Journal of Advanced Informatics in Water, Soil, and Structure 2025 2 (2) 254-266

Journal of Advanced Informatics in Water, Soil, and Structure

journal homepage: www.wss.torbath.ac.ir

Research Article

Dynamic K-Nearest Neighbors (DKNN) Approach Optimized by PSO for Daily River Inflow Prediction

Ebrahimi Ehsan[a], *Shourian Mojtaba*[*a]

[a] Faculty of Civil Engineering, Water, and Environmental Engineering, Shahid Beheshti University, Tehran, Iran

ARTICLE INFO

Received date: 11 Sep 2025
Accept date: 17 Oct 2025
Published date: 27 Oct 2025

Keywords:
Data-Driven Model, Dynamic KNN, Particle Swarm Optimization, River Inflow Prediction, Support Vector Machine

Abstract

Accurate prediction of river inflow is one of the most critical components of water resources management. Long-term forecasts are essential for water supply and storage planning, while short-term predictions are vital for anticipating extreme flow events and operating flood warning systems. Data-driven models have been widely adopted as relatively simple yet powerful tools for streamflow prediction. The K-Nearest Neighbor (KNN) method is an effective non-parametric learning approach employed in various hydrological applications; however, its performance is highly sensitive to the selection of the number of neighbors (K), particularly when forecasting extreme values. In this study, a novel approach called Dynamic K-Nearest Neighbor (DKNN) is proposed, which leverages a Support Vector Machine (SVM) model to determine optimal distance thresholds for neighbor selection. Unlike the classical KNN that uses a fixed number of neighbors, DKNN dynamically selects neighbors within an optimized distance for each prediction instance. The Particle Swarm Optimization (PSO) algorithm is employed during calibration to identify optimal distance thresholds that minimize prediction error. The proposed method was evaluated using 14 years of daily inflow data to the Ghashlagh Dam in western Iran. Results demonstrate that DKNN improves overall prediction accuracy by reducing RMSE by 6% compared to classical KNN, with the improvement reaching 8.7% for extreme flow events. The dynamic neighbor selection mechanism enables the model to extract more informative features from the training data and substantially reduces underestimation bias in high-flow conditions, making it a robust tool for flood warning applications.

Homepage: www.wss.torbath.ac.ir

*Corresponding Author:
Shourian, Mojtaba
Email: m_shourian@sbu.ac.ir
ORCID: 0000-0002-4099-9758

https://doi.org/10.22048/wss.2026.575197.1021

How to cite this article:
Ebrahimi, E. 7 Shourian, M . (2025). Dynamic K-Nearest Neighbors (DKNN) Approach Optimized by PSO for Daily River Inflow Prediction. *Journal of Advanced Informatics in Water, Soil, and Structure*, *2*(2), 254-266

1. Introduction

Accurate prediction of river inflow is one of the most fundamental steps in water resources planning, enabling efficient reservoir management for future demands (Araghinejad et al., 2018). Various studies have employed physics-based and data-driven models for streamflow forecasting (Sivakumar et al., 2002). Physics-based models are generally time-consuming, complex, and difficult to implement (Hadi and Tombul, 2018), which has led to the widespread adoption of data-driven approaches. Although data-driven models do not simulate physical and hydrological parameters explicitly, their results are sufficiently accurate to provide significant support for water resources planning and management. Commonly used models for flow prediction include Multiple Linear Regression (MLR), non-parametric methods such as K-Nearest Neighbor (KNN), and parametric approaches including Artificial Neural Networks (ANN), Adaptive Neuro-Fuzzy Inference Systems (ANFIS), and Support Vector Machines (SVM).

Galeati (1990) employed KNN for predicting daily inflow to a hydroelectric dam and compared the results with the ARX method. The findings indicated that both methods performed well, but KNN possesses a simpler structure and is therefore more suitable for larger-scale applications. Shamseldin and O'Connor (1996) utilized the Nearest Neighbor Linear Perturbation Model (NNLPM) for river flow forecasting, combining the concept of nearest neighbors with a linear perturbation model. Their proposed model demonstrated superior accuracy in predicting non-seasonal flows across six different river tests compared to simple linear and linear perturbation models. Souza Filho and Lall (2003) applied KNN to disaggregate annual or seasonal streamflow forecasts into monthly or higher-resolution flows. Their method preserved spatial-temporal consistency across different regions and sub-periods and demonstrated the ability to predict river flow up to 18 months in advance. Laio et al. (2003) showed that KNN slightly outperforms ANN in short-term flood estimation, whereas ANN models perform better in long-term predictions.

The primary strength of KNN lies in its simple algorithmic structure and its applicability to both linear and nonlinear problems. However, the results of this method depend heavily on the selection of the optimal number of neighbors (K), and error spikes may occur at extreme values (Araghinejad, 2013). Domeniconi et al. (2002) found that when query points are not uniformly distributed, finding the optimal value of K becomes extremely difficult. Different applications and scenarios require distinct optimal K values, and determining this remains an ongoing challenge. Huda et al. (2004) proposed a novel approach for dynamic KNN that employed a new type of Euclidean distance to ensure that heterogeneous features have equal influence on the model output. Their proposed method achieved better accuracy than conventional KNN models. Wu et al. (2010) presented a new dynamic nearest neighbor approach that combined three methods: dynamic selection, distance weighting, and attribute weighting. Their method outperformed other dynamic nearest neighbor approaches in classification tasks.

In the present study, an improvement to the classical KNN method is proposed for river flow prediction. Instead of using a fixed number of neighbors, all neighbors within an optimal distance from the input variable are utilized. An SVM model trained on calibration data is employed to determine this optimal distance. Consequently, for each river flow prediction sample, the number of participating neighbors may vary. The advantages of this approach include: (1) improved prediction accuracy, particularly for extreme values, and (2) the ability to leverage more training vectors, enabling the extraction of richer information and features from the data. The proposed DKNN method addresses a fundamental limitation of classical KNN by introducing a data-driven mechanism for adaptive neighbor selection, thereby reducing sensitivity to the fixed K parameter and improving robustness across varying hydrological conditions.

2. Material and Methods

2.1. Study Area and Data

The objective of this study is to predict daily inflow to the Ghashlagh Dam reservoir in western Iran. The Ghashlagh (Vahdat) Dam is located 12 km north of Sanandaj, the capital of Kurdistan Province, Iran, and supplies drinking and agricultural water to the region. The dam was constructed between 1974 and 1983. It is built on the Ghashlagh River, approximately 20 km north of Sanandaj along the Saqqez road, situated in the

Satleh and Tirgareh mountain region. The Ghashlagh River is one of the main tributaries in the Kurdistan Province catchment area, and the dam reservoir plays a crucial role in regulating the regional water supply for both municipal and irrigation purposes throughout the year.

The climate of the study area is characterized by cold, snowy winters and warm, dry summers, which is typical of the Zagros mountain region in western Iran. Annual precipitation in the Ghashlagh Dam catchment ranges between 400 and 700 mm, with the majority of rainfall occurring during the winter and spring months (November through May). The seasonal variability of precipitation directly influences the inflow regime, resulting in pronounced high-flow periods during spring snowmelt and low-flow conditions during the dry summer months. This hydrological pattern makes accurate inflow prediction particularly challenging, as the flow distribution is highly skewed with extreme events concentrated in specific seasons.

Daily average inflow to the Ghashlagh Dam reservoir for 14 years from 2003 to 2017 was utilized for training, calibration, and evaluation of the proposed method. In addition to the average reservoir inflow, daily precipitation recorded by the meteorological station in the dam's catchment area was used as an independent input variable. The average flow from two days prior to the prediction day and precipitation from one day prior were considered as predictor (independent) variables. The one-day-lag precipitation was selected because it exhibited the highest correlation coefficient with flow data in the training dataset. The predictor variables used in this study are formulated as:

$$X = \{Qt\text{-}1, Qt\text{-}2, Pt\text{-}1\}, Y = Qt \quad (1)$$

where X represents the input variables, Q is the inflow discharge, P is the average recorded precipitation in millimeters, t is the time step under examination, and Y is the predicted variable. The prediction time step was set to one day, which is consistent with the nature of reservoir inflow dynamics. For calibration and training of the SVM model, 2 years (730 days) of data were used, and 730 days were processed as evaluation data. Figure 1 shows the 14-year daily average inflow to the Ghashlagh Dam.

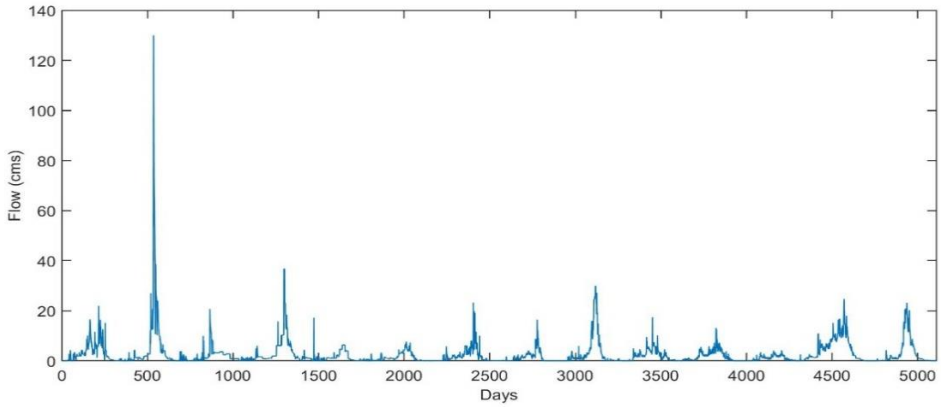

Fig. 1. Time series of average daily inflow to Ghashlagh Dam

2.2. K-Nearest Neighbor (KNN) Machine Learning Method

The K-Nearest Neighbor (KNN) algorithm is one of the most fundamental and widely used machine learning methods in the field of data-driven modeling. As a non-parametric, instance-based learning technique, KNN does not make explicit assumptions about the underlying data distribution, which makes it particularly suitable for hydrological applications where the relationships between input and output variables may be complex and non-linear. Unlike parametric models that require the specification of a functional form during training, KNN relies entirely on the local structure of the training data, making predictions based on the similarity between the current input and historical observations stored in the training set. This lazy learning characteristic means that the model does not undergo a traditional training phase; instead, it defers all computation to the prediction stage, where distances between the query point and all training samples are evaluated.

In the context of regression problems, which is

the focus of this study, the KNN algorithm operates by identifying the K closest training samples to a given query point in the feature space and computing a weighted average of their target values as the prediction. The fundamental premise is that samples with similar input features are likely to have similar output values, a principle that holds particularly well in hydrological time series where temporal autocorrelation creates natural clustering in the feature space. The choice of distance metric is critical to the performance of KNN, as it defines the notion of similarity in the feature space. The most commonly used distance metric is the Euclidean distance, which measures the straight-line distance between two points in the multidimensional feature space. In this study, the Euclidean distance function is employed (Karlsson and Yakowitz, 1987), expressed as:

$$\Delta_{xy} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_m - y_m)^2} \quad (2)$$

where X = {x1, x2, ..., xm} is the predictor vector and Xt = {x1t, x2t, ..., xmt} is a training vector. The Euclidean distance provides an intuitive and computationally efficient measure of similarity, and its effectiveness in hydrological KNN applications has been well documented in the literature. Alternative distance metrics such as Manhattan distance, Mahalanobis distance, and correlation-based distances have been explored in various studies, but the Euclidean distance remains the standard choice for streamflow prediction applications due to its simplicity and robust performance.

Once the distances between the query point and all training samples are computed, the K training samples with the smallest distances are selected as the nearest neighbors. The predicted value is then calculated as a weighted average of the target values associated with these K neighbors. The weighting scheme plays a crucial role in the final prediction quality. In the unweighted version, all K neighbors contribute equally to the prediction, whereas in the weighted version, closer neighbors are assigned higher weights based on their proximity to the query point. The kernel weighting function used in this study assigns weights inversely proportional to the squared distance, which is computed as:

$$f_k(\Delta_k) = \frac{1/\Delta_k}{\sum_{k=1}^{K} 1/\Delta_k} \quad (3)$$

The final predicted value is obtained through the weighted aggregation of the neighbor target values:

$$Y = \sum_{k=1}^{K} f_k(\Delta_k) \times y_k \quad (4)$$

The selection of the optimal K value represents the most critical hyperparameter in KNN modeling and has a profound impact on prediction performance. A small K value makes the model sensitive to noise and local fluctuations in the data, leading to high variance and overfitting, whereas a large K value smooths out the prediction but may include irrelevant or distant neighbors that introduce bias and reduce accuracy. This tradeoff between bias and variance is a well-known challenge in KNN applications, and the optimal K is typically determined through cross-validation procedures on the training data. In hydrological applications, the challenge is further compounded by the non-uniform distribution of flow values in the training data. Extreme flow events, such as floods and droughts, are inherently rare, meaning that the feature space is sparsely populated in these regions. Consequently, a fixed K value that performs well for average flow conditions may lead to significant errors when predicting extreme events, as the K nearest neighbors may include samples from very different hydrological conditions. This limitation serves as the primary motivation for the dynamic neighbor selection approach proposed in this study.

2.3. Developed Dynamic KNN (DKNN) Model

The Dynamic K-Nearest Neighbor (DKNN) model proposed in this study represents a fundamental departure from the classical KNN approach by replacing the fixed number of neighbors (K) with a dynamic, data-driven mechanism for neighbor selection. The core innovation lies in the introduction of an optimal distance threshold that varies for each prediction instance, thereby allowing the number of participating neighbors to adapt automatically based on the specific characteristics of the input vector. This approach directly addresses the primary limitation of classical KNN, namely its sensitivity to the fixed K parameter, particularly in the context of extreme value prediction where the uniform application of a single K value across all hydrological conditions leads to suboptimal

performance.

The theoretical motivation for the DKNN model stems from the observation that the density of training samples in the feature space is highly non-uniform for hydrological time series data. In regions of the feature space corresponding to normal or average flow conditions, training samples are densely distributed, and a moderate K value can easily identify relevant neighbors. However, in regions corresponding to extreme high-flow or low-flow events, the training samples are sparse, and the same K value may force the inclusion of neighbors from different hydrological regimes, leading to significant prediction errors. By transitioning from a fixed-K paradigm to an adaptive distance-based paradigm, the DKNN model allows the neighborhood boundary to expand or contract dynamically based on the local data density around each query point.

The DKNN model operates in two distinct phases: the calibration phase and the prediction phase. During the calibration phase, the PSO optimization algorithm is employed to determine the optimal distance threshold for each prediction variable in the calibration dataset. The objective function for PSO is defined as the minimization of the Root Mean Square Error (RMSE) between the predicted and observed flow values. Specifically, for each calibration sample, PSO searches for the distance threshold d that, when used to select neighbors within that distance from the query point, produces the minimum prediction error. The decision variable in the PSO optimization is the distance threshold d, subject to the constraints that d must be non-negative and must not exceed the maximum feasible neighborhood size determined by the range of the training data. This process is repeated 10 times for each configuration to ensure that the global optimum is reached, and the best result is retained.

Once the optimal distance thresholds are determined for all calibration samples, an SVM model is trained to learn the mapping between the predictor variables and their corresponding optimal distances. The SVM model serves as a meta-model that generalizes the relationship between input conditions and the optimal neighborhood size, enabling the system to predict appropriate distance thresholds for new, unseen data points. The calibration procedure can be summarized as follows:

Step 1: Select a calibration dataset that is separate from both the training and validation datasets to avoid information leakage.

Step 2: For each prediction variable in the calibration dataset, use PSO to find the optimal distance d such that neighbors selected within that distance produce the minimum RMSE for that specific prediction instance.

Step 3: Construct an SVM model using the prediction variables as inputs and the corresponding optimal distances found by PSO as target outputs. The SVM model is trained using the Sequential Minimal Optimization (SMO) algorithm with a linear kernel function.

During the prediction phase, the DKNN model operates as follows:

Step 1: The predictor variables for the new prediction instance are fed into the trained SVM model, which outputs an optimal distance threshold specific to that input vector.

Step 2: The Euclidean distance between the input vector and all training samples is computed, and all neighbors falling within the SVM-predicted optimal distance are selected.

Step 3: The kernel-weighted averaging procedure (identical to classical KNN) is applied to the selected neighbors to compute the final predicted value.

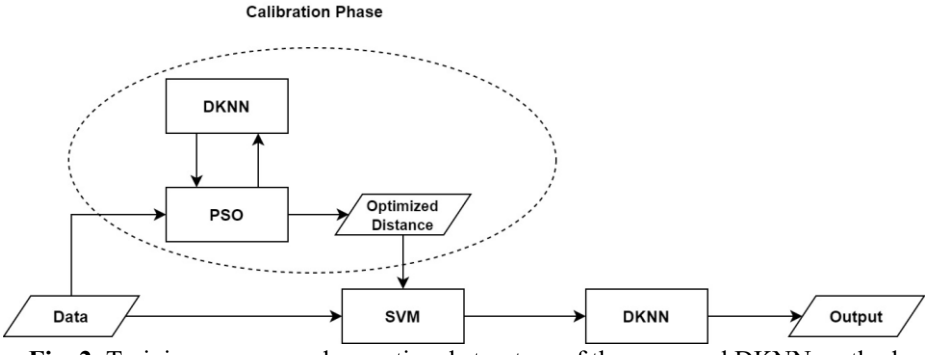

Fig. 2. Training process and operational structure of the proposed DKNN method

Figure 2 illustrates the complete training and operational workflow of the DKNN method. During calibration, PSO identifies optimized distances for different prediction variables, and these variables along with the optimized distances are used to train the SVM model. For each new data point presented to the model, the SVM computes the optimized distance, and DKNN selects all neighbors within that optimal distance. The computation of the optimal distance means that for each input vector of predictor variables, the developed algorithm returns a specific distance that represents the valid neighborhood boundary. Consequently, for each input sample, a different number of neighbors may be selected, effectively creating an optimal prediction pathway tailored to that specific vector. This adaptive mechanism is fundamentally different from the fixed-K approach, as it allows the model to automatically adjust the tradeoff between bias and variance based on the local data density around each query point.

The proposed DKNN method offers several distinct advantages over classical KNN. First, it eliminates the need for an arbitrary selection of the K parameter, replacing it with a principled, data-driven mechanism for determining the neighborhood size. Second, the dynamic neighbor selection allows the model to incorporate more training samples when the local data density supports it, enabling richer feature extraction without the risk of including irrelevant or detrimental neighbors. Third, and most importantly for hydrological applications, the adaptive mechanism reduces the underestimation bias for extreme flow events, as the model can adjust the neighborhood boundary to capture the most relevant information available in the training data for each specific prediction condition. This property makes the DKNN model particularly suitable for flood warning systems, where accurate prediction of peak flows is essential for public safety and infrastructure protection.

2.4. Support Vector Machine (SVM)

Support Vector Machine is among the relatively modern methods that have demonstrated superior efficiency compared to older classification methods, including perceptron neural networks, in recent years. The fundamental principle of SVM classification is linear data separation, where the objective is to select a separating hyperplane that maximizes the margin of confidence. For separating two classes of data points, numerous potential hyperplanes can be chosen. The goal is to find the hyperplane with the maximum margin. To find an appropriate hyperplane, it may be necessary to map variables to a higher dimension using kernel functions. Various algorithms have been proposed to solve the SVM optimization problem. Since quadratic programming typically requires significant memory for kernel function computation and is difficult to implement, it is not suitable for large-scale problems. To mitigate this issue, subset selection methods have been developed, and Sequential Minimal Optimization (SMO) employs a subset selection algorithm to optimize the kernel function. In this study, the SMO algorithm was used for training the SVM model, and after testing different functions, the Linear kernel was selected as the kernel function based on its superior performance on the calibration dataset.

In the context of the DKNN framework, the SVM model serves as the distance-prediction meta-model that maps input predictor vectors to their corresponding optimal neighborhood distances. The SVM regression model is trained using the optimal distances determined by PSO during the calibration phase as target values, with the predictor variables as inputs. The linear kernel was selected after systematic evaluation of multiple kernel functions (including polynomial, radial basis function, and sigmoid kernels) on the calibration data, as it provided the best generalization performance while maintaining computational efficiency.

2.5. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an intelligent swarm-based method used for solving optimization problems. PSO can be easily implemented, and since its computational and memory requirements are relatively low, it is computationally efficient. In PSO, a group of candidate solutions moves around the search space using velocity and weight formulas. These formulas are influenced by each particle's best-known position in the search space as well as the group's best position at each iteration. Therefore, if a better position is found in a new iteration, it directs the group's movement toward that position. The two main PSO equations that compute the

velocity and position of each particle in the search space at each iteration are expressed as:

$$v_{i+1} = K\left[w_i \times v_i + C_1 r_1 \left(p_{best_i}{}^{-x_i}\right) + C_2 r_2 \left(G_{best_i}{}^{-x_i}\right)\right] \quad (5)$$

$$x_{i+1} = x_i + v_{i+1} \quad (6)$$

where K is the constriction coefficient, w is the weight coefficient, c1 and c2 are the cognitive and social parameters respectively, and r1 and r2 are random values uniformly generated between 0 and 1. pbesti is the best position found by particle i, while gbest is the best position visited by all particles.

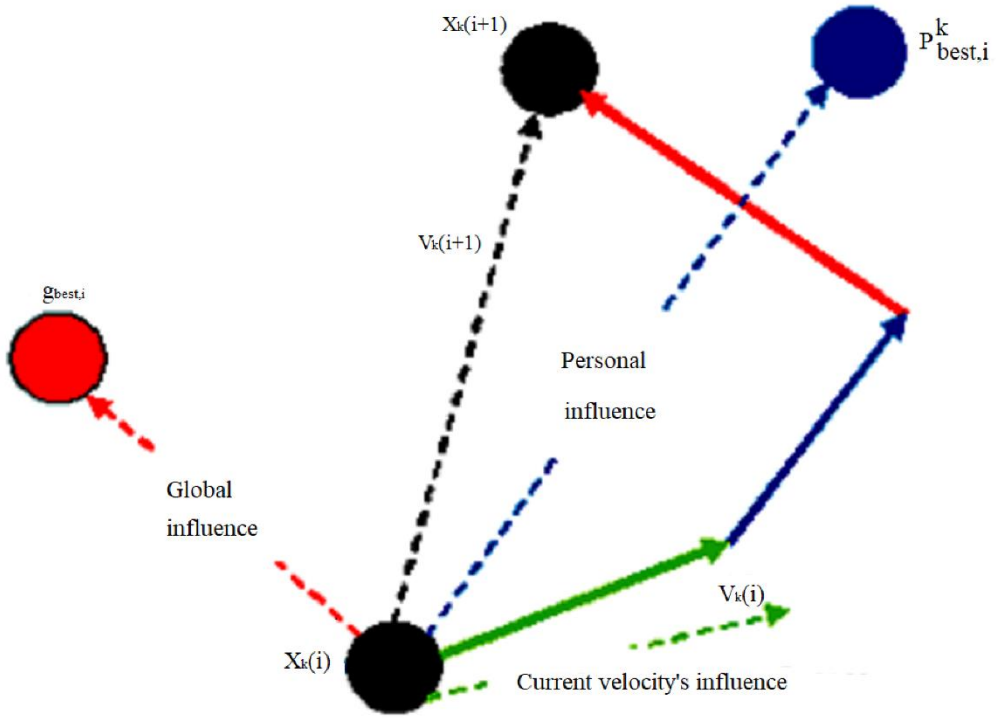

Fig.1 Generation scheme of the particles (Adopted from Jain et al., 2022)

In this study, the objective function in the PSO algorithm is to minimize the RMSE index between predicted and observed river flow values in the calibration data. The decision variables include the neighborhood distance (d) for selecting neighbors in the KNN algorithm, and the model constraints include non-negativity of the distance and the maximum neighborhood size limit (based on the range of training data). To ensure the best result is obtained with PSO, each configuration was optimized 10 times and the best result was selected. The combination of PSO with the KNN algorithm enables the systematic determination of optimal neighborhood distances, replacing the traditional trial-and-error approach for K selection with a principled optimization framework.

2.6. Operational Steps of the Research and Evaluation Criteria

This subsection provides a comprehensive overview of the operational steps followed in this research, along with the evaluation criteria used to assess model performance. Presenting the research workflow in a structured, step-by-step manner enables the reader to develop a clear mental image of the various stages involved and to understand the logical connections between different components of the methodology. Figure 3 presents the research operational flowchart, which summarizes the entire workflow from data collection through to the final comparative analysis.

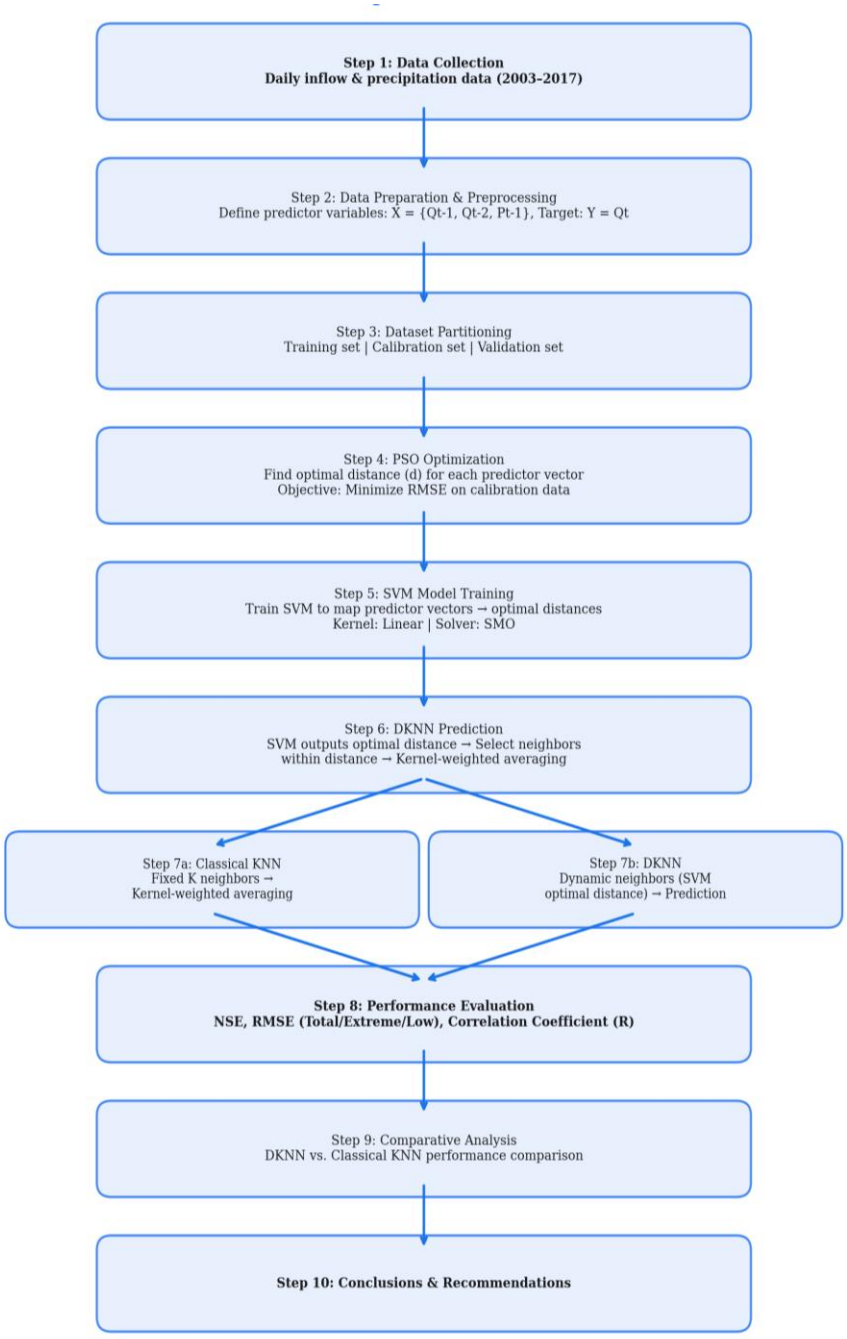

Fig. 3. Research operational flowchart illustrating the sequential steps from data collection to final evaluation

The operational steps of this research can be described as follows. In Step 1, the daily inflow and precipitation data for the Ghashlagh Dam catchment are collected from the relevant water resources authorities, covering the period from 2003 to 2017. In Step 2, the collected data are preprocessed and the predictor variables are defined. The input feature vector X = {Qt-1, Qt-2, Pt-1} is constructed, where Qt-1 and Qt-2 represent the inflow discharge from one and two days prior, respectively, and Pt-1 represents the precipitation from one day prior. The target variable is defined as Y = Qt, the inflow discharge on the prediction day. In Step 3, the dataset is partitioned into three subsets: the training set, the calibration set, and the validation set. The training set provides the historical database from which neighbors are selected, the calibration set is used to determine the optimal distance thresholds through PSO optimization, and the validation set serves as

the independent test set for final model evaluation.

In Step 4, the PSO optimization algorithm is applied to the calibration dataset. For each calibration sample, PSO searches for the optimal distance threshold d that minimizes the RMSE between the KNN prediction (using neighbors within distance d) and the observed flow value. This step produces a set of optimal distances, one for each calibration sample, which represents the ideal neighborhood boundary for the corresponding input conditions. In Step 5, the SVM model is trained using the predictor variables from the calibration set as inputs and their corresponding optimal distances as target outputs. The SVM model learns to generalize the relationship between input conditions and optimal neighborhood distances, enabling it to predict appropriate distances for new, unseen data points. In Step 6, the DKNN model is applied to make predictions on the validation dataset. For each validation sample, the SVM model first predicts the optimal distance, then all neighbors within that distance are selected from the training set, and finally the kernel-weighted averaging procedure computes the predicted flow value.

In Step 7, the classical KNN model is also applied to the same validation dataset for comparative purposes. The classical KNN model uses a fixed K value determined through cross-validation on the training data. Both the classical KNN and DKNN predictions are generated for the same validation period to ensure a fair comparison. In Step 8, the performance of both models is evaluated using the evaluation criteria described below. In Step 9, a comparative analysis is conducted to assess the relative performance of DKNN versus classical KNN, with particular attention to the prediction of extreme flow events and low-flow conditions. Finally, in Step 10, conclusions and recommendations are drawn based on the comparative results.

The following evaluation criteria are employed to quantitatively assess the performance of the proposed DKNN method and the classical KNN model. These criteria provide complementary perspectives on model performance and collectively offer a comprehensive evaluation framework.

Nash-Sutcliffe Efficiency (NSE): This criterion measures the relative magnitude of the residual variance compared to the measured data variance, with values closer to 1 indicating better model performance. NSE is widely used in hydrological modeling as it provides a normalized measure that facilitates comparison across different catchments and flow regimes.

$$NSE = 1 - \frac{\sum_{t=1}^{n}(T_t - Y_t)^2}{\sum_{t=1}^{n}(T_t - \bar{T})^2} \quad (7)$$

Root Mean Square Error (RMSE): This criterion quantifies the standard deviation of prediction residuals, with lower values indicating better model accuracy.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(T_t - Y_t)^2}{n}} \quad (8)$$

Correlation Coefficient (R): This criterion measures the linear relationship between observed and predicted values, with values closer to 1 indicating stronger correlation.

$$R = \frac{\sum_{t=1}^{n}(T_t - \bar{T})(Y_t - \bar{Y})}{\sum_{t=1}^{n}(T_t - \bar{T}) \cdot \sum_{t=1}^{n}(Y_t - \bar{Y})} \quad (9)$$

where n is the number of prediction samples, Qobs is the observed flow, Qest is the estimated value, Qobs,mean is the mean observed flow, and Qest,mean is the mean estimated value. To evaluate the performance of the proposed method under extreme conditions and for low and high flow values, RMSE was calculated separately for validation data greater than 10 m^3/s (extreme values) and for data below 1 m^3/s (low values). This disaggregated evaluation provides critical insight into model behavior across the full spectrum of hydrological conditions, which is essential for practical flood warning and drought management applications. The evaluation under different flow regimes is particularly important because a model that performs well on average may still fail catastrophically at extreme values, which are precisely the conditions where accurate predictions matter most.

2.7. Modeling Environment and Platform Specifications

The entire modeling framework proposed in this study was implemented in the MATLAB programming environment (version R2019b, MathWorks Inc.), which provides robust built-in functions for data analysis, machine learning, and optimization. The KNN algorithm was implemented using custom scripts developed specifically for this research, enabling full control over the distance computation, neighbor selection,

and kernel weighting procedures. The SVM model was constructed using the MATLAB Statistics and Machine Learning Toolbox, with the fitrsvm function employed for regression model training. The Sequential Minimal Optimization (SMO) algorithm was used as the solver for the SVM quadratic programming problem, and the linear kernel function was selected after systematic evaluation of multiple kernel options.

The PSO optimization was implemented using the MATLAB Global Optimization Toolbox, specifically the particleswarm function, with the following parameter configuration: population size of 50 particles, maximum iteration limit of 200, cognitive coefficient c1 = 1.49, social coefficient c2 = 1.49, inertia weight w linearly decreasing from 0.9 to 0.4, and constriction coefficient K = 1.0. The objective function was set to minimize the RMSE between predicted and observed flow values on the calibration dataset. Each PSO optimization run was repeated 10 times with different random initializations to reduce the risk of convergence to local optima, and the best solution across all runs was retained as the final optimal distance threshold.

All computations were performed on a desktop computer equipped with an Intel Core i7-8700 processor (3.20 GHz, 6 cores), 16 GB of RAM, and the Windows 10 operating system. The total computational time for the complete DKNN modeling workflow, including PSO optimization for all calibration samples, SVM training, and validation prediction, was approximately 45 minutes. The PSO optimization phase accounted for the majority of the computational burden, as it was repeated for each calibration sample, with 10 independent runs per sample. The SVM training and DKNN prediction phases were computationally efficient, each requiring only a few seconds. The classical KNN model, in contrast, required negligible computation time due to its simpler structure and the absence of an optimization step.

The data preprocessing and analysis tasks, including temporal lag construction, dataset partitioning, and statistical evaluation, were performed using standard MATLAB functions and custom scripts. The predictor variables were standardized to a zero mean and unit variance before model training to ensure that all features contributed equally to the distance computations and to improve the convergence of the optimization algorithms. No additional feature selection or dimensionality reduction techniques were applied, as the predictor set was already limited to three variables based on domain knowledge and preliminary correlation analysis.

3. Results and Discussion

To evaluate the performance of the proposed method, the inflow to Ghashlagh Dam was predicted using both classical KNN and DKNN. Figure 4 presents the regression diagrams for the classical KNN and DKNN models. Based on the R value and the proximity of the fitted line to the Y = T identity line, the proposed DKNN model outperforms the classical model. The DKNN regression points exhibit tighter clustering around the identity line, particularly in the high-flow region, indicating reduced underestimation bias compared to the classical approach.

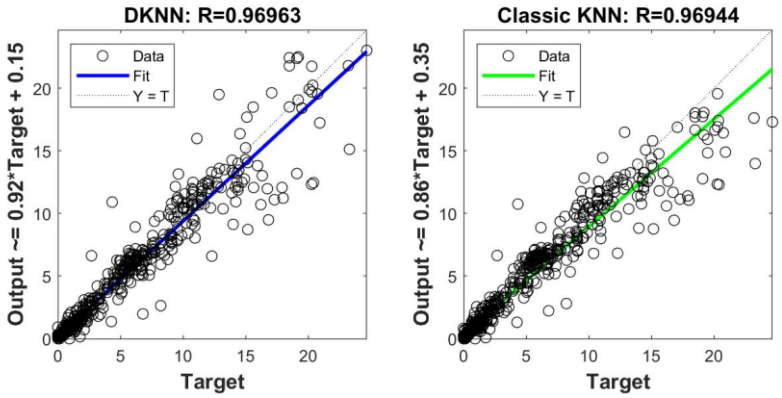

Fig. 4. Regression diagrams for classical KNN model and proposed DKNN model

Figure 5 shows the time series comparison between the DKNN model results, the classical model results, and the observational data. The DKNN model generally performs better than the classical model and exhibits lower error in predicting extreme values. Notably, the classical KNN model produces underestimations for large flow values, whereas in many of these instances, the DKNN model achieves predictions closer to the observational data. This improvement is attributable to the dynamic selection of neighbors tailored to each prediction variable, which allows the model to adaptively incorporate relevant information from the training data based on the specific hydrological conditions of each prediction instance.

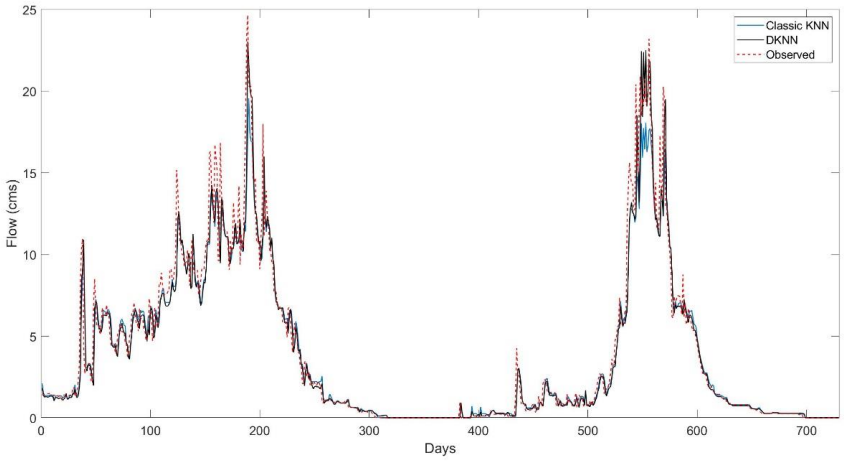

Fig. 5. Time series of observed flow, classical KNN model, and proposed DKNN model

Table 1 presents the performance indices for the classical and DKNN models. The proposed DKNN model shows a 6% improvement over classical KNN in terms of overall RMSE, demonstrating the effectiveness of this new approach. For extreme values above 10 m^3/s, which is a critical indicator of model performance in river flow prediction, the proposed method achieves even better performance with the accuracy improvement increasing to 8.7%. This highlights the strength of the DKNN method for extreme flow prediction. The improvement in extreme value prediction is particularly noteworthy because accurate forecasting of peak flows is essential for flood warning systems and reservoir operation during flood events. The classical KNN model's tendency to underestimate extreme flows poses significant risks, as it may lead to inadequate preparation for flood events and potentially catastrophic dam safety concerns.

Table 1. Statistical performance criteria for classical KNN and DKNN models

Phase	Model	NSE	Total RMSE	Extreme RMSE	Low RMSE	R
Validation	DKNN	0.939	1.260	2.864	0.152	0.970
Validation	Classic KNN	0.931	1.340	3.105	0.140	0.969
Calibration	DKNN	0.826	1.517	7.272	0.286	0.675
Calibration	Classic KNN	0.810	1.793	7.632	0.360	0.651

The results clearly demonstrate that the proposed DKNN method not only enhances prediction accuracy but also enables the extraction of more informative features from the data through dynamic neighbor selection. The sensitivity of prediction accuracy to the number of neighbors for extreme values is effectively addressed by the dynamic approach, which improves the precision of extreme flow forecasts. This makes the model particularly suitable for flood warning systems. The underestimation by the classical model for extreme events is dangerous and can lead to dam failure and irreparable damage, whereas DKNN mitigates this critical limitation through its

adaptive neighbor selection mechanism. The NSE values above 0.93 in the validation phase for DKNN indicate excellent model performance according to the commonly accepted classification in hydrological modeling, while the improvement in extreme RMSE from 3.105 to 2.864 represents a meaningful reduction in prediction error for the most critical flow conditions.

4. Conclusions

This study proposed a Dynamic K-Nearest Neighbor (DKNN) method optimized by Particle Swarm Optimization (PSO) for daily river inflow prediction. The key innovation of DKNN lies in replacing the fixed K parameter of classical KNN with a dynamic, data-driven neighbor selection mechanism based on optimal distance thresholds determined through SVM modeling. The proposed method was evaluated using 14 years of daily inflow data to the Ghashlagh Dam in western Iran, and the following conclusions can be drawn:

First, DKNN achieves a 6% improvement in overall RMSE compared to classical KNN, demonstrating that adaptive neighbor selection provides a more robust prediction framework than fixed-neighbor approaches. Second, the improvement is more pronounced for extreme flow events, reaching 8.7% reduction in RMSE for flows exceeding 10 m^3/s. This is particularly significant for flood warning applications, where underestimation of peak flows can have catastrophic consequences. Third, the dynamic neighbor selection mechanism allows the model to leverage more training vectors when appropriate, enabling richer feature extraction from the data without the risk of including irrelevant or detrimental neighbors that would compromise prediction accuracy.

The proposed DKNN framework addresses a fundamental limitation of classical KNN and demonstrates that dynamic, context-aware neighbor selection can substantially improve prediction performance in hydrological forecasting. Future research directions include extending the DKNN approach to multi-step-ahead forecasting, evaluating its performance across diverse climatic and hydrological regimes, and integrating it with other machine learning techniques for hybrid modeling frameworks.

Funding Sources

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

Declaration of Competing Interest

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

Author's Contribution Statement

All authors contributed to the study conception and design. Model development, data collection, and analysis were performed collaboratively. All authors read and approved the final manuscript.

References

Araghinejad, S. (2013). Data-driven modeling: using MATLAB® in water resources and environmental engineering (Vol. 67). Springer Science & Business Media.

Araghinejad, S., Fayaz, N., & Hosseini-Moghari, S.M. (2018). Development of a hybrid data driven model for hydrological estimation. *Water Resources Management, 32*(11), 3737-3750.

Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(9), 1281-1285.

Galeati, G. (1990). A comparison of parametric and non-parametric methods for runoff forecasting. *Hydrological Sciences Journal, 35*(1), 79-94.

Hadi, S.J., & Tombul, M. (2018). Monthly streamflow forecasting using continuous wavelet and multi-gene genetic programming combination. *Journal of Hydrology, 561*, 674-687.

Huda, M.S., Alam, K.M.R., Mutsuddi, K., Rahman, M.K.S., & Rahman, C.M. (2004). A dynamic k-nearest neighbor algorithm for pattern analysis problem. *In 3rd International Conference on Electrical & Computer Engineering*.

Karlsson, M., & Yakowitz, S. (1987). Nearest-neighbor methods for nonparametric rainfall-runoff forecasting. *Water Resources Research, 23*(7), 1300-1308.

Laio, F., Porporato, A., Revelli, R., & Ridolfi, L. (2003). A comparison of nonlinear flood forecasting methods. *Water Resources Research, 39*(5).

Shamseldin, A.Y., & O'Connor, K.M. (1996). A nearest neighbour linear perturbation model for river flow forecasting. *Journal of Hydrology, 179*(1-4), 353-375.

Sivakumar, B., Jayawardena, A.W., & Fernando, T.M.K.G. (2002). River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. *Journal of Hydrology, 265*(1-4), 225-245.

Souza Filho, F.A., & Lall, U. (2003). Seasonal to interannual ensemble streamflow forecasts for Ceara, Brazil: Applications of a multivariate, semiparametric algorithm. *Water Resources Research, 39*(11).

Wu, J., Cai, Z., & Gao, Z. (2010). Dynamic K-Nearest-Neighbor with distance and attribute weighted for classification. *In 2010 International Conference on Electronics and Information Engineering* (Vol. 1, pp. V1-356). IEEE.

Jain, M., Saihjpal, V., Singh, N., & Singh, S. B. (2022). An Overview of Variants and Advancements of PSO Algorithm. Applied Sciences, 12(17), 8392. https://doi.org/10.3390/app12178392