# ID3802 - Open Ended Lab Project
# AI ML Applications In Forecasting Water Resources

**Mentor:**Dr.Subhasis Mitra
Sasidhar Reddy Navuluri
Pawan Kumar Narayanapu

*Abstract*—**This study examines the effectiveness of artificial intelligence (AI) and machine learning models in improving precipitation forecasting for the purpose of optimizing water resource management. The project intends to reduce forecasting errors in precipitation predictions by employing a variety of strategies, such as XGBoost, Random Forest, Convolutional Neural Networks (CNNs), and Long Short-Term Memory networks (LSTMs). By conducting careful testing and analyzing the performance of the model, we rigorously examine factors such as selecting the most relevant features, fine-tuning the hyperparameters, and preparing the data. This method aims to improve the accuracy of predictions and optimize computing efficiency.**

## I. Problem Statement

Using AI and machine learning models like XGBoost, Random Forest, CNNs, and LSTMs in this work aims to help water resource managers deal with the problem of wrong predictions of rainfall. It tries to find the best ways to lower forecasting mistakes through careful analysis so that decisions can be made more reliably and resources can be used in a way that minimizes damage to the environment.

## II. Methodology

### A. Data Collection

Dataset containing historical weather data, including precipitation measurements and other relevant features such as temperature, humidity, wind speed, 27 other data of a particular location from 2007-2020 and also containing PFA Precipitation (PCP) data.

### B. Data Preprocessing

Data preprocessing is a crucial step in the machine learning pipeline, ensuring that the data is clean, consistent, and suitable for the chosen algorithm. In the context of developing a movie recommender system, data preprocessing involves handling missing values, dealing with categorical data, and normalizing numerical data.

*1) Handling Missing Values:* Missing values can significantly impact the performance of machine learning algorithms. Therefore, it's essential to identify and address missing values in the dataset. Common strategies for handling missing values include:

- Imputation:Replacing missing values with estimated values based on statistical methods or patterns in the data.
- Deletion:Removing instances with missing values, especially if they constitute a small proportion of the dataset.

### C. Model Selection and Training

Used 4 different Machine learning models to train the data such as XgBoost,Random forest,LSTM,CNN. Trained the selected models using the training data and optimize their hyper-parameters through techniques like grid search and random search.Pre-processed the data for model training, which involved normalization, scaling, and feature engineering.

## III. Model Selection and Training

### A. XgBoost

XGBoost is a machine learning algorithm that falls under the ensemble learning category, more specifically the gradient boosting framework. The system incorporates decision trees as the foundational learners and utilizes regularization approaches to improve the generalization of the model. XGBoost is a widely used machine learning algorithm that is highly efficient in terms of computing. It is particularly known for its ability to analyze feature importance and handle missing values. XGBoost is commonly applied to tasks such as regression, classification, and ranking.

It effectively manages missing data and utilizes parallel processing to improve efficiency. Furthermore, XGBoost offers valuable information regarding the significance of features, which assists in the understanding of models and facilitates decision-making. These characteristics contribute to its widespread use in applications such as weather prediction and water resource management.

*1) **Model Training**:* The data is divided into 80 % for training and 20% for the testing.

Using the below mentioned techniques we efficiently trained the model.

1) **Time series forecasting** Time series forecasting is a technique used to predict future values based on historical data points ordered sequentially over time.
   Time series data consists of observations collected at regular intervals, such as daily, monthly, or yearly. This data format captures temporal dependencies and trends, making it suitable for forecasting future values.It helps make informed decisions, plan resources effectively, and anticipate future trends and events.
   The TimeSeriesSplit is used for cross-validation method. Splits the data into multiple folds.Each fold contains a contiguous segment of the time series data. This includes:
   - Day of the week,year,month
   - Quarter
   - Month
   - Week of the year

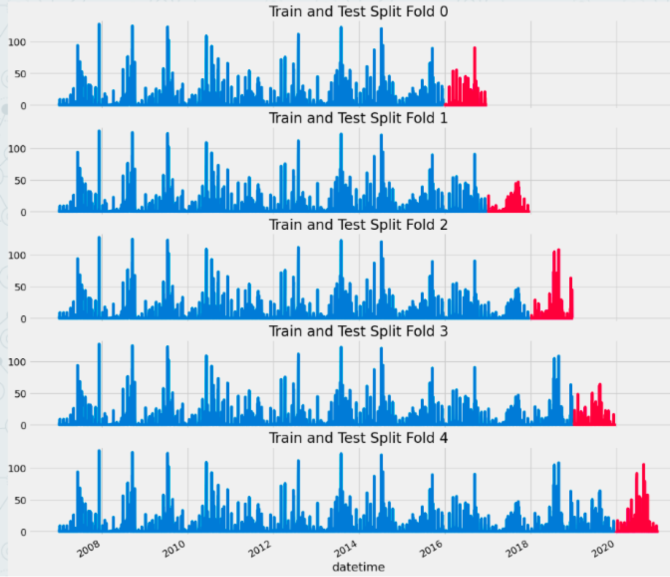


Fig. 1. Time Series Split

2) **Lag Features** By tracking temporal trends and relationships, adding lag features to a time series dataset makes the model work better. These features show what the goal variable or relevant predictors looked like in the past. This lets the model see trends, time delays, and seasonality. By adding lagged numbers, the model gets useful historical information, which makes predictions more accurate.

The first step is to split the time series data so that an XGBoost model can be trained. Each confirmation set has data from the following year, so the total amount of data keeps growing. The dataset is split into training and validation sets after each fold of time series cross-validation. The training data are used to teach the model, and root mean squared error (RMSE) is used to test both the training and validation sets.

Random search is used to fix the hyper-parameters.The search parameters include the number of estimators, maximum depth, and minimum leaf and split sample sizes.

Iteratively retraining the model makes it more efficient, which leads to better performance and the ability to adapt to changing data trends over time. This repetitive method makes sure that predictions of future values are strong and accurate.

### B. *Random Forest*

Random Forest operates by constructing multiple decision trees during the training phase and outputting the mean or average prediction of the individual trees. This ensemble approach helps to reduce overfitting, a common problem in decision tree models, and generally improves the accuracy of predictions.

It utilizes Bootstrap Aggregating (Bagging).Through bootstrapping, it picks random parts of the training data and uses a random set of traits at each node to split them up. Randomness makes sure that trees are different, which makes them stronger. Random Forest captures complex data patterns well because it grows trees to their fullest depths without pruning like some other methods do.

*1) **Model Training**:* Time Series Cross-Validation techniques is used similar to the XgBoost one.

1) **Model optimization:** GridSearchCV is used to fine-tune the Random Forest model's hyperparameters. The search parameters include the number of estimators, maximum depth, and minimum leaf and split sample sizes. This stage is crucial for improving model performance by determining the best combination of parameters.
2) **Feature Importance** Feature importance refers to a technique that assigns scores to input features based on how useful they are at predicting a target variable.
   In Random Forest, these importance scores are calculated during the training process and help in identifying which features contribute most to the prediction accuracy of the model.

### C. *Convolution Neural Network(CNN)*

Convolutional Neural Networks (CNNs) represent a category of deep neural networks renowned for their exceptional capability in processing data exhibiting a grid-like structure, including images. CNNs are utilized extensively in a variety of cognitive tasks, including video and image recognition, recommender systems, image classification, medical image analysis, and natural language processing.

*1) Model Training:*

1) **Feature Scaling:** Standardizing features with StandardScaler ensures equal contribution to the model, reducing the impact of high-value features on training.
2) **Feature Reshaping:** The data is transformed into a 3D array where each feature vector is handled as sequential input to the model by reshaping it into manner appropriate for CNNs.

**CNN Setup:** The sequence data model has numerous critical parts. Starting with a 64-filter Conv1D layer with a kernel size of 3 helps extract patterns from sequential data. This layer uses ReLU functions.

Batch Normalization stabilizes and speeds network training by normalizing activations. A 50% dropout layer randomly deactivates hidden unit connections during training to reduce overfitting.

Next, a Flatten layer converts convolutional layer output into a format suited for fully linked dense layers. Finally, the model has two thick layers: a primary layer with 50 ReLU-activated neurons and an output layer with one neuron for continuous target variable prediction.

This complete architecture efficiently processes sequence data, extracts essential information, and makes accurate predictions.

### D. *Long short-term memory(LSTM)*

Long Short-Term Memory (LSTM) is a specialized architecture of recurrent neural networks (RNNs) that aims to solve the issue of vanishing gradient in regular RNNs. By doing so,

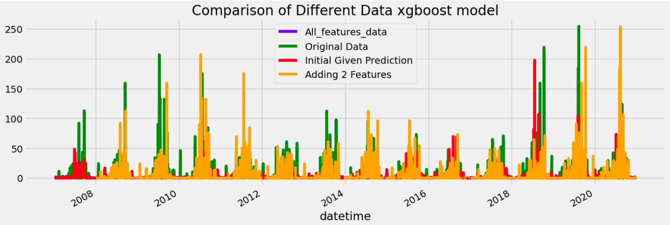


Fig. 2. Plot of XgBoost model with timeline

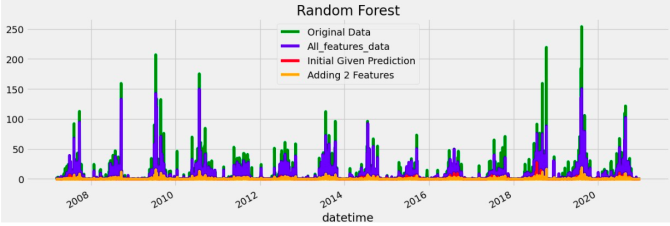


Fig. 3. Plot of Random forest model with timeline

LSTM allows for improved learning and modeling of long-range relationships in sequential data. LSTM networks consist of memory cells that retain information across time, enabling them to capture temporal patterns and relationships in the data.

*1) **Model Training**:*

- **Features of LSTM**
  - Memory Cells:Memory cells hold information over numerous time steps in LSTM networks. These cells' internal states can be updated or forgotten based on input data and previous states.
  - Gates:LSTM cells have input, forget, and output gates that control information flow in and out.

**LSTM Setup**

It has two 64- and 32-unit LSTM layers, dropout regularization, and batch normalization. For regression tasks, the model has two dense ReLU-activated layers and a linear activation output layer. Using the Adam optimizer, the model minimizes training MSE loss.

## IV. Results

The Results of the each model is provided here,mostly XgBoost has done quite a good job in decreasing the RMSE.While LSTM barely decreases the RMSE value because the LSTM model may not be complex enough to capture all data patterns and relationships.

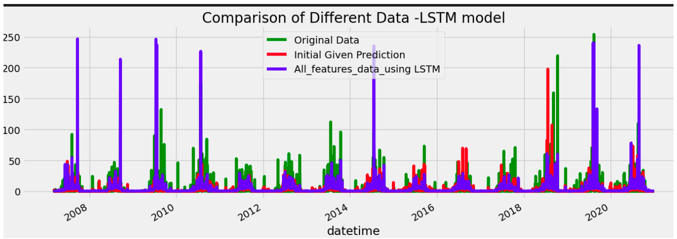


Fig. 4. Plot of LSTM model with timeline

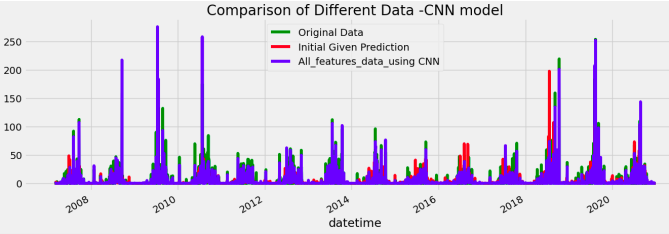
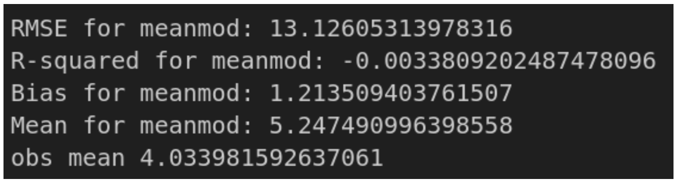
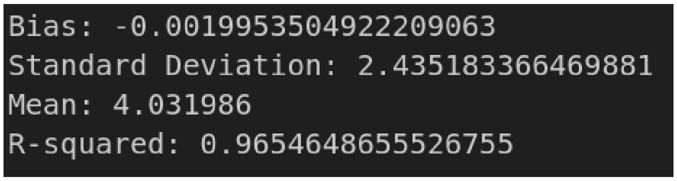


Fig. 5. Plot of CNN model with timeline



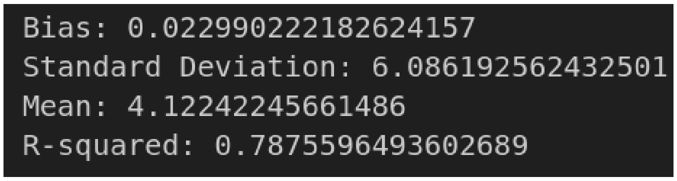
```
RMSE for meanmod: 13.12605313978316
R-squared for meanmod: -0.0033809202487478096
Bias for meanmod: 1.213509403761507
Mean for meanmod: 5.247490996398558
obs mean 4.033981592637061
```

Fig. 6. Initial PCP forecast



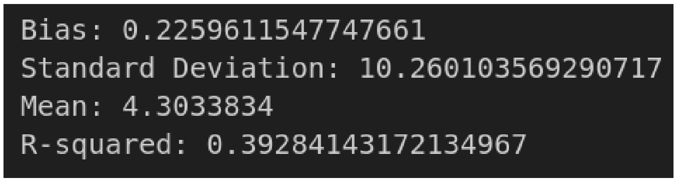
```
Bias: -0.0019953504922209063
Standard Deviation: 2.435183366469881
Mean: 4.031986
R-squared: 0.9654648655526755
```

Fig. 7. XGBoost error



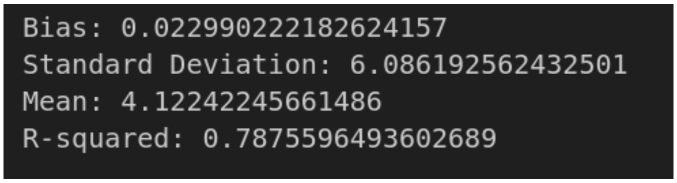
```
Bias: 0.022990222182624157
Standard Deviation: 6.086192562432501
Mean: 4.12242245661486
R-squared: 0.7875596493602689
```

Fig. 8. Random forest error



```
Bias: 0.2259611547747661
Standard Deviation: 10.260103569290717
Mean: 4.3033834
R-squared: 0.39284143172134967
```

Fig. 9. LSTM error



```
Bias: 0.022990222182624157
Standard Deviation: 6.086192562432501
Mean: 4.12242245661486
R-squared: 0.7875596493602689
```

Fig. 10. CNN error

# References

[1] @INPROCEEDINGS8734193, author=Cherif, Iyad Lahsen and Kortebi, Abdesselem, booktitle=2019 Wireless Days (WD), title=On using eXtreme Gradient Boosting (XGBoost) Machine Learning algorithm for Home Network Traffic Classification, year=2019, volume=, number=, pages=1-6, keywords=Prediction algorithms;Feature extraction;Training;Boosting;Cryptography;Machine learning algorithms;Testing;Traffic Classification;Home Networks;Machine Learning;XGBoost., doi=10.1109/WD.2019.8734193

[2] @articlearticle, author = Ali, Jehad and Khan, Rehanullah and Ahmad, Nasir and Maqsood, Imran, year = 2012, month = 09, pages = , title = Random Forests and Decision Trees, volume = 9, journal = International Journal of Computer Science Issues(IJCSI)

[3] @articlearticle, author = Hochreiter, Sepp and Schmidhuber, Jürgen, year = 1997, month = 12, pages = 1735-80, title = Long Short-term Memory, volume = 9, journal = Neural computation, doi = 10.1162/neco.1997.9.8.1735

[4] @articlearticle, author = O'Shea, Keiron and Nash, Ryan, year = 2015, month = 11, pages = , title = An Introduction to Convolutional Neural Networks, journal = ArXiv e-prints

[5] @inbookinbook, author = Cutler, Adele and Cutler, David and Stevens, John, year = 2011, month = 01, pages = 157-176, title = Random Forests, volume = 45, isbn = 978-1-4419-9325-0, journal = Machine Learning - ML, doi = 10.1007/978-1-4419-9326-7$_5$