Streamflow prediction using different Machine Learning Models based on Discharge and Precipitation Time series

BY

KILARI VENKATA SAI HARSHITA
(20JE0485)

Dissertation

SUBMITTED TO

INDIAN INSTITUTE OF TECHNOLOGY
(INDIAN SCHOOL OF MINES), DHANBAD

For the award of the degree of

BACHELOR OF TECHNOLOGY

May, 2024

CERTIFICATE

This is to certify that Ms. KILARI VENKATA SAI HARSHITA (20JE0485), a student of B.Tech, Department of Environmental Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad has worked under my guidance and completed his Dissertation entitled “Streamflow prediction using Different Machine Learning Models Based on Discharge and Precipitation Time series” in fulfillment of the requirement for award of degree of B.Tech. in Environmental Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad. This work has not been submitted for any other degree, award, or distinction elsewhere to the best of my knowledge and belief.

Dr. Tinesh Pathania
Asst. Professor
Dept. of Environmental Engineering

DECLARATION

I declare that this project report titled Streamflow prediction using Different Machine Learning Models Based on Discharge and Precipitation Time series submitted in partial fulfillment of the degree of B. Tech in Environmental Engineering is a record of original work carried out by me under the supervision of Prof. Tinesh Pathania, and has not framed the reason for the honor of some other degree, in this or some other Institution or University. With regards to the moral practice in detailing logical data, due affirmations have been made any place the discoveries of others have been referred to.

KILARI VENKATA SAI HARSHITA

20JE0485

ACKNOWLEDGMENTS

This was a wonderful journey of learning new things and building up this project I am eternally thankful to my mentor Assistant Professor Dr. Tinesh Pathania for providing me continuous guidance, support, and time from his busy schedule for weekly meetings and clearing my doubts and providing needful resources for my research work, and I am also thankful to my friends for their help and motivation. This research work presented has been completed with the help of many well-wishers directly or indirectly.

KILARI VENKATA SAI HARSHITA

Table Of Contents

1. INTRODUCTION

The sustainable management of water resources is paramount for ensuring environmental, economic, and societal well-being. Central to this management is the accurate prediction and understanding of streamflow dynamics, which directly impacts water availability, flood risk mitigation, hydropower generation, and ecosystem health. Traditional hydrological models, while effective, often face limitations in capturing the complex and nonlinear relationships inherent in streamflow processes. In recent years, the integration of machine learning techniques with hydrology has emerged as a promising approach to address these challenges.

In this context, this project focuses on leveraging machine learning algorithms to model and predict daily streamflow at the "Santa Coloma de Gramenet" hydrometric station. Located in Catalonia, Spain, this station plays a crucial role in monitoring water flow within its catchment area. By harnessing historical data on discharge and precipitation from multiple stations, along with upstream flow data, this project aims to develop robust predictive models capable of accurately forecasting streamflow patterns.

The objective of this endeavor is twofold: firstly, to explore the efficacy of various machine learning algorithms in modeling streamflow dynamics, and secondly, to provide actionable insights that can inform water resource management decisions. By employing techniques such as Multiple Linear Regression, Support Vector Regressor, Random Forest. We seek to uncover hidden patterns and relationships within the data that traditional hydrological models may overlook.

The significance of this project lies in its potential to enhance our understanding of streamflow processes and improve the reliability of streamflow predictions. By harnessing the power of machine learning, we aim to develop models that not only accurately capture the temporal and spatial variability of streamflow but also offer insights into the factors driving these fluctuations. Such models can serve as valuable tools for water managers, policymakers, and stakeholders, enabling them to make informed decisions regarding water allocation, flood preparedness, and environmental conservation efforts.

2. LITERATURE REVIEW

Several research endeavors have explored the application of machine learning (ML) techniques to forecast streamflow based solely on meteorological data across various geographical regions. Adnan et al. (2019) conducted a study on modeling monthly streamflow at the Swat River Basin in Pakistan, utilizing precipitation and temperature inputs. Their findings indicated that monthly streamflows at Kalam Station could be accurately predicted using temperature data alone. Additionally, precipitation inputs alone yielded satisfactory accuracy for Kalam Station but were less reliable for predicting streamflow at the Chakdara Station [24]. Tongal and Booij (2018) conducted a comparative analysis of ML methods, including SVM, and RF, utilizing precipitation, temperature, and evapotranspiration data for streamflow modeling across four rivers in the USA [30]. Their results corroborated the effectiveness of incorporating meteorological data for precise streamflow predictions

3. OBJECTIVES

3.1 Model Development:

The primary objective of this project is to develop accurate and reliable predictive models for daily streamflow at the "Santa Coloma de Gramenet" hydrometric station. Leveraging historical data on discharge and precipitation from multiple stations, along with upstream flow data, we aim to construct machine learning models capable of forecasting streamflow dynamics with high precision and confidence.

3.2 Algorithm Evaluation:

Another key objective is to evaluate the performance of various machine learning algorithms in modeling streamflow processes. By implementing and comparing algorithms such as Multiple Linear Regression, Support Vector Regressor, Random Forest Regressor. We seek to identify the most effective approach for streamflow prediction under different scenarios and conditions.

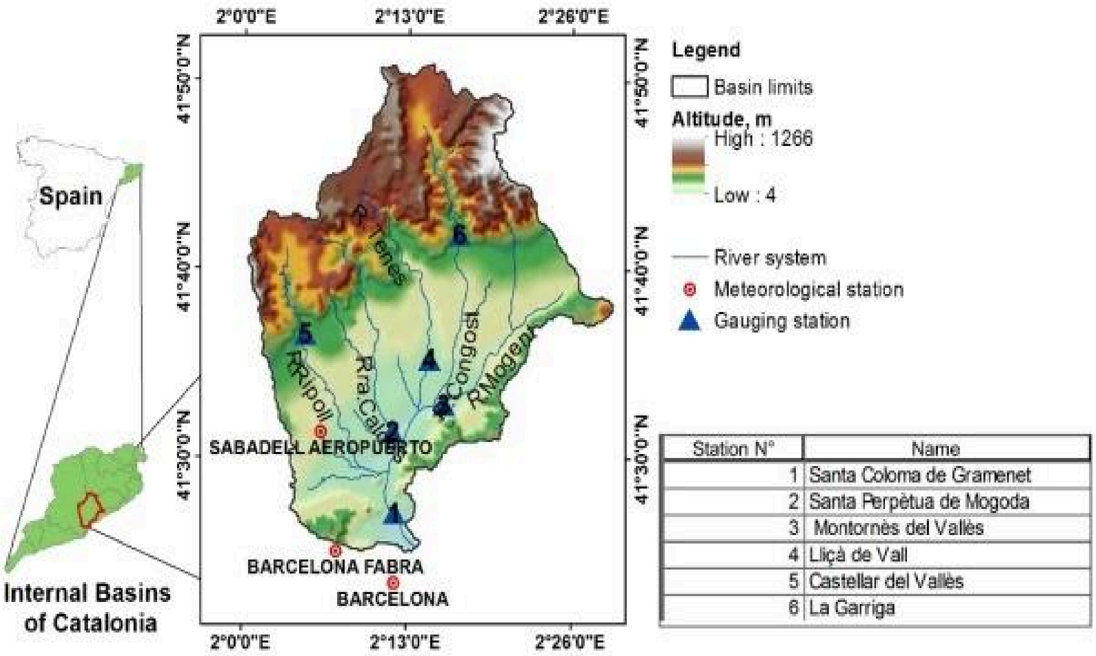
4. STUDY AREA, DATA, MODELS USED

4.1. Study area:

The study area considered was Besos river flowing through catalonia, spain formed by the confluence of the Mogent and congost river having coordinates 41°27′29″N, 2°11′39″E.

The Besòs river has five main tributaries.

- Congost River
- Ripoll River
- Mogent River
- Tenes River
- de Caldes stream



Station N°	Name
1	Santa Coloma de Gramenet
2	Santa Perpètua de Mogoda
3	Montornès del Vallès
4	Lliçà de Vall
5	Castellar del Vallès
6	La Garriga

Figure 1. Study area and location of the meteorological and gauging stations.

4.2. Data used:

In this work the data we have used obtained from Kaggle. It is continous data from 2003-01-01 to 2010-12-31. In the dataset the columns represent: -

- Gramenet: daily discharge at the "Santa Coloma de Gramenet" gauging station. Units are in m3/s.
- Barcelona, Barcelona_fabra and Sabadell_aero: daily rainfall in the "Barcelona", "Barcelona Fabra" and "Sabadell Aeropuerto" rain stations. Units are in mm.
- Garriga, Castellar, Llica, el_Mogent, Mogoda: daily upstream flow discharge at the "La Garriga", "Castellar Valles", "Lliça de Vall", "Montornes Valles", "Santa Perpetua de Mogoda" gauging stations. Units are in m3/s.

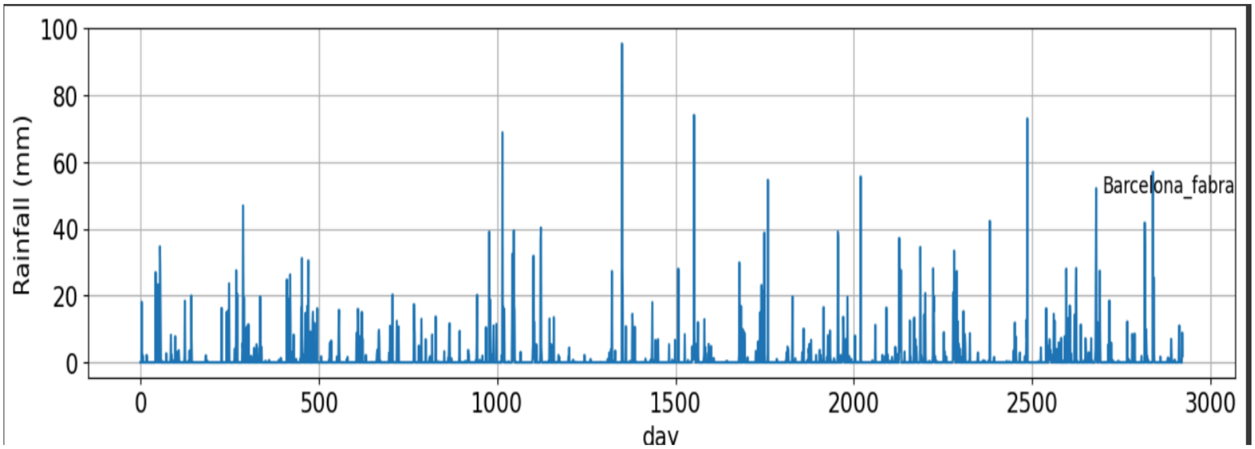
4.3. Data Exploration:

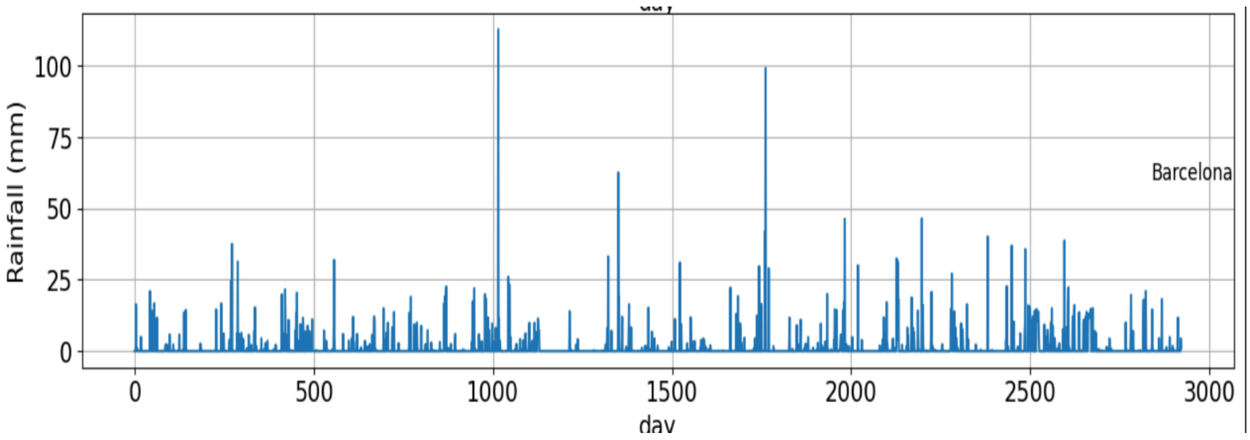
The data taken is distributed over a period from 1 January 2003 to 31 December 2010 i.e. upto 2011, as some stations no longer have records after this date. There are many missing values from 6 May 2008 onwards in almost every station. Therefore, we have filled the blank values with 0 (mm|m3/s) as 3 columns are precipitation values having units mm and others are the streamflow values to the station we considered having units m3/s . The resulting data time series contained a total of 2922 rows daily rainfall and flow records.

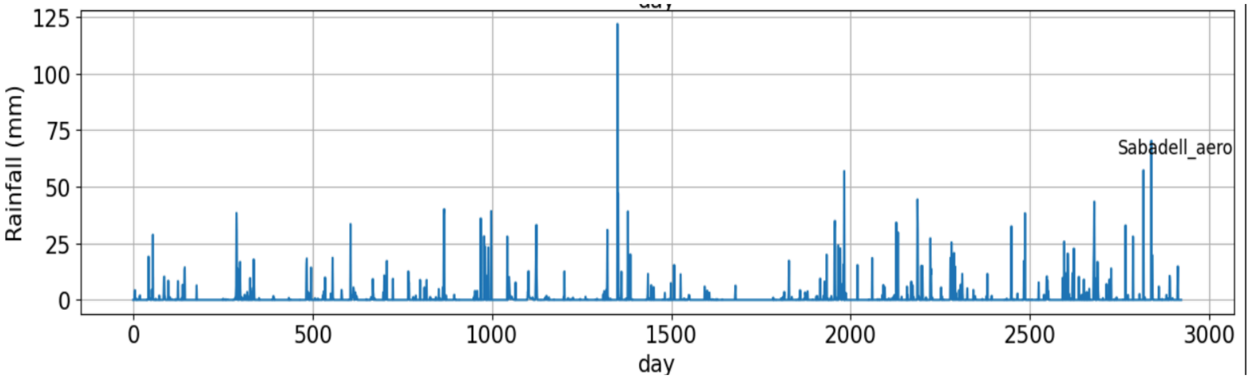
4.4. Data Visualization:

Visualizations help us in exploring the data and identify patterns, trends, and relationships, and can be used to evaluate the performance of streamflow models. For example, in time series data plots can be used to compare the predicted and observed streamflow data.

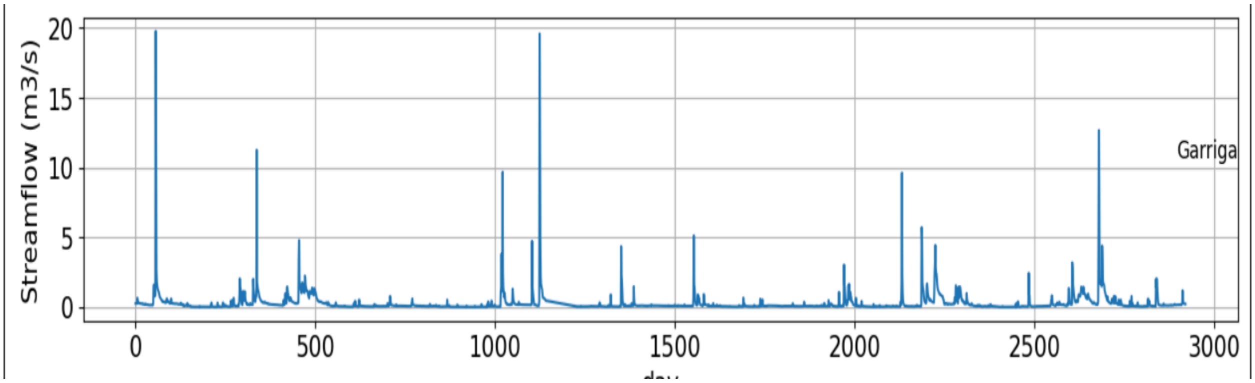
Have done the visualization between the Rainfall(mm) | Streamflow(m3\s) with respect to the days. Given as

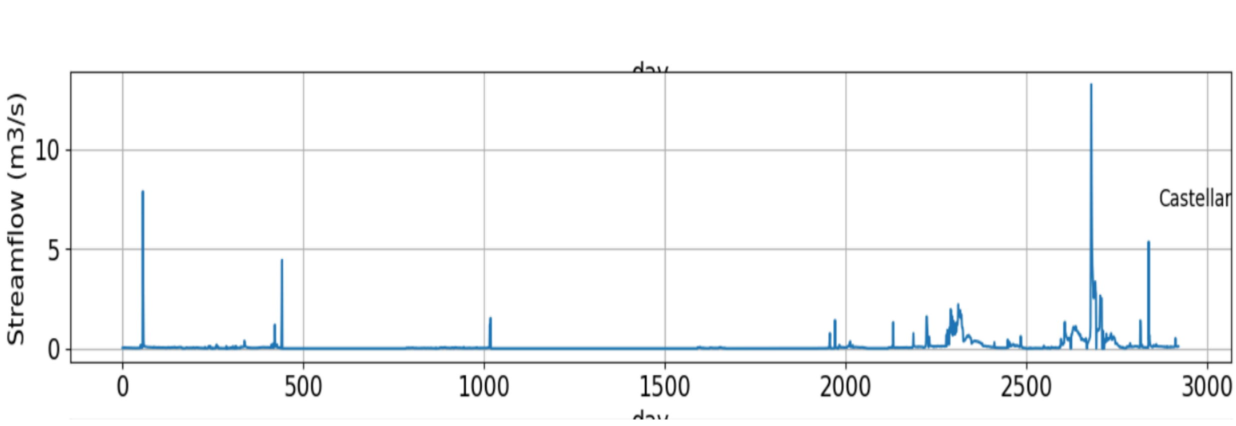


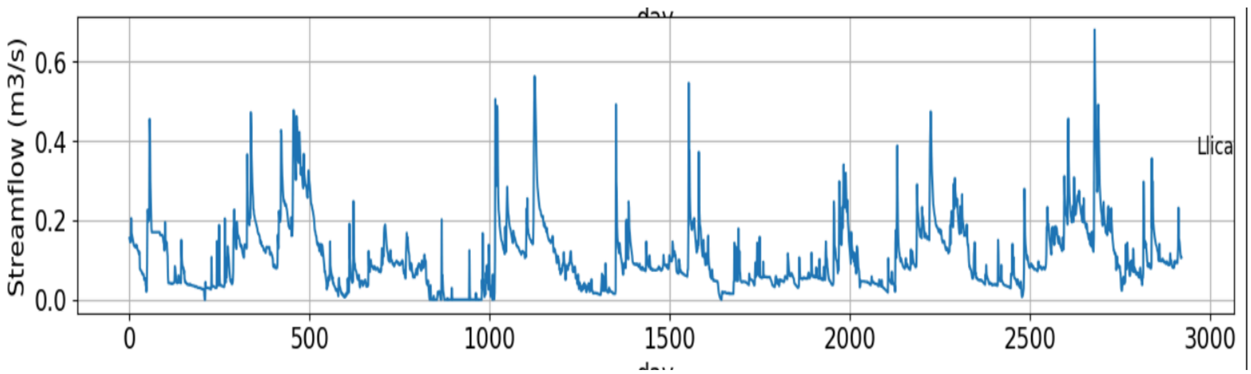


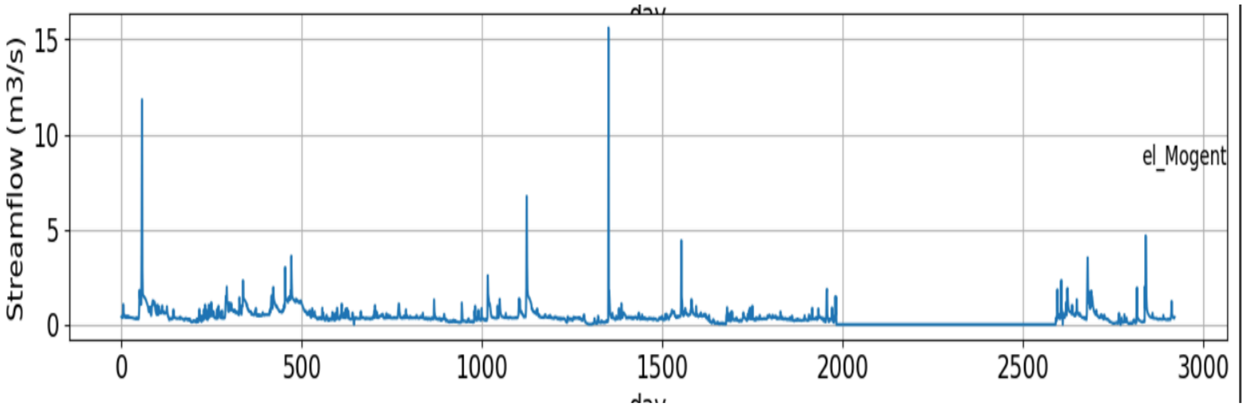


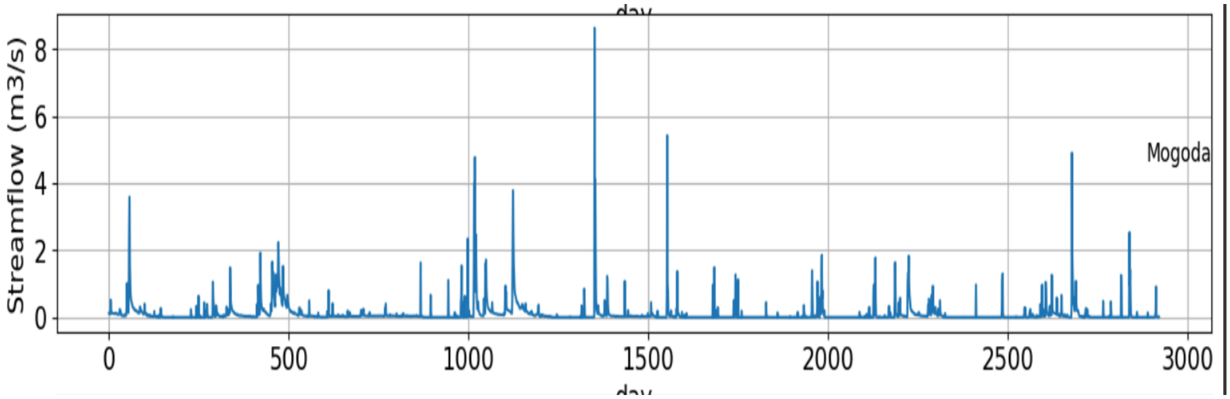
The above are the rainfall vs time graph. where the graph shows the daily rainfall for Barcelona_fabra, Barcelona, Sabadell_aero, with some days having no rainfall and others having significant rainfall events, and the graph shows that rainfall can be highly variable, with some days having very little rainfall and others having significant rainfall events.



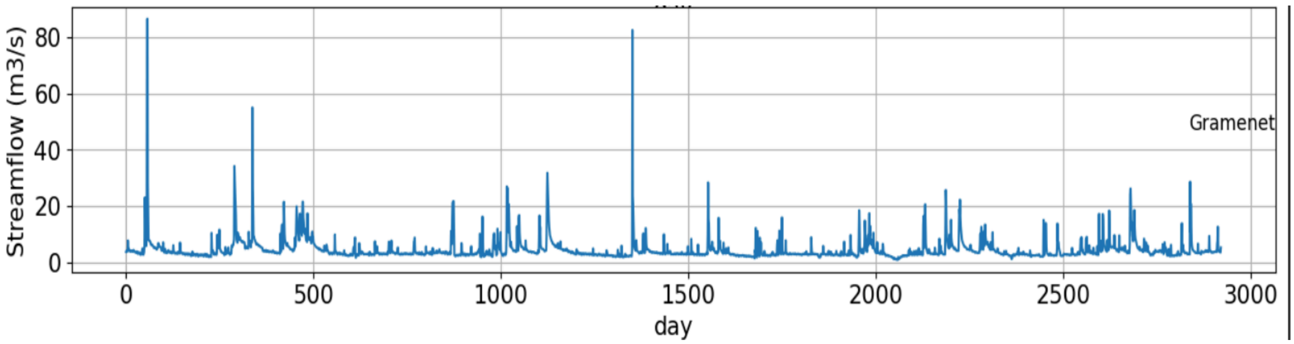








The above are the streamflow vs time graph. where the graph shows the daily streamflow for Garriga, Castellar, Llica, el_Mogent, and Magoda which are from different river streams where all meet at a point "Santa Coloma de Gramenet" gauge station where daily discharge is found out.



4.5. Models used:

Various ML and recent deep learning techniques are being developed to better model processes in different domains. In the present study, we have used Support

Vector Regression (SVR), Random Forest (RF), and Multiple Linear Regression a brief description of each algorithm is as follows:

4.5.1. Support Vector Regression :

Support Vector Regression (SVR) is a type of machine learning algorithm used for analysis. The goal of the support vector recession model is to find a function that approximates the relation between the input and target(output) variables and also minimizes the error.

And is also a popular algorithm for classification problems.The basic idea is to find the best possible hyperplane passing among the data points thereby minimizing the error.

In SVR, the hyperplane is taken such that it has the maximum margin(the distance between the hyperplane and the nearest data points).These nearest data points are called support vectors, hence the name of the algorithm.

It uses a technique called the kernel trick to transform the data into a higher-dimensional space, where it is easier to find the best hyperplane.There are many types of kernels – linear, Gaussian, radial basis function(rbf) etc. Each is used depending on the dataset. Here we have considered the kernel as rbf because when compared with kernels linear and rbf linear showed a poor model performance where when kernel is rbf it showed a better model performance.

Implementing Support Vector Regression (SVR) in Python: There are few steps for implementation of SVR in python firstly we import the libraries required. And then a standard scalar is used which basically helps to normalize the data within a particular range. Normally several common class types contain the scaling function so that they make scaling automatically. However, the SVR class is not a commonly used class type so we should perform feature scaling using Python.

Then we fit the SVR into the model where the Kernel is the most important feature as discussed the importance of it above.

And have compared between 'C': [0.1, 1, 10, 100,1000], and 'gamma': [0.001, 0.01, 0.1, 1] where the best parameters have appeared to be {'C': 1000, 'gamma': 0.01}

4.5.2. Multi Linear Regression:

Multi Linear regression is a technique to learn(train), model(forecast/predict) a linear function between (one or many)independent variables and a dependent variable. In this study we used a Multi Linear regression technique. As our input parameters are 3 precipitation values and 4 streamflow values where our dependent feature the streamflow discharge.
And if there is any noise in the Data the random error occurs where the MLR also checks for it.

For example, runoff-rainfall mapping for linear regression would look like:

$$Q = a.P + b$$

where Q = runoff or discharge i.e Y
P = precipitation i.e P
a,b = coefficients of the function

To implement linear regression in Python, we used the LinearRegression class from the sklearn.linear_model library.

4.5.3. Random Forest Regression:

Random Forest algorithm is also a machine learning algorithm used for regression(forecasting/Prediction) techniques.This also combines the multiple Decision trees logic to predict.

We have used the sklearn module for training our random forest regression model, specifically the RandomForestRegressor function. Some of the important parameters which we have used are **N_estimators, max_depth** etc.

And to implement the model we import RandomForestRegressor from Sklearn.ensemble library.

4.6. Hyperparameter Optimisation:

Models are generally trained with a range of hyperparameter values that was determined by examining various hydrological studies. Afterwards, the hyperparameter optimal values were determined through a trial-and-error process using the K- fold cross validation and Grid Search technique in the training and test datasets. Where (k-fold cross validation and Grid search techniques are two of the best techniques in hyperparameter tuning/optimization) Then, the optimal hyperparameters were maintained and the best models were used to predict the flow rate.

In the model we have used K-fold cross validation for SVR model and Grid search CV for RFR after cross checking the model performance by both the methods.

5.Methods

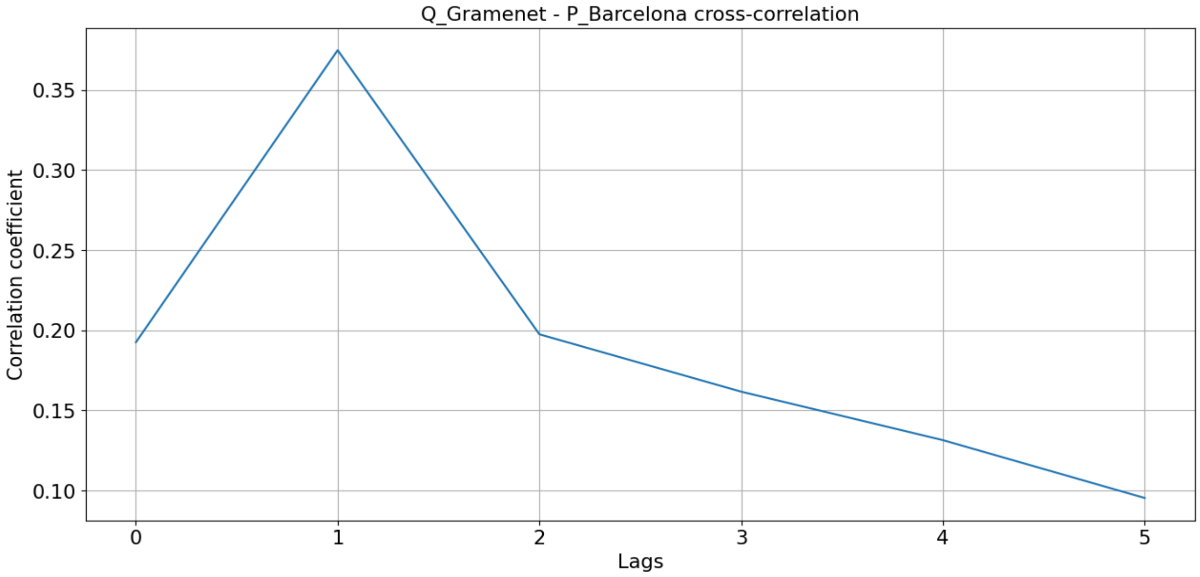
5.1. Lag creation and Cross correlation:

Lag creation is a technique used in time series analysis to create new features from existing ones by shifting the time series data by a certain number of time steps. In the context of streamflow modeling, lag creation is used to capture the temporal relationships between the streamflow data and its past values. This is important because streamflow data often exhibits autocorrelation, meaning that the current value is influenced by past values.

In the code, lag creation is implemented using the **shift** function, which shifts the streamflow data by a specified number of time steps.

Cross-correlation is a statistical measure that calculates the similarity between two time series signals at different time lags. In the context of streamflow modeling, cross-correlation is used to analyze the relationships between the streamflow data and its lagged features.The cross-correlation coefficient ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation.

For Example, in the code we have considered the cross correlation between the discharge point(Q) Gramenet and precipitation(P) at Barcelona station



Given that flow is effectively made up of contributions from different subareas whose travel time covers a range of values, the next step was the determination of the appropriate lag time concerning the prediction output. This was carried out through a cross-correlation analysis between the flow at the outlet and the upstream and downstream rainfall and flow. The cross-correlation showed the considerable influence of the previous day's (t−1) rainfall on the current value of the outlet flow for the three meteorological stations. From time t−2, this correlation decreased to less than 0.3. Also, regarding the flow at the input stations, it was seen that there was a decreasing correlation from the same day of recording. The antecedent flow after time instant t−2 did not contribute significantly to the outflow generation. Therefore, the antecedent values of flow and rainfall corresponding to time instants t and t−1 were considered.

Now, it is then necessary to use variables that extract and preserve hidden information within cyclical data, such as the distance between two events: day 30 or 31 and day 1, or month 12 and month 1. This seems important as the missing values were removed. To do this, "sin" and "cos" were used to assign each cyclic variable (day and month) to a circle so that the smallest value for that variable appeared right next to the largest value. Four cyclic features (daysin, daycos, monthsin, and monthcos) with respect to the day and the month of the year were created to obtain a total of 18 input features.

5.2. Data Normalization:

Data normalization is a preprocessing technique used to rescale numerical data to a common range, usually between 0 and 1, to prevent features with large ranges from dominating the model. Normalization is essential in machine learning and statistical modeling to ensure that all features contribute equally to the model.
Common normalization techniques used in streamflow modeling include Min-max scaling, standardization, log transformation. Here we have used the Min-max scalar where the calculation is done using the formula;

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A$$

To prevent input data in larger numerical ranges from dominating those in smaller numerical ranges and to avoid numerical difficulties during computation, all input variables were scaled to the range [0, 1] using the Sklearn library importing Minmaxscalar.

The complete Time series dataset was then divided into training and test datasets to obtain an approximate training/test split ratio of 80%/20%. With considering Test size as 0.2 in the code representing the 80% to 20% Train test split ratio.

6.Results and Discussion

6.1. Model Evaluation:

Two quantitative validation metrics, including: the coefficient of determination (R2), and the Root Mean Squared Error (RMSE), were used to assess the prediction accuracy and to compare the different data-driven models based on their relative performance.

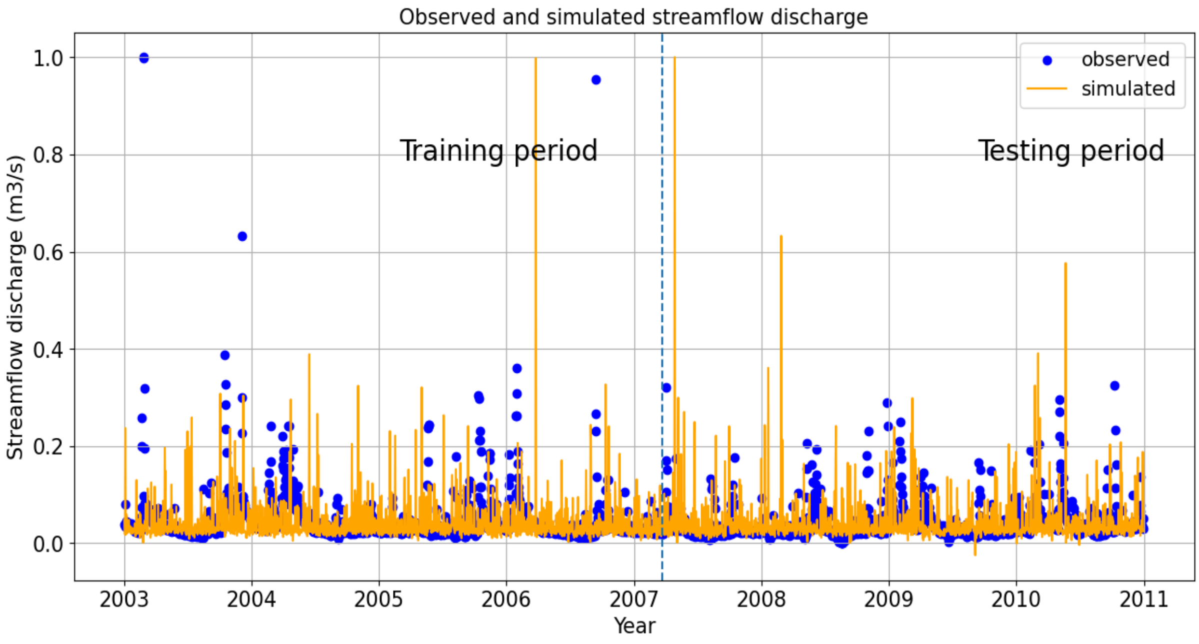
The values of the statistical performance metrics for the training and test periods are presented in below shown Table 1

Model	**R2 score**	**RMSE**
SVR	0.8517	0.0004
MLR	0.9113	0.0003
RFR	0.7551	0.0007

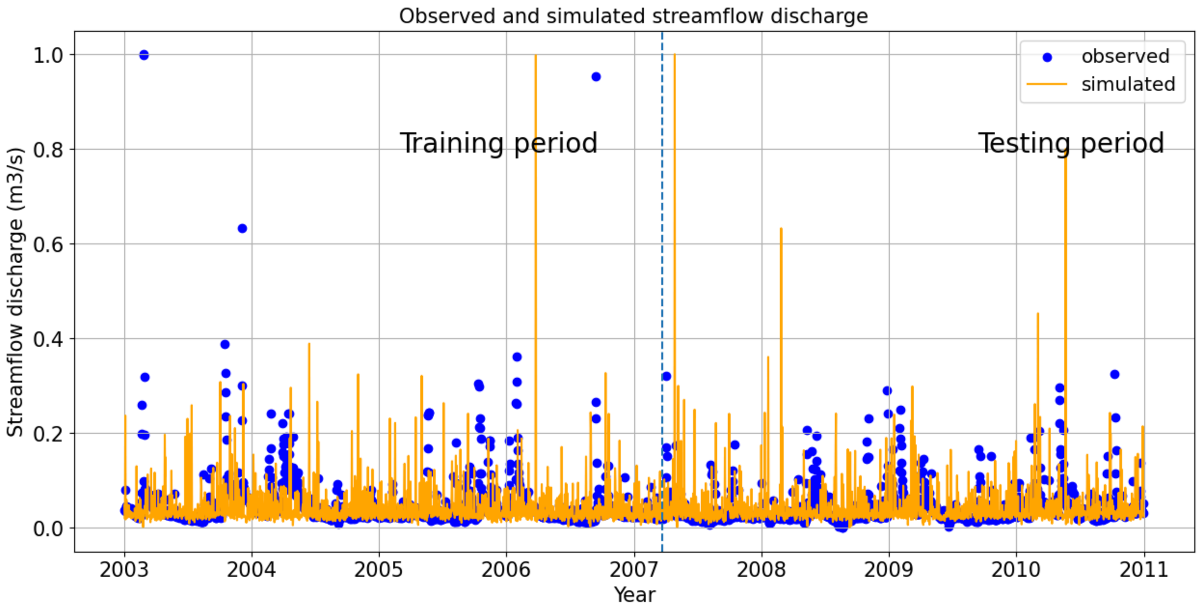
From the results obtained we can say that the Multi linear model fits well with the data we have taken.

6.2. Hydrograph:

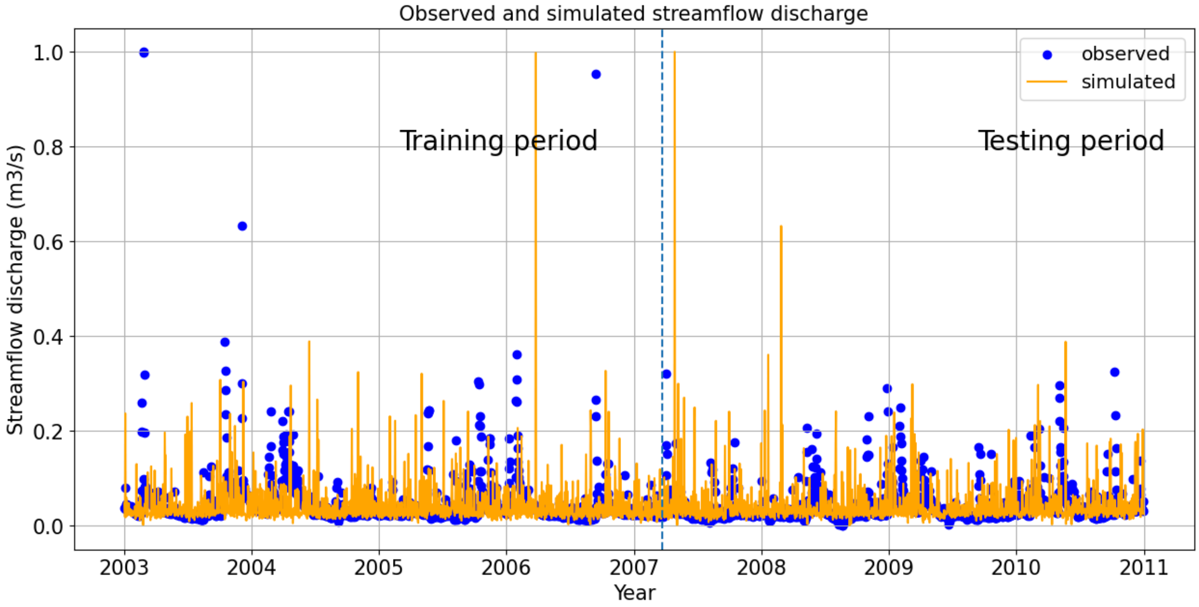
Hydrographs were also plotted to visualize the model behavior, particularly for extreme values as shown below



The hydrograph in the above figure of the observed and predicted values for each model in the training period, as well as the test period at the target station, indicate that, in general, the predicted flow fits well with the observed flow but not exactly so we can say that the SVR model didn't predict well.



The training period, as well as the test period at the target station, indicate that, in general, the predicted flow fits well with the observed flow for the MLR model.



The training period, as well as the test period at the target station, indicate that, in general, the predicted flow fits well with the observed flow for the RFR model.

7. Conclusion

To predict the daily flow at the outlet of the Besós river basin, three data-driven ML models, SVR, RFR and MLR, were used.

The obtained results show that the MLR model outperformed the other models. MLR, as well as the decision tree ensemble model (RFR), has also shown a good flow prediction capacity.

It is worth noting that the proposed Data Models have demonstrated high efficiency in capturing the real trend and the underlying phenomena of rising and falling flow curves. The use of the antecedent flows in the target gauging station had a positive impact on improving the performance of all models.

To improve the prediction capabilities of ML models, in future work, it is recommended

1. to understand clean Data correctly so that the unstructured data do not affect the performance of the model.
2. to use other variables to build a strong relationship with the streamflow.
3. to perform a sensitivity analysis of input features to bring out those that contribute the most to the flow prediction
4. to pay close attention to the data length and split ratio
5. to ensure that the training phase experiences most of the streamflow patterns to allow the models in the test period to simulate the flow discharge with an acceptable level of accuracy

REFERENCES

[1]. Multiple Linear Regression in R – a tutorial (datascienceinstitute.net)

[2]. https://www.mdpi.com/2073-4441/10/11/1536

[3]. https://www.mdpi.com/2073-4441/13/24/3482

[4]. https://www.mdpi.com/2673-4931/25/1/30

LIST OF FIGURES

LIST OF TABLES