# Streamflow Prediction of the Chaliyar River Basin Using Machine Learning Techniques

**Rahul P[1], Sarah George[2], Paul C M[3], and Sathish Kumar D[4]**

[1] Undergraduate Student, National Institute of Technology Calicut, Kozhikode, 673601, India
[2] Undergraduate Student, National Institute of Technology Calicut, Kozhikode, 673601, India
[3] Undergraduate Student, National Institute of Technology Calicut, Kozhikode, 673601, India
[4] Associate Professor, National Institute of Technology Calicut, Kozhikode, 673601, India

*Corresponding author e-mail: sathish@nitc.ac.in

**Abstract.** Floods are one of the most destructive natural disasters, causing massive damage to life and property. The Chaliyar River, located in the Malappuram district of Kerala, India, has been prone to flooding, causing concern among the people living in the region. Developing a reliable plan for sustainable flood risk management is necessary to minimize the damages associated with floods. In this study, the stage of the river at a gauging station located in the Chaliyar River is predicted using Machine Learning (ML) techniques. A comparison of the ability of different ML techniques like Support Vector Machine (SVM), Decision Tree, Random Forest, and Artificial Neural Network (ANN) for streamflow simulation is carried out. The correlation between meteorological attributes like rainfall, temperature, current and cumulative rainfall, discharge, and the river stage is analyzed. Correlation analysis is carried out between the hydrometeorological parameters and the measured stage at the Kuniyil gauging station. Various ML models were developed for the study area using multiple linear regression, ANN, SVM, Decision Tree and Random Forest techniques. The results indicate that the Random Forest, Decision Tree, and ANN models perform better and have Root Mean Squared Error (RMSE) values less than 0.5m for both training and testing datasets. The performance of the linear regression and SVM models was relatively poor, where the computed RMSE values were more than 0.5m for training and testing datasets. The developed ML models can be applied to the Chaliyar River basin to predict the peak stage and corresponding discharge during extreme rainfall events.

## 1. Introduction

The disastrous effects of flood catastrophes brought on by climate change include the loss of life, the destruction of infrastructure, damage to agricultural areas, and disruptions to the socioeconomic system. Flood disasters offer substantial difficulties to communities around the world. As a result, there is tremendous pressure on governments to address these issues by creating reliable and accurate flood forecasting models for efficient catastrophe risk management. The changing nature of climate conditions makes it difficult to make reliable flood predictions. Traditional physical models have been effective at foreseeing different flooding scenarios. They are less suited for producing short-term predictions since they frequently rely on large datasets from hydro-geomorphological monitoring and demand computationally costly operations. Additionally, it can be very difficult to design and execute these physically based models without extensive knowledge and skill in hydrological factors. Due to these drawbacks, data-driven models utilizing Machine Learning (ML) methods have become a viable and dependable substitute. ML models are excellent at learning from their surroundings on their own, which enables them to handle nonlinear data and produce extremely precise predictions and findings. The accuracy of flood forecasting is increased by using previous meteorological and hydrological data that have been gathered over time as input and output variables. This study intends to improve flood predicting capabilities by using ML algorithms and time series data analysis. It is now possible to create trustworthy flood prediction models that can support efforts to reduce the risk of disaster by utilizing the power of ML. These models have the potential to offer insightful information about flood vulnerability and give governments the tools they need to develop and implement long-term flood risk reduction plans. These techniques emphasize prevention, protection, and preparedness in order to lessen the harm caused by future flood occurrences while enhancing the security and resilience of nearby communities.Overall, by utilizing the advantages of ML and time series data, this study seeks to advance the field of flood forecasting and improve the accuracy and effectiveness of disaster risk management in vulnerable regions such as the Chaliyar River in Kerala, where previous floods have caused significant devastation.

## 2. Related work

Linear regression is widely applied in water resource engineering, particularly in river flow prediction. Supriya et al. (2015) utilized linear regression for flood forecasting in the Vellar river basin, while Ahmad et al. (2022) used multiple linear regression to predict discharge in the Var river basin, achieving an R-square value of 0.39. Decision tree algorithms, a type of supervised learning, are versatile for both categorical and continuous output variables. Their application in flood modelling is extensive due to the involvement of multiple variables, and they have demonstrated high accuracy. Khosravi et al. (2018) and Ahmad et al. (2022) employed decision tree methods for flood forecasting. Support Vector Machines (SVM) have also been used for flood prediction models. Samantaray et al. (2022) developed an SVM-based model for runoff prediction in the Baitirani river, yielding an R-square value close to 0.9. Artificial Neural Networks (ANN) have been successfully employed by Samantaray et al. (2022) for flood modelling in the Baitarani river and by Ahmad et al. (2022) for the Var river, showcasing high accuracy using multiple input parameters. These different techniques and models contribute to the advancement of flood prediction and management in various river basins.

## 3. Methodology and study area

The study area chosen is the Chaliyar river basin in Kerala, India, situated between 11° 30'N and 11° 10'N latitudes and 75° 50'E and 76°30'E longitudes falling in Survey of India (SOI) degree sheets 58A and 49M. Chaliyar River forms the third largest river in Kerala and originates from the Elambalari hills, Nilgiri District of Tamil Nadu, at an elevation of about 2066 m above mean sea level (MSL). This river has a total drainage area of 2918 km$^{2,}$ out of which 2530 km$^2$ lies in Kerala State, and the remaining area falls in Tamil Nadu. The temperature in the region ranges between 23° and 37°C.(Thakural et al.2018). . For this river segment, meteorological data include daily time series of rainfall, humidity, temperature, and surface pressure. Meteorological and Hydrological data of Kuniyil station is collected from various websites and stations. Kuniyil river station is selected based on the availability of data and the effect of floods in the nearby area. These data, after processing, will be used in diverse ML techniques. Meteorological data, including daily time series of rainfall, humidity, temperature, and surface pressure, was collected from Power Data Access Viewer by NASA of Kuniyil station. Hydrological data daily time series of discharge and the water level was collected from the Indian Water Resources Information System Website, where the data are measured at the station of Kuniyil. The meteorological and hydrological data were pre-processed to fit into the model for the best prediction. The data are trimmed, and the data from 1982 to 2020 are used for further steps. The gap and irregularities data were analyzed, and subsequent actions were taken. Data missing in one column was filled by taking the average of the above and below cells. Some dates were cleared from the dataset due to a high margin of error, and the dataset was cleaned. The data obtained had to be crunched into an appropriate tabular format in Microsoft Excel. By plotting an initial time-series graph, the trends and periodicity of the different parameters were studied. Based on the trend so obtained, the missing data points were filled, and the outliers present were removed. A summary of statistics of different variables was found and followed by univariate, bivariate, and multivariate analysis.

Figure 1,2,3 conveys that meteorological data like rainfall follow a periodic trend, and rainfall becomes peaks during the monsoon season. It is also concluded that the rainfall, discharge, and level follow the same trend, which is further solidified using the correlation values. From the level graph, it is visible that the level also follows a similar trend, but the minimum value of the level has come down over the years and gradually increased over the years.

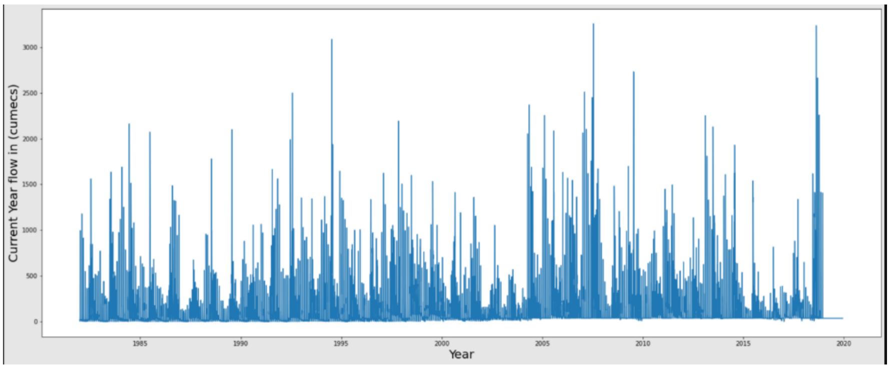


Figure 1. Time-series Plot of Discharge period of 30 years.

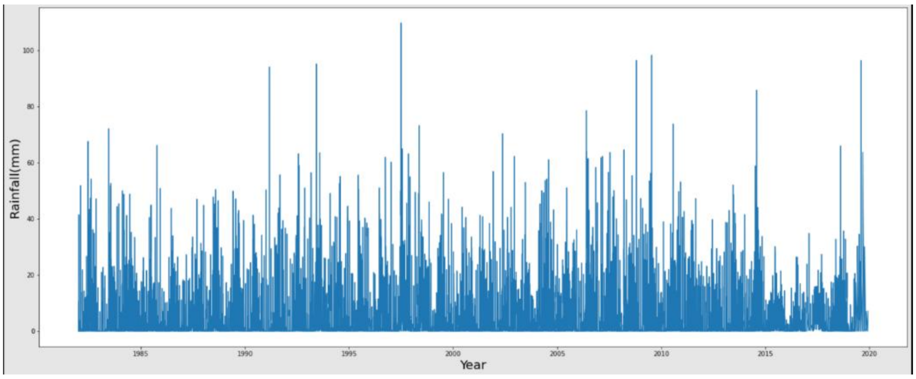


Figure 2. Time series Plot of Daily Rainfall period of 30 years

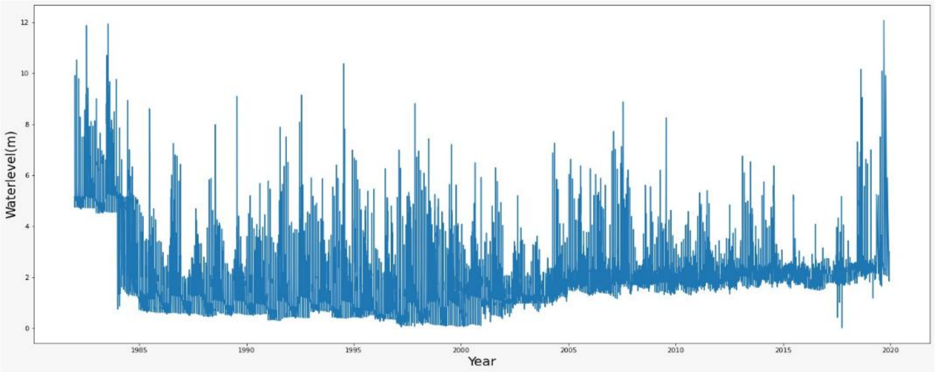


Figure 3. Time series Plot of level over a period of 30 years

According to Table 1, the correlation of discharge and level with all the meteorological parameters has been found. Rainfall has a large positive correlation with the current year's flow, which means the discharge increases with an increase in rainfall. At the same time, the temperature has a negative correlation with both level and flow, which means that temperature has inverse proportionality with discharge and level. The last ten years' flow also has a significant correlation with the discharge and level values. Humidity also has a non-significant positive correlation similar to the negative correlation of surface pressure.

Table 1. Correlation of different parameters with flow stage value.

| Parameters | Correlation value | |
|---|---|---|
| | Current Year flow ( cumec) | Level (m) |
| Specific humidity at 2m(g/kg) | 0.33 | 0.25 |
| Relative humidity (%) | 0.45 | 0.34 |
| Rainfall(mm) | 0.53 | 0.38 |
| The temperature at 2m height | -0.31 | -0.23 |
| Earth skin temperature | -0.30 | -0.23 |
| The temperature at 2m height minimum | -0.49 | -0.37 |
| The temperature at 2m height maximum | -0.47 | -0.34 |
| Surface pressure(kPa) | -0.26 | -0.16 |
| Last ten years' average flow (cumec) | 0.65 | 0.47 |
| Last year's flow (cumec) | 0.38 | 0.29 |
| Current year flow previous day (cumec) | 0.89 | 0.56 |
| Current year flow in (cumec) | 1 | 0.62 |
| Level (m) | 0.62 | 1 |

Based on the observation and metadata in Table 2, the mean discharge from 1982–2020 is 128.3 $m^3/s$, and peak discharge was recorded as 3256 $m^3/s$ on 17 July 2007. The average precipitation is 4.987 mm, and maximum precipitation was observed at 109.89 mm; the minimum precipitation is 0 showing that there was no rainfall on that day. The mean temperature is 23.4 ℃ which is close to the average temperature of the autumn season. The minimum temperature was which is close to the average temperature of the autumn season. The minimum temperature was recorded at 11 ℃ on 13 December 1994, and the maximum temperature was 38.78℃ on 24 April 2016.

Table 2. Meteorological Data of Chaliyar Basin

| Parameters | Mean | Standard deviation | Min | Max |
|---|---|---|---|---|
| Specific humidity at 2m(g/kg) | 14.552 | 2.595 | 4.330 | 19.040 |
| Relative humidity (%) | 78.085 | 14.378 | 27.810 | 97.120 |
| Rainfall(mm) | 4.987 | 9.053 | 0.000 | 109.890 |
| The temperature at 2m (℃) | 23.427 | 2.120 | 16.890 | 30.000 |
| Earth skin temperature (℃) | 23.710 | 2.449 | 16.570 | 31.000 |
| The temperature at 2 m minimum (℃) | 9.004 | 4.303 | 1.530 | 20.490 |
| The temperature at 2 m maximum (℃) | 28.594 | 3.915 | 20.980 | 38.780 |
| Temperature min (℃) | 19.5905 | 2.185 | 11.050 | 24.690 |
| Surface pressure (kPa) | 93.532 | 0.177 | 92.730 | 94.110 |
| Last 10-Year Average (cumec) | 133.679 | 169.204 | 0 | 1268.95 |
| Last Year (cumec) | 130.283 | 247.585 | 0 | 3256 |
| Current year flow previous day(cumec) | 128.319 | 250.247 | 0 | 3256 |
| Current Year (Flow in cumec) | 128.319 | 250.247 | 0 | 3256 |
| Level (m) | 2.066 | 1.423 | 0.010 | 12.08 |

## 4. Results and discussions

### *4.1 Linear Regression*

After splitting the data into training and testing multiple linear regression model was created using linear regression in the Sklearn python package. In this case, the testing set contains 10% of the data, and the random state is set to 2 for reproducibility. In this case, the 'StandardScaler' is used to standardize the data, which means that the features are scaled so that they have zero mean and unit variance. The process of standardization involves subtracting the mean from each data point and then dividing it by the standard deviation. All the parameters were taken in as input parameters, and the following results were obtained.

R square value of 0.794 and Mean Square Error (MSE) value of 0.415 $m^2$and Root Mean Square Error (RMSE) of 0.548 m for testing data and R square value of 0.761 and Mean Square Error (MSE) value of 0.384 $m^2$ and Root Mean Square Error (RMSE) of 0.548 m for training data.

The following equation was obtained for the above regression model.

$$y = 2.124 + 0.0624x_1 + 0.0946x_2 - 0.000228x_3 + 0.0353x_4 + 0.0773x_5 + 0.0371x_6 + 0.860x_7 \qquad (1)$$

Where,

$y$ = Water level (m), $x_{1=}$ Specific humidity (g/kg), $x_{2=}$ Rainfall (mm), $x_{3=}$ Temperature (at 2m), $x_{4=}$ Surface Pressure (kPa),$x_{5=}$ Last ten-year average flow (cumec), $x_{6=}$ Last year's flow (cumec), $x_{7=}$ Current year flow (cumec).

The following results are comparable to the multiple linear models created by Ahmad et al.(2022) on the Var river having an R square value of 0.794 for testing data and 0.761 for training data. Figure 4 shows the validation plot for testing data, while figure 5 shows the scatter plot between predicted and actual water level data for the training and testing dataset, respectively. The validation plot of predicted and actual water level values shows that the model is able to predict some of the peaks and predicting lower stage values for testing data. From the scatter plot with regression line, it is clear that for higher stage values are lower than the actual values and the model is not that accurate.

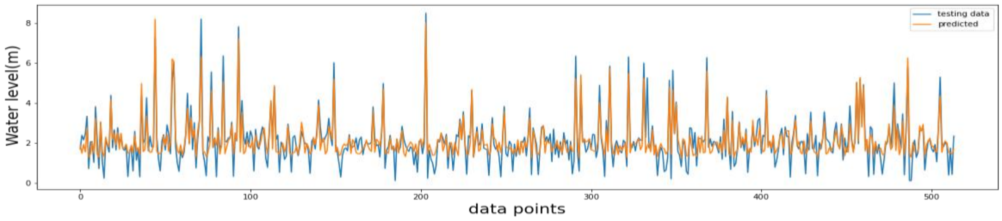


Figure 4. Validation of Linear Regression model on testing data

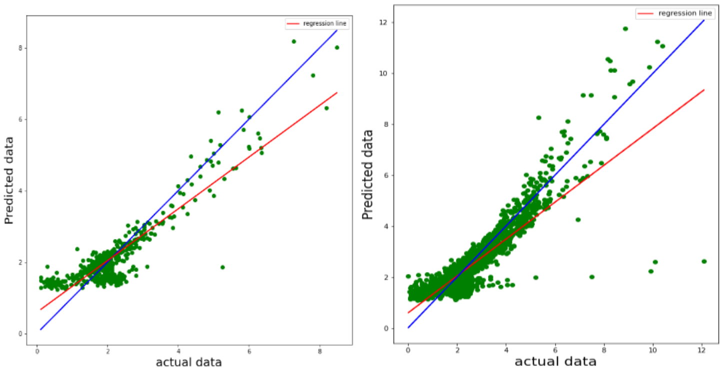


Figure 5. Scatterplot of predicted vs actual data of linear regression model using training and testing data respectively.

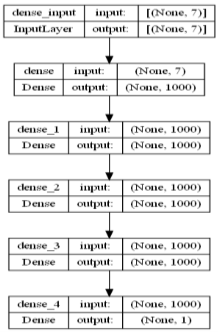
### *4.2 Artificial Neural Network*

Artificial Neural Networks (ANNs) are a type of machine learning algorithm that are designed to simulate the behaviours of the human brain. They are composed of layers of nodes, or neurons, which are interconnected through weighted connections. The neurons receive input from other neurons and perform mathematical operations on the input to produce an output, which is transmitted to other neurons in the network.

The connections between the neurons are adjusted through a process called backpropagation, which involves iteratively adjusting the weights in the network to minimize the difference between the actual output and the desired output for a given input. This process allows the network to learn from training data and improve its performance over time.

The ANN has an input layer with an input dimension equal to the number of features in the input data, followed by three hidden layers with 1000 neurons each. The activation function used for the first three hidden layers is Rectified Linear Unit (ReLU).

Rectified Linear Unit (ReLU) is an activation function that is commonly used in Artificial Neural Networks (ANNs) for deep learning. The function is defined as f(x) = max(0,x), which means that it returns the input value if it is positive, and 0 if it is negative. Figure 9 represents the ReLU function. This nonlinear activation function is widely used in deep learning models because it allows the model to introduce non-linearity and to learn complex relationships between input features. The final layer has 1000 neurons and uses a linear activation function, which is appropriate for regression tasks. Figure 6 represents the model layers with a number of nodes, inputs, and outputs.



| | | |
|---|---|---|
| dense_input | input: | [(None, 7)] |
| InputLayer | output: | [(None, 7)] |
| dense | input: | (None, 7) |
| Dense | output: | (None, 1000) |
| dense_1 | input: | (None, 1000) |
| Dense | output: | (None, 1000) |
| dense_2 | input: | (None, 1000) |
| Dense | output: | (None, 1000) |
| dense_3 | input: | (None, 1000) |
| Dense | output: | (None, 1000) |
| dense_4 | input: | (None, 1000) |
| Dense | output: | (None, 1) |

Figure 6. Visualisation of ANN model.

R square value of 0.887 and Mean Square Error (MSE) value of 0.2819 $m^2$ and Root Mean Square Error (RMSE) of 0.407 m for testing data and R square value of 0.866 and Mean Square Error (MSE) value of 0.265 $m^2$ and Root Mean Square Error (RMSE) of 0.410 m for training data. Figure 7 shows the validation plot for testing data, while figures 8 shows the scatter plot between predicted and actual water level data for training and testing dataset, respectively. The validation plot of predicted and actual water level values shows that the model is able to predict most of the peaks and also predicting accurately for testing data. From the scatter plot with regression line, it is clear that for higher stage values are lower than the actual values and the model is accurate.

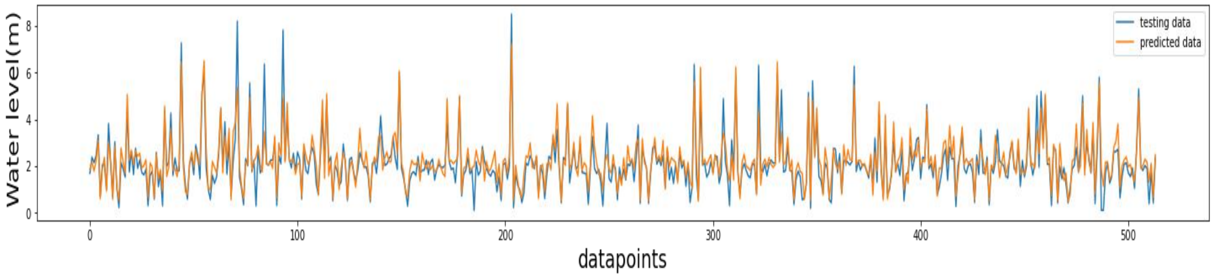


Figure 7. Validation of ANN model on testing data.

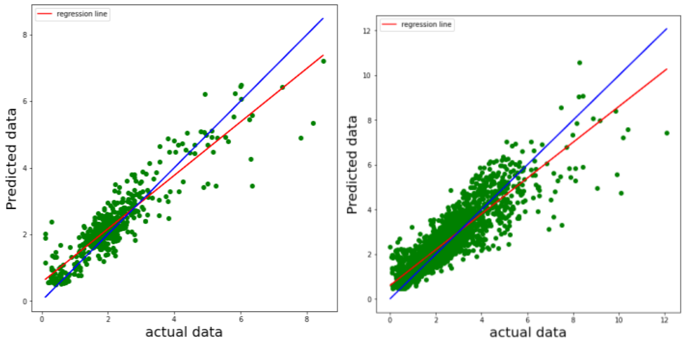


Figure 8. Scatterplot of predicted vs actual data of ANN model using testing data.

### *4.3 Support Vector Machine*

SVM regression is a supervised learning algorithm that identifies a hyperplane that best fits the training data. It can handle linear and nonlinear relationships between input and output variables and is constructed by solving a convex optimization problem. We used Support Vector Regression (SVR) from the sklearn library in Python. It first imports different types of scalers and the training-testing split function from sklearn. Then, it splits the input data X and output target y into training and testing sets using the train_test_split() function. Next, the MaxAbsScaler() function is used to standardize the features for both the training and testing sets.

MaxAbsScaler is a data normalization method that scales each feature to the range of [-1, 1] using the maximum absolute value of the feature. This scaler is commonly used when the data contains outliners or when the feature values have very different ranges.

The scaling formula for each feature x is:

$$x_{scaled} = x/\max\,(abs(x)) \qquad (2)$$

Where max(abs(x)) is the maximum absolute value of the feature.

Finally, the SVR() model is trained on the standardized training data and output target using the fit() method. The trained model can then be used to make predictions on new data.

R square value of 0.716 and Mean Square Error (MSE) value of 0.414 $m^2$ and Root Mean Square Error (RMSE) of 0.643 m, for testing data and R square value of 0.751 and Mean Square Error (MSE) value of 0.314 $m^2$ and Root Mean Square Error (RMSE) of 0.560 m for training data. Figure 9 shows the validation plot testing data, while figure 10 shows the scatter plot between predicted and actual water level data for the training and testing dataset, respectively. The validation plot of predicted and actual water level values shows that the model is able to predict some of the peaks, in some cases model predicts a higher value while in some case it predicts a lower value. From the scatter plot with regression line, it is clear that the predicted values are higher than the actual stage values and is highly accurate, regression line is parallel to the 45-degree line but not coinciding with it, it means that the predicted values have a constant bias or offset from the actual values.

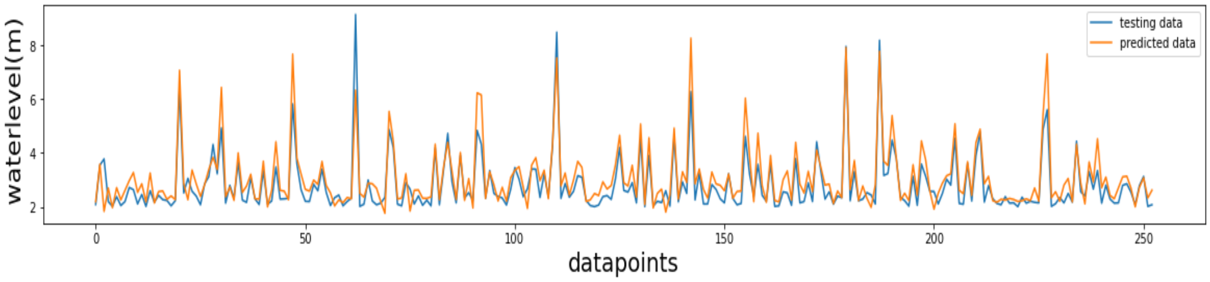


Figure 9. Validation of SVM model on training data.

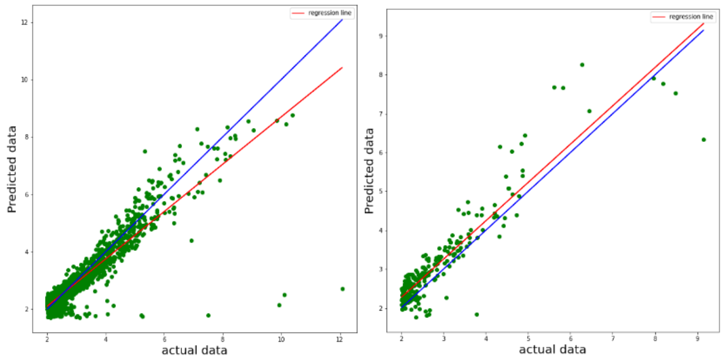


Figure 10. Scatterplot of predicted vs actual data of SVM model using training and testing data respectively.

*4.4 Decision Tree*

A decision tree regressor is a type of decision tree used for regression tasks. It divides the feature space into regions, with each region corresponding to a leaf node in the tree. The target variable values in each leaf node are predicted by taking the average of the training samples in that region. The algorithm recursively splits the data based on the feature that provides the highest information gain until some stopping criterion is met. The resulting tree can be used to predict the target variable for new data by traversing the tree based on the feature values of the new data and outputting the predicted value at the reached leaf node.

The given code creates a decision tree regressor with a maximum depth of 7. The algorithm for decision tree regression starts with the entire dataset at the root node of the tree. For each feature, the information gain is calculated, which measures the reduction in entropy or impurity of the data when split based on that feature. The feature that provides the highest information gain is selected to split the data at that node. This process of recursively splitting the data based on the feature with the highest information gain is repeated until some stopping criterion is met, such as reaching the maximum depth of the tree or a minimum number of samples in a leaf node. At each leaf node, the target variable is predicted by taking the average of the training samples in that region. Information gain is a measure of the reduction in entropy or impurity of the data after splitting based on a particular feature. It is used to determine the best feature to split the data at each node of the decision tree. The resulting decision tree regressor can be used to predict the target variable for new, unseen data by traversing the tree based on the feature values of the new data and outputting the predicted value at the leaf node reached. The feature that provides the highest information gain is chosen as the split feature at that node.

R square value of 0.831 and Mean Square Error (MSE) value of 0.246 $m^2$ and Root Mean Square Error (RMSE) of 0.496 m for testing data and R square value of 0.928 and Mean Square Error (MSE) value of 0.089 $m^2$ and Root Mean Square Error (RMSE) of 0.299 m for training data.  Figure 11 shows the validation plot for testing data, while figure 12 shows the scatter plot between predicted and actual water level data for the training and testing dataset, respectively. The validation plot of predicted and actual water level values shows that the model is able to predict most  of the peaks and the predicted values are slightly higher than the actual values .From the scatter plot with regression line it is clear that the predicted values are higher than the actual stage values and is highly accurate, regression line is parallel to the 45-degree line but not coinciding with it, it means that the predicted values have a constant bias or offset from the actual values.

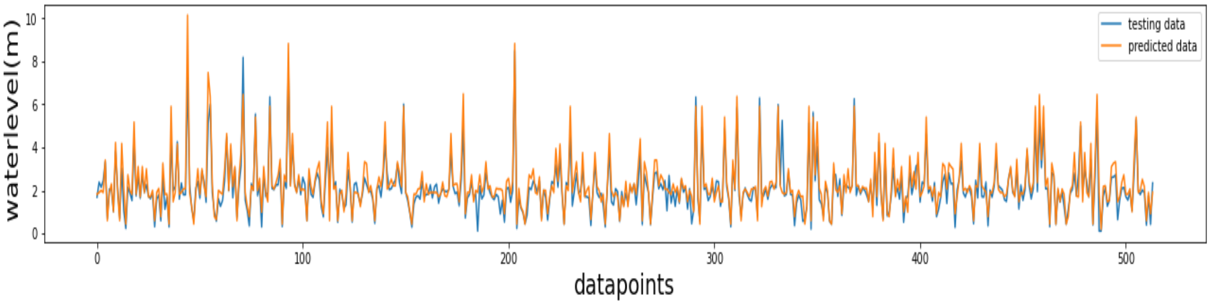


Figure 11. Validation of Decision tree model on testing data.

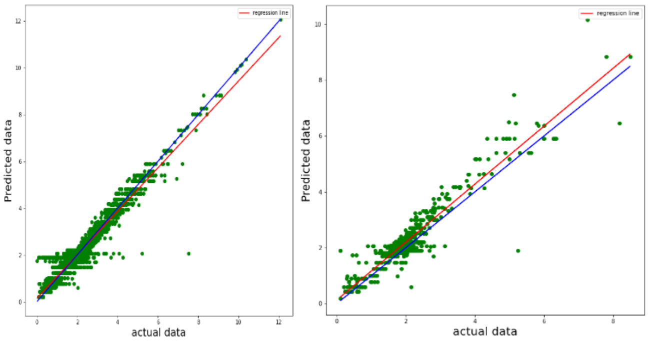


Figure 12. Scatterplot of predicted vs actual data of Decision tree model using testing and testing data respectively.

### *4.5 Random Forest*

Random Forest Regressor is a popular machine learning algorithm used for regression tasks. It is based on the concept of ensemble learning, which combines multiple models to improve the accuracy of predictions. The Random Forest Regressor algorithm combines the outputs of multiple decision trees to create a final prediction. Each tree in the forest is trained on a randomly selected subset of features and observations from the training data. By building multiple trees in this way, the algorithm can capture complex nonlinear relationships between the input features and the target variable. Random Forest Regressor is efficient with large datasets and reduces overfitting by using a subset of features for each tree. It is also robust to noisy and outlier data by averaging the predictions of all trees. The used model creates an instance of the 'RandomForestRegressor' class with 1000 decision trees and sets the random state to 0. Training a Random Forest Regressor involves several steps. First, subsets of the original dataset are randomly selected with replacement, creating multiple training sets. For each training set, a subset of features is randomly selected to use for training a decision tree. Decision trees are built for each training set using the selected features, and the tree is trained using the training data. This process is repeated for the specified number of decision trees. To make a prediction, a new input data point is passed through each decision tree, and a prediction is obtained for each tree. The predictions from all decision trees are then averaged to get the final prediction for the input data point. During the training process, the algorithm also employs techniques to prevent overfitting, such as limiting the maximum depth of the decision trees and using a minimum number of samples required to split an internal node. R square value of 0.848 and Mean Square Error (MSE) value of 0.222 $m^2$ and Root Mean Square Error (RMSE) of 0.470 m for testing data and R square value of 0.988 and Mean Square Error (MSE) value of 0.014 $m^2$ and Root Mean Square Error (RMSE) of 0.122 m for training data.  Figure 13 shows the validation plot for testing data, while figure 14 shows the scatter plot between predicted and actual water level data for the training and testing dataset, respectively. The validation plot of predicted and actual water level values shows that the model is able to predict most  of the peaks and the predicted values are slightly higher than the actual values .From the scatter plot with regression line it is clear that the predicted values are higher than the actual stage values and is highly accurate, regression line is parallel to the 45-degree line but not coinciding with it, it means that the predicted values have a constant bias or offset from the actual values.

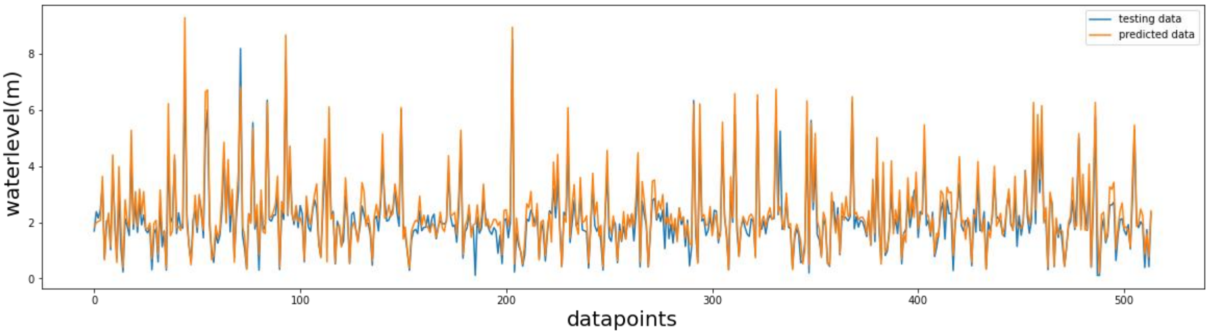


Figure 13. Validation of Random Forest model on testing data.

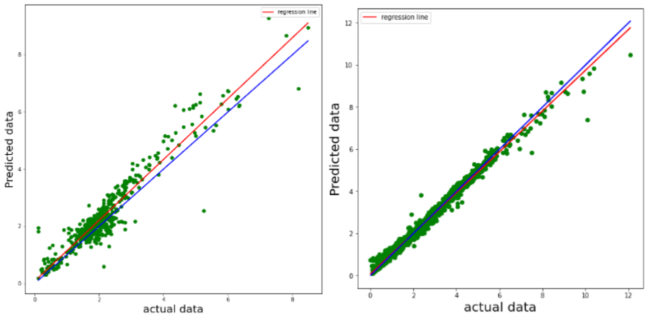


Figure 14. Scatterplot of predicted vs actual data of Random Forest model using testing and training data respectively.

## 5.Summary and Conclusion

A substantial review of the vast literature comprising particulars on the various Machine Learning methods ANN, which showcased the influence of different meteorological and hydrological parameters on the discharge and water level of a given geographical region, was done. Data from graded sources were obtained containing various hydrological and meteorological parameters across the Chaliyar River basin. The datasets were analysed and cleaned thoroughly to remove all the errors.

The correlation of different meteorological parameters with discharge and stage was found using the correlation function present in python. A meta table containing standard deviation, mean, maximum and minimum values of all the parameters were found and analysed.

The model was created in order to predict the level of water in the Chaliyar river basin using multiple linear regression, ANN, SVM, decision tree and random forest. Table 3 represents the comparison between the results of all three models. ANN is the most accurate and precise of all three models

Table 3. Comparative Results of the models

| Models | Training Data | | | Testing Data | | |
|---|---|---|---|---|---|---|
| | $R^2$ | MSE $(m^2)$ | RMSE (m) | $R^2$ | MSE $(m^2)$ | RMSE (m) |
| Linear Regression | 0.794 | 0.415 | 0.548 | 0.761 | 0.384 | 0.548 |
| ANN | 0.887 | 0.281 | 0.407 | 0.866 | 0.265 | 0.410 |
| SVM | 0.716 | 0.414 | 0.643 | 0.751 | 0.314 | 0.560 |
| Decision Tree | 0.928 | 0.246 | 0.496 | 0.831 | 0.089 | 0.299 |
| Random Forest | 0.988 | 0.014 | 0.122 | 0.848 | 0.222 | 0.470 |

Based on the provided results, it can be summarized that the Random Forest and ANN models have the highest accuracy scores and the lowest risk of overfitting. The Random Forest model performed very well on both the training and testing data with high accuracy scores and low error metrics. The ANN model also showed high accuracy scores and low error metrics on both training and testing data. The Decision Tree and Linear Regression models have a higher risk of overfitting with a significant difference between the training and testing metrics. Although the Decision Tree model performed well on the testing data, it did not perform as good as the Random Forest or ANN models. The Linear Regression model performed reasonably well on the testing data, but not as good as the Random Forest or ANN models. The SVM model had the lowest accuracy scores and the highest risk of overfitting among all the models with lower accuracy scores and higher error metrics on both training and testing data. In conclusion, the Random Forest and ANN models appear to be the most suitable for this particular dataset, while the SVM model may not be appropriate for this particular problem. The Decision Tree and Linear Regression models may also perform well, but extra measures should be taken to avoid overfitting.

**Nomenclatures**

| | |
|---|---|
| ANN | Artificial neural network |
| ML | Machine learning |
| SVM | Support vector machines |
| RMSE | Root Mean Square Error |
| MSE | Mean Square Error |
| ReLU | Rectified Linear Unit |

## References

[1] Ahmad, M., Al Mehedi, M. A., Yazdan, M. M. S., & Kumar, R. (2022). Development of Machine Learning Flood Model Using Artificial Neural Network (ANN) at Var River. *Liquids*, *2*(3), 147-160.

[2] Dtissibe, F. Y., Ari, A. A. A., Titouna, C., Thiare, O., & Gueroui, A. M. (2020). Flood forecasting based on an artificial neural network scheme. *Natural Hazards*, *104*(2), 1211-1237.

[3] Khosravi, K., Pham, B. T., Chapi, K., Shirzadi, A., Shahabi, H., Revhaug, I., Prakash, I., & Tien Bui, D. (2018). A comparative assessment of decision trees algorithms for flash flood susceptibility modelling at Haraz watershed, northern Iran. Science of the Total Environment, 627, 744–755.

[4] Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. *Water*, *10*(11), 1536

[5] Samantaray, S., Das, S., Sahoo, A., & Satapathy, D. P. (2022). Monthly runoff prediction at Baitarani river basin by support vector machine based on Salp swarm algorithm. *Ain Shams Engineering Journal*, *13*(5), 101732.

[6] Supriya, P., Krishnaveni, M., & Subbulakshmi, M. (2015). Regression Analysis of Annual Daily Rainfall and Stream Flow for Flood Forecasting in Vellar River Basin. *Aquatic Procedia*, *4*, 957–963.

[7] Thakural, L. N. (2018). Trend Analysis of Rainfall for the Chaliyar Basin, South India. *International Journal for Research in Applied Science and Engineering Technology*, *6*(7), 91–100.

**Acknowledgement**

Authors are highly thankful to faculty in charge Dr. Sathish Kumar D whose guidance and instructions were of immense help while making this report. I also acknowledge my gratitude towards my parents and member of my family, who has always supported me morally as well as economically.