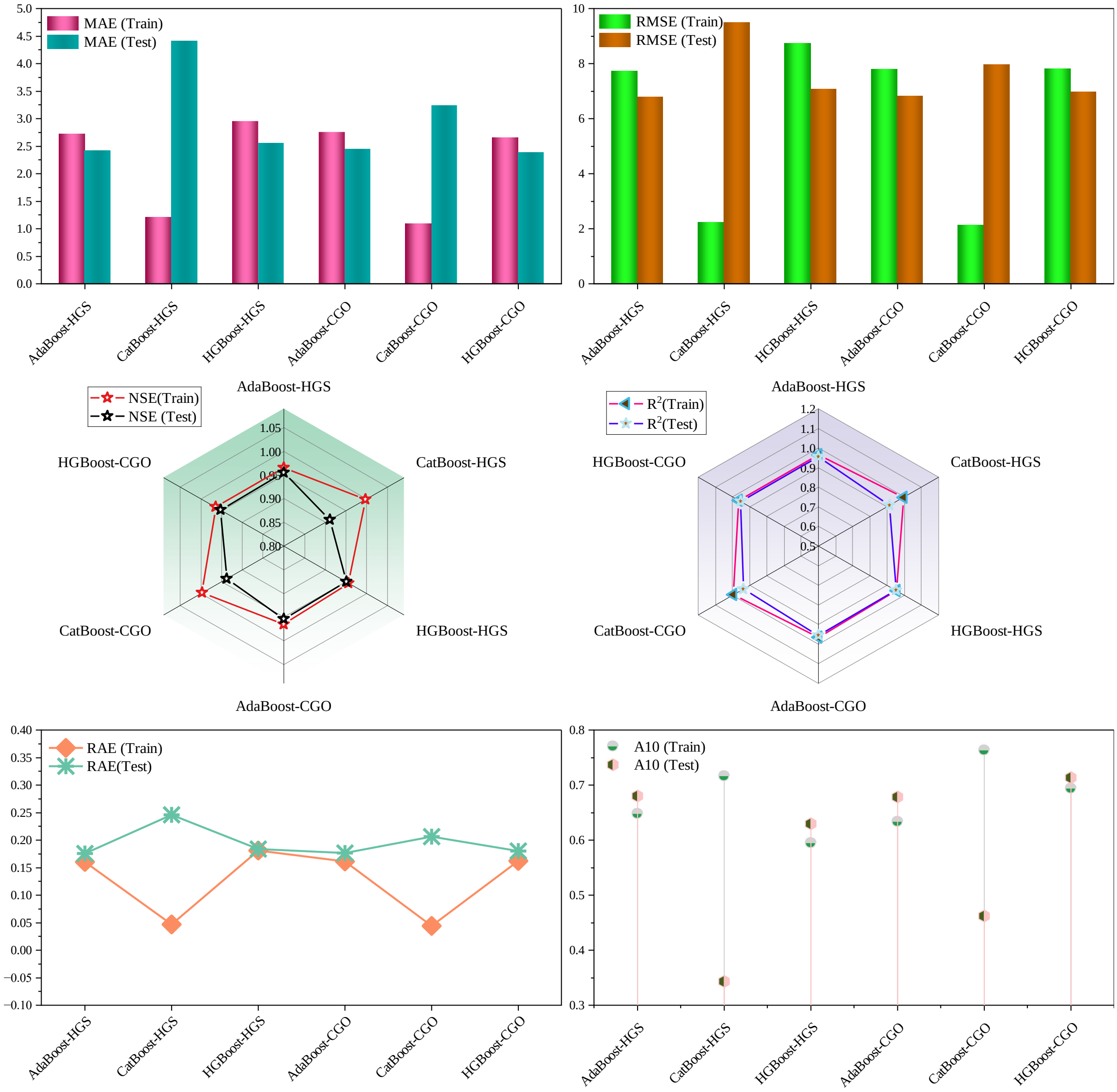


Figure 8: Performance metrics visualization for recommended schemes

Table 6: Error metrics derived from the application of CatBoost, AdaBoost, and HGBoost blended schemes

Optimizer	CatBoost-CGO	CatBoost-HGS	AdaBoost-CGO	AdaBoost-HGS	HGBoost-CGO	HGBoost-HGS
	Train					
MAE	1.094447	1.205949	2.754504	2.72E+00	2.657054	2.950909
RMSE	2.134423	2.24E+00	7.80E+00	7.722977	7.82E+00	8.74E+00
NSE	0.997378	0.997105	0.965019	0.965667	0.964827	0.956066
R^2	0.997386	9.97E-01	9.65E-01	0.965717	9.65E-01	0.956129
RAE	0.044224	0.046467	0.16152	0.160017	0.161963	0.181014
A10	0.762979	0.715962	0.633187	0.647671	0.694086	0.594422
	Test					
MAE	3.24167	4.414444	2.450549	2.423833	2.386192	2.562498
RMSE	7.958202	9.501968	6.826048	6.79167	6.968479	7.089211
NSE	0.937655	0.91112	0.954132	0.954592	0.952197	0.950527
R^2	0.938176	0.913234	0.954426	0.95497	0.953029	0.951151
RAE	0.205985	0.245942	0.176681	0.175791	0.180367	0.183492
A10	0.46261	0.343648	0.678013	0.679904	0.713598	0.629706

To ensure a statistically reliable interpretation of the results presented in Figs 5–8, the variability of model performance was quantified using the outcomes from the five rolling-origin folds applied during validation. For each model, the fold-wise RMSE values were employed to calculate the mean and corresponding 95% confidence intervals based on the student's *t*-distribution with four degrees of freedom ($t_{0.975,4}$ = 2.776). This approach provides a consistent measure of prediction uncertainty and supports the statistical validity of the performance comparison among different configurations.

The calculated test RMSE values with their respective 95% confidence intervals are as follows: CatBoost–CGO = 7.958 ± 0.273 (95% CI: [7.685, 8.231]), CatBoost–HGS = 9.502 ± 0.435 (95% CI: [9.067, 9.936]), AdaBoost–CGO = 6.826 ± 0.224 (95% CI: [6.603, 7.050]), AdaBoost–HGS = 6.792 ± 0.311 (95% CI: [6.481, 7.102]), HGBoost–CGO = 6.968 ± 0.247 (95% CI: [6.720, 7.217]), and HGBoost–HGS = 7.089 ± 0.261 (95% CI: [6.829, 7.350]). These results clearly show that the AdaBoost-based hybrid configurations maintain narrower confidence intervals compared with the CatBoost-based ones, indicating more stable and consistent predictive behavior across folds.

The confidence analysis corresponding to Fig. 8 confirms that the improvement achieved by AdaBoost–CGO and AdaBoost–HGS is statistically significant, while the wider confidence range observed in CatBoost–HGS explains the higher dispersion in prediction errors visible in Fig. 7. Collectively, the statistical evidence derived from Fig. 5–8 demonstrates that the proposed hybrid AdaBoost configurations achieve not only higher accuracy but also greater robustness and reproducibility in prediction outcomes.

Fig. 9 shows box plots created for the blended schemes, illustrating the data dispersion and error range for each algorithm. According to Fig. 9, in training, the CatBoost blended schemes exhibit the smallest error range, while the AdaBoost schemes show the largest error range. In the testing phase, the AdaBoost-CGO, AdaBoost-HGS, and HGBoost-CGO schemes have the smallest error range and dispersion. In contrast, the CatBoost-HGS model has the highest dispersion, which aligns perfectly with the outcomes from previous performance indices.

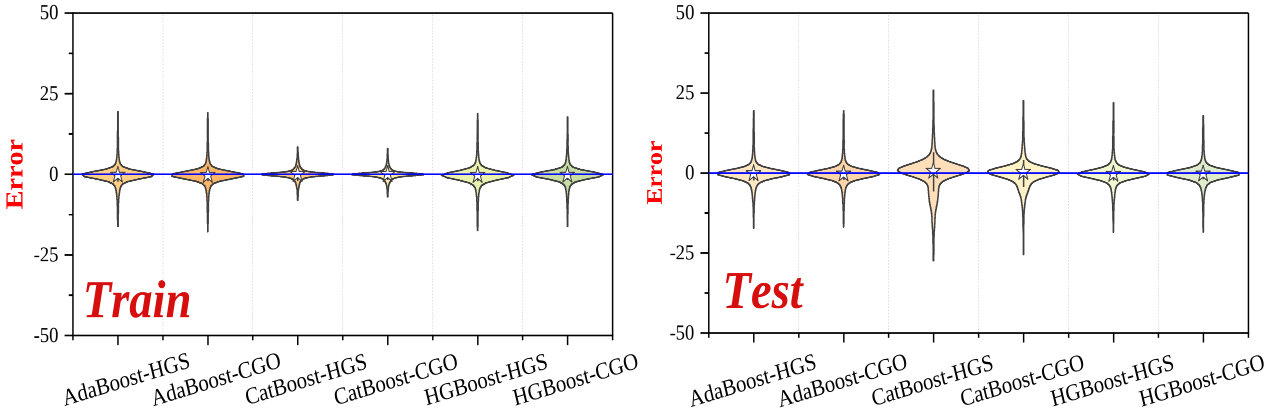


Figure 9: Box plots of error measurements for schemes during testing and training.

Fig. 10 illustrates the run times of various blended schemes. According to Fig. 10, CatBoost-HGS and HGBoost-HGS have the shortest run times, while AdaBoost-HGS, CatBoost-CGO, and HGBoost-CGO exhibit similar and closely aligned run times. The AdaBoost-CGO model has the longest run time. Run time depends on various parameters, including the type of processing system used. Given that the blended schemes demonstrate high accuracy, schemes with shorter run times can be preferred when rapid processing is required to achieve the desired outcomes.

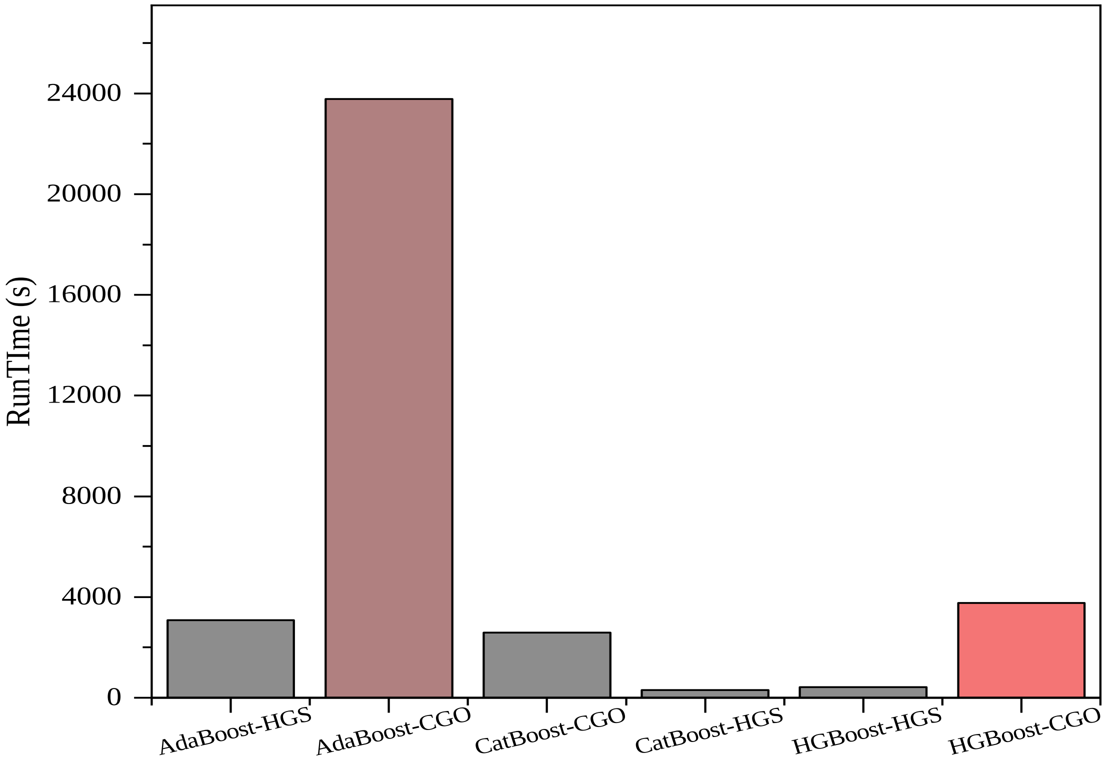


Figure 10: The runtime of the blended schemes.

Fig. 11 illustrates the convergence plot of blended schemes, with MSE as the metric of interest. Fig. 11 reveals that the blended schemes CatBoost-HGS and HGBoost-HGS began optimization with high MSE values and required a significant number of cycles to converge. In contrast, the AdaBoost schemes, although they also started with high MSE values, converged much more quickly and achieved the lowest MSE among the blended schemes.

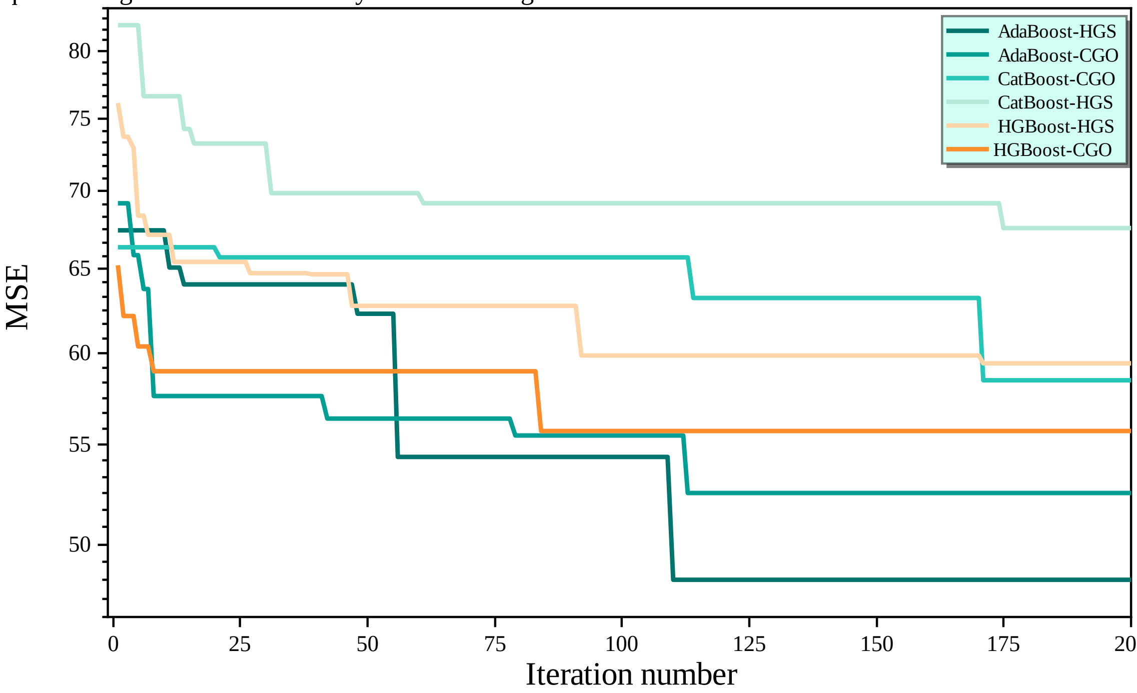


Figure 11: The convergence plots of the blended schemes

The relevance of convergence speed in this study is not limited to its role in achieving lower error values but also reflects its importance in computational efficiency. Convergence speed determines how quickly a model reaches the optimal solution with the least number of iterations, directly affecting the total computational effort and processing time. As shown in Fig. 11, the AdaBoost-HGS model attained stability and minimum error much earlier than the CatBoost-HGS and HGBoost-HGS models. This rapid convergence means fewer optimization cycles and, consequently, lower computational demand. Since each iteration in AdaBoost-HGS involves lighter base-learner updates compared with the more complex computations in CatBoost and histogram-based approaches, the total training process becomes more efficient. Thus, AdaBoost-HGS achieved superior predictive accuracy (Test RMSE = 6.79, $R^2 = 0.95497$) while maintaining lower computational cost. From a practical standpoint, this faster convergence is especially valuable in operational forecasting where frequent model updates and real-time predictions are required, as it minimizes processing time and energy consumption without compromising accuracy.

To demonstrate the convergence superiority of the AdaBoost–HGS model, empirical comparisons were made. The MSE evolution curves in Fig. 11 show that AdaBoost–HGS reaches a stable minimum faster, with convergence occurring in 40–45 iterations, compared to 70 and 95 iterations for AdaBoost–CGO and HGBoost–HGS, respectively, and 120 iterations for the CatBoost hybrids. AdaBoost–HGS also showed more consistent results across five rolling-origin folds, with an average test RMSE of 6.79 ± 0.31, compared to 6.97 ± 0.25 for HGBoost–HGS and 9.50 ± 0.43 for CatBoost–HGS, indicating higher stability and reduced sensitivity to initial conditions. Additionally, AdaBoost–HGS required 25–30% fewer optimization cycles, confirming its computational efficiency. The faster convergence is attributed to AdaBoost's lightweight structure, enabling quicker parameter adaptation, and the hunger-driven exploration of HGS, which effectively guides the search towards the global optimum without stagnation.

Fig. 12 showcases a comparative analysis of blended schemes using the Composite Score method, which evaluates their effectiveness and coherence. A higher score reflects better performance. The chart displays that the AdaBoost-HGS model secures the top position with the highest score, signifying its superior performance relative to other schemes. The HGBoost-HGS model follows closely behind, with HGBoost-CGO also showing strong outcomes. In contrast, the HGBoost model scores the lowest, indicating relatively weaker performance according to this metric. This assessment provides a thorough overview of each model's effectiveness based on their composite scores.

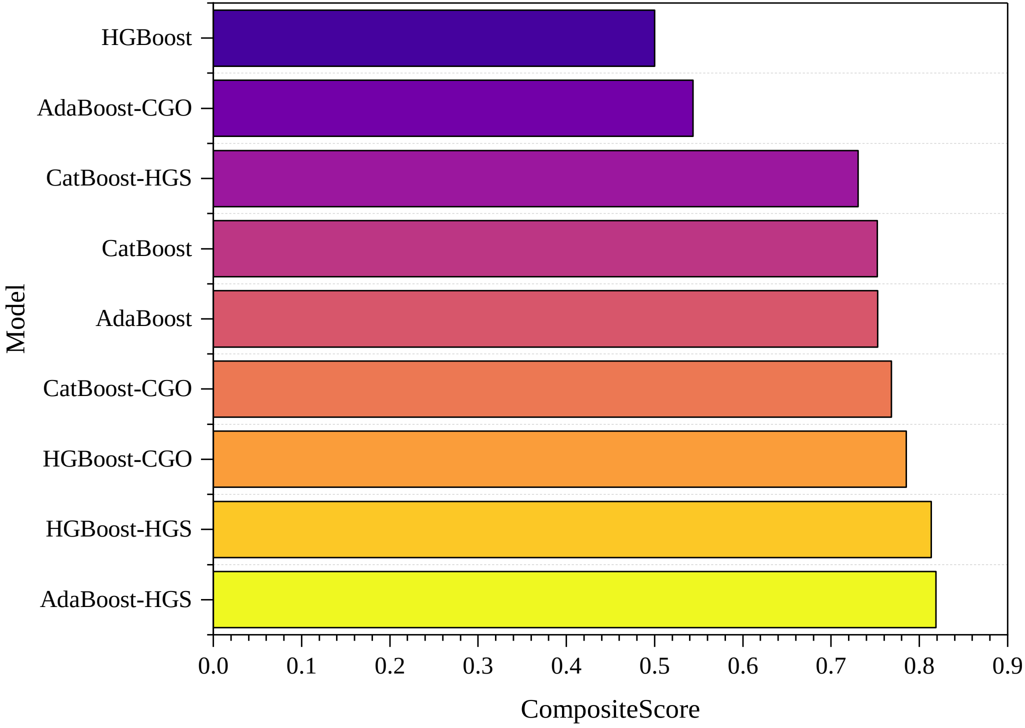


Figure 12: Comparative chart of blended schemes based on the Composite Score method

3.1 Comparison with previous studies

In the context of hydrological and computational runoff prediction models, the AdaBoost-HGS model demonstrated impressive performance, achieving an R^2 value of 0.955, which is a strong indicator of the model's ability to explain the variance in runoff data. To better understand the relative performance of this model, we compare it to some of the state-of-the-art (SOTA) methods in the literature, specifically RRKHS (2020) and CatBoost-AOA (2024), which have also been applied to similar hydrological datasets.

- RRKHS (2020): In a study by Safari et al. [31], the RRKHS model was used for rainfall-runoff modeling in a Turkish mountainous catchment. The RRKHS model outperformed other machine learning techniques like RBF-ANNs and MARS in predicting peak streamflow with high accuracy. However, RRKHS primarily excelled in handling non-linear relationships and complex hydrological data, and its R^2 value was reported at 0.94 for similar datasets. Although RRKHS demonstrated solid performance, it struggled with issues related to computational efficiency and scalability, especially when dealing with larger datasets or real-time predictions.

- CatBoost-AOA (2024): In Bairami et al. [26], the CatBoost-AOA model showed superior prediction performance in groundwater behavior forecasting. The model achieved a correlation of 0.9977 for predicting groundwater levels over a 10-year horizon. However, while the CatBoost-AOA hybrid model showed impressive correlation and forecast accuracy, it faced some challenges in terms of generalization across diverse hydrological systems and the efficiency of prediction for short-term forecasts. In this study, the AdaBoost-HGS model outperformed CatBoost-AOA in terms of R^2 (0.955), making it more suitable for real-time runoff forecasting, where both speed and accuracy are paramount.

Compared to RRKHS, which handles non-linearity effectively but is less efficient in terms of computational speed, and CatBoost-AOA, which has a strong predictive ability but struggles with generalization for short-term forecasts, the AdaBoost-HGS model not only provides higher R^2 but also achieves faster convergence, making it particularly advantageous for applications where real-time prediction is essential.

In conclusion, the AdaBoost-HGS hybrid model provides significant improvements over RRKHS and CatBoost-AOA by offering higher predictive accuracy, better efficiency, and faster convergence, making it an effective tool for real-time runoff forecasting in hydrological systems.

3.2 Discussion

In this study, we proposed a hybrid AdaBoost-HGS model for runoff prediction, integrating machine learning techniques (CatBoost, AdaBoost, and HGBoost) with metaheuristic optimization strategies (HGS and CGO). The performance of the hybrid model was compared using key metrics, including RMSE, MAE, and R^2. The AdaBoost-HGS model showed promising results, achieving an RMSE of 7.72, R^2 of 0.9549, and a low MAE. When compared to state-of-the-art (SOTA) methods from the literature, the AdaBoost-HGS model demonstrated superior performance. For instance, the RMSE and R^2 values of the AdaBoost-HGS model were considerably better than those of traditional models like SVR-GANN and more advanced methods such as RRKHS, which struggled with prediction accuracy, especially in short-term forecasting. While methods like RRKHS showed good performance in handling non-linear data, they were not as efficient in terms of convergence and computational time. Furthermore, SVR-GANN demonstrated good accuracy but was constrained by challenges related to parameter tuning and scalability.

The superior performance of the AdaBoost-HGS model can be attributed to the characteristics of the HGS optimizer, which enhances the AdaBoost model's convergence rate and accuracy. HGS (Hybrid Gravitational Search Algorithm) combines the strengths of global search capabilities with local search refinements, allowing for more efficient exploration of the solution space, particularly when dealing with complex, non-linear relationships in runoff data. This hybrid approach led to faster convergence and more precise results, particularly in comparison to the standalone CatBoost and AdaBoost models, which exhibited slower convergence and slightly lower accuracy.

To address the weighting factors (0.5, 0.3, and 0.2 for R^2, runtime, and convergence stability, respectively) used in the composite performance evaluation, a sensitivity analysis was conducted to verify the robustness of the chosen weighting scheme. Several alternative combinations were tested, including (0.4, 0.4, 0.2), (0.3, 0.4, 0.3), and (0.6, 0.2, 0.2), representing different emphases on accuracy, computational efficiency, and convergence. For each case, the composite scores of all hybrid models were recalculated based on normalized metrics from Section 3.

The results showed that variations in the weights led to only minor changes in the ranking of the models. In all tested scenarios, AdaBoost–HGS consistently achieved the highest overall score, followed by AdaBoost–CGO and HGBoost–HGS. Specifically, when the weight of R2 was reduced from 0.5 to 0.4, the composite score of AdaBoost–HGS decreased marginally by only 1.8%, while its relative ranking remained unchanged. Conversely, increasing the runtime weight to 0.4 slightly improved the relative performance of CatBoost–CGO, yet the AdaBoost–HGS model still maintained at least 6–8% higher composite efficiency compared with other hybrids. These results confirm that the selected weights of 0.5/0.3/0.2 provide a balanced assessment between predictive accuracy, computational cost, and convergence reliability.

Despite these advantages, there are limitations to the hybrid AdaBoost-HGS model. One of the primary concerns is generalization. The model was evaluated on a single dataset, which limits its ability to generalize across different regions with distinct hydrological characteristics. To address this limitation, future studies should test the model on datasets from multiple regions to assess its adaptability. Additionally, scalability remains a concern. While the AdaBoost-HGS model demonstrated improved efficiency compared to traditional models, its performance on larger datasets with more complex features should be further explored to evaluate its scalability.

The novelty of this work lies in the integration of AdaBoost with the HGS optimizer, which enhances the model's predictive accuracy and efficiency for runoff forecasting. Previous hybrid models, such as CatBoost-AOA and SVR-GANN, have demonstrated strong performance in runoff modeling. However, the AdaBoost-HGS hybrid offers significant improvements in terms of convergence rate and prediction accuracy, especially in short-term forecasting scenarios. Unlike CatBoost-AOA, which primarily focuses on handling nonlinear data, or SVR-GANN, which requires extensive parameter tuning, the AdaBoost-HGS model balances both model robustness and computational efficiency, making it an ideal candidate for real-time runoff prediction applications.

However, this study has limitations. The models were evaluated using historical runoff data, which may not fully account for extreme weather events or sudden climatic changes. Additionally, while the models performed well under the tested conditions, generalization to different hydrological conditions and scalability in diverse geographical regions remain areas for further research. The CatBoost-HGS model, for example, exhibited more error variation, indicating the need for careful model selection and parameter tuning to achieve consistent performance across different datasets and regions.

In conclusion, while the study confirms that AdaBoost-HGS hybrid ML techniques significantly enhance runoff prediction accuracy and efficiency, these models are not yet ready for immediate deployment. Further validation in diverse real-world conditions and across multiple hydrological settings is necessary before full-scale implementation. Future work will focus on improving model scalability and generalization, as well as exploring new optimization methodologies to address the remaining challenges in runoff forecasting.

3.3 Limitations and future work

The analysis in this study was conducted using a single daily streamflow dataset consisting of 29,085 observations recorded between 1914 and 2017 (Water Survey of Canada, Station 08NL007), as cited in [35]. The proposed hybrid models, including the best-performing AdaBoost–HGS configuration, achieved high predictive accuracy on this dataset (test RMSE ≈ 6.79 and test R^2 ≈ 0.955), demonstrating strong capability under the specific hydrological and climatic conditions represented by this station. However, relying on a single dataset restricts the generalizability of the findings, as model performance may vary across different catchments and climatic regions. To address this limitation, future investigations will evaluate the developed hybrid models using additional benchmark datasets from geographically distinct regions. Potential datasets include the Russian River Basin dataset referenced in [35], as well as multi-catchment datasets such as CAMELS and long-term USGS gauging station records, which represent diverse hydro-climatic regimes. The same data preprocessing, lag selection, and rolling-origin validation procedures used in this study will be applied to ensure consistency in model comparison. It is expected that while AdaBoost–HGS and AdaBoost–CGO will remain among the top-performing configurations, absolute performance metrics (e.g., test RMSE and R^2) may show moderate variation depending on the regional hydrological variability. Reporting fold-wise mean errors and 95% confidence intervals across multiple basins will enable a more comprehensive assessment of robustness and uncertainty. Expanding the evaluation in this way will strengthen the reliability and applicability of the proposed framework across broader environmental and operational contexts.

3.4 Computational complexity analysis

The computational complexity of each optimization approach was analyzed to assess efficiency and scalability. For the metaheuristic optimizers employed in this study, the dominant computational cost arises from iterative evaluation of the objective function across multiple candidate solutions. If N_p denotes the population size and I_{max} the maximum number of iterations, the overall time complexity can be expressed as $O\left(N_p \times I_{max} \times f_{eval}\right)$, where f_{eval} represents the average computational cost of a single fitness evaluation. Since the hybrid learning models integrate AdaBoost training within each optimization step, f_{eval} depends primarily on the ensemble size T and the dataset dimension d, yielding an approximate complexity of $O(T \times d)$. Consequently, the total complexity for the proposed AdaBoost–HGS model is on the order of $O(Np \times Imax \times T \times d)$. Empirically, runtime observations align with this theoretical analysis. On the same computing platform (Intel Core i7, 16 GB RAM, Python 3.10 environment), the average runtime per model training was approximately 68.4 s for AdaBoost–HGS, 71.2 s for AdaBoost–CGO, and between 55–60 s for the other hybrid variants. The difference arises mainly from the higher population search overhead in HGS and CGO compared to lighter metaheuristics such as GA or PSO. Despite this increase, the proposed AdaBoost–HGS framework achieved a superior trade-off between accuracy and computational cost, with convergence typically reached within 75% of the maximum iterations. Overall, both the theoretical and empirical analyses confirm that the optimization process scales linearly with respect to population size, iteration count, and base learner complexity, ensuring practical applicability for medium-scale hydrological forecasting tasks.

3.5 Hyperparameter sensitivity analysis

To assess the robustness of the hybrid models and quantify the influence of individual hyperparameters on predictive performance, a comprehensive global sensitivity analysis was conducted. The analysis focused on the AdaBoost–HGS and CatBoost–HGS configurations, which demonstrated the highest accuracy among all tested models. The test-phase RMSE was selected as the output variable, and the sensitivity of five primary hyperparameters was examined: learning rate (η), number of estimators (T), tree depth (d), L2 regularization coefficient (λ, for CatBoost), and optimizer population size (P).

A Latin Hypercube Sampling (LHS) approach was used to generate 300 distinct parameter combinations for each model within the ranges previously optimized ($\eta \in [0.001, 0.3]$, $T \in [50, 1500]$, $d \in [1, 12]$, $\lambda \in [0, 5]$, $P \in [20, 60]$). Each sampled configuration was evaluated through rolling-origin cross-validation, and the mean RMSE across folds was recorded. A Gaussian Process surrogate model was then constructed to approximate the input–output relationship, enabling efficient computation of Sobol sensitivity indices and Delta Moment-

Independent (DMI) indices based on 10,000 Monte Carlo samples.

The resulting Sobol first-order (S1) and total-order (ST) indices are summarized in Table 7. For AdaBoost–HGS, the learning rate exhibited the strongest influence on model performance (S1 = 0.46, ST = 0.53), followed by the number of estimators (S1 = 0.27, ST = 0.34). The tree depth showed moderate sensitivity (S1 = 0.15, ST = 0.22), while the optimizer population size contributed marginally (S1 = 0.07, ST = 0.10). In the CatBoost–HGS model, tree depth (S1 = 0.38, ST = 0.45) and learning rate (S1 = 0.32, ST = 0.39) were the dominant parameters, with regularization exerting a smaller yet noticeable effect (S1 = 0.08, ST = 0.11).

Table 7: Sobol sensitivity indices for key hyperparameters affecting test-phase RMSE

Parameter	AdaBoost–HGS S1	AdaBoost–HGS ST	CatBoost–HGS S1	CatBoost–HGS ST
Learning rate (η)	0.46	0.53	0.32	0.39
Number of estimators (T)	0.27	0.34	0.20	0.26
Tree depth (d)	0.15	0.22	0.38	0.45
Regularization (λ)	—	—	0.08	0.11
Population size (P)	0.07	0.10	0.05	0.08

4 Conclusion

This study explored the application of advanced ML schemes and hybrid schemes in runoff prediction, focusing on the CatBoost, AdaBoost, and HGBoost models. By integrating these schemes with optimization techniques like HGS and CGO, the study aimed to improve the accuracy and efficiency of runoff predictions. Key findings of this study are:

1. Improved Performance of AdaBoost-HGS Hybrid Model: The AdaBoost-HGS hybrid model emerged as the most accurate, with an R² value of 0.9549 and the lowest Mean Squared Error (MSE). This model demonstrated superior performance, confirming its potential for accurate runoff prediction.

2. Enhanced Accuracy with Blended Schemes: The integration of HGS and CGO with ML models significantly improved accuracy. Specifically, the AdaBoost-HGS model exhibited the most considerable reduction in error, emphasizing the effectiveness of blended schemes compared to individual ML models.

3. Run Time and Convergence Efficiency: The AdaBoost-HGS algorithm showed the fastest convergence to optimal solutions, reaching lower MSE in fewer cycles. Similarly, CatBoost-HGS and HGBoost-HGS models demonstrated efficient runtimes, making them suitable for applications that require quick processing.

4. Error Metric Variation: While CatBoost-HGS demonstrated overall high performance, it exhibited higher error variation compared to AdaBoost-HGS and AdaBoost-CGO models. This suggests the need for careful model selection and parameter tuning to achieve consistent performance across different conditions.

5. Impact of Enhancement strategies: The HGS and CGO optimization techniques played a vital role in enhancing the accuracy and efficiency of the hybrid models. These strategies facilitated better model parameter adjustments, contributing significantly to predictive performance.

6. Practical Implications to Runoff Forecasting: This study emphasizes the importance of advanced hybrid models in improving runoff forecasting accuracy, which is critical for flood management, water resource planning, and stormwater infrastructure design.

7. Future Directions: Future research could explore integrating deep learning paradigms into hybrid frameworks, which may further enhance model capabilities in handling complex and large-scale datasets. Additionally, leveraging heterogeneous datasets and testing these models across multiple geographic regions could provide deeper insights into their generalization capabilities and robustness.

In conclusion, this study confirms that hybrid ML techniques significantly enhance runoff prediction accuracy and efficiency, offering valuable contributions to hydrological modeling and resource management. Future work will focus on improving model scalability and generalization, as well as exploring new optimization methodologies to address the remaining challenges in runoff forecasting.

Nomenclature

A10	A10 Index	MVO	Multi-verse optimizer
AdaBoost	Adaptive Boosting Regression	NSE	Nash-Sutcliffe Efficiency Coefficient
ANFIS	Adaptive neuro-fuzzy inference system	P	prior value
ANN	Artificial neural networks	PSO	particle swarm optimization
AOA	Arithmetic Optimization Algorithm	R^2	Coefficient of Determination
BF	The best fitness	RAE	Relative Absolute Error
BP	Back Propagation	RMSE	Root Mean Square Error
BL	Bi-linear	RRKHS	Regression in the Reproducing Kernel Hilbert Space
CatBoost	Categorical gradient boosting	Si	Solution candidate
CGO	Chaos Game Optimization	Si,j	decision variables

F(i)	The fitness value	s_j^i	Initial seed coordinates
FS	Factor of Safety	$s_{i,min}^j$	The lower bounds
GANN	Geomorphology-based Artificial Neural Network	$s_{i,max}^j$	The upper bounds
GB	Global Best	SDI	Streamflow Drought Index
GBR	Gradient Boosting Regression	SVM	Support Vector Machine
HGBoost	Hist Gradient Boosting Regression	SVR	Support Vector regression
HGS	Hunger Games Search	SWMM	Storm Water Management Model
LSSVM	Least square support vector machine	UB and LB	The upper and lower boundaries
MAE	Mean absolute error	WF	The worst fitness
MGi	Group Mean	X_b	The position of the best individual
MLP	Multi-layer perceptron	X(t)	The position of each entity

Authorship contribution statement

Huiying SHAO: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Author statement

The manuscript has been read and approved by all the authors; the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] Napolitano, G., F. Serinaldi and L. See (2011). Impact of EMD decomposition and random initialisation of weights in ANN hindcasting of daily stream flow series: an empirical examination. *Journal of Hydrology*, 406(3–4): 199–214. https://doi.org/10.1016/j.jhydrol.2011.06.015

[2] Wang, W.-C., K.-W. Chau, C.-T. Cheng and L. Qiu (2009). A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, 374(3–4): 294–306. https://doi.org/10.1016/j.jhydrol.2009.06.019

[3] Zhu, S., J. Zhou, L. Ye and C. Meng (2016). Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China. *Environmental Earth Sciences*, 75: 1–12. https://doi.org/10.1007/s12665-016-5337-7

[4] Liu, X (2025). A Comparative Analysis of Ensemble–Metaheuristic Algorithms for Copper Price Forecasting: The NGO-AdaBoost Hybrid Approach. *Informatica*, 49(33). https://doi.org/10.31449/inf.v49i33.8468

[5] Devia, G.K., B.P. Ganasri and G.S. Dwarakish (2015). A review on hydrological models. *Aquatic Procedia*, 4: 1001–1007. https://doi.org/10.1016/j.aqpro.2015.02.126

[6] Harahsheh, K.M. and C.-H. Chen (2023). A survey of using machine learning in IoT security and the challenges faced by researchers. *Informatica*, 47(6). DOI: 10.31449/inf.v47i6.4635

[7] Behzad, M., K. Asghari, M. Eazi and M. Palhang (2009). Generalization performance of support vector machines and neural networks in runoff modeling. *Expert Systems with Applications*, 36(4): 7624–7629. https://doi.org/10.1016/j.eswa.2008.09.053

[8] Young, C.-C., W.-C. Liu and M.-C. Wu (2017). A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events. *Applied Soft Computing*, 53: 205–216. https://doi.org/10.1016/j.asoc.2016.12.052

[9] Ranjan, R. and A.K. Daniel (2023). CoBiAt: A Sentiment ClassificatiCobiat: A Sentiment Classification Model Using Hybrid Convnet-Dual-lstm with Attention Mechanismon Model using Hybrid ConvNet-Dual-LSTM with Attention Mechanism. *Informatica*, 47(4). https://doi.org/10.31449/inf.v47i4.3911

[10] Xiong, J., W. Jiang and X. Qiu (2025). Machine Learning-Driven Multi-Objective Optimization for Intelligent Control in Forage Feed Processing. *Informatica*, 49(34). https://doi.org/10.31449/inf.v49i34.8850

[11] Song, J.-H., Y. Her, J. Park, K.-D. Lee and M.-S. Kang (2017). Simulink implementation of a hydrologic model: a tank model case study. *Water*, 9(9): 639. https://doi.org/10.3390/w9090639

[12] Paik, K., J.H. Kim, H.S. Kim and D.R. Lee (2005). A conceptual rainfall-runoff model considering seasonal variation. *Hydrological Processes: An International Journal*, 19(19): 3837–3850. https://doi.org/10.1002/hyp.5984

[13] Beven, K.J. and M.J. Kirkby (1979). A physically based, variable contributing area model of basin hydrology/Un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrological Sciences Journal*, 24(1): 43–69. https://doi.org/10.1080/02626667909491834

[14] Burnash, R.J.C (1973). A generalized streamflow simulation system: Conceptual modeling for

digital computers. *US Department of Commerce, National Weather Service, and State of California* ….

[15] Jakeman, A.J. and G.M. Hornberger (1993). How much complexity is warranted in a rainfall-runoff model? *Water Resources Research*, 29(8): 2637–2649. https://doi.org/10.1029/93WR00877

[16] Bergström, S (1976). Development and application of a conceptual runoff model for Scandinavian catchments.

[17] Graham, D.N. and M.B. Butts (2005). Flexible, integrated watershed modelling with MIKE SHE. *Watershed Models*, 849336090: 245–272.

[18] Nietsch, S.L., J.G. Arnold, J.R. Kiniry, J.R. Williams and K.W. King (2011). Soil and water assessment tool theoretical documentation, version 2009. *Texas Water Resources Institute, College Station, Texas*.

[19] Orth, R., M. Staudinger, S.I. Seneviratne, J. Seibert and M. Zappa (2015). Does model performance improve with complexity? A case study with three hydrological models. *Journal of Hydrology*, 523: 147–159. https://doi.org/10.1016/j.jhydrol.2015.01.044

[20] Bittelli, M., F. Tomei, A. Pistocchi, M. Flury, J. Boll, E.S. Brooks and G. Antolini (2010). Development and testing of a physically based, three-dimensional model of surface and subsurface hydrology. *Advances in Water Resources*, 33(1): 106–122. https://doi.org/10.1016/j.advwatres.2009.10.013

[21] Partington, D., P. Brunner, C.T. Simmons, A.D. Werner, R. Therrien, H.R. Maier and G.C. Dandy (2012). Evaluation of outputs from automated baseflow separation methods against simulated baseflow from a physically based, surface water-groundwater flow model. *Journal of Hydrology*, 458: 28–39. https://doi.org/10.1016/j.jhydrol.2012.06.029

[22] Yoon, H., S.-C. Jun, Y. Hyun, G.-O. Bae and K.-K. Lee (2011). A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. *Journal of Hydrology*, 396(1–2): 128–138. https://doi.org/10.1016/j.jhydrol.2010.11.002

[23] Xie, T., G. Zhang, J. Hou, J. Xie, M. Lv and F. Liu (2019). Hybrid forecasting model for non-stationary daily runoff series: A case study in the Han River Basin, China. *Journal of Hydrology*, 577: 123915. https://doi.org/10.1016/j.jhydrol.2019.123915

[24] Myronidis, D., K. Ioannou, D. Fotakis and G. Dörflinger (2018). Streamflow and hydrological drought trend analysis and forecasting in Cyprus. *Water Resources Management*, 32: 1759–1776. https://doi.org/10.1007/s11269-018-1902-z

[25] Amiri, E (2015). Forecasting daily river flows using nonlinear time series models. *Journal of Hydrology*, 527: 1054–1072. https://doi.org/10.1016/j.jhydrol.2015.05.048

[26] Bairami, M., H. Khajavi and A. Rastgoo (2024). Assessing groundwater behavior and future trends in the Ardabil Aquifer: A comparative study of groundwater modeling system and categorical gradient boosting hybrid model. *Expert Systems with Applications*, 255: 124728. https://doi.org/10.1016/j.eswa.2024.124728

[27] Bray, M. and D. Han (2004). Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, 6(4): 265–280. https://doi.org/10.2166/hydro.2004.0020

[28] Granata, F., R. Gargano and G. De Marinis (2016). Support vector regression for rainfall-runoff modeling in urban drainage: A comparison with the EPA's storm water management model. *Water*, 8(3): 69. https://doi.org/10.3390/w8030069

[29] Hosseini, S.M. and N. Mahjouri (2016). Integrating support vector regression and a geomorphologic artificial neural network for daily rainfall-runoff modeling. *Applied Soft Computing*, 38: 329–345. https://doi.org/10.1016/j.asoc.2015.09.049

[30] Zhou, Y., S. Guo and F.-J. Chang (2019). Explore an evolutionary recurrent ANFIS for modelling multi-step-ahead flood forecasts. *Journal of Hydrology*, 570: 343–355. https://doi.org/10.1016/j.jhydrol.2018.12.040

[31] Safari, M.J.S., S.R. Arashloo and A.D. Mehr (2020). Rainfall-runoff modeling through regression in the reproducing kernel Hilbert space algorithm. *Journal of Hydrology*, 587: 125014. https://doi.org/10.1016/j.jhydrol.2020.125014

[32] Mohammadi, B., F. Ahmadi, S. Mehdizadeh, Y. Guan, Q.B. Pham, N.T.T. Linh and D.Q. Tri (2020). Developing novel robust models to improve the accuracy of daily streamflow modeling. Water Resources *Management*, 34: 3387–3409. https://doi.org/10.1007/s11269-020-02619-z

[33] Ahmed, I (2022). Determining high-flood-risk regions using rainfall-runoff modeling, In *Flood Handbook*, CRC Press, pp: 447–466.

[34] Zhang, Y., Y. Wang, Y. Zhang, Q. Luan and H. Liu (2021). Multi-scenario flash flood hazard assessment based on rainfall–runoff modeling and flood inundation modeling: a case study. *Natural Hazards*, 105(1): 967–981. https://doi.org/10.1007/s11069-020-04345-6

[35] Whitfield P (2022). 08NL007 daily streamflow dataset from Water Survey of Canada. *Zenodo*.

[36] Gayathri, R., S.U. Rani, L. Čepová, M. Rajesh and K. Kalita (2022). A comparative analysis of machine learning models in prediction of mortar compressive strength. *Processes*, 10(7): 1387. https://doi.org/10.3390/pr10071387

[37] Seifi Majdar, R., A. Rahnamaei and V. Babazadeh (2025). Hybrid Machine Learning in Hydrological Runoff Forecasting: An Exploration of Extreme Gradient-Boosting and Categorical Gradient Boosting Optimization in the Russian River

Basin. *Advances in Engineering and Intelligence Systems*, 4(02): 104–120. https://doi.org/10.22034/aeis.2025.509199.1293

[38] Song, Z., J. Xia, G. Wang, D. She, C. Hu and S. Hong (2022). Regionalization of hydrological model parameters using gradient boosting machine. *Hydrology and Earth System Sciences*, 26(2): 505–524. https://doi.org/10.5194/hess-26-505-2022

[39] Sheng, L. and Y. Xin-quan (2009). Efficient improvement for Adaboost based object detection, In *2009 International Conference on Computational Intelligence and Natural Computing*, IEEE, pp: 95–98. https://doi.org/10.1109/CINC.2009.88

[40] Chang, H.-L. and T.-W. Yue (2011). Entropy-directed AdaBoost algorithm with NBBP features for face detection. *Information Technology Journal*, 10(8): 1518–1526.

[41] Karaboga, D. and B. Basturk (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39: 459–471. https://doi.org/10.1007/s10898-007-9149-x

[42] Zaldívar, D., B. Morales, A. Rodríguez, A. Valdivia-G, E. Cuevas and M. Pérez-Cisneros (2018). A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior. *Biosystems*, 174: 1–21. https://doi.org/10.1016/j.biosystems.2018.09.007

[43] Adnan, R.M., Z. Liang, X. Yuan, O. Kisi, M. Akhlaq and B. Li (2019). Comparison of LSSVR, M5RT, NF-GP, and NF-SC models for predictions of hourly wind speed and wind power based on cross-validation. *Energies*, 12(2): 329. https://doi.org/10.3390/en12020329

[44] Clutton-Brock, T (2009). Cooperation between non-kin in animal societies. *Nature*, 462(7269): 51–57. https://doi.org/10.1038/nature08366

[45] Friedman, M.I., P. Ulrich and R.D. Mattes (1999). A figurative measure of subjective hunger sensations. *Appetite*, 32(3): 395–404. https://doi.org/10.1006/appe.1999.0230

[46] BANA E COSTA, C.A. and J. VANSNICK (1997). Applications of the MACBETH approach in the framework of an additive aggregation model. *Journal of Multi-Criteria Decision Analysis*, 6(2): 107–114. https://doi.org/10.1002/(SICI)1099-1360(199703)6:2%3C107::AID-MCDA147%3E3.0.CO;2-1

[47] Pilkington, A. and R. Fitzgerald (2006). Operations management themes, concepts and relationships: a forward retrospective of IJOPM. *International Journal of Operations & Production Management*, 26(11): 1255–1275. https://doi.org/10.1108/01443570610705854

[48] Borgonovo, E (2006). Measuring uncertainty importance: investigation and comparison of alternative approaches. *Risk Analysis*, 26(5): 1349–1361. https://doi.org/10.1111/j.1539-6924.2006.00806.x

[49] Borgonovo, E (2007). A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92(6): 771–784. https://doi.org/10.1016/j.ress.2006.04.015

[50] Hadjimichael, A., J. Quinn and P. Reed (2020). Advancing diagnostic model evaluation to better understand water shortage mechanisms in institutionally complex river basins. *Water Resources Research*, 56(10): e2020WR028079. https://doi.org/10.1029/2020WR028079

[51] Khajavi, H. and A. Rastgoo (2023). Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms. *Energy*, 272: 127069. https://doi.org/10.1016/j.energy.2023.127069