

https://doi.org/10.51885/3134-8025_IICS_2026_3_7

SRSTI 28.23.25

APPLICATION OF MACHINE LEARNING ALGORITHMS FOR DAILY WATER LEVEL PREDICTION IN KAZAKHSTAN RESERVOIRS

ҚАЗАҚСТАН СУ ҚОЙМАЛАРЫНДАҒЫ ТӘУЛІКТІ СУ ДЕҢГЕЙІН БОЛЖАУ ҮШІН МАШИНАЛЫҚ ОҚЫТУ АЛГОРИТМДЕРІН ҚОЛДАНУ

ПРИМЕНЕНИЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ЕЖЕДНЕВНОГО ПРОГНОЗИРОВАНИЯ УРОВНЯ ВОДЫ В КАЗАХСТАНСКИХ ВОДОХРАНИЛИЩАХ

Z.O. Oralbekova ¹, P.A. Pertsev ¹, Z.T. Khassenova ^{2*}, D.O. Oralbekova ³,
E.A. Aitmukhanbetova ¹

¹Astana IT University, Astana, Kazakhstan

²D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan

³Institute of information and computational technologies, Almaty, Kazakhstan

*Corresponding author: Zarina Khassenova, e-mail: zkhasanova@edu.ektu.kz

Keywords:

Machine Learning,
K-Nearest Neighbors,
Hybrid Model, Water Level
Prediction, Kazakhstan
Reservoirs.

ABSTRACT

Accurate water-level prediction is essential for effective water resource management in Kazakhstan. Reservoirs play an important role in irrigation, hydropower generation, and water supply. Traditional forecasting methods often struggle to capture complex interactions between natural processes and human activities, especially when data are limited or conditions change rapidly. This study applies three baseline machine learning models, Decision Tree, Random Forest, and K-Nearest Neighbors, as well as a hybrid model based on Random Forest Regressor and Multilayer Perceptron, to predict daily water levels using historical Kazhydromet data. The data were preprocessed by handling missing values and applying feature scaling. The experimental results showed that the hybrid RF+MLP model achieved the best performance among the baseline and proposed models. Additional TimeSeriesSplit validation and comparison with advanced models confirmed that RF-based models provide relatively stable results under limited data conditions. The findings indicate that machine learning methods, especially hybrid approaches, can support daily water-level forecasting and improve water resource management in Kazakhstan.

Түйінді сөздер:

машиналық оқыту,
К-жақын көршілер,
гибридті модель, су
деңгейін болжау,
Қазақстан су қоймалары.

ТҮЙІНДЕМЕ

Қазақстанда су ресурстарын тиімді басқару үшін су деңгейін дәл болжау өте маңызды. Су қоймалары суаруда, гидроэнергетика өндірісінде және сумен жабдықтауда маңызды рөл атқарады. Дәстүрлі болжау әдістері, әсіресе деректер шектеулі болғанда немесе жағдайлар тез өзгерген кезде, табиғи процестер мен адам қызметі-



нің арасындағы күрделі өзара әрекеттесуді анықтауда қиындықтарға тап болады. Бұл зерттеуде Қазгидрометтің архивті деректеріне негізделген күнделікті су көлемін болжау үшін үш негізгі машиналық оқыту моделдері – шешім ағаштары, кездейсоқ орман мен К-ең жақын көршілер, сондай-ақ кездейсоқ орман регрессоры мен көп қабатты перцептронға негізделген гибриді модель қолданылды. Деректер жетіспейтін мәндерді өңдеу және мүмкіндіктерді масштабтауды қолдану арқылы алдын ала өңделді. Тәжірибелік нәтижелер гибриді RF+MLP моделі базалық және ұсынылған модельдер арасында ең жақсы нәтижеге қол жеткізгенін көрсетті. Қосымша TimeSeriesSplit валидациясы және озық модельдермен салыстыру РФ негізіндегі модельдер шектеулі деректер жағдайында салыстырмалы түрде тұрақты нәтижелер беретінін растады. Зерттеу нәтижелері машиналық оқыту әдістерінің, әсіресе гибриді тәсілдердің, күнделікті су деңгейін болжауды қолдай алатынын және Қазақстанда су ресурстарын басқаруды жақсартуға алатынын көрсетеді.

Ключевые слова:

Машинное обучение,
метод К-ближайших
соседей, гибридная
модель, прогнозирование
уровня воды,
водохранилища
Казахстана.

АННОТАЦИЯ

Точное прогнозирование уровня воды имеет важное значение для эффективного управления водными ресурсами в Казахстане. Водохранилища играют важную роль в ирригации, гидроэнергетике и водоснабжении. Традиционные методы прогнозирования часто с трудом улавливают сложные взаимодействия между природными процессами и деятельностью человека, особенно когда данные ограничены или условия быстро меняются. В этом исследовании используются три базовые модели машинного обучения – Дерево решений, Случайный лес и К-ближайших соседей – а также гибридная модель на основе Случайного леса регрессора и Многослойного перцептрона для прогнозирования суточных объёмов воды на основе архивных данных Казгидромета. Данные были предварительно обработаны путем удаления пропущенных значений и масштабирования признаков. Экспериментальные результаты показали, что гибридная модель RF+MLP показала наилучшие результаты среди базовой и предложенной моделей. Дополнительная проверка с использованием TimeSeriesSplit и сравнение с улучшенными моделями подтвердили, что модели на основе случайного леса обеспечивают относительно стабильные результаты в условиях ограниченных данных. Полученные результаты показывают, что методы машинного обучения, особенно гибридные подходы, могут поддерживать ежедневное прогнозирование уровня воды и улучшать управление водными ресурсами в Казахстане.

INTRODUCTION

Research indicates that Random Forest and deep learning models are well-suited for forecasting water levels. Random Forest provides both strong accuracy and interpretability, which makes it beneficial for day-to-day application.

Effective water resource management is increasingly crucial due to shifts in climate, population growth, and economic needs. In Kazakhstan, reservoirs support irrigation, power generation, and urban water supply, thus accurate prediction of daily water levels is vital. Water processes are dynamic and affected by factors like rainfall, temperature, and human activities. Data limitations can also reduce standard water models.

Traditional mathematical and statistical methods often rely on assumptions that may not fully reflect regional hydrological conditions. As a result, their performance may decrease under rapidly changing conditions or when data availability is limited. In recent years, machine learning has become a viable alternative. Machine learning models can adjust for variability, missing data, and large datasets by learning from historical data, without requiring specific physical rules.

Evidence indicates that machine-learning models, such as Random Forests and XGBoost, outperform conventional models for the purpose of water resources forecasting. Decision Tree remain widely used of their interpretability, while K-Nearest Neighbors offers an uncomplicated model for benchmarks. In addition, with the incorporation of machine learning to remote-sensing data and Internet of Things (IoT) sensors, the use of data-driven approaches is becoming more prevalent over the use of statistical methods in water management globally.

These trends have important implications for Kazakhstan. The country's arid climate, dependence on international rivers, and variable water availability create specific challenges in prediction. Adaptable and precise models could significantly improve reservoir management and water security.

This study explores the application of three machine learning techniques—Decision Tree, Random Forest, K-Nearest Neighbors, and hybrid model based on Random Forest Regressor and Multilayer Perceptron - to forecast daily water levels in Kazakhstan's reservoirs, drawing on Kazhydromet data from 1995 to 2022. By evaluating their predictive performance, this study aims to identify reliable methods for regional water management and to contribute to the understanding of machine learning applications in water resources.

LITERATURE REVIEW

Machine learning (ML) has become an important tool in hydrological research, providing data-driven alternatives to traditional physically based models. Hydrological systems are inherently complex, non-linear, as well as subject to numerous interactive factors such as rainfall, temperature, land use, as well as human activity. Traditional models often struggle with such dynamics, especially under spatial settings where data is scarce or climatic conditions rapidly changing. Machine learning presents a different way by learning patterns from existing data, and its use in runoff modeling has grown (Mohammadi, 2021).

In contrast, ML algorithms are excellent at predicting outcomes such as the water level in a reservoir, flood modeling, and streamflow predictions since they are able to learn patterns directly from prior observations. Hydrologists are especially interested in fast and scalable methods such as ML because of its ability to learn from complex inputs, nonlinear relationships, and missing data.

For Kazakhstan, where hydrological data may be fragmented or incomplete, ML can support water resource management and improve prediction accuracy.

Machine learning (ML) is becoming more important in operational hydrology. For instance, Tyralis et al. (2019) looked at how ML is used in water resources and found that ensemble methods and deep learning models are being used more often instead of standard statistical methods. Jayatilake et al. (2022) showed how ML is integrated with remote sensing and IoT to allow for real-time observation and forecasting of hydrologic parameters.

ML could help improve water resource management and prediction accuracy in Kazakhstan, where hydrological data might be scattered or not complete. ML, aside from flow prediction, has also found use in assessing water quality, as shown in hydrological work (Zhu et. al, 2022).

Decision Tree (DT) and Random Forest (RF) are common ML algorithms in hydrology because they are easy to understand, strong, and can model nonlinear relationships. DT makes a flowchart-like structure with decisions based on feature thresholds. RF creates a set of decision

trees to lower overfitting and improve generalization. Pham et. al. (2021) showed that RF works well for short-term streamflow forecasts in areas where rainfall and snowmelt affect water flow.

In this study, DT, RF, K-Nearest Neighbors, and hybrid model based on Random Forest Regressor and Multilayer Perceptron were used to predict daily water levels in Kazakhstan's reservoirs using the Kazhydromet dataset (1995–2022). X. Wang et.al. (2018) used similar methods to predict groundwater levels with enhanced random forest regression. RF did better than DT in all train-test splits. This shows that RF is better at understanding complex patterns in time series data.

Abozweita et al. (2024) reported similar results in a Malaysian case study, where RF did better than DT and other models in predicting monthly inflows. That study showed that RF is resistant to noise and can deal with missing data by using internal averaging.

Jibril et al. (2022) showed that RF is good at streamflow prediction across many basins, and that it can adapt to different hydrological conditions. These results support RF as a key model in hydrological forecasting.

K-Nearest Neighbors (KNN) is a simple instance-based algorithm. It predicts values using the most similar observations in the training dataset. In hydrology, KNN has been used for streamflow prediction, water quality assessment, and rapid estimation of hydrological indicators.

Recent advances in hydrological forecasting have increasingly explored hybrid models that combine conventional machine-learning methods, such as RF, with neural-network components, such as MLP or other deep architectures. The rationale is to combine the ensemble-based predictive capability of RF with the nonlinear modeling capacity of neural networks.

One common hybrid structure trains a RF model first on the input features, obtains its predictions or residuals, and then uses those predictions (or residuals) as additional inputs to a neural network.

This two-stage process can be described as:

1. First stage: the RF regressor learns from the preprocessed input features and produces predicted values (or errors) on training data.
2. During the second stage, a neural network is trained on the original features and the predictions from the RF model to refine the final result. Thus, the neural network may correct systematic errors captured by a RF but missed by the RF ensemble, and capture relationships that the ensemble did not capture.

Empirical findings indicate that hybrids very often provide reliable but limited improvements in major metrics (e.g., RMSE reduction, increases in R^2), compared to either RF or neural models independently. For instance, Zhou et al. (2023) developed a SA-CNN-LSTM hybrid for rainfall-runoff simulation that outperformed either RF or single models in terms of RMSE and R^2 .

These results support the inclusion of a hybrid RF with MLP model in the study's methodology. For reservoir data in Kazakhstan, which may be incomplete or non-stationary, this approach provides a flexible way to capture nonlinear relationships and improve the initial predictions generated by RF.

Comparative evaluation of ML models is essential for selecting appropriate methods for hydrological forecasting tasks. Previous studies have evaluated DT, RF, KNN, and neural network models using different datasets and study regions.

Jayathilake et al. (2022) conducted a comparative study of ML and deep learning models for streamflow prediction. They found that CNN-LSTM combos beat all the other models, plus random forests and KNN, particularly in capturing long-term temporal patterns. However, random forest remained competitive in terms of training efficiency and practical implementation.

Ahmed et al. (2022) evaluated XGBoost, random forests, and KNN for river inflow prediction in India. RF and XGBoost did the best, with the random forest being better at working

with data it hadn't seen before. The study showed that using a mix of methods is helpful when dealing with noisy and incomplete hydrological data.

Machine learning applications in Central Asia research are relatively new but show positive signs. Kazakhstan's large and varied water systems create specific problems for predicting water levels. The Kazhydromet data set in this work is a rare public source covering many reservoirs for over 20 years.

Kalashnikova et al. (2024) studied water modeling in Central Asia, noting the need for local models that consider regional climate changes and infrastructure limits. It backs up using ML to fill missing data and make better predictions.

Jibril et al. (2022) emphasized adaptive models for changing data. RF's group approach fits these tasks because it can follow changes in data patterns over time.

Also, Kazhydromet and regional water groups have started adding ML to their forecasting, specifically in the Aral Sea and Syrdarya areas. This study contributes to this research direction by demonstrating the applicability of ML-based water-level forecasting for Kazakhstan.

Previous studies indicate that RF and deep learning models are effective for water-level prediction. RF is accurate and easy to understand, which makes it a good option for daily use.

MATERIALS AND METHODS

Dataset Description

The dataset used in this study was obtained from Kazhydromet's open hydrological data source. The data are provided in separate CSV files, with each file covering a different hydrological basin in Kazakhstan: Balkhash-Alakol, Irtysh (Ertis), Esil, Shu-Talas, Syrdarya, Tobol-Torgay, and Ural. Each file contains the following columns:

- Station code – ID of the hydrological station (removed during preprocessing);
- Date – measurement date;
- Value – recorded daily water level;
- Conditional mark – annotation symbol for special cases (removed during preprocessing).

The dataset sample with example values provided in Table 1.

Table 1. Dataset Sample

Station code	Date	Value	Conditional mark
14002	2001-01-12	108	I
14002	2001-01-13	145	I

To prepare the data for modeling, we combined all the basin files into one main dataset. To prepare dataset Code 1 was implemented. During this combination, we dropped the "Station code" and "Conditional mark" columns. We then renamed the "Value" column in each file to show which basin it came from (for example, Value Balkhash-Alakol, Value Irtysh). After that, we merged all the datasets using the Date field to create a unified time series.

The final dataset includes data from 1995 to 2022 and has synchronized daily water level measurements from different basins. This structure enables comparison and prediction of reservoir water levels across Kazakhstan.

Data Preprocessing

Before training the model, several steps were taken to get the data ready for regression analysis. The Date column was changed from text to datetime format in Code 2. The year, month, and day were extracted as they reflect the time of year water levels change. To capture the cyclical nature of time, seasonal features were engineered using sine-cosine transformations for both month and day of year, ensuring that temporally adjacent periods (such as December and January) are recognized as similar by the models.

```
import pandas as pd
import glob
import os

path = "./"

all_files = glob.glob(os.path.join(path, "*.csv"))

dfs = []
for file in all_files:
    df = pd.read_csv(file)
    df = df.drop(columns=["Код поста", "Условный знак"], errors="ignore")

    reservoir_name = os.path.splitext(os.path.basename(file))[0]

    df = df.rename(columns={"Значение": f"Value {reservoir_name}"})

    dfs.append(df)

final_df = dfs[0]
for df in dfs[1:]:
    final_df = pd.merge(final_df, df, on="Дата", how="outer")

final_df = final_df.rename(columns={"Дата": "Date"})
final_df.to_csv("water.csv", index=False, encoding="utf-8-sig")
```

Code 1. Dataset consolidation snippet

```
if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
    df['year'] = df['Date'].dt.year
    df['month'] = df['Date'].dt.month
    df['day'] = df['Date'].dt.day

    df['month_sin'] = np.sin(2 * np.pi * df['month'] / 12)
    df['month_cos'] = np.cos(2 * np.pi * df['month'] / 12)
    df['day_of_year'] = df['Date'].dt.dayofyear
    df['day_sin'] = np.sin(2 * np.pi * df['day_of_year'] / 365)
    df['day_cos'] = np.cos(2 * np.pi * df['day_of_year'] / 365)
    df['quarter'] = df['Date'].dt.quarter
```

Code 2. Date transformation

Next, all numerical columns were converted to numeric format. At this time, it was clear that the data had missing values across basins because of incomplete records. Figure 1 shows distribution of missing values throw columns of the table. Because most machine learning models cannot work with missing values, it was important to fill these gaps.

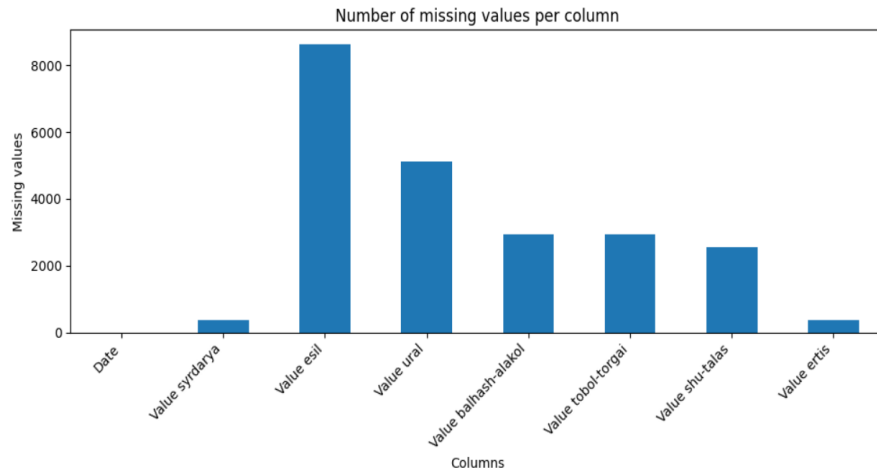


Figure 1. Number of missing values per column

To reduce the influence of outliers, all missing values were filled with the column median instead of the mean using Code 3. This approach helped preserve the overall data distribution and reduced sensitivity to extreme water-level values.

```
num_cols = [c for c in df.columns if c not in ['Date', TARGET_COL] and pd.api.types.is_numeric_dtype(df[c])]

for c in num_cols + [TARGET_COL]:
    if c in df.columns:
        df[c] = df[c].fillna(df[c].median())

X = df[num_cols]
y = df[TARGET_COL]
```

Code 3. Missed values filling

Additionally, as shown in Code 4, the quarter feature was extracted to represent broader seasonal patterns. Lag features (7, 14, and 30 days) and rolling averages (7 and 30 days) were created to incorporate temporal dependencies and trends into the prediction models.

```
for lag in [7, 14, 30]:
    df[f'{TARGET_COL}_lag_{lag}'] = df[TARGET_COL].shift(lag)

df[f'{TARGET_COL}_rolling_7'] = df[TARGET_COL].rolling(window=7).mean()
df[f'{TARGET_COL}_rolling_30'] = df[TARGET_COL].rolling(window=30).mean()

num_cols = [c for c in df.columns if c not in ['Date', TARGET_COL] and pd.api.types.is_numeric_dtype(df[c])]

for c in num_cols:
    if c in df.columns:
        df[c] = df[c].fillna(df[c].median())
```

Code 4. Lag features and rolling averages

Because machine learning models can be sensitive to how variables are scaled, all input variables were normalized with scikit-learn's StandardScaler, which changes each feature to have a zero mean and unit variance.

Methodology

1. Baseline models

To evaluate predictive performance, three commonly used regression models were applied in Code 5 snippet:

1. The DT Regressor was used as a simple and interpretable baseline. It divides the feature space into smaller regions using a sequence of threshold-based rules. Although decision trees are easy to understand, they can overfit the training data and may perform less reliably on more complex patterns.

2. The RF Regressor is an ensemble method based on multiple decision trees trained on different subsets of the data. By combining multiple trees and introducing randomness during training, the model usually produces more stable and accurate predictions than a single decision tree.

3. The KNN Regressor predicts the target value based on the values of the most similar observations in the dataset. Although this method is computationally straightforward, its performance depends strongly on feature scaling and representation.

Hyperparameter tuning was performed to improve predictive performance. The DT model employed a fixed random state to ensure reproducibility of results. The KNN model used five nearest neighbors to balance stability and sensitivity to local patterns. For RF, the number of trees, maximum tree depth, minimum split size, minimum leaf size, and feature selection strategy were optimized.

```
algs = {  
    'Decision Tree': DecisionTreeRegressor(random_state=RANDOM_STATE),  
    'Random Forest': RandomForestRegressor(  
        n_estimators=300,  
        max_depth=15,  
        min_samples_split=5,  
        min_samples_leaf=2,  
        max_features='sqrt',  
        random_state=42  
    ),  
    'KNN': KNeighborsRegressor(n_neighbors=5)  
}  
results = []  
  
for i, (train_frac, test_frac) in enumerate(splits, start=1):  
    X_train, X_test, Y_train, Y_test = train_test_split(X_scaled, y, train_size=train_frac, random_state=RANDOM_STATE)  
    for name, model in algs.items():  
        model.fit(X_train, Y_train)  
        Y_pred = model.predict(X_test)  
        rmse = np.sqrt(mean_squared_error(Y_test, Y_pred))  
        r2 = r2_score(Y_test, Y_pred)  
        results.append({  
            'Algorithm': name,  
            'Iteration': i,  
            'Num_features': X.shape[1],  
            'Train_size_%': int(train_frac*100),  
            'Test_size_%': int(test_frac*100),  
            'RMSE': rmse,  
            'R2': r2  
        })
```

Code 5. Model training and evaluation snippet

Hyperparameter Selection

The model hyperparameters were selected through preliminary experimental tuning. Different model configurations were tested, and the final settings were chosen based on RMSE, while R^2 was used as an additional evaluation metric. A fixed random state of 42 was used to make the results reproducible.

For the Decision Tree Regressor, the default structure was used with a fixed random state. For the Random Forest Regressor, the tree depth, minimum split size, minimum leaf size, and feature selection strategy were tuned. For the KNN Regressor, the number of neighbors was set to 5. In the hybrid RF+MLP model, the Random Forest component used the same selected RF settings, while the MLP component used two hidden layers with 64 and 32 neurons, ReLU activation, the Adam optimizer, and a maximum of 1000 iterations.

The selected hyperparameters are summarized in Table 2.



Table 2. Selected hyperparameters of the models

Model	Hyperparameter	Selected value
DT	random_state	42
	max_depth	15
	min_samples_split	5
RF	min_samples_leaf	2
	max_features	sqrt
	random_state	42
KNN	n_neighbors	5
	max_depth	15
	min_samples_split	5
Hybrid RF component	min_samples_leaf	2
	max_features	sqrt
	hidden_layer_sizes	(64, 32)
Hybrid MLP component	activation	relu
	solver	adam
	max_iter	1000
	random_state	42

2. Hybrid model

To improve the baseline results, a two-stage hybrid model combining RF and MLP was developed. This model was designed to combine the similarity-based prediction capability of RF with the error-correction ability of a neural network.

1) In the first stage, the RF regressor was trained on the preprocessed and standardized features and generated an initial prediction of daily water levels. Next, the residual error between the true target value and the RF prediction was computed and used to train the neural network.

2) In the second stage, the neural network learned a correction term for the initial RF estimate. The final hybrid prediction was obtained by adding the neural-network-based correction to the RF prediction.

Hyperparameter tuning of the hybrid model was performed for both the RF Regressor and the Multilayer Perceptron (MLP). The MLP included two hidden layers with 64 and 32 neurons, used the ReLU activation function, an adaptive learning rate, and a maximum of 1000 iterations. Thus, the proposed hybrid model can be viewed as a two-stage regression framework in which RF provides the initial estimate and the MLP refines it through residual correction.

Figure 2 illustrates the architecture of the proposed hybrid model. The training and inference procedure of the proposed model is summarized in Algorithm 1, while Code 6 presents its implementation:

Algorithm 1. Hybrid RF and MLP model

Input:

X_{train}, y_{train}

X_{test}

Output:

y_{hybrid}

1. Train RF regressor on (X_{train}, y_{train})
2. $y_{rf_train} \leftarrow RF.predict(X_{train})$
3. $residuals \leftarrow y_{train} - y_{rf_train}$
4. Train MLP regressor on $(X_{train}, residuals)$
5. $y_{rf_test} \leftarrow RF.predict(X_{test})$
6. $\Delta y \leftarrow MLP.predict(X_{test})$
7. $y_{hybrid} \leftarrow y_{rf_test} + \Delta y$
8. Return y_{hybrid}

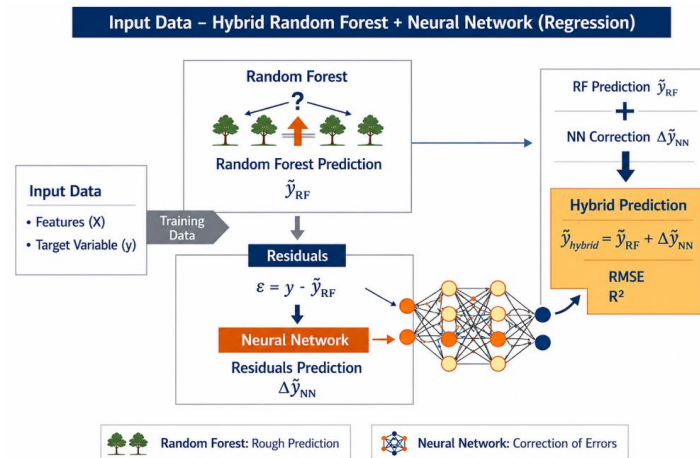


Figure 2. Architecture of the proposed hybrid model

```
from sklearn.neural_network import MLPRegressor

for i, (train_frac, test_frac) in enumerate(splits, start=1):
    X_train, X_test, Y_train, Y_test = train_test_split(
        X_scaled, y, train_size=train_frac, random_state=RANDOM_STATE
    )

    RandomForest = RandomForestRegressor(
        n_estimators=300,
        max_depth=15,
        min_samples_split=5,
        min_samples_leaf=2,
        max_features='sqrt',
        random_state=42
    )

    RandomForest.fit(X_train, Y_train)

    residuals = Y_train - RandomForest.predict(X_train)

    nn = MLPRegressor(
        hidden_layer_sizes=(64, 32),
        activation='relu',
        solver='adam',
        max_iter=1000,
        random_state=RANDOM_STATE
    )
    nn.fit(X_train, residuals)

    Y_pred_hybrid = RandomForest.predict(X_test) + nn.predict(X_test)

    rmse = np.sqrt(mean_squared_error(Y_test, Y_pred_hybrid))
    r2 = r2_score(Y_test, Y_pred_hybrid)

    results.append({
        'Algorithm': 'Hybrid RandomForest+MLP',
        'Iteration': i,
        'Num_features': X.shape[1],
        'Train_size_%': int(train_frac * 100),
        'Test_size_%': int(test_frac * 100),
        'RMSE': rmse,
        'R2': r2
    })
}}
```

Code 6. Hybrid model training and evaluation snippet

To validate the models under a realistic forecasting setting, TimeSeriesSplit was additionally applied. This method preserves the temporal order of observations: each model is trained on earlier data and tested on later data.

3. Evaluation Metrics

Each model was tested by splitting the data into training and testing sets in different ways: 60/40, 70/30, 75/25, 80/20, and 85/15. TimeSeriesSplit validation was also applied to assess the models under a chronological forecasting scenario. Each time, the models were trained and tested using four standard metrics:

1. Root Mean Squared Error (RMSE): It measures the average magnitude of prediction errors.
2. Coefficient of Determination (R^2): This indicates how much of the target's variance is explained by the model.
3. Mean Absolute Error (MAE): It measures the average absolute prediction error.
4. Pearson correlation coefficient: It measures the agreement between observed and predicted water levels.

RESULTS AND DISCUSSION

After training and testing the models, the RMSE and R^2 results for different train-test splits are presented in Table 3.

Table 3. Model performance results

Algorithm	Number of iterations	Number of features	Number of targets	Train Size, %	Test size, %	RMSE/ R^2 (for regression tasks)
DT	1	21	9	60	40	17.889357 / 0.941465
	2	21	9	70	30	17.952092 / 0.939741
	3	21	9	75	25	18.643088 / 0.935279
	4	21	9	80	20	18.619866 / 0.936053
	5	21	9	85	15	17.683095 / 0.940499
RF	1	21	9	60	40	13.230117 / 0.96798
	2	21	9	70	30	12.568647 / 0.970463
	3	21	9	75	25	12.427935 / 0.971239
	4	21	9	80	20	12.032438 / 0.973296
	5	21	9	85	15	11.812112 / 0.973450
KNN	1	21	9	60	40	15.182268 / 0.957840
	2	21	9	70	30	13.760381 / 0.964596
	3	21	9	75	25	13.483024 / 0.966148
	4	21	9	80	20	12.539609 / 0.970998
	5	21	9	85	15	12.423364 / 0.970631
Hybrid model (RF+MLP)	1	21	9	60	40	11.515105 / 0.975747
	2	21	9	70	30	10.843687 / 0.978014
	3	21	9	75	25	10.795978 / 0.978296
	4	21	9	80	20	10.305398 / 0.980412
	5	21	9	85	15	10.582080 / 0.978692

The DT Regressor showed the weakest performance. Its RMSE decreased from 17.89 (60/40 split) to 17.68 (85/15 split), and its R^2 stayed high, between 0.935 and 0.94. These results indicate that DT performed less accurately than ensemble-based methods.

The RF Regressor showed steady, average performance. Its RMSE decreased 13.23 to 11.81, and its R^2 went up from 0.968 to 0.973 as the training size got bigger. This result demonstrates the effectiveness of ensemble learning in reducing variance and improving prediction accuracy.

The KNN Regressor also demonstrated strong performance and improved as the training-set size increased. However, RF consistently achieved lower RMSE and higher R^2 than KNN across all evaluated splits, indicating better baseline predictive performance.

The hybrid model, based on RF Regressor and MLP, showed good results. Its RMSE decreased from 11.52 to 10.3, and the R^2 increased from 0.975 to 0.98.

In summary, RF was the strongest baseline model in predicting daily water levels in Kazakhstan's reservoirs. It consistently achieved lower RMSE and higher R^2 than DT and KNN across all evaluated train-test splits. KNN also improved as the training-set size increased, but it did not outperform RF. DT showed the weakest predictive performance among the tested models and therefore served primarily as a simple interpretable baseline for comparison. The accuracy of all models improved with a larger training set, demonstrating that having enough data is crucial in making accurate predictions of water levels.

To statistically assess the reliability of the results, the mean and standard deviation of RMSE, MAE, R^2 , and Pearson correlation were calculated across the TimeSeriesSplit folds. Table 4 summarizes these results.

Table 4. Summary of TimeSeriesSplit validation results, mean $\pm$ std

Algorithm	RMSE, mean $\pm$ std	MAE, mean $\pm$ std	R^2 , mean $\pm$ std	Pearson, mean $\pm$ std
DT	44.493 $\pm$ 20.684	32.423 $\pm$ 15.449	0.423 $\pm$ 0.454	0.828 $\pm$ 0.057
RF	43.198 $\pm$ 25.588	32.397 $\pm$ 18.826	0.433 $\pm$ 0.530	0.889 $\pm$ 0.036
KNN	55.617 $\pm$ 21.742	42.494 $\pm$ 17.646	0.126 $\pm$ 0.482	0.711 $\pm$ 0.059
Hybrid model (RF+MLP)	42.921 $\pm$ 23.646	32.429 $\pm$ 17.014	0.451 $\pm$ 0.497	0.760 $\pm$ 0.279

The TimeSeriesSplit results show that chronological validation is more difficult than random train-test splitting. The lower R^2 values and higher standard deviations indicate that model performance changes across different time periods. This reflects the non-stationary behavior of water-level data. Therefore, the reported model performance should be interpreted not only by the best RMSE and R^2 values, but also by the stability of the results across time folds.

Additional advanced models were evaluated using the same TimeSeriesSplit validation strategy. The comparison with Hybrid RF+LSTM+GRU and Hybrid RF+XGB+LSTM+GRU is presented in Table 5.

Table 5. Comparison with advanced models using TimeSeriesSplit validation

Algorithm	RMSE, mean $\pm$ std	MAE, mean $\pm$ std	R^2 , mean $\pm$ std	Pearson, mean $\pm$ std
Hybrid RF+MLP	42.921 $\pm$ 23.646	32.429 $\pm$ 17.014	0.451 $\pm$ 0.497	0.760 $\pm$ 0.279
Hybrid RF+LSTM+GRU	42.554 $\pm$ 21.467	33.106 $\pm$ 16.376	0.473 $\pm$ 0.412	0.830 $\pm$ 0.099
Hybrid RF+XGB+LSTM+GRU	42.554 $\pm$ 21.467	33.106 $\pm$ 16.376	0.473 $\pm$ 0.412	0.830 $\pm$ 0.099

Table 5 shows that the advanced models gave results close to the proposed Hybrid RF+MLP model. Although the Hybrid RF+XGB+LSTM+GRU model was slightly better in RMSE and R^2 , the improvement was very small. The Hybrid RF+MLP model was slightly better in MAE. For this reason, the more complex models cannot be considered clearly better in this study. One

possible reason is that the dataset includes only limited input variables, while LSTM and GRU models usually need more detailed hydrometeorological data to achieve stronger performance.

Since RF showed increasingly competitive performance as the amount of training data grew, it was selected as the first-stage component of the hybrid architecture. As shown in Table 3, the hybrid model, which combines RF Regressor and MLP, outperformed the standalone RF Regressor in all tests. These results show that the hybrid model improved prediction accuracy over the standalone RF Regressor, indicating that the second-stage neural component provided a consistent refinement of the initial RF predictions (Figure 3).

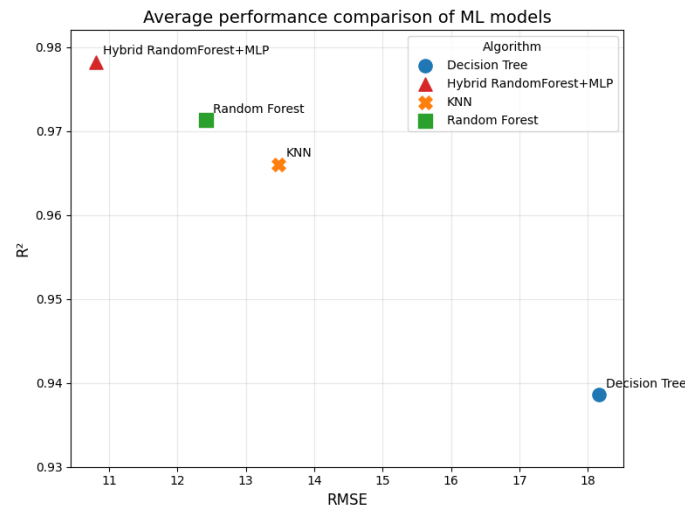


Figure 3. Average performance comparison of baseline and proposed models by RMSE and R^2

Figure 4 shows how different train/test splits affect RMSE. All models showed better performance with bigger training sets, and hybrid model showed the steadiest gains.

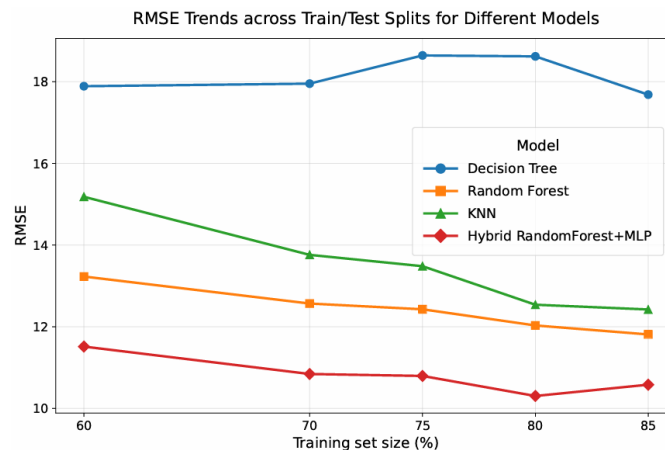


Figure 4. RMSE trends across train/test splits for different models

Overall, the results show that RF was the strongest baseline model, consistently outperforming DT and KNN across the evaluated train-test splits. The hybrid model delivered the best overall predictive accuracy, outperforming all baseline models in both RMSE and R^2 . The additional TimeSeriesSplit validation showed that RF-based models remained relatively stable under a stricter chronological evaluation.

LIMITATIONS OF THE STUDY

This study has several limitations. First, the analysis is based on historical hydrological observations and does not explicitly incorporate external meteorological variables such as precipitation, temperature, snowmelt, or reservoir discharge operations. Second, missing values were imputed using median values, which may reduce the representation of extreme hydrological events. Third, TimeSeriesSplit was added to test the models in a more realistic time-based setting. However, validation on independent future data would further strengthen the evaluation. The differences across folds indicate that water levels change over time and are influenced by factors not included in the dataset.

CONCLUSION

Among the baseline and proposed models, the hybrid RF with MLP model gave the best overall performance, with the lowest RMSE and the highest R^2 values across the different train-test splits. This shows that combining RF with an MLP can improve prediction accuracy for this task.

Among the baseline models, RF achieved the strongest overall performance, with consistently lower RMSE and higher R^2 across all train-test splits. KNN also improved as the training-set size increased, but it did not outperform RF. DT showed the lowest predictive accuracy and served mainly as an interpretable baseline.

TimeSeriesSplit showed that forecasting future periods is more challenging. Although the results became lower, RF-based models remained more stable than the others. Additional comparison with LSTM, GRU, and XGBoost-based models showed similar results, but these more complex models did not provide a clear advantage over the proposed hybrid model.

The results also showed that all models performed better when more training data were available. This highlights the importance of using sufficient historical data for reliable forecasting. In addition, the improved performance of RF after adding temporal and seasonal features shows that feature design is an important part of hydrological prediction.

Overall, the study shows that machine learning methods, especially the proposed hybrid model, can improve daily water level forecasting and may be useful for better water resource management in Kazakhstan.

CONFLICT OF INTEREST: The authors declare no conflict of interest.

FUNDING: This research did not receive any external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT: Not applicable.

INFORMED CONSENT STATEMENT: Not applicable.

DATA AVAILABILITY STATEMENT: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENTS: The authors acknowledge the anonymous reviewers for their helpful comments that improved the article.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES: The authors declare that no artificial intelligence technologies were used in the preparation of this manuscript.

REFERENCES

- Abozweita, O. A., Ahmed, A. N., Sidek, L. B. M., Basri, H. B., Zawawi, M. H. B., Huang, Y. F., & El-Shafie, A. (2024). Enhancing hydrological predictions: optimized decision tree modelling for improved monthly inflow forecasting. *Journal of Hydroinformatics*, 26 (11), 2683–2703. <https://doi.org/10.2166/hydro.2024.205>
- Ahmed, A. N., Yafouz, A., Birima, A. H., Kisi, O., Huang, Y. F., Sherif, M., Sefelnasr, A., & El-Shafie, A. (2022). Water level prediction using various machine learning algorithms: a case

- study of Durian Tunggal River, Malaysia. *Engineering Applications of Computational Fluid Mechanics*, 16(1), 422–440. <https://doi.org/10.1080/19942060.2021.2019128>
- Jayathilake, T., Sarukkalige, R., Hoshino, Y., & Rathnayake, U. (2022). Wetland water level prediction using Artificial Neural Networks—A case study in the Colombo Flood Detention Area, Sri Lanka. *Climate*, 11(1), 1. <https://doi.org/10.3390/cli11010001>
- Jibril, N. M., Bello, N. A., Aminu, N. I. I., Ibrahim, N. A. S., Bashir, N. A., Malami, N. S. I., Habibu, N., MA, & Magaji, N. M. (2022). An overview of streamflow prediction using random forest algorithm. *GSC Advanced Research and Reviews*, 13(1), 050–057. <https://doi.org/10.30574/gscarr.2022.13.1.0112>
- Kalashnikova, O., Niyazov, J., & Nurbatsina, A. (2024). Hydrological modeling in studies of mountain river basins of Central Asia. *Georgian Geographical Journal*, 4(2), 38–47. <https://doi.org/10.52340/ggj.2024.04.02.05>
- Mohammadi, B. (2021). A review on the applications of machine learning for runoff modeling. *Sustainable Water Resources Management*, 7, 98. <https://doi.org/10.1007/s40899-021-00584-y>
- Pham, L. T., Luo, L., & Finley, A. (2021). Evaluation of random forests for short-term daily streamflow forecasting in rainfall- and snowmelt-driven watersheds. *Hydrology and Earth System Sciences*, 25(6), 2997–3015. <https://doi.org/10.5194/hess-25-2997-2021>
- Tyralis, H., Papacharalampous, G., & Langousis, A. (2019). A brief review of random forests for water scientists and practitioners and their recent history in water resources. *Water*, 11(5), 910. <https://doi.org/10.3390/w11050910>
- Wang, X., Liu, T., Zheng, X., Peng, H., Xin, J., & Zhang, B. (2018). Short-term prediction of groundwater level using improved random forest regression with a combination of random features. *Applied Water Science*, 8, 125. <https://doi.org/10.1007/s13201-018-0742-6>
- Zhou, F., Chen, Y., & Liu, J. (2023). Application of a new hybrid deep learning model that considers temporal and feature dependencies in Rainfall–Runoff simulation. *Remote Sensing*, 15(5), 1395. <https://doi.org/10.3390/rs15051395>
- Zhu, M., Wang, J., Yang, X., Zhang, Y., Zhang, L., Ren, H., Wu, B., & Ye, L. (2022). A review of the application of machine learning in water quality evaluation. *Eco-Environment & Health*, 1(2), 107–116. <https://doi.org/10.1016/j.eehl.2022.06.001>
- Kaggle. (2025). Kazakhstan Basin Water Level Kazhydromet Dataset. [Online]. Available: <https://www.kaggle.com/datasets/sabinayesniyazova/kazakhstan-basin-water-level-kazhydromet> [Accessed on: Oct. 20, 2025]
- GitHub repository. (2025). Jupyter Notebook file with source code used in article [Online]. Available: <https://github.com/perpavbek/ml-kazakhstan-water-level.git> [Accessed on: Nov. 12, 2025]

Авторлар туралы мәліметтер
Информация об авторах
Information about authors



Оралбекова Жанар Орымбаевна – PhD, Astana IT University, Астана қ., Қазақстан

Оралбекова Жанар Орымбаевна – PhD, Astana IT University, г.Астана, Казахстан

Oralbekova Zhanar – PhD, Astana IT University, Astana, Kazakhstan, e-mail: zh.oralbekova@astanait.edu.kz, ORCID: <https://orcid.org/0000-0002-3726-8882>



Перцев Павел Александрович – 3 курс студенті, Astana IT University, Астана, Қазақстан

Перцев Павел Александрович – студент 3 курса, Astana IT University, г. Астана, Казахстан

Pertsev Pavel – third-year student, Astana IT University, Astana, Kazakhstan,

e-mail: 230308@astanait.edu.kz,

ORCID: <https://orcid.org/0009-0003-1219-8093>



Хасенова Зарина Төлеубековна – PhD, Д. Серікбаев атындағы Шығыс Қазақстан техникалық университеті, Өскемен қ., Қазақстан

Хасенова Зарина Төлеубековна – PhD, Восточно-Казахстанский технический университет им. Д. Серикбаева, г. Усть-Каменогорск, Казахстан

Khassenova Zarina – PhD, D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan,

e-mail: zkhasenova@edu.ektu.kz,

ORCID: <https://orcid.org/0009-0001-1417-7793>



Оралбекова Дина Орымбаевна – PhD, Ақпараттық және есептеу технологиялар институты, Алматы, Қазақстан

Оралбекова Дина Орымбаевна – PhD, Институт информационных и вычислительных технологий, г. Аматы, Казахстан

Oralbekova Dina – PhD, Institute of information and computational technologies, Almaty, Kazakhstan,

e-mail: d.oralbekova@ipic.kz,

ORCID: <https://orcid.org/0000-0003-4975-6493>



Айтмұханбетова Эльвира Айтмұханбетқызы – компьютерлік ғылымдар магистрі, Astana IT University, Астана, Қазақстан

Айтмұханбетова Эльвира Айтмұханбетқызы – магистр компьютерных наук, Astana IT University, г. Астана, Казахстан

Aitmukhanbetova Elvira – Master of Science in Computer Science, Astana IT University, Astana, Kazakhstan,

e-mail: elvira.aitmukhanbetova@astanait.edu.kz,

ORCID: <https://orcid.org/0000-0001-7835-873X>