Forecasting rainfall using STL-Decomposition and Deep-Learning models

Prerak Patel

***Abstract*—The analysis and prediction of rainfall is crucial for a specific region, as it provides deep insights to researchers. It also helps to plan and manage water resources and public transports. This report focuses on daily data with a combined approach of STL-decomposition , machine learning and deep learning models to forecast rainfall a day-ahead, to identify the underlying information in the data Exploratory Data Analysis (EDA) was performed on the raw data. The proposed hybridSTL-DL approach mainly consists of three steps : (1) Decompose the data into trend, seasonality and remainder. (2)Two different deep learning models and one machine learning model,namely Gated Recurrent Unit (GRU) network, multi-time-scale GRU network ,and Light Gradient Boosting Machine(LightGBM) model, are built for modeling and predicting the three components, respectively. (3) The predicted rainfall is eventually acquired by adding up the predicted values of the three components, and several metrics are used to evaluate themodel performance. [1]**

***Index Terms*—IEEE, STL-decompostion, Loess, Trend ,Seasonality, remainder, GRU, multi-timescale-GRU, LightGBM**

I. Introduction

RAINFALL prediction is crucial for any region and thus studying it helps in improving the crop production and resource management. It also helps us in disaster management by predicting landslides and flood situations.

Given past weather data and trying to predict the future rainfall using machine learning or deep learning or any other statistical method is called rainfall prediction.

Jan 30, 2025

II. Approach/Methods/Materials

Prior to any application of algorithm preprocessing and Exploratory data analysis is performed on the data. Followed by a combined approach is used here to decompose the rainfall into its three components trend, seasonality and remainder. Then Deep-learning and machine learning algorithms are used to predict the each of it.

The dataset used in study is available on kaggle [2] and is a cleaned version. Python language and Jupyter notebook is used to perform coding.

The steps performed are as below:

- Exploratory Data Analysis
- STL-Decomposition
- Gated Recurrent Unit network
- Multi-time scale Gated Recurrent Unit network
- LightGradient Boosting Machine (LightGBM)

Department of Electrical and Computer Engineering, Florida Institute of Technology, Florida,FL,32901,USA
Professor Anthony Smith

A. Exploratory data analysis

The granuality of the data was in hours and to do daily prediction resampling it was necessary so based on each column mean or sum was calculated and resampled for the whole dataset

EDA started with the descriptive statistics of each column and then plotting histograms to find out the distribution of the each column data. Correlation matrix is used to find the relation between variables in the data.

B. STL (Seasonal-Trend decomposition using LOESS)

STL decomposes time series into three constituents, namely season, trend, and remainder. Most of the variants are represented using the additive model, though the multiplicative models are possible. It uses locally weighted regression techniques to perform smoothing operations on different aspects of time series. The mathematical representation of the STL decomposition is as follows:

$$Y_i = S_i + T_i + R_i \tag{1}$$

Where:

- Y_i: Observed value
- S_i: Seasonal part
- T_i: Trend part
- R_i: Remainder

It can be found as follows. let $x_i$and y_i for i=1 to n be measurement for dependent variable then choose an integer q as the number of values that are closest to x and then set a closest weight for x_i according to the distance between x_i and x when $q < n$. W denotes the weight function as follows [3]:

$$W\left(\text{u}\right) = \begin{cases} \left(1-u^3\right)^3 & \text{for } 0 \le u < 1 \\ 0 & \text{for } u \ge 1 \end{cases} \tag{1}$$

$vi(x)$ is the neighborhood weight of x_i , $q(x)$ is the distance between x_i and x :

$$v_i\left(x\right) = W\left(\left|x_i - x\right|\right)/\lambda_q\left(x\right) \tag{2}$$

Normally, the graphic representation of the seasonal component should be smooth and repetitive with similar amplitude over time.

C. Gated Recurrent Unit (GRU)

This method is used as it has a special ability to model temporal dependencies in sequential data. Its most widely used in forecasting long-term patterns in the data. It comes under the category of RNN's. As illustrated in Fig. 1, there is an input layer composed of multiple neurons, the number of neurons is determined by the size of the feature space. Similarly, the number of neurons in the output layer corresponds to the output space. [4]

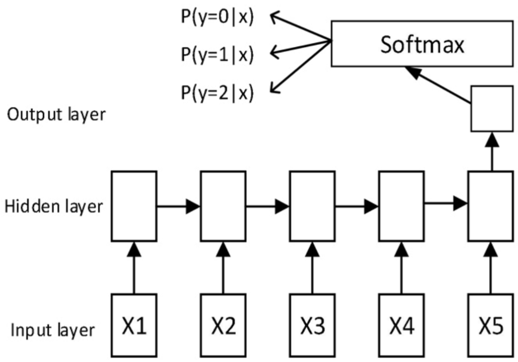

Fig. 1: GRU Model Structure

The hidden layer(s) containing memory cells cover the main functions of the GRU networks. Changes and maintenance of cell status depend on two gates in the cell: a reset gate t_r and an update gate t_u [4] . The structure of a memory cell is illustrated as a circuit diagram in Fig. 2.

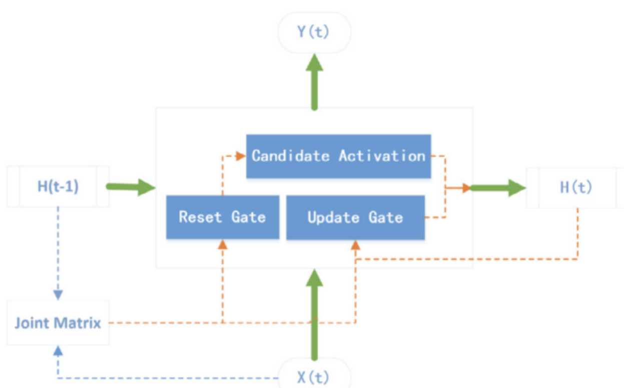

Fig. 2: GRU Memory Cell Structure

D. Multi-time scale Gated Recurrent Unit

The multiple timescales in an MTGRU network is implemented with the help of a timescale variable implemented inside a conventional GRU, thereby adding another gating unit. The timescale gating unit essentially modulates the mixture of the past and current hidden states. [5] Its used to handle multiple sequential data at a time and is an advance version of normal GRU. It can be seen as Fig.3

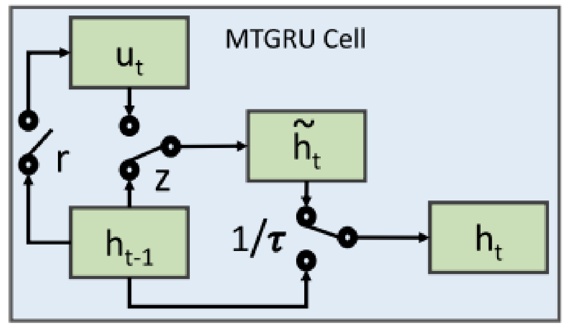

Fig. 3: MTGRU Cell

E. LightGBM

Its an open-source gradient boosting framework developed by Microsoft based on decision trees to improve model efficiency. Select the instances with larger gradients during the training process to optimize memory usage and training time.

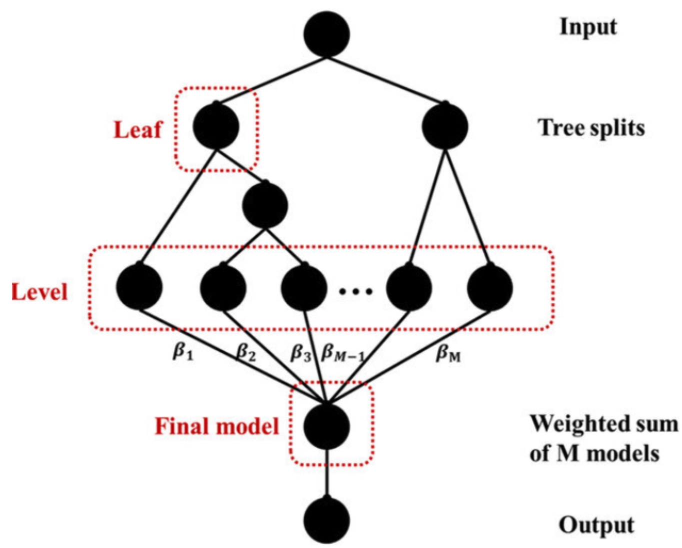

Fig. 4: LightGBM tree structure

III. Experiments/Results

A. Exploratory data analysis

The work started by plotting the data's rainfall, temperature, and other raw variables. We can see a clear peak in the data on 17 June 1993 and 26 June 2002, with rainfall above 200mm/day. It can be seen through the figure. Wind speed, cloud cover, and surface pressure show a relationship with each other. Wind speed and cloud cover are directly proportional to each other, and both are inversely proportional to surface pressure. It can be illustrated in the figure 5.

It can be visualized that the temperature, pressure, and surface pressure are nearly normally distributed, and there's

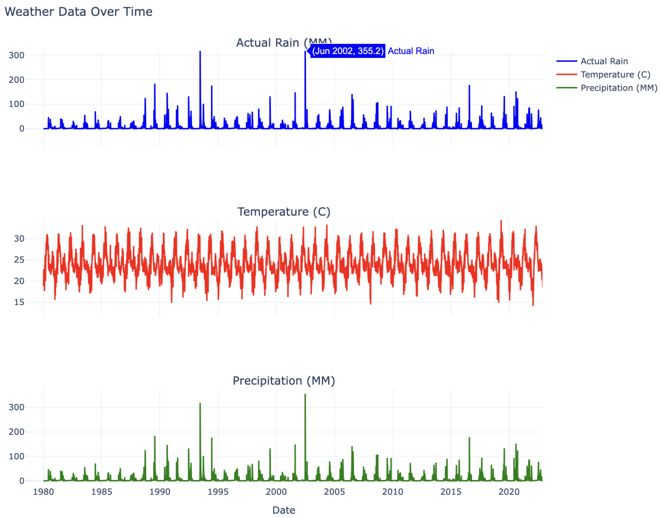

Fig. 5: Raw daily data plot

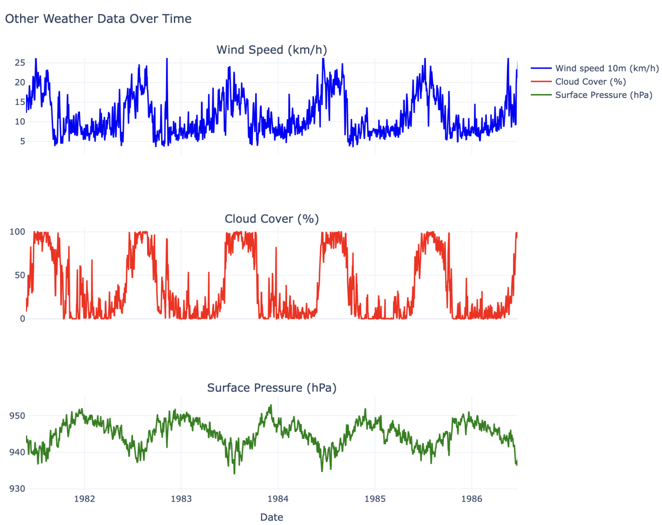

Fig. 6: wind speed, cloud cover and surface pressure EDA

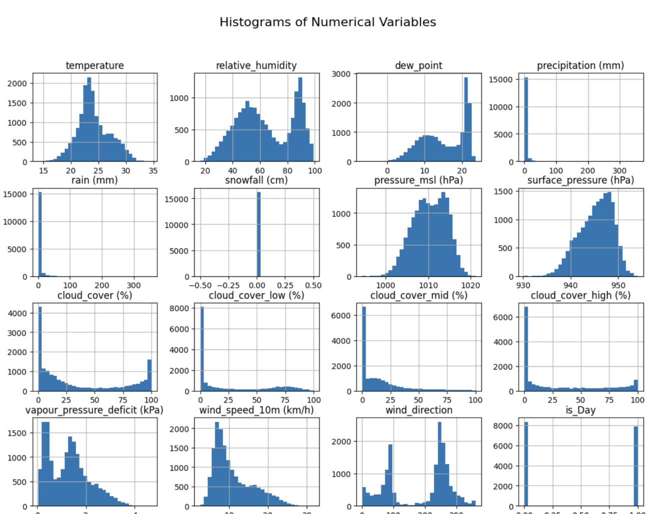

Fig. 7: Histogram

no snowfall, so it is just a bar. Vapor pressure and wind speed are right-skewed data.

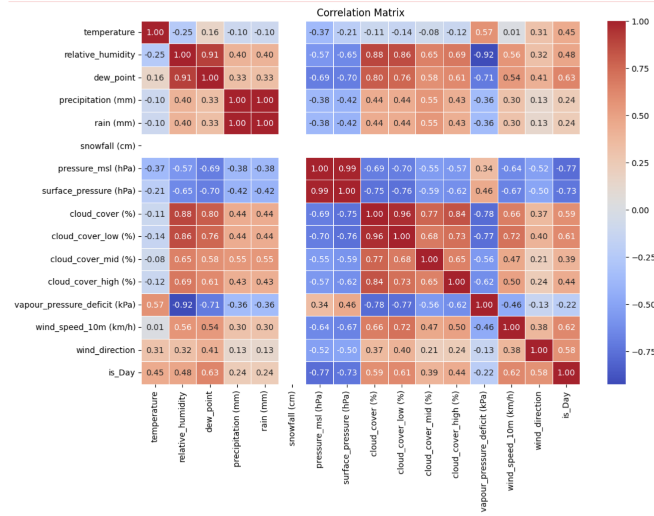

Fig. 8: Correlation matrix

B. Seasonal-Trend decomposition using LOESS

The amplitude of seasonality increased and decreased in years around 1995 and 2003 The result of the STL decomposition is as follows:

Fig. 9: STL-decomposition

Three different time series are separated from the original rainfall data namely the trend, seasonal and the remainder part. The last three years of data is kept seperate for the testing purposes.

C. Gated Recurrent Unit

The model visualization can be seen in the Figure .10 The training is done for 40 epochs. Past three values of the trend was given as input to the model and it was able to predict the next trend value. Training was done on different number of time steps and 3 was found to be the best.

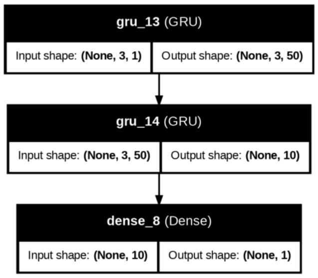

Fig. 10: GRU model

The model performed really well on the test data and result is as follows :

Metric	Value
R^2 Score	0.9982
RMSE	0.0022
MAE	0.0018

TABLE I: Performance Metrics of the Model

we can compare the combined model result with a conventional multi-timescale GRU model with input of past rainfall along with other data as baseline model.

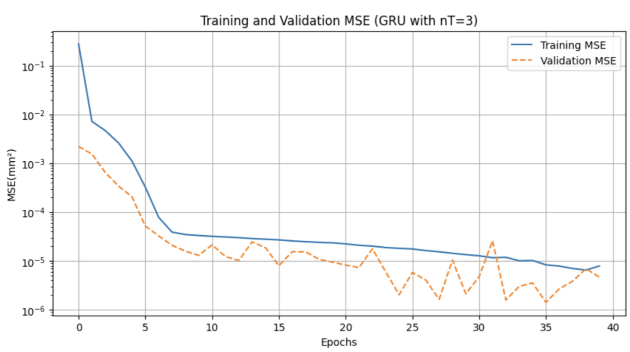

Fig. 11: Trend component Training loss and Validation loss

Fig. 12: Tend prediction

D. Multi time-scale GRU

The autocorrelation plots were created for finding the best seasonal lags for the data. The ACF and PACF plots can be seen in Fig 13 and Fig 14. Three years of previous lags showed a good correlation. The model took past 4 seasonal values and same day of last three years as input and predicted the next days seasonality component. Behind the scenes grid search was used to find the optimal parameters. The training took for the 100 epochs with early stopping and best epoch was 58.we can see it in fig 16

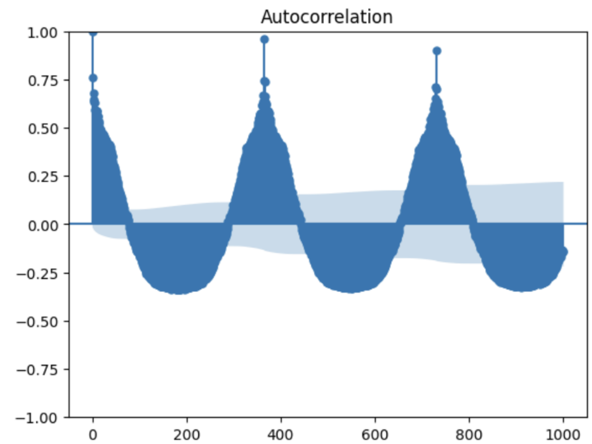

Fig. 13: Seasonal ACF Plot

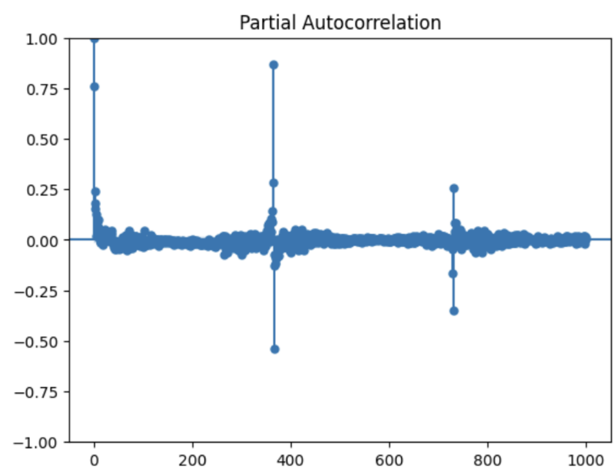

Fig. 14: Seasonal PACF plot

We can clearly see that the 98% of the variance of the data was explained by the model and the prediction were on and off by 0.28mm. The predictions were so far good but with the performance metrics as follows II:

Metric	Value
R^2	0.9897
RMSE	0.5049 mm
MAE	0.2858 mm

TABLE II: Performance Metrics for the Seasonal Model

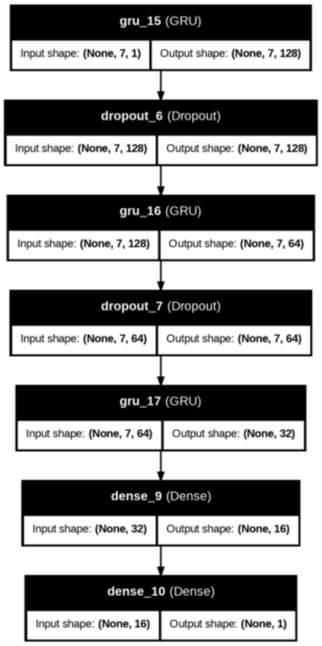

Fig. 15: Multi timse scale GRU model architecture

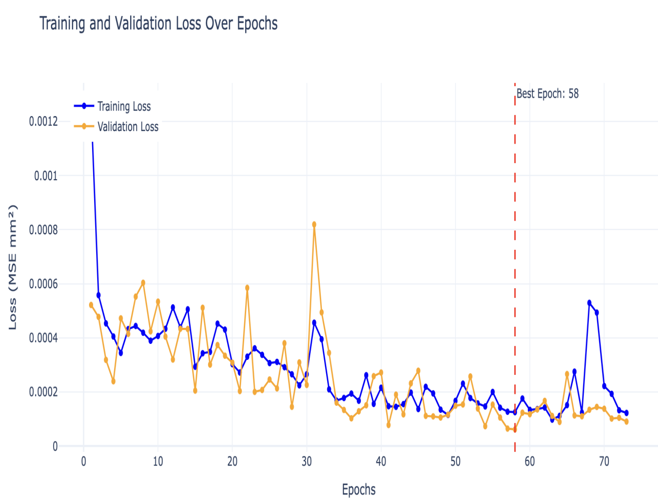

Fig. 16: Seasonal component Validation and training loss

We can see the three years predicted seasonal component and the actual component in this plot17

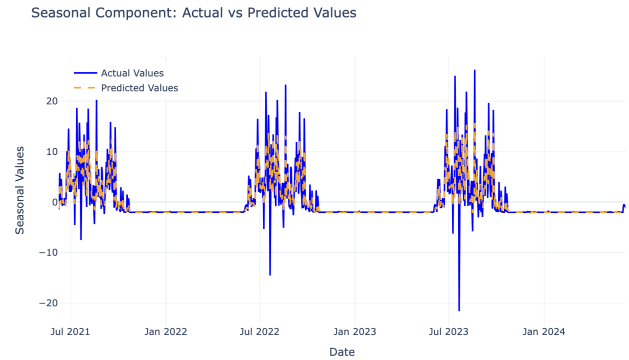

Fig. 17: Seasonal Component Prediction

E. LightGBM

The highly correlated features were taken out from the series, and the remaining features lagged versions were passed to it along with the remainder of the highest ACF value lagged data. The plot for the best features is as shown in table The performance metrics is as follows in tableIII:

Grid search was used to find the best parameters for model it can be seen as follows:18

Parameter tuning for the LightGBM model.

Parameter	Candidate values	Optimal result
Maximal number of leaves in one tree	32, 64, 128, 256	128
Minimal data number in one leaf	8, 16, 32, 64	16
Learning rate	0.05, 0.1, 0.15, 0.2	0.1

Fig. 18: Best Parameters for the model

Metric	Value
R^2	0.6790
RMSE	3.4081 mm
MAE	1.5664 mm

TABLE III: Remainder Performance Metrics for the Model

The prediction results of the remainder component are as follows in fig19

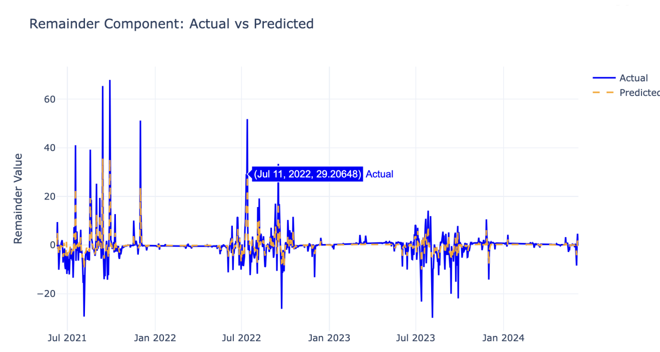

Fig. 19: Remainder Prediction and actual

We can clearly see that the remainder model is performing poorly on the datapeaks which is explained by its low r2 score.

F. Combined Model Predictions

The predictions of three models were added with their index to get the final rainfall prediction here.

Model	R^2	RMSE (mm)	MAE (mm)
GRU	0.9998	0.0077	0.0061
Multi-timescale GRU	0.9853	0.6036	0.2895
LightGBM	0.6790	3.4081	1.5664
Combined Model	0.7902	3.3351	1.3846
Baseline GRU	0.1618	6.6703	2.2121

TABLE IV: Performance Metrics for Different Models

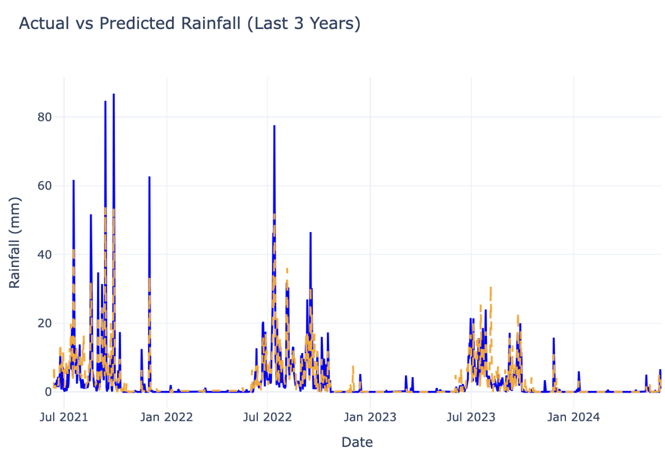

Fig. 20: Rainfall Prediction Vs Actual

The weekly random plot gives a deep explaination about the predictions we can see that its able to predict the normal rainy days with minimal errors but it struggles to accurately predict the sudden rainy day although the likelihood event of rainfall is indicated here correctly.

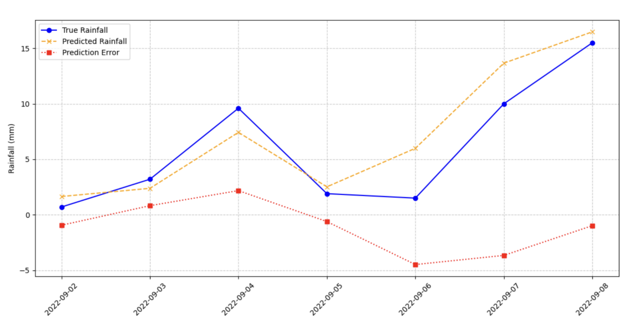

Fig. 21: Random week prediction

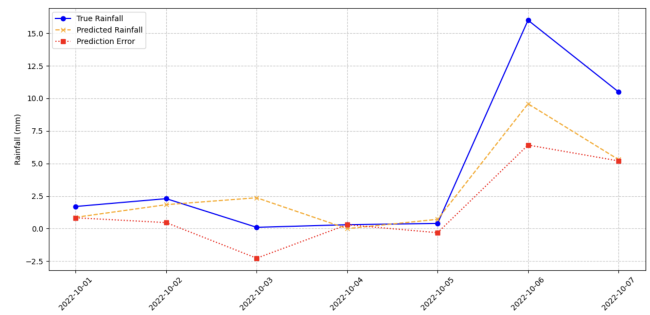

Fig. 22: Random week prediction

IV. Discussion

We can clearly see that the rainfall prediction using hybrid approach is a better than the conventional approach. The results showed the combined model was able to explain the 79% of varaince in the data. The future directions for this research work are to use hourly data for the study and to use the multi-time-scale CONV1D-GRU network for the remainder series prediction. The idea is to work in the direction of trying to predict the remainder series as it shows unpredictable meterological events like storm or cyclone.

V. Conclusion

This study revealed unlikely events of heavy rainfall due to cyclone or storm are still less likely to be predictable using this approach.

- The trend model performed really good for the given data and explained 99% variance of the data which is a good indication
- The seasonal model performed moderately but it still underestimates the peak seasonality in the data.
- The remainder model performed poorly as expected. The reason being it tries to predict the noise in the data.
- Though the overall model performed better than the baseline model explaining the 79% of the data variance with rmse of 3.3mm which is a pretty good thing for the given data scenario.

References

[1] R. He, L. Zhang, and A. W. Z. Chew, "Modeling and predicting rainfall time series using seasonal-trend decomposition and machine learning," *Knowledge-Based Systems*, vol. 251, p. 109125, 2022.

[2] P. Dande, "Time series weather dataset," https://www.kaggle.com/datasets/parthdande/timeseries-weather-dataset, 2024, accessed: 01-Mar-2025.

[3] D. Chen, J. Zhang, and S. Jiang, "Forecasting the short-term metro ridership with seasonal and trend decomposition using loess and lstm neural networks," *IEEE Access*, vol. 8, pp. 91 181–91 187, 2020.

[4] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep learning with gated recurrent unit networks for financial sequence predictions," *Procedia Computer Science*, vol. 131, pp. 895–903, 2018, recent Advancement in Information and Communication Technology:. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918306781

[5] D. S. Moirangthem and M. Lee, "Hierarchical and lateral multiple timescales gated recurrent units with pre-trained encoder for long text classification," *Expert Systems with Applications*, vol. 165, p. 113898, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742030693X