

**FLOOD FORECASTING IN THE KUSHIYARA RIVER USING
SUPERVISED MACHINE LEARNING**

Alvee Bin Hannan

Student ID: 1616016



**DEPARTMENT OF WATER RESOURCES ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA**

MAY, 2022

MAY

2022

**FLOOD FORECASTING IN THE KUSHIYARA RIVER USING
SUPERVISED MACHINE LEARNING**

B.Sc. Engineering Thesis
(Course no. WRE 400: Project and Thesis)

By

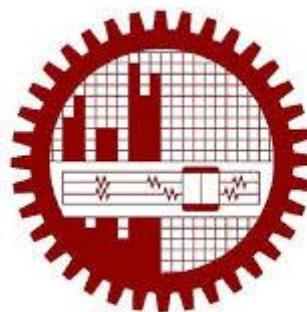
Alvee Bin Hannan

Under the Supervision of

Dr. Nasreen Jahan, Professor

In partial fulfilment of the requirement for the degree of

BACHELOR OF SCIENCE IN WATER RESOURCES ENGINEERING



DEPARTMENT OF WATER RESOURCES ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND
TECHNOLOGY

May 2020

Declaration

I, Alvee Bin Hannan, hereby declare that, the work presented in this thesis entitled “Use of multiple machine learning and deep learning algorithms in forecasting floods in the Kushiyara River”, being submitted to department of Water Resources Engineering, BUET for degree of Bachelor of Science, is the outcome of the original work done by me under the supervision of Dr. Nasreen Jahan, Professor, Department of Water Resources Engineering, BUET. Neither this thesis nor any part of it has been submitted anywhere for the award of any degree, diploma or other similar title or recognition from this or any other institution.

.....
(Name of the candidate)

Alvee Bin Hannan
Student ID: 1616016

.....
(Name of the supervisor)

Dr. Nasreen Jahan
Professor

Department of Water Resources Engineering,
Bangladesh University of Engineering and Technology
Dhaka-1000

Acknowledgement

First, I acknowledge the blessing of Almighty Allah who has provided me with the opportunity to complete this project.

I would like to express my heartiest gratitude to my thesis supervisor Dr. Nasreen Jahan, Professor of Department of Water Resources Engineering, Bangladesh University of Engineering and Technology, for her helpful suggestion, comments and continuous support and encouragement throughout this thesis work.

I gratefully acknowledge the librarians and administrative staffs of Department of Water Resources engineering for their kind help during this thesis work.

I would like to express my sincere gratitude to Sumit Kumar Saha, Siam Maksud and Maesha Mushsharat Aadrita who encouraged me to complete the research work directly.

I would also like to express my gratitude towards Afiya Narzis for her helpful instructions regarding my thesis.

At last, I want to acknowledge the immense support of my parents and other members of family for supporting me mentally to complete this thesis.

Alvee Bin Hannan

May, 2022

Table of Contents

Declaration	i
Acknowledgement	ii
Table of Contents	iii
List of Figures	v
List of Tables	vi
List of Abbreviations	vii
Abstract	ix
Chapter 1 Introduction	1
1.1 General	1
1.2 Objective of Study:	3
1.3 Organization of the thesis	3
1.4 Description of Study area.....	3
Chapter 2 Literature review	5
2.1 River system of Bangladesh.....	5
2.2 Floods in Bangladesh.....	5
2.3 Review of past studies on Flood forecasting in Bangladesh.....	6
2.4 Review of past studies on Flood forecasting using machine learning	8
Chapter 3 Description of models	11
3.1 General	11
3.2 Random Forest Regressor	11
3.2.1 Hyperparameter Tuning	12
3.3 Catboost Regressor	13
3.4 Deep Neural Network (DNN)	13
3.4.1 Activation Function	14
3.4.2 Loss Function.....	14

3.4.3 Optimizers:.....	15
3.5 Long Short-Term Memory (LSTM):	16
Chapter 4 Methodology	18
4.1 General	18
4.2 Data collection	18
4.2.1 ERA5 dataset	19
4.2.2 Water Level data (WL)	21
4.3 Watershed delineation:.....	23
4.4 Data extraction and pre-processing.....	23
4.5 Construction of the models	24
4.6 Model training and testing	25
4.7 Evaluation of model performance.....	25
Chapter 5 Results and Discussions	27
5.1 Evaluation of model performance:.....	27
5.2 Evaluation of model performance for recent flood events:	34
Chapter 6 Conclusion and recommendation	37
6.1 Conclusion	37
6.2 Recommendation	37
References.....	38
Appendix A	42
Appendix B	66

List of Figures

Figure 1.1 Map Showing Bangladesh and the Study area	4
Figure 3.1:Architecture of a typical Random Forest regressor model.....	11
Figure 3.2: Hyperparameter tuning using GridsearchCV	12
Figure 3.3 Architecture of the DNN model used to create forecast for 1-day using all variables as input.....	16
Figure 3.4: Architecture of the LSTM model used to forecast WL for day t with all variables as input.	17
Figure 4.1 Flowchart showing methodology of the study	18
Figure 4.2 Grid locations for ERA5 data collection	22
Figure 5.1 RMSE values for different models with different combinations of input features for different lead times	30
Figure 5.2 Coefficient of determination (R^2) values for different models with different combinations of input features for different lead times	31
Figure 5.3 Predicted WL (m) vs Observed WL (m) for train data set with input combination 4 for different lead times.....	32
Figure 5.4 Predicted WL (m) vs Observed WL (m) for validation data set with input combination 4 for different lead times.....	32
Figure 5.5 Predicted WL (m) vs Observed WL for test data set with input combination 4 for different lead times.....	33
Figure 5.6 Comparison of water levels using DNN with combination 4 for different lead times.....	33
Figure 5.7 Comparison water levels of actual flash flood events with simulated water levels (Input combination 4, Lead time 3-days)	36
Figure 5.8 Comparison water levels of actual monsoon flood events with simulated water levels (Input combination 4, Lead time 3-days)	36

List of Tables

Table 4.1 Summary of data collection	21
Table 4.2 Summary of flood events that took place from 2017 to 2020	22
Table 4.3 Hyperparameters used for model training	24
Table 4.4 Combinations of input variables	25
Table 5.1: Model evaluation based on RMSE and R ² scores.....	29
Table 5.2: Evaluation of models for recent Monsson flood events based on RMSE values	34
Table 5.3 Evaluation of models for recent Flash Flood events based on RMSE values	35

List of Abbreviations

Adaptive moment estimation	
(Adam)	17
Artificial Neural Networks	
(ANN)	2
Bangladesh Water Development Board	
(BWDB)	1
Categorical Boosting	
(CB)	2
Climate Data Operator	
(CDO)	22
coefficient of determination.	
(R²)	25
Copernicus climate data store	
(cds)	21
deep learning	
(DL)	ix
deep neural network	
(DNN)	2
digital elevation model	
(DEM)	20
fifth generation European Centre for Medium-Range Weather Forecasts Reanalysis	
(ERA5)	2
Flood Forecasting and Warning Center	
(FFWC)	2
Ganges-Brahmaputra-Meghna	
(GBM)	1
Gradient Boosted Decision Tree	
(GBDT)	14
Integrated Forecasting System	
(IFS)	10
Long Short-Term Memory	

(LSTM).....	2
machine learning	
(ML)	2
mean squared error function	
(MSE)	16
Multilayer Perceptron	
(MLP).....	5
Random Forest Regressor	
(RF).....	2
Rectified Linear Unit	
(ReLU).....	8
Recurrent Neural Network	
(RNN)	2
root mean square error	
(RMSE)	25
total column rainwater	
(TCRW)	2
total column water vapour	
(TCWV)	2
total precipitation	
(TP).....	2
volumetric soil water	
(SWVL).....	2
water level	
(WL)	ix

Abstract

The northeastern region of Bangladesh is highly susceptible to flooding almost every year mainly due to its geographical location. Heavy flooding in this region causes immense damages to lives, property and mostly, agricultural crops. This region suffers from both flash flood and monsoon flood. However, the lack of rainfall information from the upstream catchment areas outside of Bangladesh makes flood forecasting in this region difficult. In addition, inaccuracies in rainfall forecasts and a lack of high-resolution bathymetry and topography data limit the applicability of hydrologic and hydrodynamic models to forecast floods in this region. Currently, the Flood Forecasting and Warning Centre (FFWC) of Bangladesh Water Development Board (BWDB) is producing short-range deterministic flood forecasts with a lead time of up to three days. However, medium-range (3 to 7 days) forecasts are critical for minimizing flood-related losses because they allow greater time for decision-making and planning. With a view to addressing these issues, four machine learning (ML) algorithms have been chosen in this study for flood forecasting in the Kushiyara river, which is one of the major rivers of the northeastern region of Bangladesh. Random Forest Regressor (RF), Categorical boosting regressor (CB), Deep Neural Network (DNN) and Long Short-Term Memory neural network (LSTM) have been chosen as the ML algorithms. Daily total precipitation (TP), volumetric soil water up to 100cm depth (SWVL), total precipitable water (TCRW), and total column water vapor (TCWV) from the fifth generation European Centre for Medium-Range Weather Forecasts Reanalysis (ERA5) has been used as inputs to the model in different combinations with historical water level (WL) data collected from BWDB. The models have been trained and tested using data from 2007 to 2020 to forecast floods with a lead time of 1, 3, 5 and 7 days, taking four different combinations of the input variables. When all the variables are taken as input together, the RMSE test scores for a forecast with a lead time of 1 day are found to be 0.31m, 0.38m, 0.52m and 0.39m for DNN, RF, CB and LSTM respectively. For a 7-day lead time, the values are 1.09m, 1.35m, 1.29m and 1.29m respectively for DNN, RF, CB and LSTM. The R^2 values for the models are found to be 0.90, 0.84, 0.86, and 0.86, respectively, for DNN, RF, CB and LSTM for 7-day lead time. The study finds that the deep learning algorithms perform with better consistency regardless of lead times and input combinations. The LSTM models show more consistent results for a

shorter lead time (1-day and 3-day) with only precipitation as input. However, for multiple variables, and when the lead time increases, the DNN models show a better overall performance considering training, validation and testing scenarios. The study suggests that despite the scarcity of rainfall data from the upstream areas, using ERA5 products in conjunction with deep learning algorithms may allow us to forecast and monitor high-volume flood occurrences in this region.

Chapter 1

Introduction

1.1 General

Bangladesh is highly susceptible to flooding almost every year, having 405 rivers flowing through it (Bangladesh Water Development Board (BWDB), 2022). Of these rivers, 57 are transboundary, meaning that flooding in these rivers occurs due to precipitation outside the administrative boundary of Bangladesh. The prime contributor to flow in these rivers is the Ganges-Brahmaputra-Meghna (GBM) basin, which consists of a whopping area of over 1.7 million sq. km. Only 7 per cent of this huge area lies within the administrative boundaries of Bangladesh, the remaining is distributed among India (64 per cent), China (18 per cent), Nepal (9 per cent) and Bhutan (3 per cent) (FAO AQUASTAT, 2011). The smallest of the three basins, the Meghna basin, drains an area of 82,000 sq. km. spanning over Assam, Manipur, Nagaland, Meghalaya of India and the northeastern part of Bangladesh. The average annual rainfall in the Meghna basin is around 5800mm, making it one of the雨iest regions of the world (Japan International Cooperation Agency, 2011). Heavy rainfall in this region causes devastating floods in the rivers downstream.

Floods in Bangladesh can be categorized into a few groups depending on their nature of origin: Monsoon floods, tidal floods, urban floods and flash floods. Of these, flash floods cause the most damage every year. The topography of the northeastern part of Bangladesh makes it vulnerable to these flash flood events. Heavy rainfall in the upstream Indian states of Meghalaya, Manipur, Assam and Nagaland causes water to rush into the rivers of Sylhet, Sunamganj, Habiganj, Netrakona of Bangladesh and flood the lowlands of these regions. The entire process often takes only a few hours to a few days.

But Flood forecasting in this region is a challenging task owing to the lack of proper rainfall data collection in the upstream regions. Thus, the forecasts that are prepared for this area are often reliant on rainfall forecast data. But the rainfall forecasts may contain inaccuracies, which often results in erroneous forecasts. Besides, the lack of high-resolution bathymetry and topographic data also puts major barriers to the preparation of accurate flood forecasts using hydrologic and hydrodynamic models. Currently, the

Flood Forecasting and Warning Center (FFWC) of the Bangladesh Water Development Board (BWDB) is preparing short term (1-3 days) deterministic forecasts and long term (10-15 days) probabilistic forecasts for the monsoon period. However, flash flood events are very difficult to predict, as often these floods take only a few hours to reach.

This study attempts to tackle both these issues. The objective of this study is to create a medium-range (3-7 days) forecast for the Kushiyara river situated in the northeastern region of Bangladesh. The use of rainfall forecast data was avoided to prevent errors owing to erroneous rainfall forecasts. Instead, historical daily total precipitation (TP) data was obtained from the fifth generation European Centre for Medium-Range Weather Forecasts Reanalysis (ERA5) along with other variables taken from the same source to be used in preparing a flood forecast at 1,3,5,7-day lead times. The other variables used were namely volumetric soil water up to 100cm depth, total column rainwater (TCRW), and total column water vapour (TCWV). Historical water level data recorded at the Sheola gauging station by the BWDB was also included as an input variable, owing to its auto-correlative property. The use of TCRW and TCWV helps in predicting flash flood situations with reasonable accuracy.

In recent decades, machine learning (ML) has been applied efficiently in hydrology. The popularity of ML and Artificial Neural Networks (ANNs) in hydrology can be explained by their ability to mimic highly non-linear complex problems and are now widely used in the modelling of complex hydrologic systems (Dawson & Wilby, 2001). The use of ML is seen to perform better than traditional data-driven or hydrologic models in some cases. (Mosaffa et al., 2022). In this study, four separate models have been developed. Two of them are well known supervised ML algorithms, namely Random Forest Regressor (RF), and Categorical Boosting (CB). The remaining two models are developed on deep learning algorithms. Firstly, a feed-forward deep neural network (DNN) with six dense layers has been trained to forecast water levels. Then Long Short-Term Memory (LSTM) based on Recurrent Neural Network (RNN) has been used to account for the time sequence of data during training (Kao et al., 2020). This study attempts to find the best model for flood forecasting in the region with available data.

1.2 Objective of Study:

The specific objectives of this study are:

- i. To develop a flood forecast model for the Kushiyara river based on supervised machine learning algorithms.
- ii. To evaluate the usefulness of machine learning techniques in flood forecasting by comparing the simulated data against the observed data.
- iii. To compare the performances of different algorithms in flood forecasting during monsoon and flash flood seasons.

1.3 Organization of the thesis

This thesis is divided into 6 chapters. The first chapter is an overall introduction that discusses the objectives of the study and description of the study area. The second chapter titled literature review discusses the river systems and floods of Bangladesh, and reviews past studies conducted on flood forecasting and machine learning. The third chapter describes the individual Machine learning models that have been used in the study; their governing equations, architecture and hyperparameters. The fourth chapter titled Methodology discusses the workflow of this study in chronological order starting from data collection and pre-processing, watershed delineation, to model construction and evaluation. The fifth chapter discusses the results of the models and their performance on actual flood events. The final chapter includes conclusion on this study and recommendation for future studies on this topic.

1.4 Description of Study area

The Kushiyara river originates near Karimganj, where the Barak River bifurcates into a northern branch, namely Surma, and the southern branch being Kushiyara. The Barak River rises on the southern slope of the steep Barail Range at the Manipur-Nagaland border and forms part of Manipur's northern border with Nagaland, where it is known as Kirong. It runs westerly and southerly from there to Tipaimukh, where it abruptly turns north and forms the border between Assam's Cachar district and Manipur for a substantial distance. After there, it swings west near Jirimukh and sluggishly travels over the Cachar plain, then it splits into the Surma and the Kushiyara near Karimganj. The river, which runs for 900 kilometers from source to mouth, drains a total area of 52,000 square kilometers.(Afiya Narzis, 2020)

The study area is a significant portion of the Barak Basin. It comprises an area of 26,079 km², of which only 38 km² lie within the borders of Bangladesh. The outlet for the basin has been taken at the BWDB gauge station at Sheola (SW173), located at Beanibazar, Sylhet. The coordinates of the station are 24.8873°N, 92.19°E. [fig 1.1]. Both monsoon floods and flash floods hit this station almost every year due to heavy rainfall and the topography of the basin. The average annual rainfall in this region is 3000mm. Most of this rainfall occurs in the higher elevation areas in the mountains of Assam and Manipur, which takes little time to rush downstream. Such events are called flash floods. This generally takes place between Mid-April to Mid-June. Monsoon floods, on the other hand, occur as a result of heavy monsoon rains, in July-September.

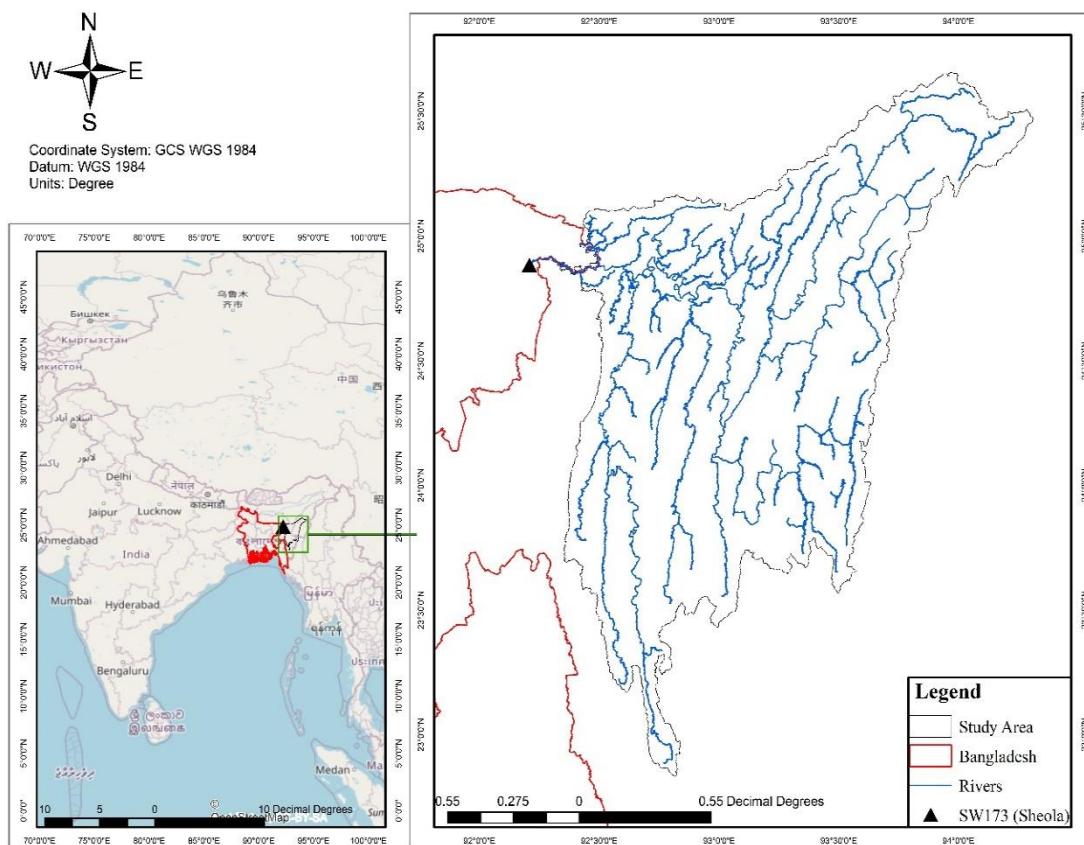


Figure 1.1 Map Showing Bangladesh and the Study area

Chapter 2

Literature review

2.1 River system of Bangladesh

Bangladesh has about 808 or more crisscrossing rivers all over the country, most of which are either tributaries or distributaries to the Ganges-Brahmaputra-Meghna (GBM) river systems (Huq, 2008). Bangladesh is located at the downstream area and covers roughly 7% area of the drainage catchments of GBM river systems. The combined discharge of the three main rivers is among the highest in the world. Peak discharges are of the order of 100 000 m³ /s in the Brahmaputra, 75 000 m³ /s in the Ganges, 20 000 m³ /s in the upper Meghna and 160 000 m³ /s in the lower Meghna. On an average, about 1,106 km³ of water crosses the borders of Bangladesh annually, of which 85% between June and October, around 54% (599 km³) is contributed by the Brahmaputra, 31% (344 km³) by the Ganges and nearly 15% (163 km³) by the tributaries of the Meghna and other minor rivers. Bangladesh has about of 58 major transboundary rivers shared with neighboring countries, India and Myanmar (Banglapedia, 2009). These transboundary rivers are very significant for cross-boundary water flow and to maintain water resources systems of Bangladesh. North-East region receives stream flow from Brahmaputra River through distributaries Climate Change and Water Resources Page 4 of 22 including the Old Brahmaputra, the Dhaleswari and Lohajang River and water also comes directly from Meghalaya, Assam and Tripura of India through the Barak (Surma, Kushyiara), Juri, Manu, Khowai and other rivers. Inflows from India into the North-West region include water flow from West Bengal, India in the rivers Teesta, Mohananda, Dharla and Dudkumar; in North-East region. The South-East region receives stream flow from the Tripura and Mizoram states of India mainly through the Karnafuli, Sangu, Mathamuhuri, Muhuri, Gumti and Feni rivers, while only Mathabhangha River enters the South West region. No major transboundary inflow occurs in the North-Central and South-Central regions (Huq, 2008).

2.2 Floods in Bangladesh

Flood is the most universal of the natural hazards and most of the remarkable floods of the globe are associated with the world's great rivers. Generally, flood can be defined

as an overflow of a large amount of water beyond its normal limits, especially over the land that is normally dry. Flooding is the natural characteristics of the rivers. So flood also can be defined as comparatively high flow of water that overflow the natural or artificial banks in any of the reaches of a stream. When the banks of the rivers are overtapped, water spreads over the floodplain and generally causes huge problems for crops, vegetation and inhabitants. In Bangladesh, the definition of flood is little bit different. In the rainy season when the water flow exceeds the holding capacity of rivers and low-lying area sit swamps the whole area causing damage to vegetables, crops, home, roads, bridge and other properties. In Bangladesh, flood can be categorized as Monsoon flood, Flash flood and Tidal/Coastal flood.

Monsoon floods are seasonal, increases slowly and decreases slowly, inundates vast areas and causes huge losses to life and property. In case of Flash floods water increases and decreases suddenly, generally happen in the valleys of the hilly areas. Coastal flood last for short duration, height is generally 3m to 6m, blocks inland flood drainage. (Baten et. al., 2018)

2.3 Review of past studies on Flood forecasting in Bangladesh

A three-year project titled “Expansion of Flood Forecasting and Warning Services (1997)” with the objective to aid national preparedness for floods and to mitigate flood impacts commenced in January 1995 with the assistance from Danida.(Institute of water modelling, 2019) Development of a pilot forecast system for two flashy rivers was one of the six important outputs of the project. The study tried to develop flash flood forecasting of the Manu River system in the Northeast region. The upstream boundaries of the pilot models were defined at stations close to Indo-Bangladesh border within Bangladesh i.e., Manu Railway Bridge in the Manu River. The Manu Railway Bridge is located some eight kilometres within Bangladesh from the Indo-Bangladesh border. The Manu, and subsequently Moulvi Bazar city, experience flash floods due to rainfall across the border and by rainfall within the Manu catchment itself. More than 80% of the Manu catchment is located in the mountainous Tripura region of India for which no rainfall/flow data is available. Due to the hydrological features of the river and its catchment, only very short lead times can be obtained, i.e., of the order of a few hours. At the time of the study, continuously measured rainfall and water level data were available via telemetry stations at Manu Railway Bridge and Sherpur. In order to provide forecast boundary conditions for the model an approach based on the use of

Artificial Neural Network (ANN) was adopted. It is based on recognition of patterns, in this case rate of rise patterns of water levels at the two gauging stations related to the observed flood intensity. The Multilayer Perceptron (MLP) algorithm was implemented in this study to forecast river water levels up to a 24-hour time horizon with an input of the past 24 hours. The inputs were presented to the ANN as hourly values (FFWC, 2005). The key findings of the study are that MIKE11 performs well enough to simulate flash flood, One of the main difficulties in flash flood forecasting is the estimation of boundaries, i.e., upstream inflows and rainfall over the catchment, Quantitative estimation of rainfall from remote sensing data is invaluable to improve estimation of flash flood boundaries and Decision support tools based on statistics or artificial neural networks (ANN) need to be developed to aid boundary estimation.

ADPC has carried out the study “Flood Forecast Technology for Disaster Preparedness in Bangladesh” under financial assistance of USAID during 2008 to develop flash flood forecast technology in Northeast Bangladesh. In order to improve the accuracy of flash flood forecasting using the ADPC rainfall forecast, this project has developed hydrological models of four cross-border catchments: the Jadukata river, the Khowai river, the Manu River and the Barak River catchments in the North-East Region of Bangladesh. Figure 3-2 shows study catchments. The hydrological models of the four catchments have been developed using MIKE11 and SWAT modelling software and found that performance of model developed based on MIKE 11 is better than the performance of the SWAT model. The major findings of the study were that The rainfall forecast produced by ADPC do not sufficiently correspond to the sudden localized intense rainfall that occur in the cross-border catchments. There is no rainfall recording station within the Jadukata river and Khowai river catchments, and inadequate number of rainfalls recording stations within the Manu River and the Barak River catchments; Except the Barak River, the basin lag time of other catchments is small, shorter than a day. Thus, daily rainfall records are not adequate for fine calibration of the models.

In a study that focuses on improving flood forecast in Bangladesh, a river stage neural network model has been developed to study and predict the water level of Dhaka city. the study was titled “Improving flood forecasting in Bangladesh using an artificial neural network (2010)’. A total of five stations located at the border area of Bangladesh on the Ganges, Brahmaputra and Meghna rivers are selected as input nodes and Dhaka on the Buriganga river is the output node for the neural network. This model is trained

with river stage data for a period of 1998 to 2004 and validated with data from 2005 to 2007. The river stage of Dhaka has been predicted for up to ten days with very high accuracy. Values of R^2 , root mean square and mean absolute error are found ranging from 0.537 to 0.968, 0.607 m to 0.206 m and 0.475 m to 0.154 m, respectively, during training and validation of the model. The results of this study can be useful for real-time flood forecasting by reducing computational time, improving water resources management and reducing the unnecessary cost of field data collection.(Islam, 2010)

2.4 Review of past studies on Flood forecasting using machine learning

(Piotrowski et al., 2006) used Multi-Layer Perceptron and Radial Basis Function Neural Networks, along with the Nearest Neighbour approach and linear regression for flash-flood forecasting in the mountainous Nysa Kłodzka River catchment. It turned out that the Radial-Basis Function Neural Network is the best model for 3- and 6-h lead time prediction and the only reliable one for 9-h lead time forecasting for the largest flood used as a test case. In the paper, Nysa Kłodzka river rainfall-runoff forecasts for flood periods are performed by means of several data analysis techniques. The obtained results clearly indicate that Radial-Basis Function Neural Networks outperform the other techniques for testing data.

(Sahoo et al., 2006) studied flash floods of Hawaii streams that pose continuous threats to the coastal environment because the streams respond rapidly to high runoff. High-frequency stream flow and water quality estimation are essential to correctly assess water quality variations and pollutant loads during flash floods, because stream flow and turbidity in Hawaii can change by a factor of 60 and 30, respectively, in 15 min. This study shows the application of artificial neural networks (ANNs) to assess flash floods and their attendant water quality parameters using measured data of a Hawaii stream. The paper illustrates that ANNs predict stream flow with a correlation coefficient (R) greater than 0.99 and turbidity and specific conductance with R -values greater than 0.80. Although the R -values for the estimation of dissolved oxygen, pH, and water temperature were somewhat low, most of the estimated stream water quality values (turbidity, specific conductance, dissolved oxygen, pH, and water temperature) were within the limits of $\pm 30\%$ deviations of the 1:1 line. The R -value for the estimation of stream water qualities could have been significantly improved if high resolution (at 15 min or lower measurement frequency), noise-free, and continuous data were available for a longer period of time. The paper demonstrates that the upstream water

quality parameters depend on weather forces and land use of the watershed and the downstream water quality parameters additionally influenced by oceanic tides. Stream stage is found to be an important input parameter for stream flow prediction using ANN; however, the predictive performance of ANN for the estimation of stream flow is improved if weather data, rainfall, and evapotranspiration are included in the input data set.

(Tien Bui et al., 2020) proposes and evaluates a new approach for flash flood susceptibility mapping based on Deep Learning Neural Network (DLNN) algorithm, with a case study at a high frequency tropical storm area in the northwest mountainous region of Vietnam. Accordingly, a DLNN structure with 192 neurons in 3 hidden layers was proposed to construct an inference model that predicts different levels of susceptibility to flash flood. The Rectified Linear Unit (ReLU) and the sigmoid were selected as the activate function and the transfer function, respectively, whereas the Adaptive moment estimation (Adam) was used to update and optimize the weights of the DLNN. A database for the study area, which includes factors of elevation, slope, curvature, aspect, stream density, NDVI, soil type, lithology, and rainfall, was established to train and validate the proposed model. Feature selection was carried out for these factors using the Information gain ratio. The results show that the DLNN attains a good prediction accuracy with Classification Accuracy Rate = 92.05%, Positive Predictive Value = 94.55% and Negative Predictive Value = 89.55%. Compared to benchmarks, Multilayer Perceptron Neural Network and Support Vector Machine, the DLNN performs better; therefore, it could be concluded that the proposed hybridization of GIS and deep learning can be a promising tool to assist the government authorities and involving parties in flash flood mitigation and land-use planning.

(Kao et al., 2020) mentioned that operational flood control systems depend on reliable and accurate forecasts with a suitable lead time to take necessary actions against flooding. This study proposed a Long Short-Term Memory based Encoder-Decoder (LSTM-ED) model for multi-step-ahead flood forecasting for the first time. The Shihmen Reservoir catchment in Taiwan constituted the case study. A total of 12,216 hourly hydrological data collected from 23 typhoon events were allocated into three datasets for model training, validation, and testing. The input sequence of the model contained hourly reservoir inflows and rainfall data (traced back to the previous 8 h) of ten gauge stations, and the output sequence stepped into 1- up to 6-hour-ahead reservoir

inflow forecasts. A feed forward neural network-based Encoder-Decoder (FFNN-ED) model was established for comparison purposes. This study conducted model training a number of times with various initial weights to evaluate the accuracy, stability, and reliability of the constructed FFNN-ED and LSTM-ED models. The results demonstrated that both models, in general, could provide suitable multi-step ahead forecasts, and the proposed LSTM-ED model not only could effectively mimic the long-term dependence between rainfall and runoff sequences but also could make more reliable and accurate flood forecasts than the FFNN-ED model. Concerning the time delay between the time horizons of model inputs (rainfall) and model outputs (runoff), the impact assessment of this time-delay on model performance indicated that the LSTM-ED model achieved similar forecast performance when fed with antecedent rainfall either at a shorter horizon of 4 h in the past ($T - 4$) or at horizons longer than 7 h in the past ($> T - 7$). We conclude that the proposed LSTM-ED that translates and links the rainfall sequence with the runoff sequence can improve the reliability of flood forecasting and increase the interpretability of model internals.

Chapter 3

Description of models

3.1 General

Four machine learning models have been used in this study. They are random forest regressor, catboost regressor, deep neural network and long short-term memory neural network. A detailed description of these models can be found in this chapter.

3.2 Random Forest Regressor

A random forest (RF) is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It is an "ensemble learning" approach that involves aggregating a large number of decision trees to reduce variance as compared to single decision trees.(Couronné et al., 2018) Only a certain amount of randomly selected features is evaluated as candidates for splitting while creating each tree at each split. Due to the high number of trees, RF is typically thought of as a black-box method. (Probst P et al., 2018) In this study, the package RandomForestRegressor from scikit-learn ensemble library was used to train and test the model. The optimal hyperparameters for the model were selected with the help of the popular GridsearchCV module used for hyperparameter tuning. [Fig. 3.1] illustrates the architecture of the model.

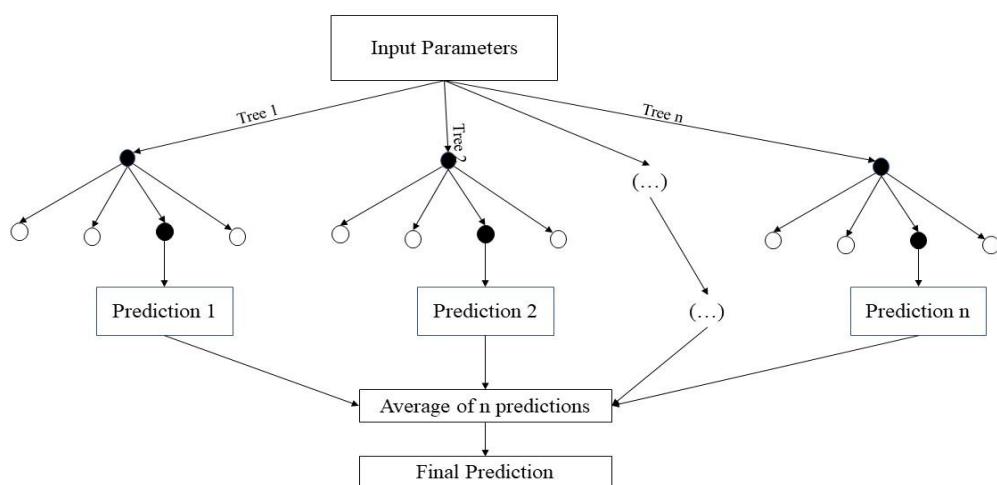


Figure 3.1: Architecture of a typical Random Forest regressor model

3.2.1 Hyperparameter Tuning

Hyperparameters contain the data that dictate the training process of ML models [26]. A judicious selection of tuning hyperparameters can minimize training model time and cost. In this study, hyperparameter tuning for the DL models was performed manually and that for the ML models was done using the popular GridsearchCV library.

3.2.1.1 GridSearchCV

GridsearchCV is a conventional technique of hyperparameters optimization that simply does a thorough search across a given portion of the training algorithm's hyperparameters space [fig 4.2]. As the parameter space for some parameters in a machine learning method may comprise spaces with actual or limitless values, it is usually required to set a boundary to do a grid search. Grid search suffers from large dimensional spaces, although it can typically be readily parallelized because the hyperparameter values used by the algorithm are frequently independent of one another (Theodoridis, 2020) The performance of RF is considered to be relatively resistant to parameter specifications, with performance relying on parameter values less than those of other machine learning methods. (Probst P et al., 2018) In rare instances, though, considerable improvements may be made. (Probst P et al., n.d.)

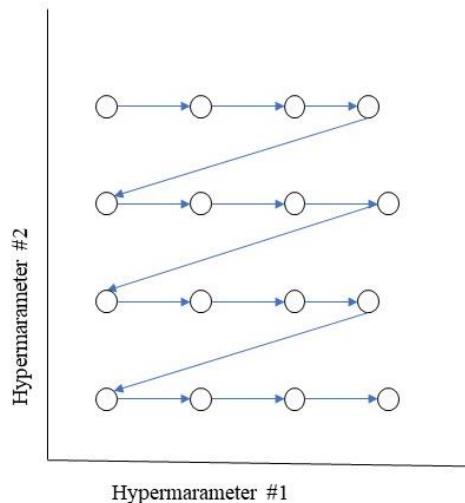


Figure 3.2: Hyperparameter tuning using GridsearchCV

3.3 Catboost Regressor

CatBoost is an open-source, Gradient Boosted Decision Tree (GBDT) implementation for Supervised Machine Learning (ML). (Hancock & Khoshgoftaar, 2020) Gradient boosting is a strong machine-learning approach that produces cutting-edge outcomes in a range of real-world applications. It has long been the preferred approach for learning problems with heterogeneous characteristics, noisy data, and complicated dependencies: online search, recommendation systems, weather forecasting, and many more. (Hancock & Khoshgoftaar, 2020; R. Caruana & A. Niculescu-Mizil, 2006; Roe et al., 2005) It is simply a procedure of generating an ensemble predictor in a functional space using gradient descent. (Prokhorenkova et al., 2017) Prokhorenkova et. al. in their study brought forward two main issues of existing gradient boosting implementations. One of them was a prediction shift in the learned model, the other was a target leakage and a prediction shift in preprocessing categorical features when they are converted to their target statistics.(Prokhorenkova et al., 2017). Prokhorenkova et. al came up with a solution to solve both the problems. They combined a modification of standard gradient boosting algorithm, namely ordered boosting, along with a novel algorithm for categorical feature processing, into an open-source library termed Categorical Boosting (or Catboost for short). They showed that this surpasses the existing state-of-the-art gradient boosted decision tree implementations — XGBoost and LightGBM — on a wide range of prominent machine learning tasks. (Prokhorenkova et al., 2017) The algorithm was chosen for this study because of its capability in handling regression problems equally as compared to classification problems.

3.4 Deep Neural Network (DNN)

Deep neural networks with multiple parameters are extremely powerful machine learning systems (Srivastava et al., 2014). They usually contain more than one non-linear hidden layer along with one input and an output layer (Kim, P.,2017). This makes them capable of learning very complex correlations between their inputs and outputs. The DNN developed for this study includes an input layer with n input nodes, and 5 dense layers (Gelenbe & Yin, 2018). with 64 nodes per layer, each followed by a dropout (Srivastava et al., 2014) of 0.1. ‘n’ is the number of input variables that are chosen for the models. The output layer that contains the forecast values, has one node. The activation function chosen for the DNN models is Rectified Linear Unit (ReLU)

function. The loss function chosen was mean squared error function (MSE). [Fig. 3.4] illustrates the architecture of the model.

3.4.1 Activation Function

Activation functions are mostly used to generate non-linear neural networks. They specify a range of values for the neuron's output. The output signal becomes a simple linear function when there is no activation function.(Nwankpa et al., 2018) One of the most common functions used in deep learning is the rectified linear unit (ReLU). This function has been selected for this study due to its reliability and speed of performance (Theodoridis, 2020). The function is quite simple (Liu, 2020) and can be expressed by the following equation:

$$ReLU(x) = \max(0, x)$$

An advantage of ReLU is that the deep neural networks can use this function to achieve sparse activation.(Liu, 2020) Following initialization, the weight can make approximately half of the output of hidden units equal to 0. The activated subset of neurons for a given input is single, and the computation of the subset is linear, resulting in the ReLU function being an activation function without the gradient disappearing (Liu, 2020). It is also sometimes a disadvantage as training stops when $x < 0$. Variants of the ReLU have been developed to solve this problem.(Theodoridis, 2020)

3.4.2 Loss Function

A loss function, also known as a cost function, is a function that converts an event or the values of one or more variables into a real number that intuitively represents some "cost" connected with the occurrence. (Raschka, 2019) The goal of loss functions is to compute the quantity that a model should try to minimize during training. (*Keras API Reference / Losses*, 2022) A loss function is minimized by an optimizer. (Raschka, 2019) Few of the most popular loss functions used in deep learning models are binary cross-entropy, AUC, categorical cross-entropy, mean squared error (MSE), etc. (*Keras API Reference / Losses*, 2022) This study makes use of the MSE as loss function for both DNNs and LSTM – NNs

$$MSE = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2$$

3.4.3 Optimizers:

Optimization is termed as the method of minimizing the errors generated by a loss function. (Raschka, 2019) Popular processes of optimization; i.e., optimizers are stochastic gradient descent (SGD), root mean square propagation (RMSProp), adaptive gradient algorithm (AdaGrad), Adaptive moment estimation (Adam) etc. This study makes use of the Adam optimizer.

Adam stands for Adaptive moment estimation. It is a first-order gradient-based stochastic objective function optimization technique based on adaptive estimations of lower-order moments. (Kingma & Ba, 2014) The approach is simple to develop, computationally efficient, requires minimal memory, is invariant to gradient diagonal rescaling, and is ideally suited for issues with huge amounts of data and/or parameters. (Kingma & Ba, 2014) The weight (w_t) updates are done as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

η is the step size/learning rate, around 1e-3 in the original paper. m and v are moving averages, g is gradient on current mini-batch. β_1 and β_2 are forgetting parameters, with typical values 0.9 and 0.999, respectively. (Kingma & Ba, 2014)

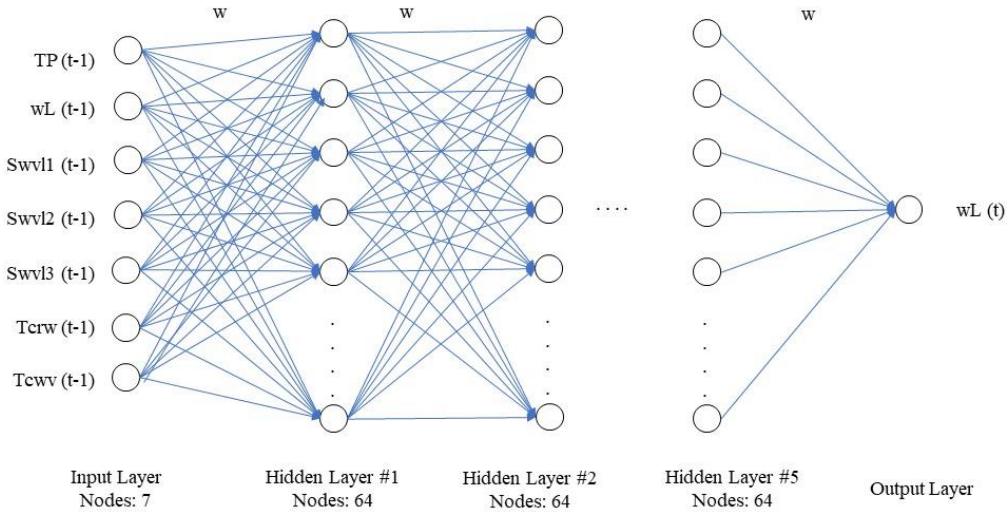


Figure 3.3 Architecture of the DNN model used to create forecast for 1-day using all variables as input. The list of abbreviations contains the explanation to the variable names.

3.5 Long Short-Term Memory (LSTM):

A limitation of DNN is that it sometimes fails to successfully handle time-series data since, after each iteration of processing data, information of the sequential order of the data is erased. (Kao et al., 2020). This is where Recurrent Neural Networks (RNN) play a crucial role in preserving the sequential information of the data when it handles time-series data. LSTMs are a special kind of DNN fitted to an RNN architecture (Kao et al., 2020) specifically processing time-series data. The LSTM unit looks back at a certain number of time steps in the past before it proceeds to prepare an output. (Gui et al., 2021) It was designed to overcome the limitation of RNNs that restricted them to looking back only approximately 10-time steps in the past. (Hochreiter & Schmidhuber, 1997) The use of LSTM is prominent in the preparation of rainfall-runoff models too. (Kratzert et al., 2018) This study uses a model with 3 dense LSTM layers with 64 neurons each, the ReLU activation function and the adam optimizer to forecast water levels at the required lead-time using a long short-term memory of 14-time steps. The loss function has been set to MSE. [Fig. 3.5] illustrates the architecture of the model.

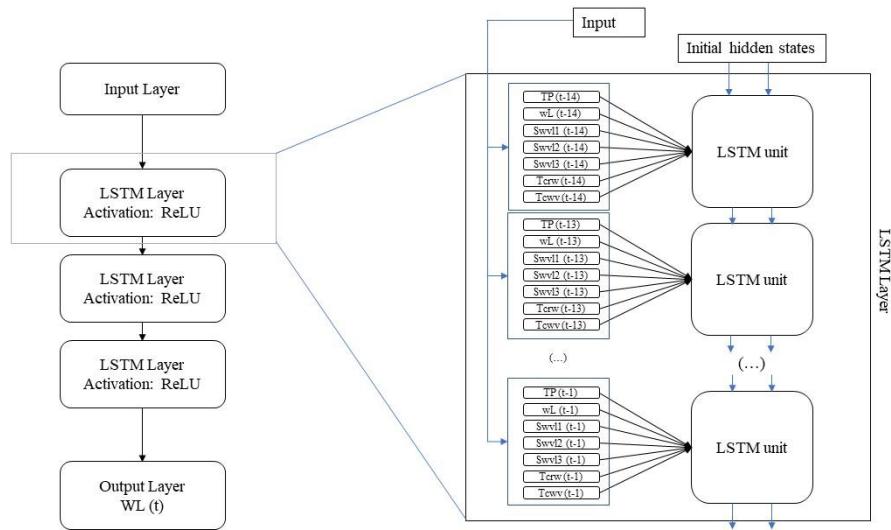


Figure 3.4: Architecture of the LSTM model used to forecast WL for day t with all variables as input. The LSTM unit in the model takes past 14 day's values to maintain sequence in the data.

Chapter 4

Methodology

4.1 General

The study aims to forecast floods for four different lead times using four separate algorithms using four combinations of input variables. For this to be done, a catchment area has been generated using digital elevation model (DEM) files, hydrological data has been collected from multiple sources, the models were built, trained and tested. Finally, the results were analyzed and compared.

The following flow chart shows the workflow of the study.

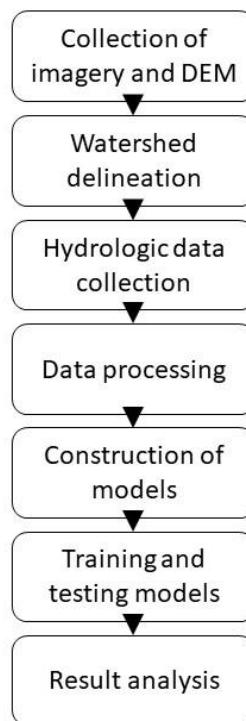


Figure 4.1 Flowchart showing methodology of the study

4.2 Data collection

Data used in the study can be divided into imagery, meteorological data and observational data. ASTGTM v003 (ASTER Global Digital Elevation Model V003 [Data Set], 2019) is DEM that has been used in the study to delineate the study area watershed. This DEM can be accessed from the USGS Earthdata store using wget

(Scrivano, 2012) software. This Study uses meteorological data from the ERA5 gridded dataset for 13 years ranging from January 1, 2007, to December 31, 2020, to train and test separate ML and DL models. 6 variables were chosen as input to the models from the product titled "ERA5 hourly data on single levels from 1979 to present" found in the Copernicus climate data store (cds) (Hersbach et al., 2018) using the cdsAPI. They are total precipitation (TP), volumetric soil water layer 1,2,3 (SWVL1, SWVL2, SWVL3), total column rainwater (TCRW) and total column water vapor (TCWV). The data was available over a gridded resolution of 0.25 degrees and had a temporal resolution of 1 hour in NetCDF format.

4.2.1 ERA5 dataset

ERA5 is the ECMWF's fifth-generation reanalysis of the global climate and weather during the previous 4 to 7 decades. Using the rules of physics, reanalysis combines model data with observations from around the world to create a globally complete and consistent dataset. This principle, known as data assimilation, is based on the method used by numerical weather prediction centers, in which a previous forecast is combined with newly available observations in an optimal way every so many hours (12 hours at ECMWF) to produce a new best estimate of the state of the atmosphere, known as analysis, from which an updated, improved forecast is issued. Reanalysis works in the same way, but at a lower resolution to provide a dataset that spans several decades.(Hersbach et al., 2018). 6 variables were chosen as input to the models from the product titled "ERA5 hourly data on single levels from 1979 to present". They are total precipitation (TP), volumetric soil water layer 1,2,3 (SWVL1, SWVL2, SWVL3), total column rainwater (TCRW) and total column water vapor (TCWV). Different combinations of these 6 variables were used along with historical water level data to forecast water levels at Sheola station at 1-, 3-, 5- and 7-day lead times.

4.2.1.1 Total precipitation (TP)

This parameter is the sum of large scale and convective precipitation. Large-scale precipitation is generated by the cloud scheme in the ECMWF Integrated Forecasting System (IFS). The cloud scheme simulates the production and dispersion of clouds and large-scale precipitation as a result of changes in atmospheric parameters (such as pressure, temperature, and moisture) predicted directly by the IFS at spatial scales of the grid box or greater. The convection system in the IFS, which models convection at geographic scales smaller than the grid box, generates convective precipitation. This

variable excludes fog, dew, and precipitation that evaporates in the atmosphere before reaching the Earth's surface. This parameter is accumulated over the 1 hour ending at the validity date and time. The units of this parameter are depth in metres of water equivalent. It's the depth of the water if it were uniformly distributed over the grid box.(Hersbach et al., 2018)

4.2.1.2 Volumetric Soil Water Layer 1,2,3 (SWVL1,2,3)

The ECMWF Integrated Forecasting System (IFS) represents soil in four layers: Layer 1: 0 - 7cm, Layer 2: 7 - 28cm, Layer 3: 28 - 100cm, and Layer 4: 100 - 289cm. The whole globe, including oceans, falls under the definition of soil water. Soil texture (or classification), soil depth, and underlying groundwater level are all associated with volumetric soil water.(Hersbach et al., 2018) The first three layers were chosen for this study as the changes in volumetric soil water in the fourth layer show little to no correlation to the target water level data. The units of this parameter are cubic meters of water per cubic meter of soil.

4.2.1.3 Total column rainwater (TCRW)

This variable represents the total mass of water in raindrop-sized droplets in a column reaching from the Earth's surface to the top of the atmosphere; i.e., water which can fall to the surface as precipitation. The area-averaged value for a grid box is represented by this parameter. Clouds are made up of a range of water droplets and ice particles of varying sizes. The cloud scheme used by the IFS, the droplet formation, conversion, and aggregation processes are also greatly simplified. reduces this to represent a variety of discrete cloud droplets/particles, such as cloud water droplets, raindrops, ice crystals, and snow (aggregated ice crystals). The droplet formation, conversion, and aggregation processes are also greatly simplified. (Hersbach et al., 2018). Units of this parameter are Kilograms of water per square meters of area over the earth surface. This variable increased the accuracy of the model in forecasting water levels in flash flood events.

4.2.1.4 Total column water vapour (TCWV)

The total quantity of water vapour in a column ranging from the Earth's surface to the top of the atmosphere is represented by this parameter. (Hersbach et al., 2018) This parameter represents the area-averaged value for a grid box, giving values in Kilograms of water vapour per square meters of area in terms of units. This variable was included

in the model to avoid using the rainfall forecasts and performed well when used as input alongside TCRW.

4.2.2 Water Level data (WL)

Water level data has been collected from the Bangladesh Water Development Board (BWDB). BWDB records water levels at 3hr intervals from 6:00 hrs to 9:00 hrs every day. This study makes use of the daily averaged value of these recorded water levels from January 1, 2007 to December 31, 2020.

It is to be noted that peak discharge is not recorded during flash flood events, and are usually interpolated using the rating curve equation. FFWC (Flood Forecast and Warning Centre) of BWDB issues forecast at 1-to-5-day lead time. This study will analyze the model results in detail for selected recent actual monsoon and flash flood events occurring in the years 2017, 2018, 2019, and 2020 for testing model performances. The summary of these flood events has been illustrated in table 4.2.

Table 4.1 Summary of data collection

Data	Units	Package name / Station name	Temporal resolution	Source	Period
DEM		ASTMT v003		Earthdata portal	
Precipitation	m	ERA5 hourly data on single levels from 1979 to present	1 day	Copernicus climate data store	2007-2020
Soil water Volume	m^3	ERA5 hourly data on single levels from 1979 to present	1 day	Copernicus climate data store	2007-2020
Precipitable water	kg/m^2	ERA5 hourly data on single levels from 1979 to present	1 day	Copernicus climate data store	2007-2020
Water vapor volume	kg/m^2	ERA5 hourly data on single levels from 1979 to present	1 day	Copernicus climate data store	2007-2020
Water level	m	Sheola (SW173)	1 day	BWDB	2007-2020

Table 4.2 Summary of flood events that took place from 2017 to 2020

Event Year	Monsoon Flood			Flash Flood		
	Duration	Max WL (m)	Min WL (m)	Duration	Max WL (m)	Min WL (m)
2017	6/1 - 10/15	14.532	11.02	3/15 - 5/31	13.878	4.752
2018	6/1 - 10/15	13.62	6.44	3/15 - 5/31	12.93	4.74
2019	6/1 - 10/15	13.48	7.6	3/15 - 5/31	9.03	3.01
2020	6/1 - 10/15	13.2	9.37	3/15 - 5/31	9.41	2.95

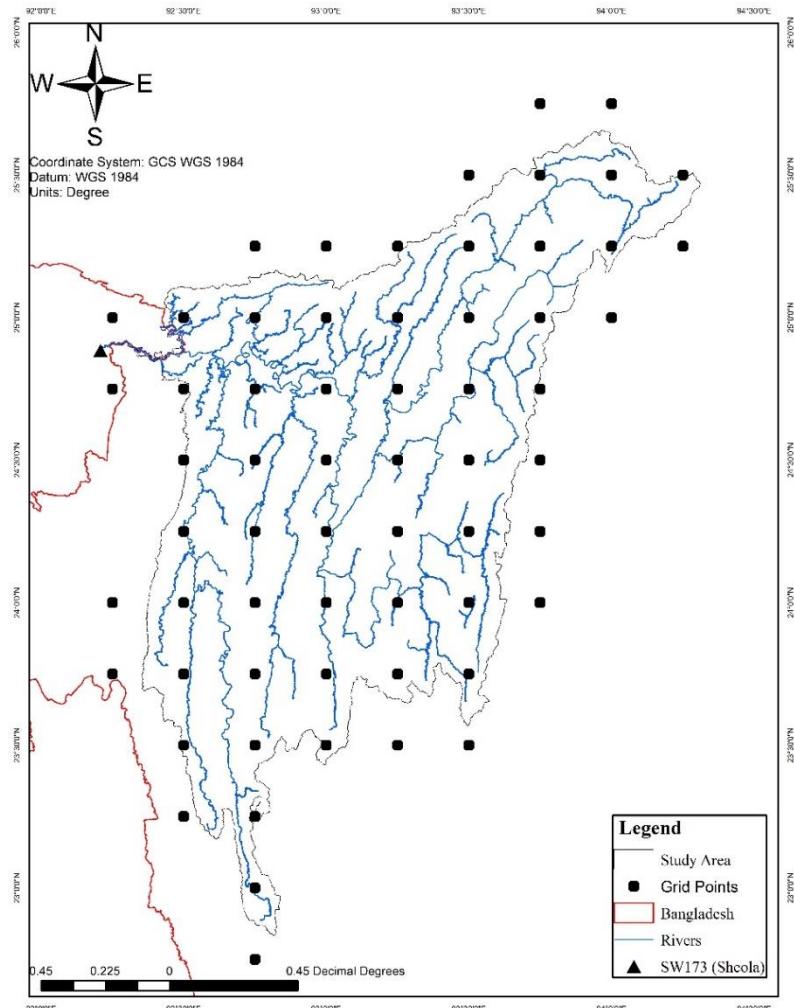


Figure 4.2 Grid locations for ERA5 data collection

4.3 Watershed delineation:

Watershed delineation has been performed using QGIS 3.16. The DEMs were collected from the ASTMT v003 package from earthdata portal. The stream network has been downloaded using the quickOSM plug-in that is built-in to the QGIS software. The outlet of the basin was taken at Sheola (SW173) river station.

4.4 Data extraction and pre-processing

The hourly precipitation data has converted to a daily sum, and the remaining variables were converted to daily mean values using the powerful Climate Data Operator (CDO) tool.(Schulzweida, 2021) Data was collected for a period spanning from January 1, 2007, to December 12, 2020, over 62 grid points that lie within the study area[fig2], and were then averaged to get a single daily time series for every variable. Data extraction from NetCDF format was done using python scripting.

Pre-processing is required to provide the ML model with suitable data for it to produce acceptable results. In this study, 6 variables were imported from the ERA5 dataset, that contains continuous, high-quality data with little to no noise. The only issue, however, was that the data is available in hourly temporal resolution in netCDF format over gridded locations. Since the model has been developed for forecasting daily water levels, the data has to be converted to daily values. The hourly precipitation data has converted to a daily sum, and the remaining variables were converted to daily mean values using the powerful Climate Data Operator (CDO) tool(Schulzweida, 2021). For the remaining variables of the ERA5 dataset, the same is done taking daily mean values.

The data ranging from 2007 to 2020 is split into three non-overlapping sets for training, validation and testing. 50% of the data was used for training, 20% for validation, and the remaining 30% for testing. The split has been done maintaining the original sequence of the data as random split may not be the way to go for sequential data.

For the RF, CB and DNN models, the input dataset is so formed that each layer includes a certain combination of all variables for day T, where each layer of the target dataset includes only water level for day T+N, where N is the lead time, i.e., number of days into the future for which water levels are to be forecast. This has been done for training, validation and also the testing sets.

For the LSTM models however, each input layer contains values of the variables for days T up to T-14, as the model has been designed to ‘look back’ into the past 14 dates,

before it proceeds to make a WL forecast for day T+N, which compares to the actual WL values for the same date set in the target dataset. The models are then built for training on this pre-processed data.

4.5 Construction of the models

A total of 64 models were developed for 4 algorithms using datasets for 4 different input combinations and 4-time steps. Google colaboratory environment chosen for developing the models. A summary of the input and output vector for the models has been shown in [table 4.1]. The different combinations of the input features while building the models are given in [table 4.2]. Shape of input and output variable can be found in table 4.3. The hyperparameters for the training of the models have been selected using GridsearchCV. The full list of hyperparameters for different models can be found in table 4.4.

Table 4.3 Hyperparameters used for model training

Model	Hyperparameter	Value
RF	number of estimators	50
	bootstrap	TRUE
	max depth	11
	out-of-bag estimators	TRUE
CB	learning rate	0.002
	depth	10
	iterations	8000
	Overfitting detector	TRUE
	patience	5
DNN	Hidden Layers	6
	Neurons per layer	64
	Activation Function	ReLU
	Loss Function	MSE
	Optimizer	Adam
LSTM	Hidden Layers	3
	Neurons per layer	64
	Activation Function	ReLU
	Loss Function	MSE
	Optimizer	Adam

Table 4.4 Combinations of input variables

Combination	Input Features	Number of features (F)
1	Total precipitation (TP), Water level (WL)	2
2	TP, WL, SWVL1, SWVL2, SWVL3	5
3	TP, WL, SWVL1, SWVL2, SWVL3, TCRW	6
4	TP, WL, SWVL1, SWVL2, SWVL3, TCRW, TCWV	7

4.6 Model training and testing

All 64 models have been trained and tested for their respective input and target dataset. A held-out dataset has been used to perform validation of the models and generate results. The reliability of the models is then checked for real flood events that took place between 2017 and 2020. Both monsoon flood and flash flood seasons have been taken into account for evaluation of model performance on real flood events.

4.7 Evaluation of model performance

To assess the prediction outcomes of the four models, this study uses the root mean square error (RMSE) and the coefficient of determination (R^2). The RMSE value indicates the error between the projected and observed values, and its unit is the same as the model output value. The RMSE value might be anything between 0 and infinity. A model with an RMSE value around zero is more likely to make accurate projections. The following equation may be used to compute the RMSE.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2}$$

where N is the number of samples, d_i is the target output value, and y_i is the model output value.

The R^2 value is the proportion of the variation in the dependent variable that can be predicted by the independent variable(s), and it is widely used to assess the linear correlation between model outputs and goal outputs. The R^2 value is a number that varies from 0 to 1. When the R^2 value of a model is near 1, it means it can predict more accurately. The following equation can be used to obtain the R^2 value.

$$R^2 = \left[\frac{\sum_{t=1}^N (d_i - \bar{d})(y_i - \bar{y})}{\sqrt{\sum_{t=1}^N (d_i - \bar{d})^2 * \sum_{t=1}^N (y_i - \bar{y})^2}} \right]^2$$

Where \bar{d} is the mean of target outputs, and \bar{y} is the mean of model outputs, N is the number of samples, d_i is the target output value, and y_i is the model output value.

The reliability of the model has then been checked against real flood events that took place between the period 2017 to 2020. Both monsoon and flash flood events have been taken into account for model performance evaluation on real flood events. The results are then analyzed and compared on the basis of error values.

Chapter 5

Results and Discussions

The models were evaluated on two popular evaluation indicators. In order to obtain the reliability of the models, they were tested against four flash flooding events and four monsoon flood events.

5.1 Evaluation of model performance:

It is to be mentioned that all the models were different from each other, and thus had very different training algorithms. Though the use of ReLU activation function and Adam optimizer on the DNN and LSTM models reduced computation time, it took the LSTM a lot more train than the other models owing to its recurrent structure. The RF and CB models were comparatively much faster. [Table 5.1] shows the performance of the models based on their RMSE and R^2 values for training, validation and testing datasets, for all four combinations of input features.

Table 5.1 shows that all the models perform well in terms of high R^2 value and low RMSE. The performance of the models improve as more features are included for training. It can also be seen that the models have almost similar performances irrespective of the algorithm chosen. Slight improvements can be seen in the LSTM models as these account for the recurrent nature of input data. Another observation is that the models show better performance as the time step is reduced. This phenomenon fits natural expectation. [fig 5.1] shows the scatter plots of forecast vs. actual water level for training, validation and testing datasets for the different algorithms with input combination 4 for a time step of T+3.

It has been observed from table 5.1 that for lead times of 1 and 3 days, the LSTM models perform relatively better with more consistent low values of RMSE and high values of R^2 for training, validation and testing. The DNN models are at a very close second place. The RF and the catboost models, despite having very low RMSE and very high R^2 values for training, perform relatively worse on validation and held-out test datasets. This is due to the well-known property of these models being prone to overfitting on train dataset.

For a lead time of 5 and 7 days, the DNN models tend to perform better in terms of consistent low RMSE and high R^2 values. There is a noticeable drop in the performance for all the models in forecasting with 5 and 7 day lead times, just as expected. Both the RF and catboost regressor maintain their trend of overfitting on training dataset, leading to good train scores but much worse test scores. A visual representation of these values can be obtained in figure 5.1 and 5.2.

Table 5.1: Model evaluation based on RMSE and R² scores

Input Combinations	Time Step	RMSE (m)				R-squared			
		DNN	RF	CB	LSTM	DNN	RF	CB	LSTM
1	Train	0.30	0.13	0.24	0.23	0.992	0.999	0.995	0.996
	T+1	Val	0.31	0.29	0.27	0.22	0.992	0.993	0.994
	Test	0.34	0.37	0.39	0.28	0.990	0.988	0.987	0.993
	Train	0.70	0.28	0.51	0.60	0.959	0.993	0.978	0.970
	T+3	Val	0.73	0.61	0.58	0.63	0.955	0.969	0.972
	Test	0.77	0.75	0.71	0.72	0.949	0.952	0.956	0.955
	Train	1.00	0.42	0.92	0.93	0.917	0.985	0.929	0.928
	T+5	Val	1.03	0.85	0.97	0.97	0.911	0.939	0.920
	Test	1.06	1.02	1.02	1.04	0.902	0.909	0.910	0.907
	Train	1.22	0.55	0.96	1.14	0.876	0.975	0.923	0.892
2	T+7	Val	1.24	1.10	0.99	1.17	0.869	0.897	0.917
	Test	1.27	1.28	1.23	1.23	0.860	0.857	0.868	0.871
	Train	0.28	0.12	0.23	0.25	0.994	0.999	0.996	0.995
	T+1	Val	0.29	0.28	0.29	0.26	0.993	0.993	0.993
	Test	0.31	0.38	0.52	0.29	0.991	0.988	0.976	0.993
	Train	0.57	0.24	0.47	0.58	0.973	0.995	0.982	0.972
	T+3	Val	0.62	0.58	0.57	0.61	0.967	0.971	0.973
	Test	0.64	0.78	0.80	0.69	0.964	0.946	0.945	0.959
	Train	0.84	0.36	0.91	0.86	0.942	0.989	0.931	0.938
	T+5	Val	0.88	0.79	0.96	0.91	0.934	0.948	0.922
3	Test	0.91	1.12	1.07	1.01	0.928	0.891	0.901	0.911
	Train	1.10	0.44	0.92	1.14	0.900	0.984	0.930	0.893
	T+7	Val	1.14	1.02	0.97	1.17	0.890	0.911	0.921
	Test	1.20	1.40	1.30	1.25	0.875	0.830	0.853	0.866
	Train	0.27	0.11	0.23	0.32	0.994	0.999	0.996	0.992
	T+1	Val	0.30	0.28	0.29	0.32	0.993	0.993	0.993
	Test	0.30	0.37	0.52	0.33	0.992	0.988	0.976	0.991
	Train	0.61	0.23	0.46	0.58	0.969	0.995	0.982	0.972
	T+3	Val	0.70	0.58	0.56	0.63	0.958	0.972	0.973
	Test	0.67	0.76	0.79	0.74	0.961	0.950	0.945	0.953
4	Train	0.81	0.35	0.91	0.87	0.945	0.990	0.932	0.937
	T+5	Val	0.84	0.77	0.95	0.94	0.939	0.950	0.923
	Test	0.91	1.09	1.07	0.94	0.928	0.897	0.900	0.925
	Train	1.06	0.44	0.88	1.03	0.907	0.984	0.935	0.912
	T+7	Val	1.08	1.01	0.95	1.11	0.900	0.913	0.923
	Test	1.12	1.34	1.28	1.44	0.891	0.843	0.857	0.821
	Train	0.27	0.11	0.22	0.27	0.994	0.999	0.996	0.994
	T+1	Val	0.29	0.28	0.29	0.30	0.993	0.993	0.993
	Test	0.31	0.38	0.51	0.39	0.992	0.988	0.977	0.987
	Train	0.60	0.23	0.43	0.52	0.970	0.996	0.984	0.977
5	T+3	Val	0.66	0.57	0.55	0.60	0.963	0.972	0.974
	Test	0.68	0.75	0.79	0.66	0.959	0.951	0.946	0.963
	Train	0.81	0.34	0.91	0.72	0.945	0.991	0.932	0.957
	T+5	Val	0.86	0.77	0.96	0.89	0.937	0.949	0.922
	Test	0.94	1.06	1.06	0.96	0.923	0.901	0.901	0.921
	Train	1.03	0.40	0.89	0.84	0.912	0.987	0.935	0.941
6	T+7	Val	1.05	0.96	0.93	1.10	0.906	0.922	0.926
	Test	1.09	1.35	1.29	1.29	0.896	0.841	0.855	0.858

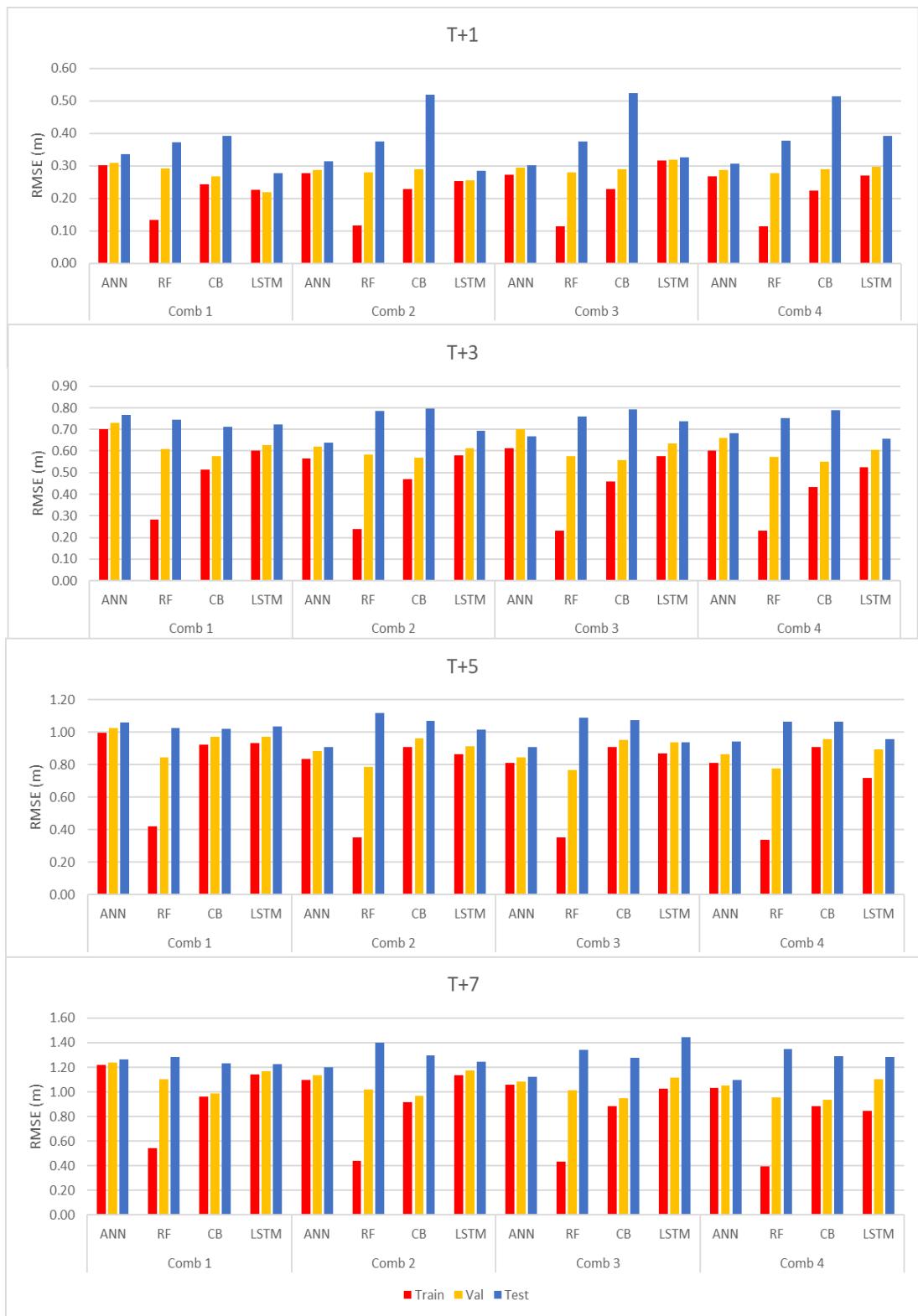


Figure 5.1 RMSE values for different models with different combinations of input features for different lead times



Figure 5.2 Coefficient of determination (R^2) values for different models with different combinations of input features for different lead times

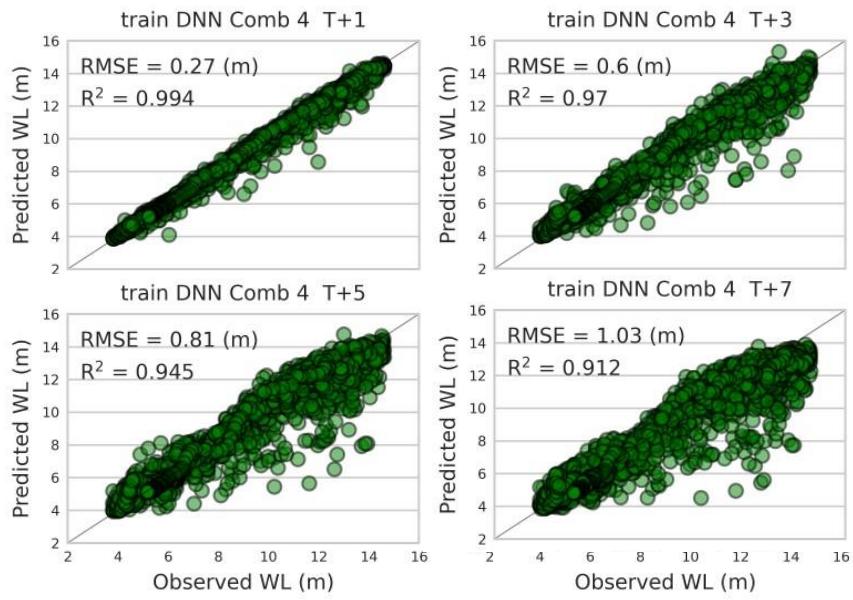


Figure 5.3 Predicted WL (m) vs Observed WL (m) for train data set with input combination 4 for different lead times

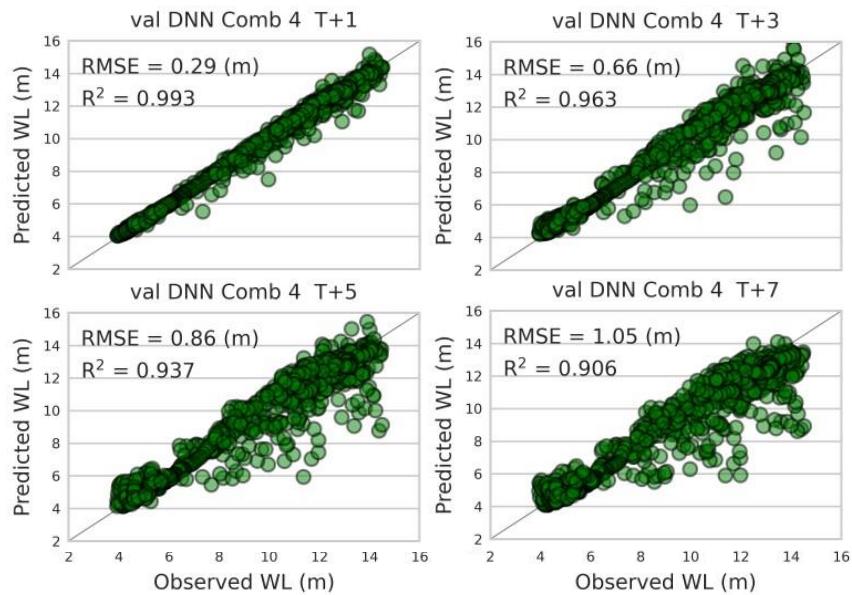


Figure 5.4 Predicted WL (m) vs Observed WL (m) for validation data set with input combination 4 for different lead times.

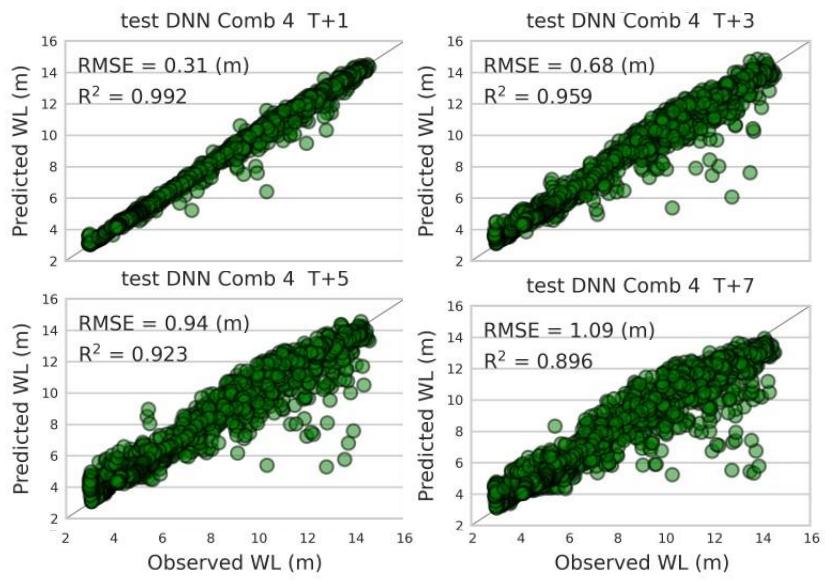


Figure 5.5 Predicted WL (m) vs Observed WL for test data set with input combination 4 for different lead times.

Forecast WLs for different lead times for the DNN model for combination 4 were then plotted against time to visually compare the forecast WLs with the actual ones. Fig 5.6 shows this plot.

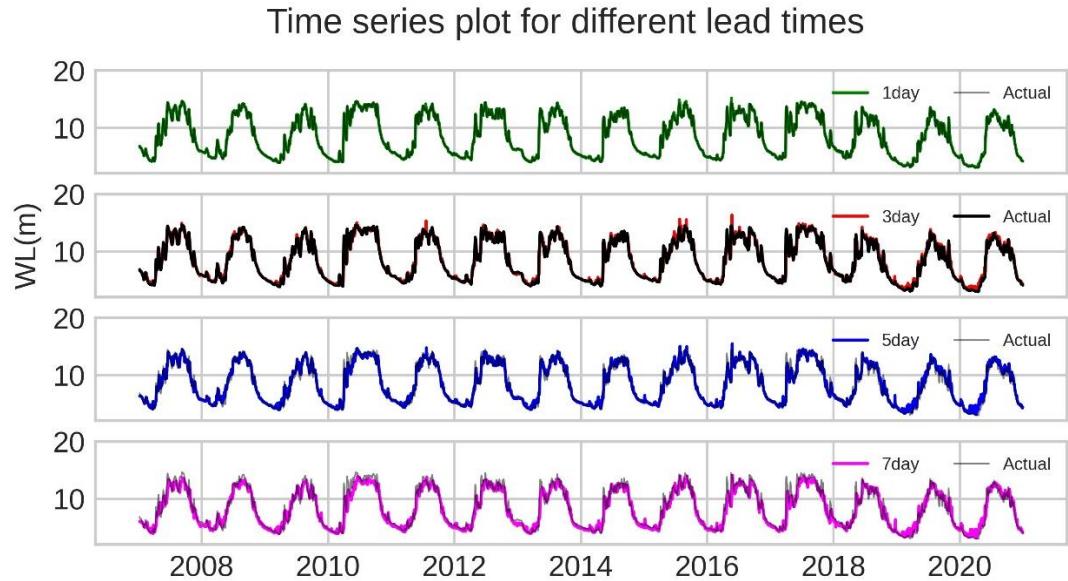


Figure 5.6 Comparison of water levels using DNN with combination 4 for different lead times.

5.2 Evaluation of model performance for recent flood events:

According to the flood forecast performances of the models in [table 6.1], the DNN models have slight superiority over the other models in terms of RMSE and R² values. This continues to be the case in actual flood events. The CB and RF models demonstrate ill performance for flash flood events, especially the 2020 event. In general, however, all the models perform better in forecasting monsoon floods compared to flash floods. Also noticeable is that the models show best performance with input combination 3 for both the monsoon and the flash flood events. [Table 6.2] and [Table 6.3] show the performance of the models in forecasting monsoon flood and flash floods respectively in terms of RMSE values.

Table 5.2: Evaluation of models for recent Monsson flood events based on RMSE values

Input combination	Time Step	2017				2018				2019				2020			
		DNN	RF	CB	LSTM												
1	T+1	0.32	0.29	0.27	0.24	0.40	0.40	0.37	0.36	0.34	0.41	0.35	0.28	0.31	0.32	0.30	0.26
	T+3	0.68	0.57	0.50	0.56	0.78	0.75	0.63	0.76	0.75	0.89	0.73	0.76	0.66	0.74	0.69	0.68
	T+5	0.93	0.73	0.86	0.93	1.06	0.97	0.99	1.03	0.98	1.19	0.93	1.02	0.89	0.96	0.86	0.94
	T+7	1.03	0.89	0.93	1.19	1.26	1.17	1.06	1.20	1.11	1.35	1.11	1.09	1.08	1.04	0.94	1.09
2	T+1	0.29	0.25	0.26	0.26	0.36	0.36	0.38	0.39	0.32	0.40	0.35	0.28	0.29	0.36	0.30	0.26
	T+3	0.57	0.50	0.49	0.60	0.60	0.66	0.64	0.70	0.67	0.90	0.79	0.71	0.69	0.77	0.68	0.63
	T+5	0.86	0.73	0.86	0.86	0.86	0.94	1.00	1.03	0.83	1.19	0.94	0.98	0.85	0.94	0.84	0.89
	T+7	0.95	0.91	0.99	1.06	1.21	1.18	1.08	1.22	1.15	1.41	1.25	1.12	1.05	1.10	0.93	1.10
3	T+1	0.30	0.25	0.26	0.40	0.31	0.36	0.38	0.39	0.31	0.39	0.35	0.32	0.30	0.35	0.31	0.34
	T+3	0.63	0.50	0.49	0.52	0.62	0.64	0.62	0.61	0.68	0.89	0.78	0.66	0.70	0.76	0.67	0.65
	T+5	0.70	0.69	0.85	0.97	0.89	0.92	1.00	0.86	0.97	1.14	0.94	0.89	0.85	0.92	0.84	0.96
	T+7	0.97	0.91	0.94	1.15	1.04	1.20	1.08	1.23	1.01	1.37	1.26	1.12	0.95	1.03	0.93	1.15
4	T+1	0.29	0.25	0.26	0.29	0.33	0.36	0.39	0.53	0.33	0.38	0.35	0.44	0.31	0.35	0.30	0.46
	T+3	0.56	0.51	0.49	0.53	0.66	0.64	0.62	0.59	0.77	0.89	0.78	0.75	0.73	0.76	0.66	0.60
	T+5	0.75	0.72	0.86	0.72	0.92	0.97	1.00	0.88	0.98	1.19	0.93	1.03	0.90	0.92	0.83	0.94
	T+7	1.10	0.93	0.98	0.74	1.05	1.35	1.19	1.61	0.97	1.46	1.28	1.64	0.96	1.07	0.92	1.39

Table 5.3 Evaluation of models for recent Flash Flood events based on RMSE values

Input Combinations	Time Step	2017				2018				2019				2020			
		DNN	RF	CB	LSTM												
1	T+1	0.68	0.63	0.54	0.48	0.49	0.43	0.41	0.36	0.36	0.45	0.55	0.36	0.25	0.58	0.73	0.30
	T+3	1.62	1.39	1.27	1.31	1.14	0.98	0.88	1.02	0.77	1.00	0.97	0.86	0.56	1.06	1.12	0.65
	T+5	2.31	1.90	2.18	2.08	1.60	1.34	1.53	1.56	0.99	1.35	1.03	1.17	0.80	1.54	1.12	0.99
2	T+1	2.80	2.43	2.32	2.54	1.93	1.61	1.58	1.87	1.14	1.60	1.56	1.25	0.84	1.91	1.83	1.05
	T+3	0.64	0.64	0.61	0.53	0.47	0.40	0.42	0.37	0.34	0.48	0.81	0.34	0.22	0.60	0.98	0.23
	T+5	1.27	1.37	1.27	1.31	0.89	0.87	0.82	0.99	0.77	1.20	1.21	0.86	0.59	1.38	1.35	0.65
3	T+1	1.87	1.91	2.17	1.97	1.26	1.16	1.54	1.44	1.07	1.79	1.16	1.14	0.81	2.17	1.17	1.09
	T+3	2.38	2.30	2.19	2.63	1.67	1.46	1.45	1.94	1.30	2.22	1.87	1.31	0.88	2.99	2.07	1.09
	T+5	0.60	0.65	0.62	0.51	0.46	0.40	0.42	0.40	0.33	0.48	0.81	0.37	0.20	0.61	0.97	0.36
4	T+1	1.33	1.35	1.28	1.20	0.98	0.81	0.83	0.93	0.72	1.12	1.17	0.95	0.50	1.29	1.32	1.21
	T+3	1.84	1.90	2.15	1.81	1.25	1.08	1.54	1.34	1.08	1.73	1.16	1.07	0.90	2.13	1.20	0.88
	T+5	2.39	2.31	2.19	2.17	1.65	1.47	1.42	1.36	1.28	2.03	1.85	2.62	0.81	2.61	2.03	2.87
4	T+1	0.60	0.63	0.60	0.48	0.45	0.40	0.43	0.38	0.34	0.48	0.77	0.53	0.26	0.68	0.95	0.53
	T+3	1.36	1.32	1.25	1.13	1.00	0.79	0.80	0.92	0.74	1.06	1.18	0.84	0.80	1.40	1.37	0.92
	T+5	1.94	1.84	2.14	1.60	1.36	1.04	1.53	1.32	1.01	1.52	1.13	1.32	1.43	2.21	1.17	1.63
	T+7	2.47	2.23	2.22	2.37	1.67	1.24	1.36	1.73	1.05	1.89	1.75	1.21	1.27	2.99	2.14	1.22

It can be observed from table 6.3 that all the models are not very efficient while tested against the flash flood events. This is because the flash flood events are catastrophic and has a very small time of concentration. For the monsoon events, however, the models are seen to perform reasonably better. Fig 6.6 and 6.7 shows the comparison of actual water level of the monsoon and flash flood events from 2017 to 2020 with forecast for a 3-day lead taking all variables as input.

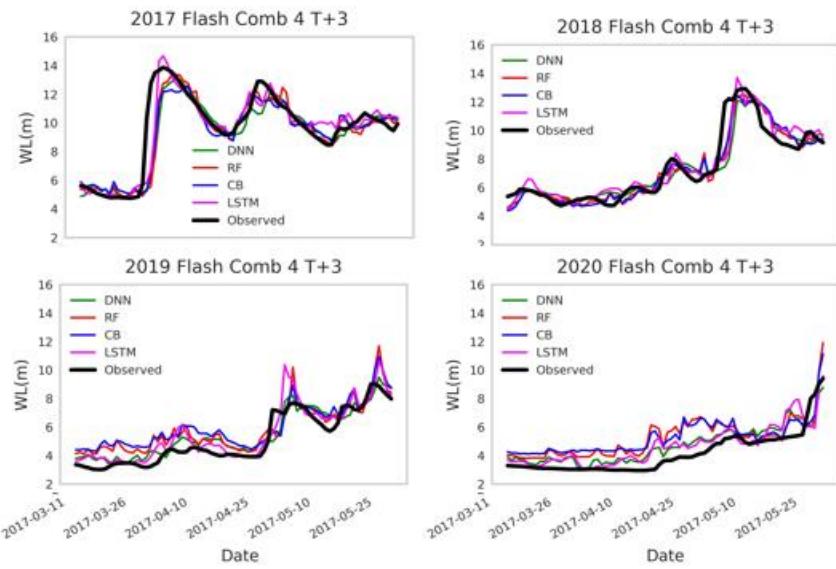


Figure 5.7 Comparison water levels of actual flash flood events with simulated water levels (Input combination 4, Lead time 3-days)

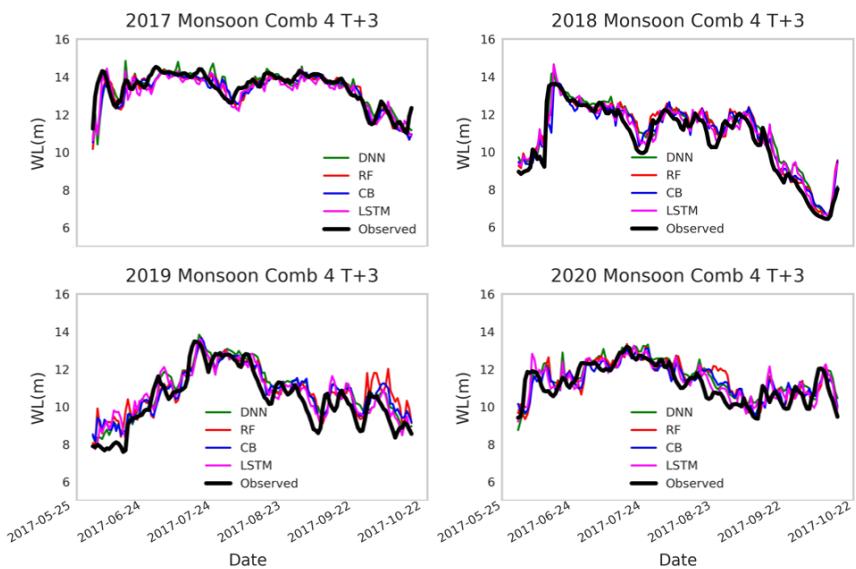


Figure 5.8 Comparison water levels of actual monsoon flood events with simulated water levels (Input combination 4, Lead time 3-days)

Chapter 6

Conclusion and recommendation

6.1 Conclusion

This study explores four supervised machine learning techniques for flood forecasting in a region where observational data required for building hydrological flood forecasting models are difficult to access.

The models developed for this study show good results for a short lead time with rmse values ranging from 0.23m to 0.3m for 1 day lead and 0.5m to 0.7m for 3-day lead; but shows a higher error for longer lead times, with rmse values 0.8m to 1m for 5-day lead time and 1m to 1.5m for 7day lead time. This can be explained by the flash flood events that take place in the region, causing the water levels to rise by almost 10m within 24 hours. The best model, however, was found to be the deep neural network model with consistent low rmse values and higher correlation values compared to the other models.

It is observed that data driven models might be a viable option for generating short to mid-range flood forecasts in such data scarce regions. RF and CB models, however, could not perform as consistently as the DNN and LSTM models, as they tended to overfit the training dataset. This statement can be supported by the fact that the error values in testing was much higher than that in training scenario for the RF and CB models, which indicates overfitting. If a larger dataset were available, we might get better results.

Though the models show inaccuracy in forecasting flash floods, the performances of the models are reasonable for monsoon flood forecasting. Further tuning and calibration may be suggested for better model performances and hence, more accurate forecasts.

6.2 Recommendation

The water level data used in the study was found to be of good quality and free of noise only from 2007. The models should be re-calibrated when more data will be available in the future for better learning of supervised machine learning models. Another suggestion that may be considered is the use of stacking ensembles for a more robust learning of the models. Although past studies reported about the inaccuracy in precipitation forecast, future study may consider bias/error correction of precipitation forecast and incorporate that as additional input variable.

References

- ASTER Global Digital Elevation Model V003 [Data set]. . (2019). NASA EOSDIS Land Processes DAAC. <https://doi.org/10.5067/ASTER/ASTGTM.003>
- Keras API reference / Losses. (2022, May 10). <https://keras.io/api/losses/> Overview of hyperparameter tuning . (2022, May 6).
- Afiya Narzis. (2020). *Impacts of climate change and upstream intervention on the hydrology of the Meghna river basin using SWAT*. Bangladesh University of Engineering and Technology.
- Bangladesh Water Development Board (BWDB). (2022, May 10). নদী সম্পর্কিত তথ্য. <https://www.bwdb.gov.bd/rivers-information>
- Banglapedia, 2009. Banglapedia: National Encyclopedia of Bangladesh, Asiatic Society of Bangladesh (<http://www.banglapedia.org/>), Dhaka, Bangladesh.
- Baten, A., Arcos González, P., & Delgado, R. C. (2018). Natural Disasters and Management Systems of Bangladesh from 1972 to 2017: Special Focus on Flood. *OmniScience: A Multi-Disciplinary Journal*, 8(3), 35–47. www.stmjournals.com
- Couronné, R., Probst, P., & Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, 19(1), 270. <https://doi.org/10.1186/s12859-018-2264-5>
- Dawson, C. W., & Wilby, R. L. (2001). Hydrological modelling using artificial neural networks. *Progress in Physical Geography: Earth and Environment*, 25(1), 80–108. <https://doi.org/10.1177/030913330102500104>
- FAO AQUASTAT. (2011). *Trans boundary River Basins – Ganges-Brahmaputra-Meghna River Basin*. FAO.
- Gelenbe, E., & Yin, Y. (2018). Deep Learning with Dense Random Neural Networks (pp. 3–18). https://doi.org/10.1007/978-3-319-67792-7_1
- Gui, J., Cong, X., Cao, Y., Ren, W., Zhang, J., Zhang, J., Cao, J., & Tao, D. (2021). A Comprehensive Survey and Taxonomy on Image Dehazing Based on Deep Learning.
- Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of Big Data*, 7(1), 94. <https://doi.org/10.1186/s40537-020-00369-8>
- Hersbach, H., Bell, B., Berrisford, P., & Biavati, (2018). ERA5 hourly data on single levels from 1979 to present. *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*.

- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Huq, I., 2008. Water Resources Management in Bangladesh, April, 2008, ISBN: 984-300- 001995-0.
- Institute of water modelling. (2019). *Flash Flood Early Warning System for North Eastern Part of Bangladesh*. www.iwmbd.org
- Islam, A. S. (2010). Improving flood forecasting in Bangladesh using an artificial neural network. *Journal of Hydroinformatics*, 12(3), 351–364. <https://doi.org/10.2166/hydro.2009.085>
- Japan International Cooperation Agency (Ed.). (2011). *Preparatory Planning Study for Meghna River Basin Management in the People's Republic of Bangladesh*. Infrastructure Development Institute - Japan Yachiyo Engineering Co., Ltd.
- Kao, I.-F., Zhou, Y., Chang, L.-C., & Chang, F.-J. (2020). Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 583, 124631. <https://doi.org/10.1016/j.jhydrol.2020.124631>
- Kim, P. (2017). Convolutional Neural Network. In *MATLAB Deep Learning* (pp. 121–147). Apress. https://doi.org/10.1007/978-1-4842-2845-6_6
- Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. <http://arxiv.org/abs/1412.6980>
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11), 6005–6022. <https://doi.org/10.5194/hess-22-6005-2018>
- Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*.
- Liu, H. (2020). Rail transit channel robot systems. In *Robot Systems for Rail Transit Applications* (pp. 283–328). Elsevier. <https://doi.org/10.1016/B978-0-12-822968-2.00007-3>
- Mohammed, K., Islam, A. K. M. S., & Khan, M. J. U. (n.d.). Flash Flood Forecasting in the Northeast Region of Bangladesh Using Artificial Neural Network. *6th International Conference on Water & Flood Management (ICWFM-2017)*.
- Mondal, S., Akter, L., Hiya, H., & Farukh, M. (2021). Effects of 2017 Early Flash Flooding on Agriculture in Haor Areas of Sunamganj. *Journal of Environmental Science and Natural Resources*, 12(1–2), 117–125. <https://doi.org/10.3329/jesnr.v12i1-2.52007>

- Mosaffa, H., Sadeghi, M., Mallakpour, I., Naghdyzadegan Jahromi, M., & Pourghasemi, H. R. (2022). Application of machine learning algorithms in hydrology. In *Computers in Earth and Environmental Sciences* (pp. 585–591). Elsevier. <https://doi.org/10.1016/B978-0-323-89861-4.00027-0>
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning.*
- Piotrowski, A., Napiórkowski, J. J., & Rowiński, P. M. (2006). Flash-flood forecasting by means of neural networks and nearest neighbour approach – a comparative study. *Nonlinear Processes in Geophysics*, 13(4), 443–448. <https://doi.org/10.5194/npg-13-443-2006>
- Probst P, Bischl B, & Boulesteix A-L. (2018). *Tunability: Importance of hyperparameters of machine learning algorithms.*
- Probst P, Wright M, & Boulesteix A-L. (n.d.). *Hyperparameters and Tuning Strategies for Random Forest. 2018.*
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). *CatBoost: unbiased boosting with categorical features.*
- R. Caruana, & A. Niculescu-Mizil. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning, ACM*, 161–168.
- Raschka, S. (2019). *Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2.* Packt Publishing, Limited.
- Roe, B. P., Yang, H.-J., Zhu, J., Liu, Y., Stancu, I., & McGregor, G. (2005). Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2–3), 577–584. <https://doi.org/10.1016/j.nima.2004.12.018>
- Sahoo, G. B., Ray, C., & De Carlo, E. H. (2006). Use of neural network to predict flash flood and attendant water qualities of a mountainous stream on Oahu, Hawaii. *Journal of Hydrology*, 327(3–4), 525–538. <https://doi.org/10.1016/j.jhydrol.2005.11.059>
- Santosh, K., Das, N., & Ghosh, S. (2022). Deep learning: a review. In *Deep Learning Models for Medical Imaging* (pp. 29–63). Elsevier. <https://doi.org/10.1016/B978-0-12-823504-1.00012-X>
- Schulzweida, U. (2021). *CDO User Guide (Version 2.0.0).* Zenodo.
- Scrivano, G. (2012, August 6). *GNU wget 1.14 released.*

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*.

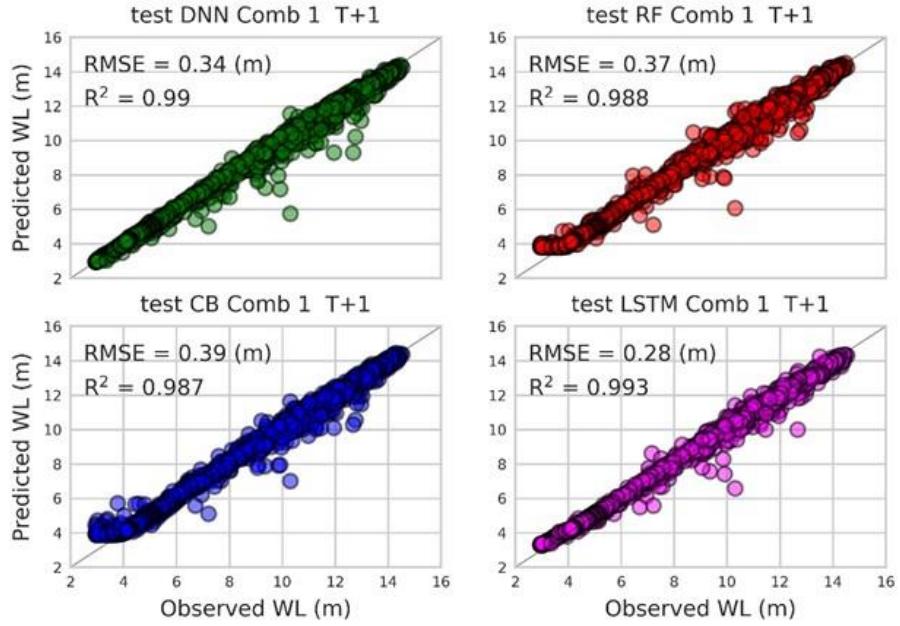
Theodoridis, S. (2020). Neural Networks and Deep Learning. In *Machine Learning* (pp. 901–1038). Elsevier. <https://doi.org/10.1016/B978-0-12-818803-3.00030-1>

Tien Bui, D., Hoang, N.-D., Martínez-Álvarez, F., Ngo, P.-T. T., Hoa, P. V., Pham, T. D., Samui, P., & Costache, R. (2020). A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Science of The Total Environment*, 701, 134413. <https://doi.org/10.1016/j.scitotenv.2019.134413>

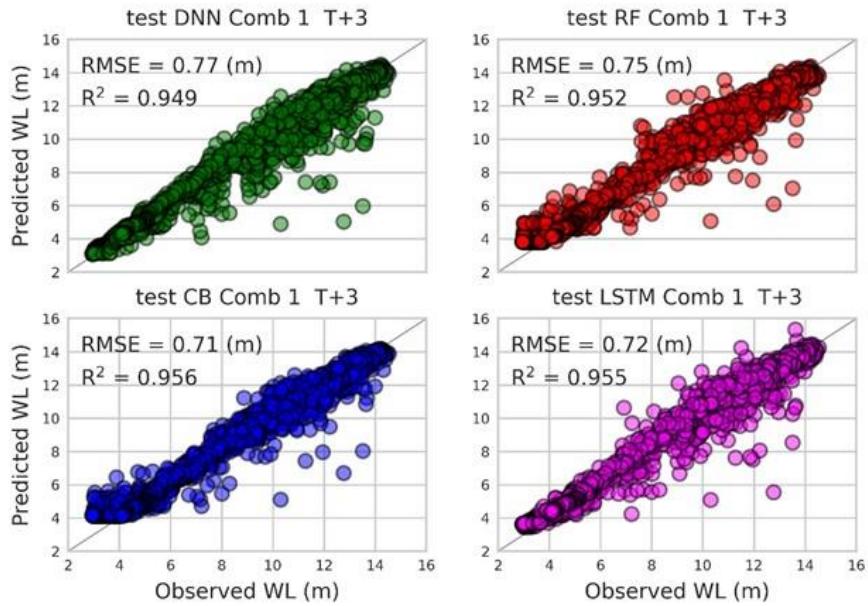
Wazed, A. (1991). *Bangladeshher Nadimala (Rivers of Bangladesh, in Bangla)* (F. Khan (Ed.)). University Press Limited.

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58, 308–324. <https://doi.org/10.1016/j.trc.2015.02.019>

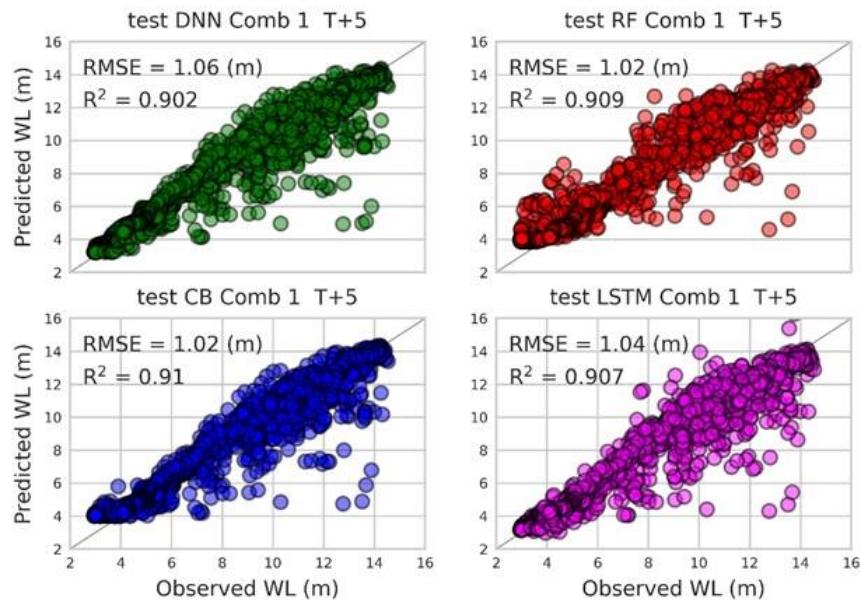
Appendix A



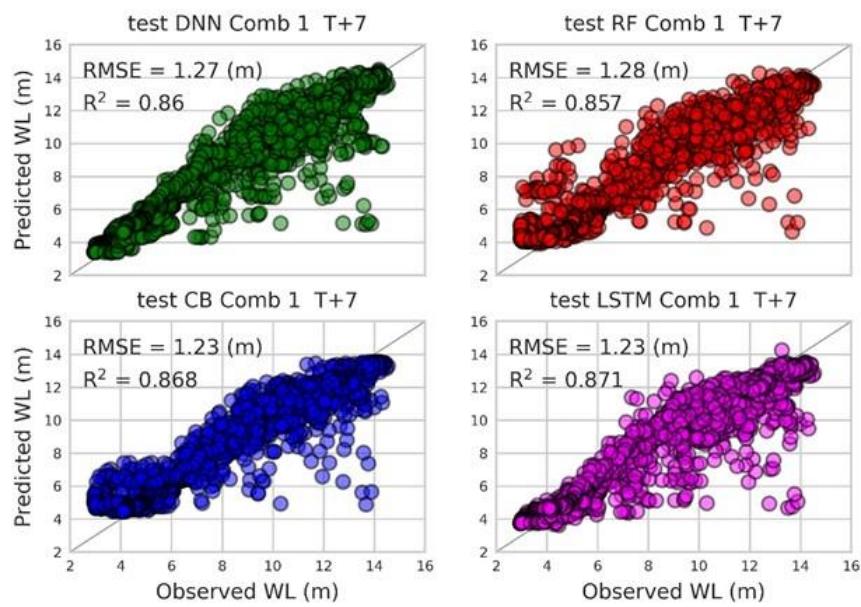
A. 1 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 1, Lead time 1day)



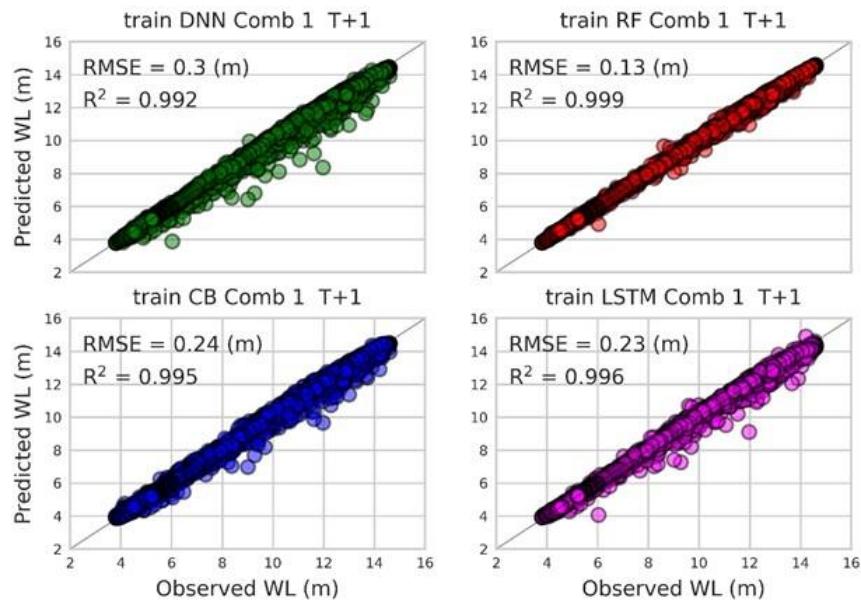
A. 2 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 1, Lead time 3days)



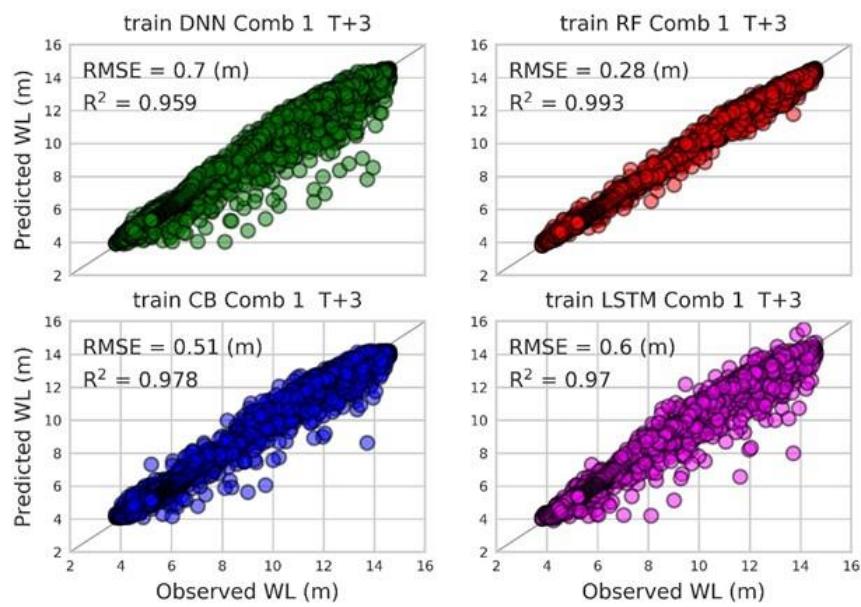
A. 3 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 1, Lead time 5 days)



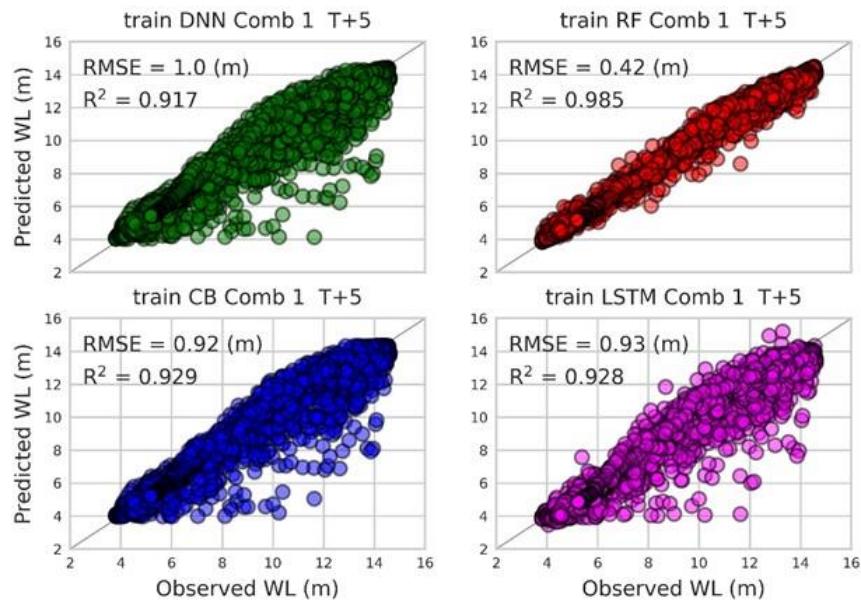
A. 4 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 1, Lead time 7 days)



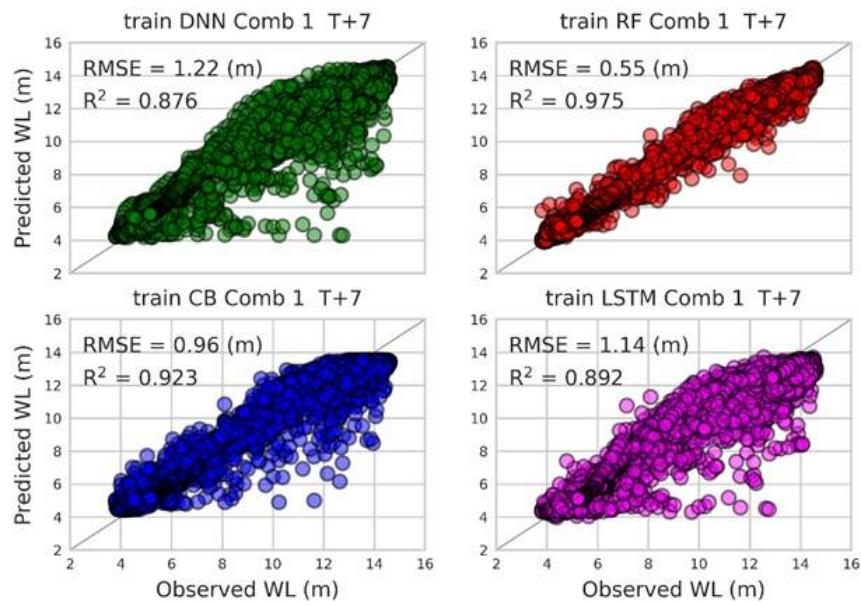
A. 5 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 1, Lead time 1 days)



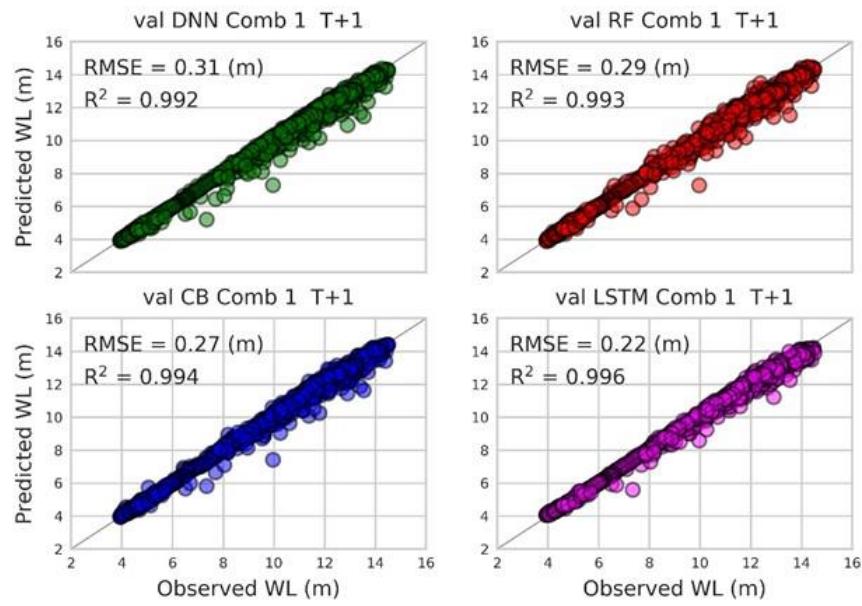
A. 6 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 1, Lead time 3 days)



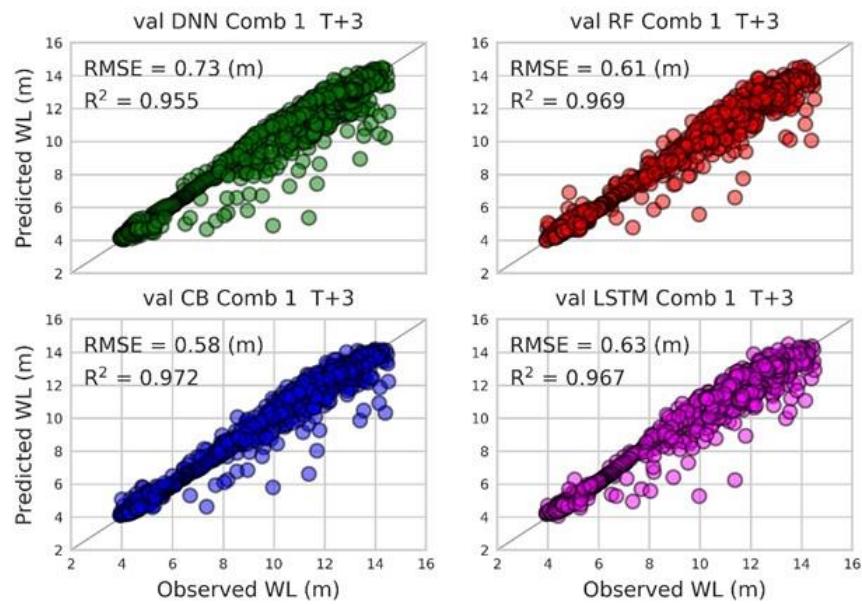
A. 7 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 1, Lead time 5 days)



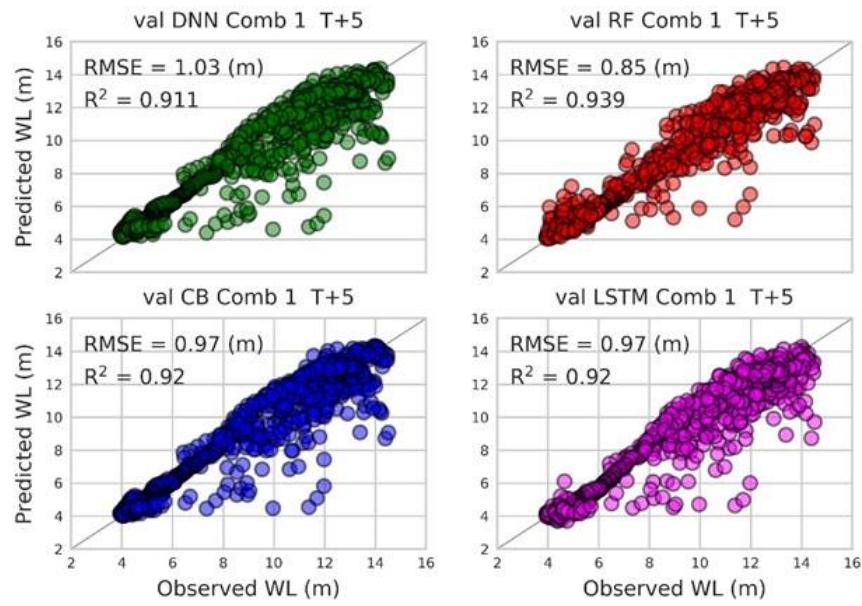
A. 8 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 1, Lead time 7 days)



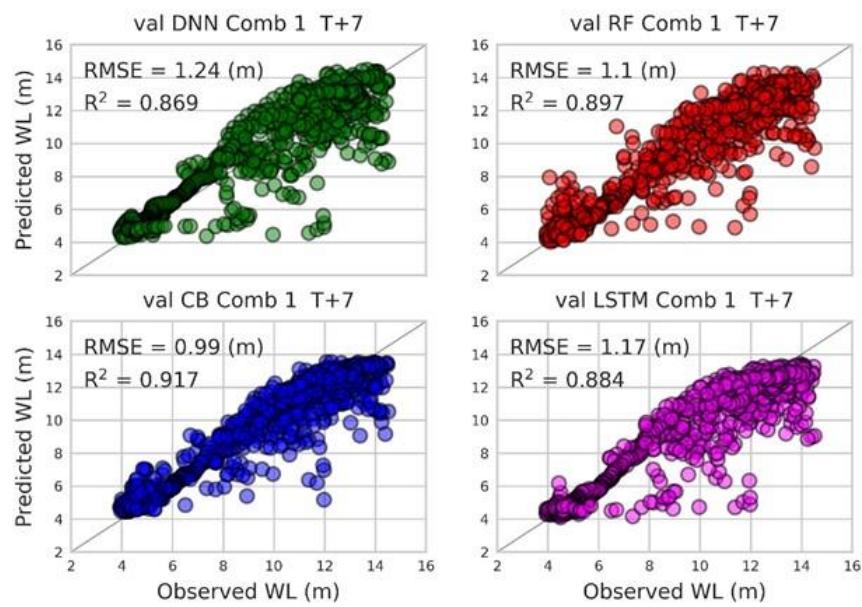
A. 9 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 1, Lead time 1 days)



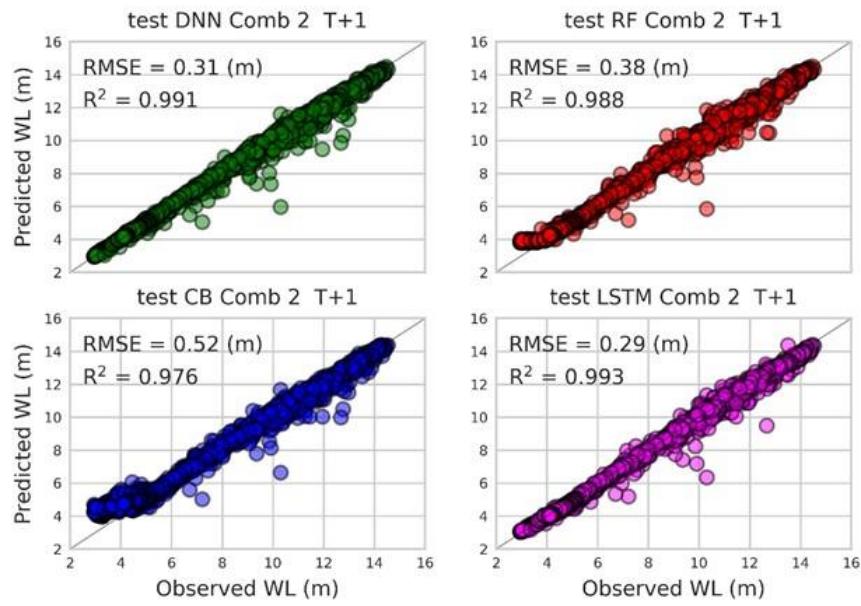
A. 10 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 1, Lead time 3 days)



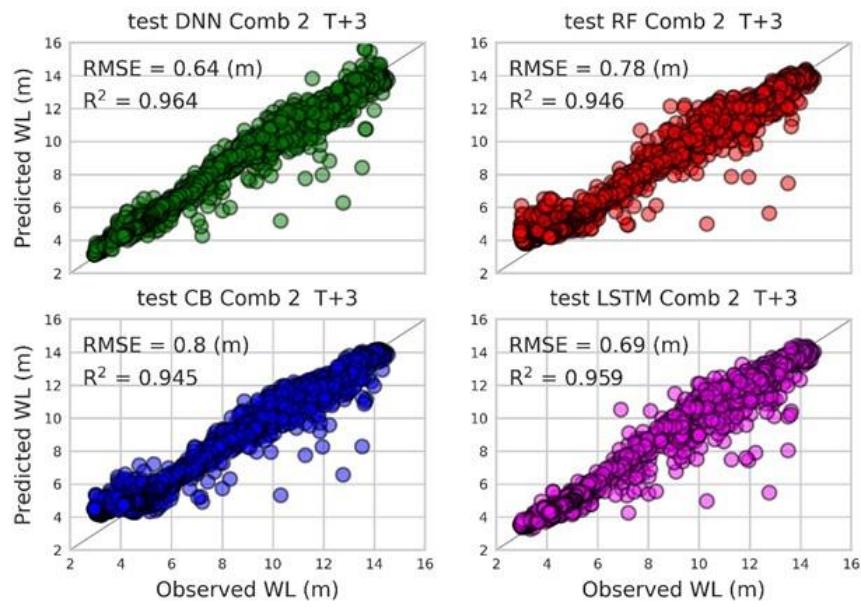
A. 11 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 1, Lead time 5 days)



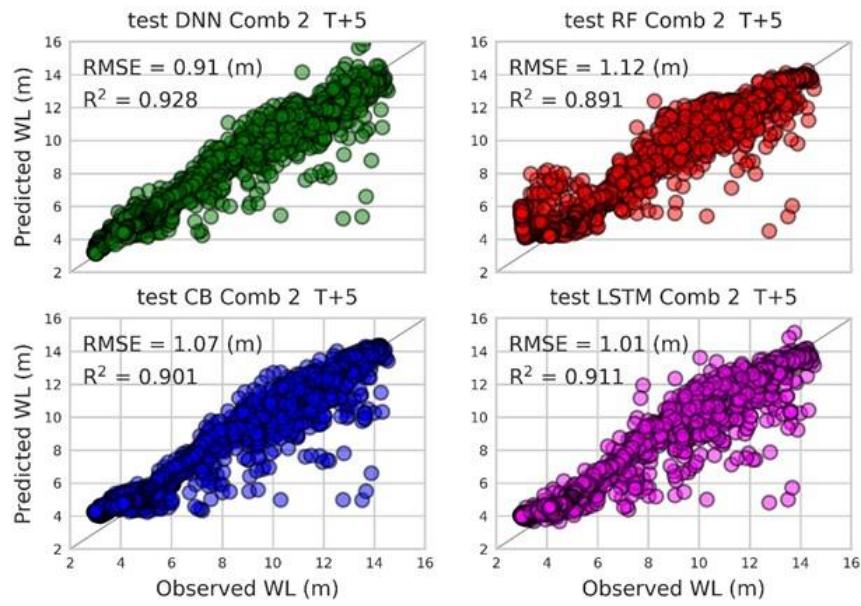
A. 12 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 1, Lead time 5 days)



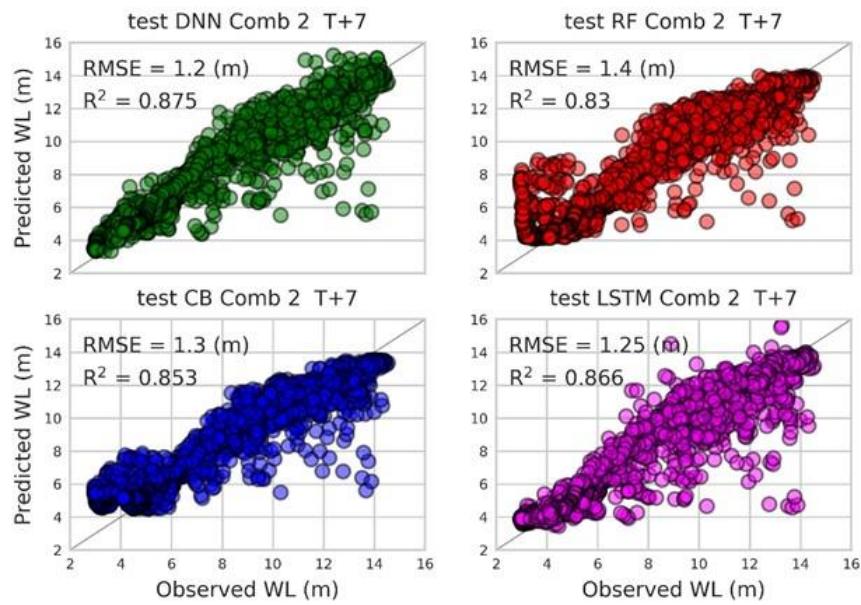
A. 13 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 2, Lead time 1 days)



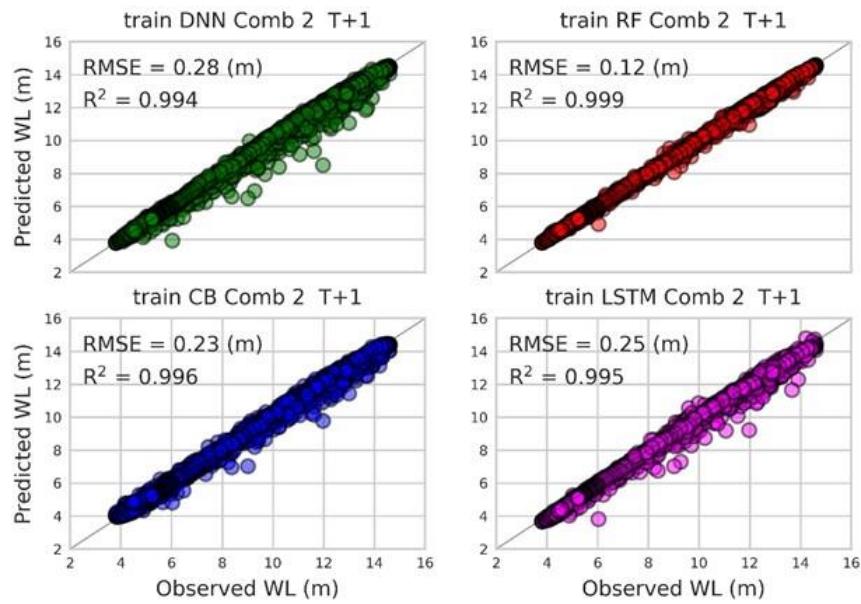
A. 14 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 2, Lead time 3 days)



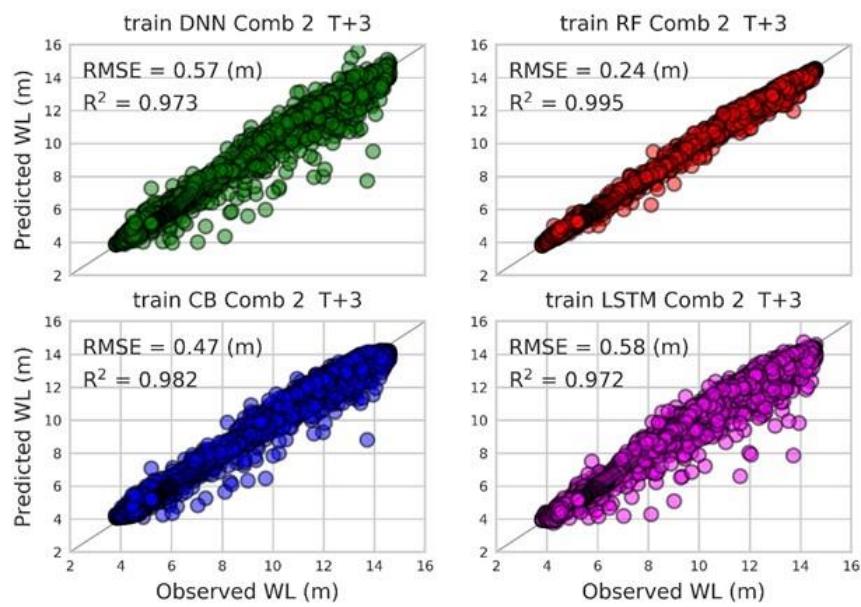
A. 15 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 2, Lead time 5 days)



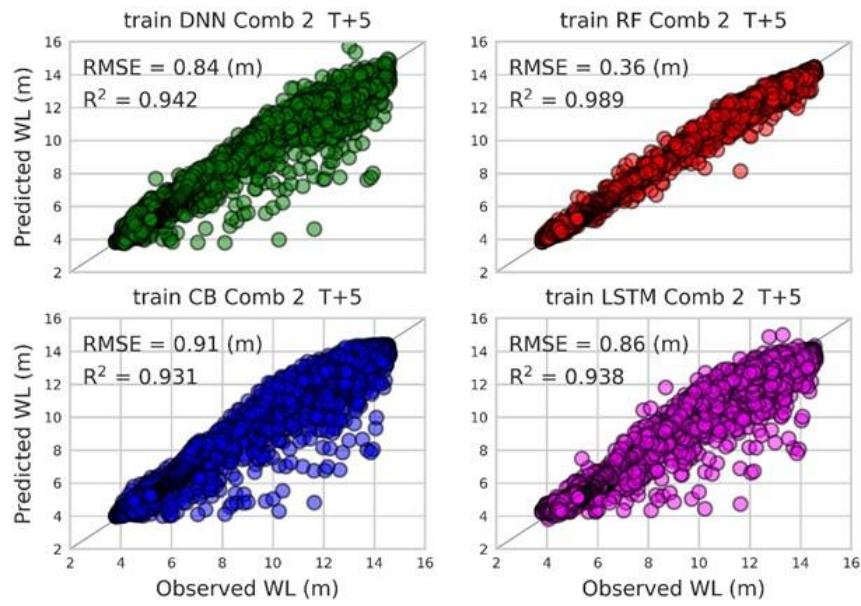
A. 16 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 2, Lead time 7 days)



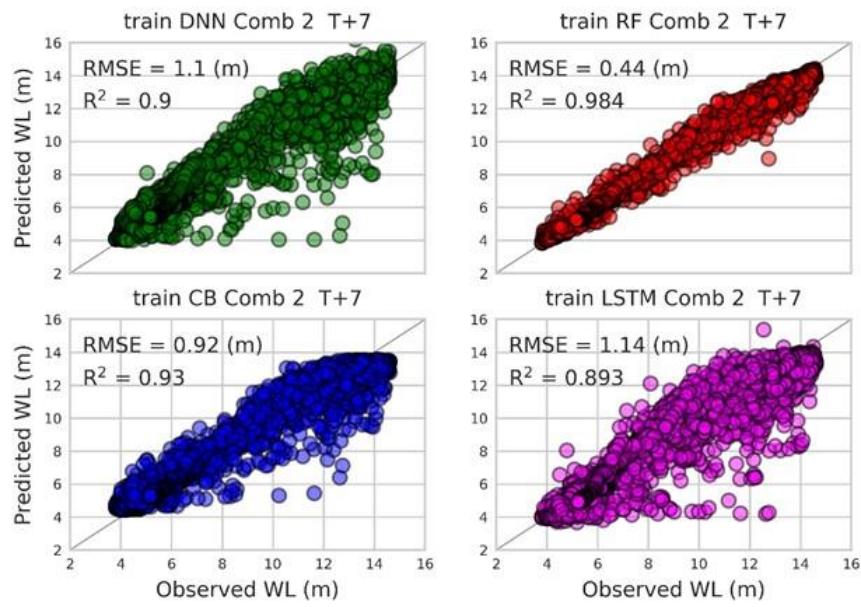
A. 17 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 2, Lead time 1 days)



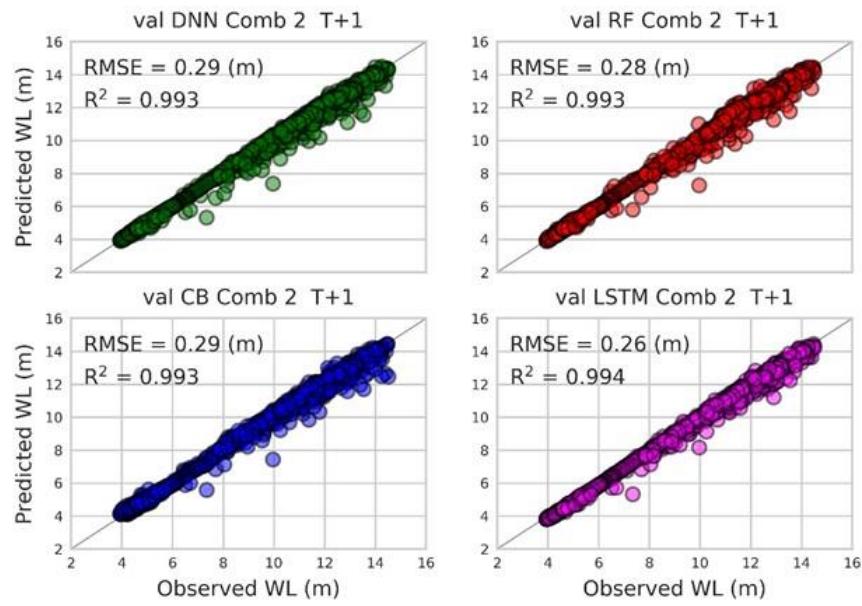
A. 18 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 2, Lead time 3 days)



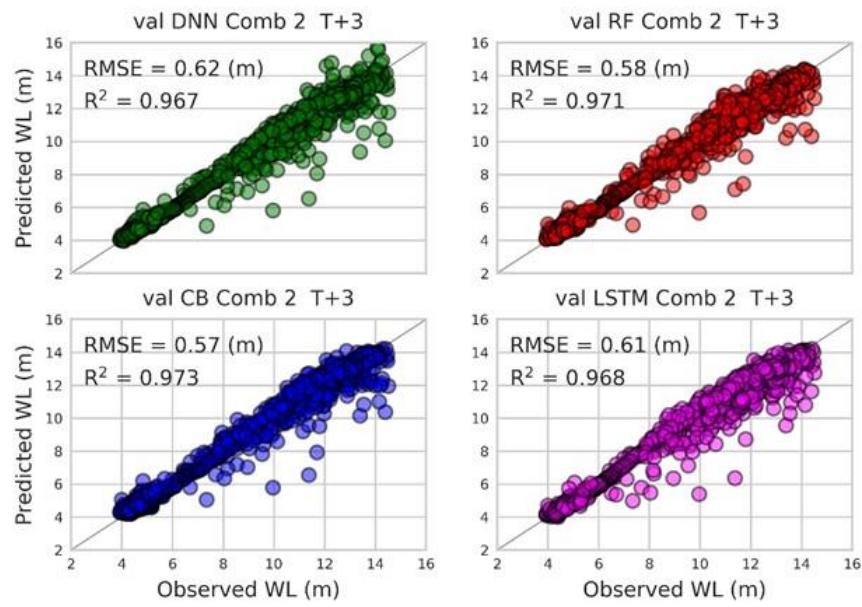
A. 19 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 2, Lead time 5 days)



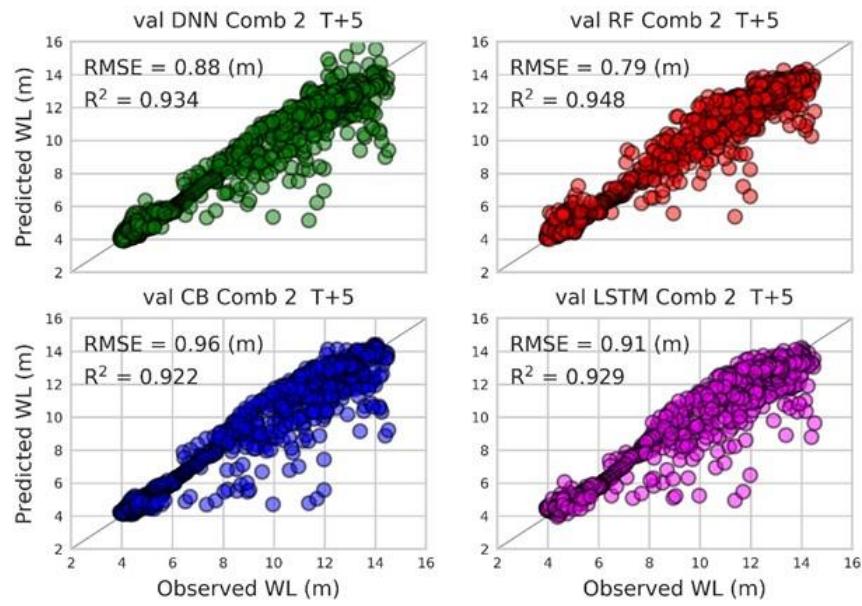
A. 20 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 2, Lead time 7 days)



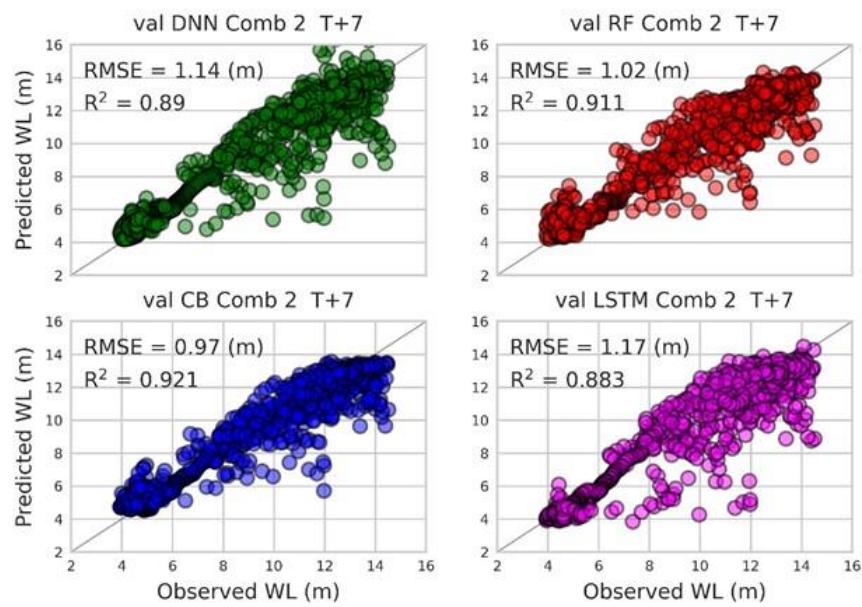
A. 21 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 2, Lead time 1 days)



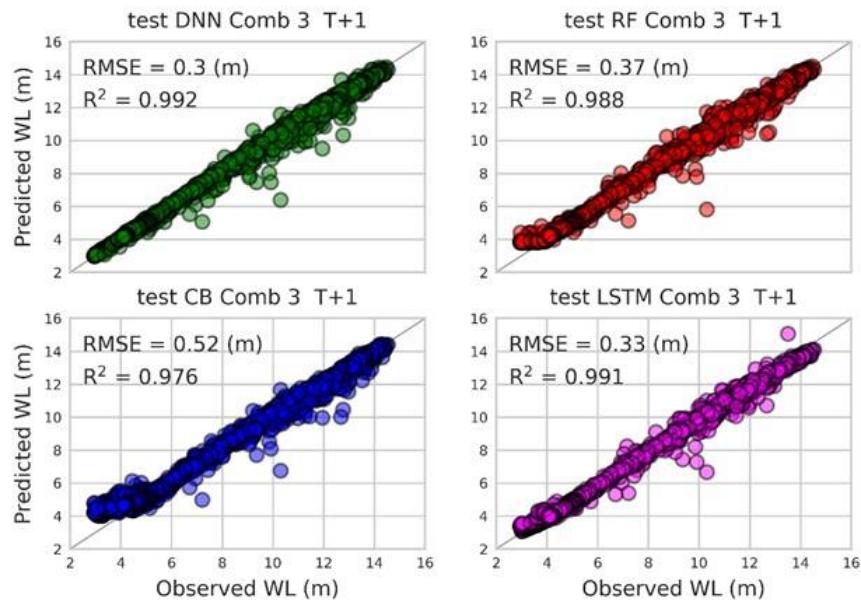
A. 22 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 2, Lead time 3 days)



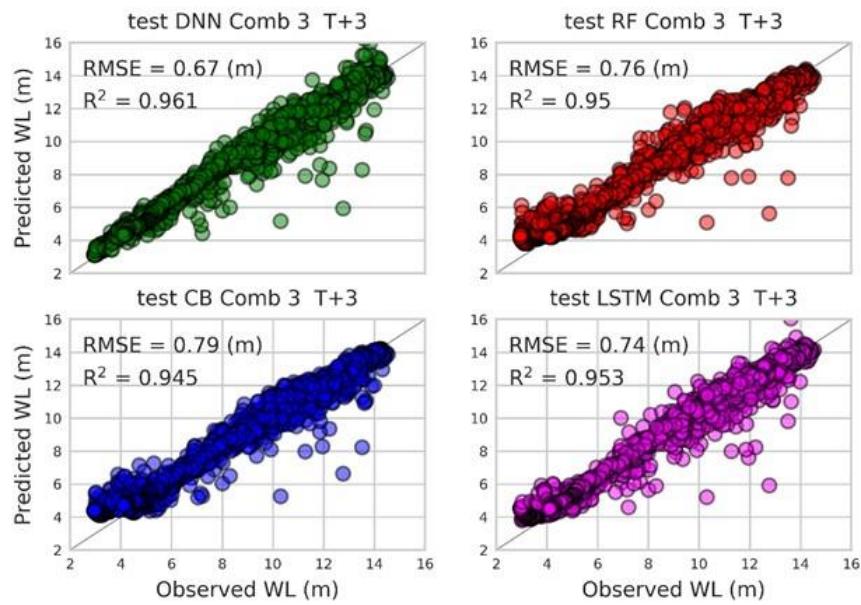
A. 23 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 2, Lead time 5 days)



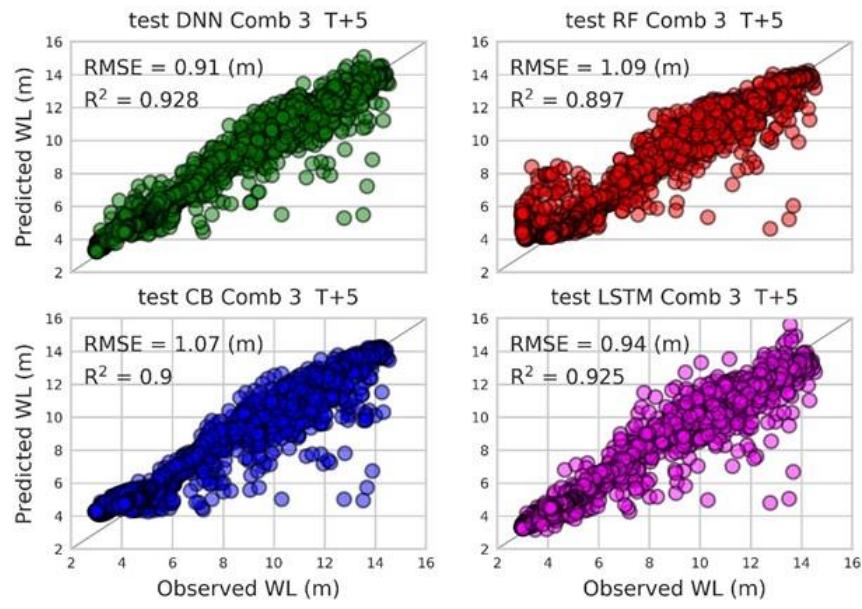
A. 24 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 2, Lead time 7 days)



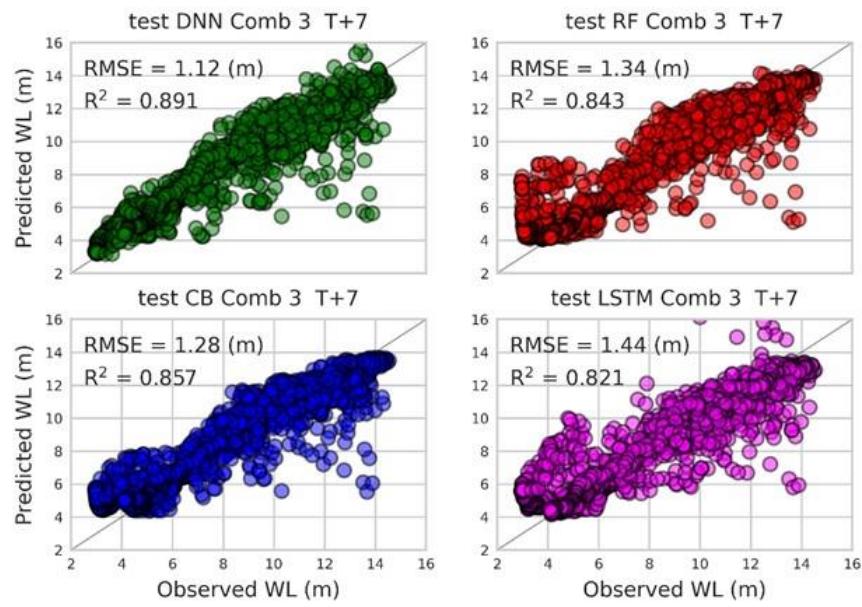
A. 25 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 3, Lead time 1 days)



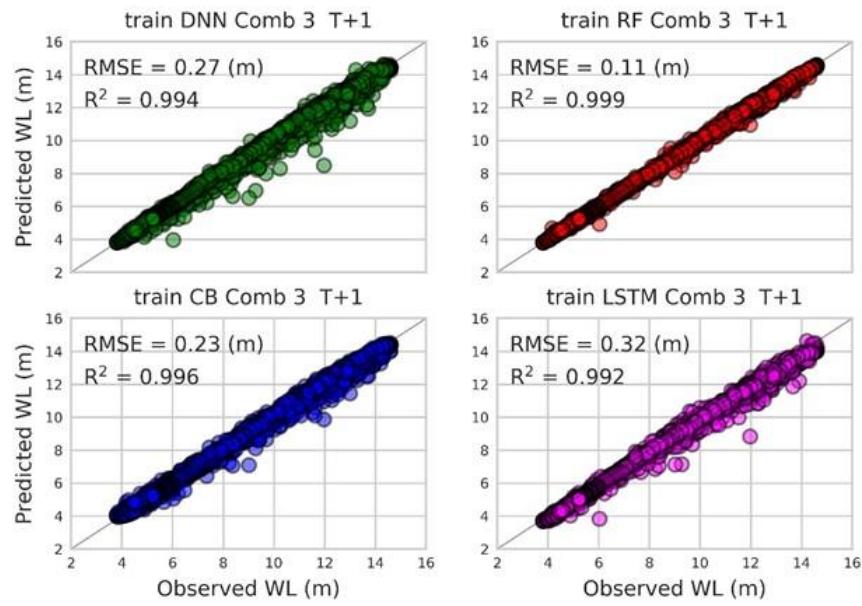
A. 26 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 3, Lead time 3 days)



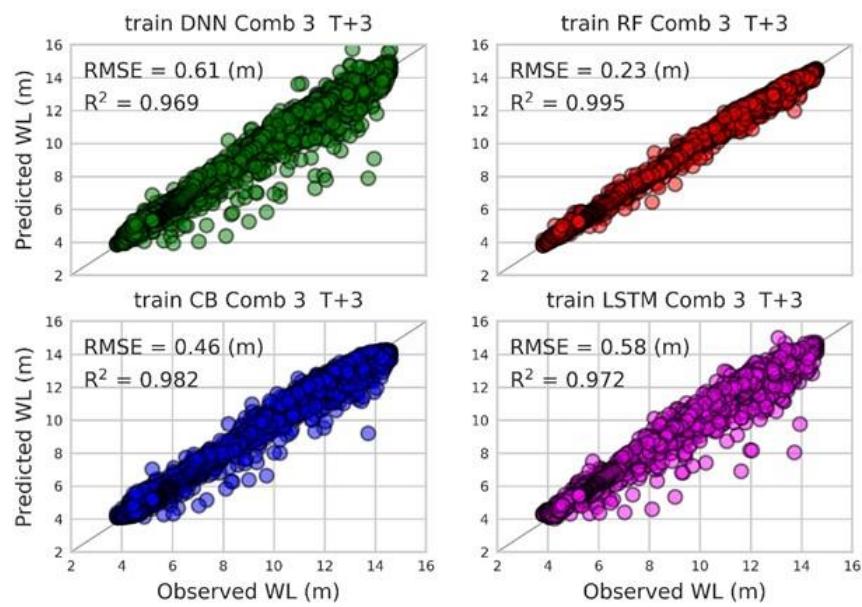
A. 27 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 3, Lead time 5 days)



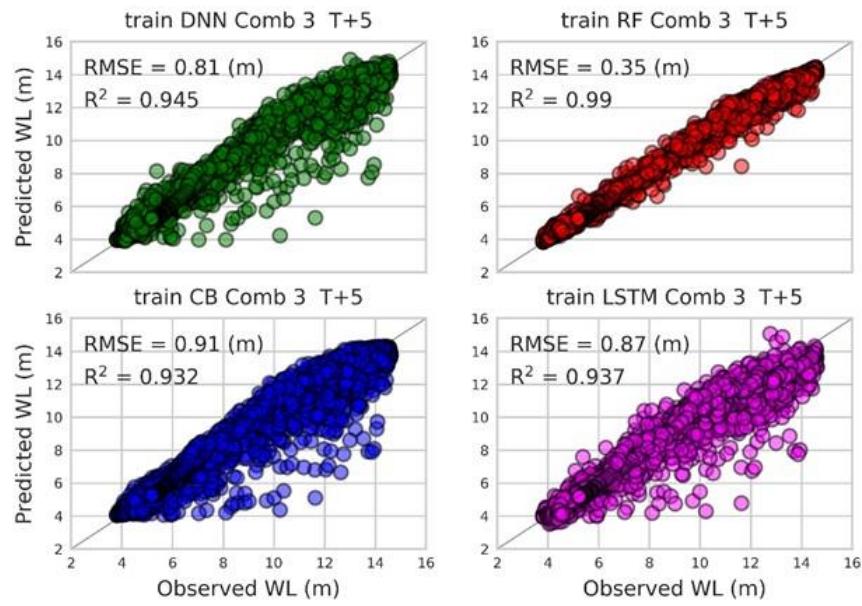
A. 28 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 3, Lead time 7 days)



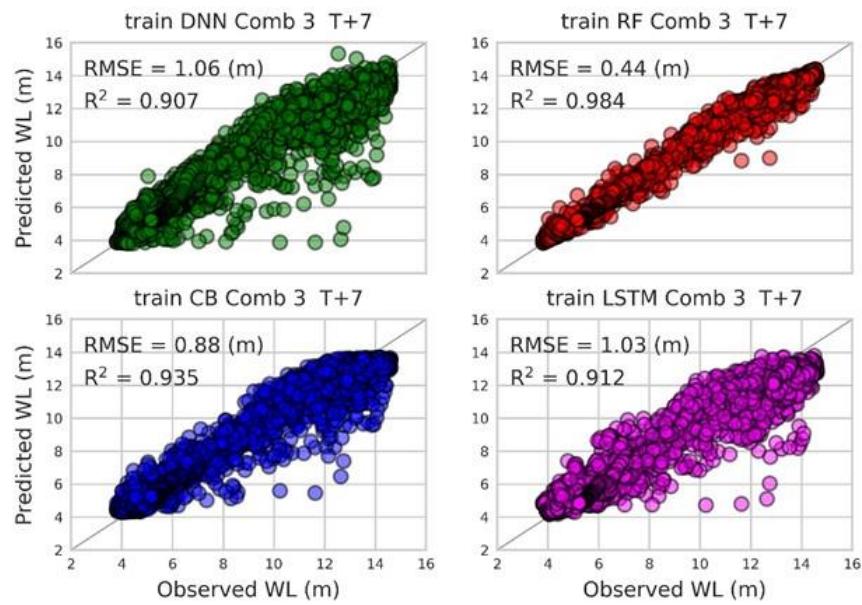
A. 29 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 3, Lead time 1 days)



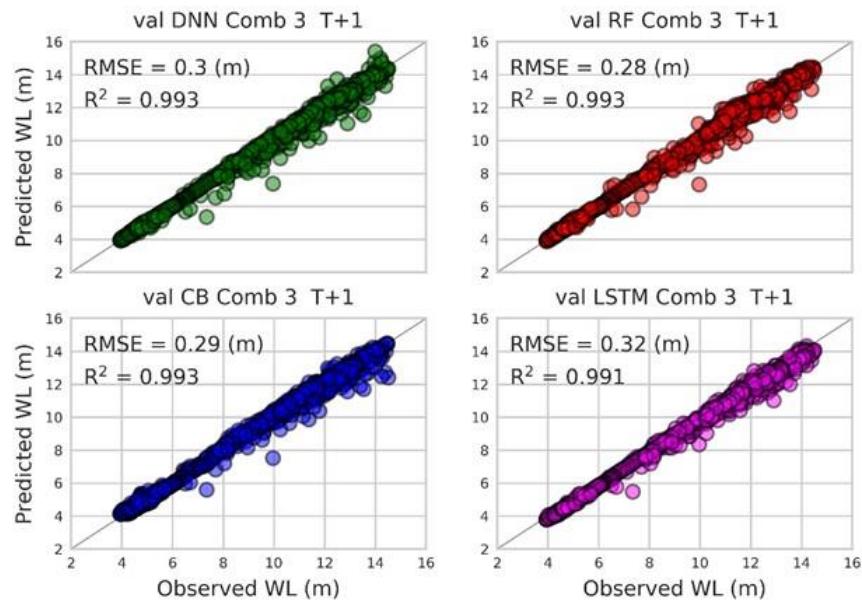
A. 30 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 3, Lead time 3 days)



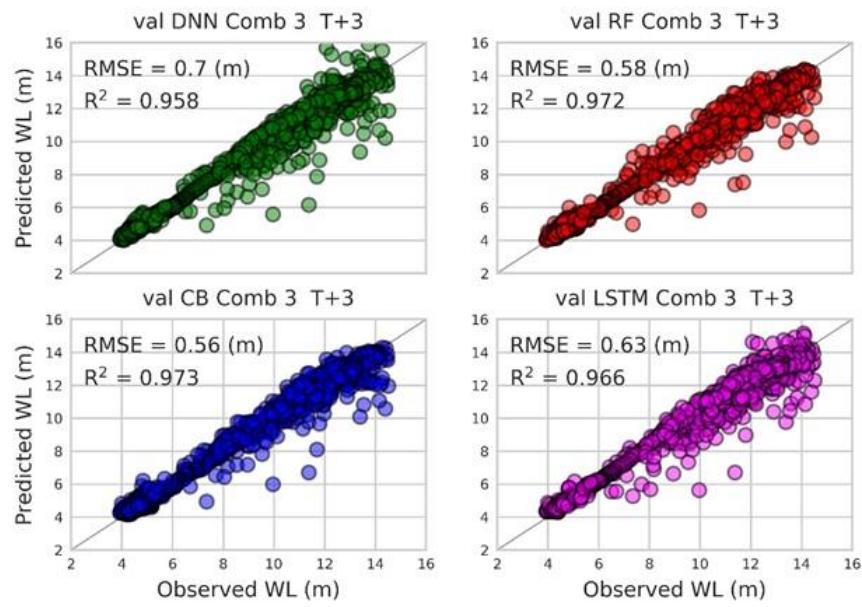
A. 31 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 3, Lead time 5 days)



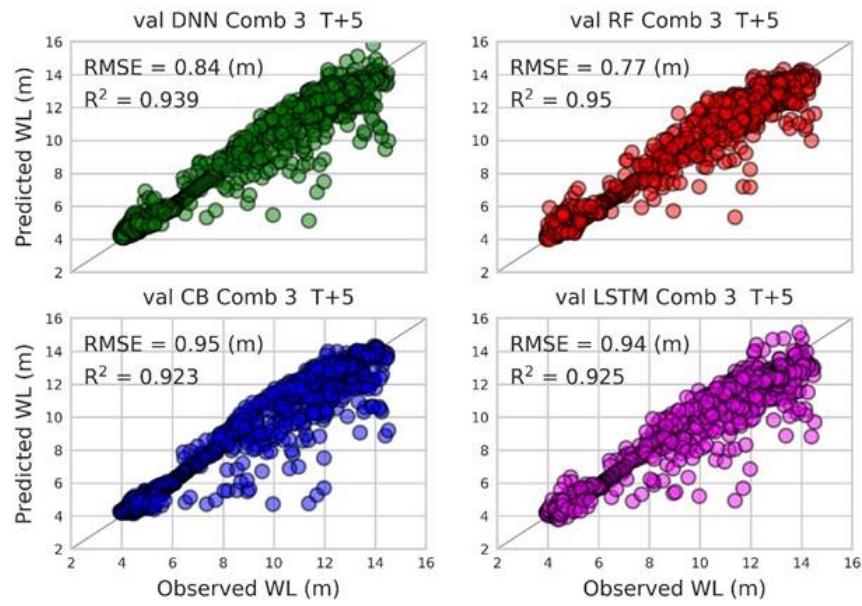
A. 32 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 3, Lead time 7 days)



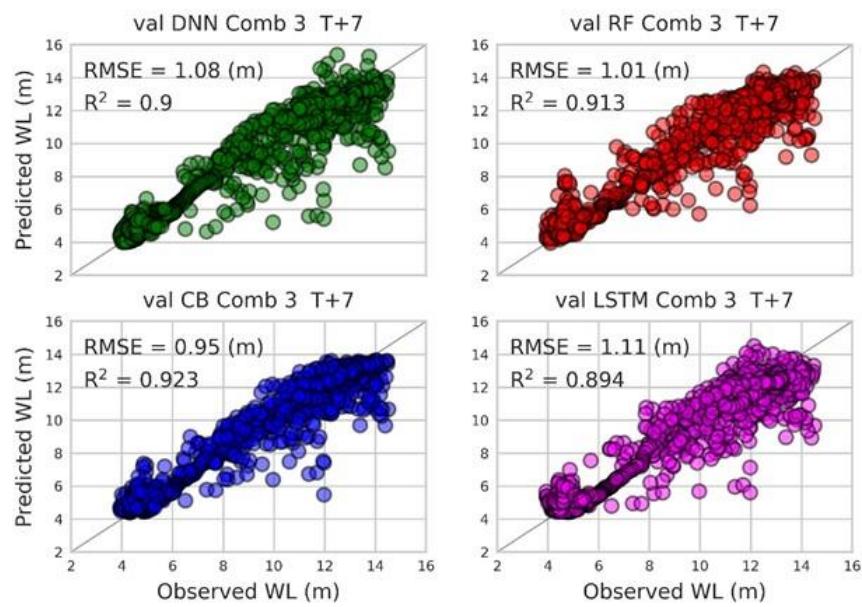
A. 33 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 3, Lead time 1 days)



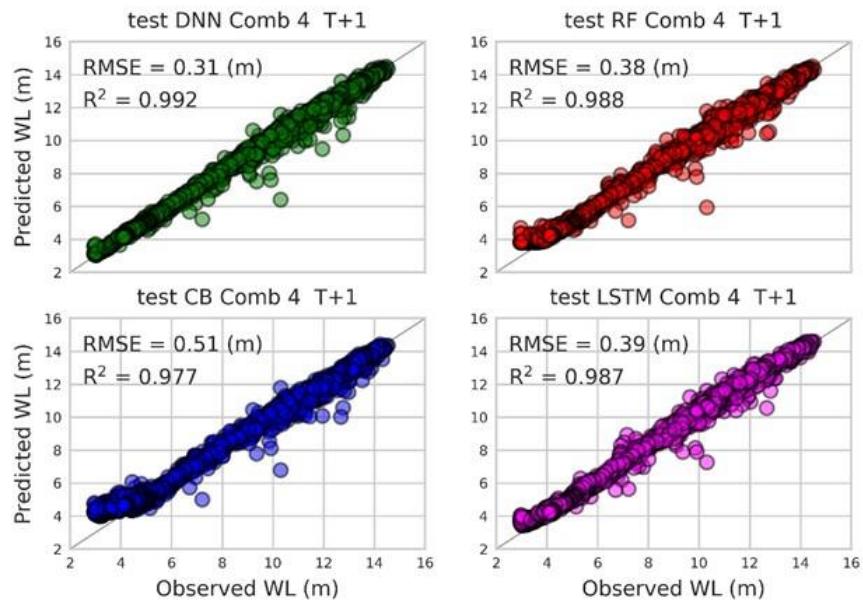
A. 34 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 3, Lead time 3 days)



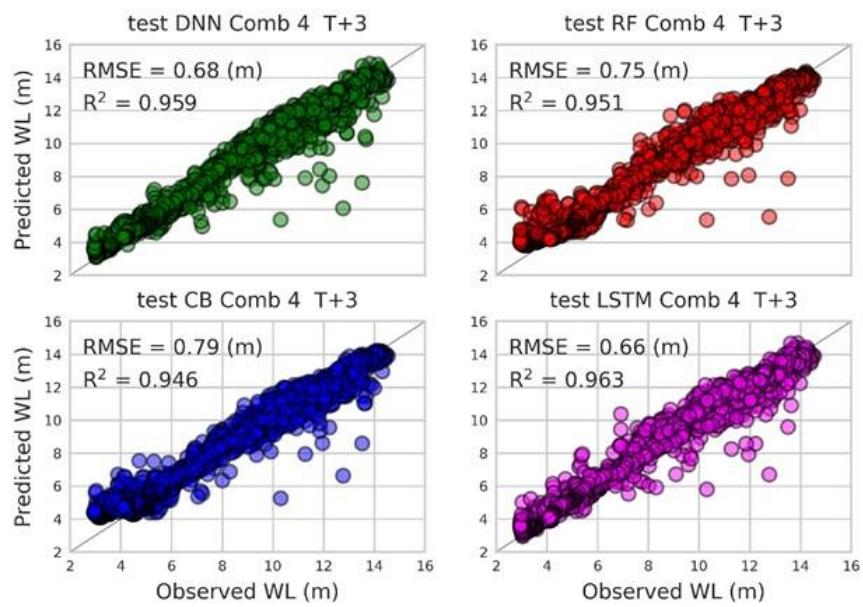
A. 35 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 3, Lead time 5 days)



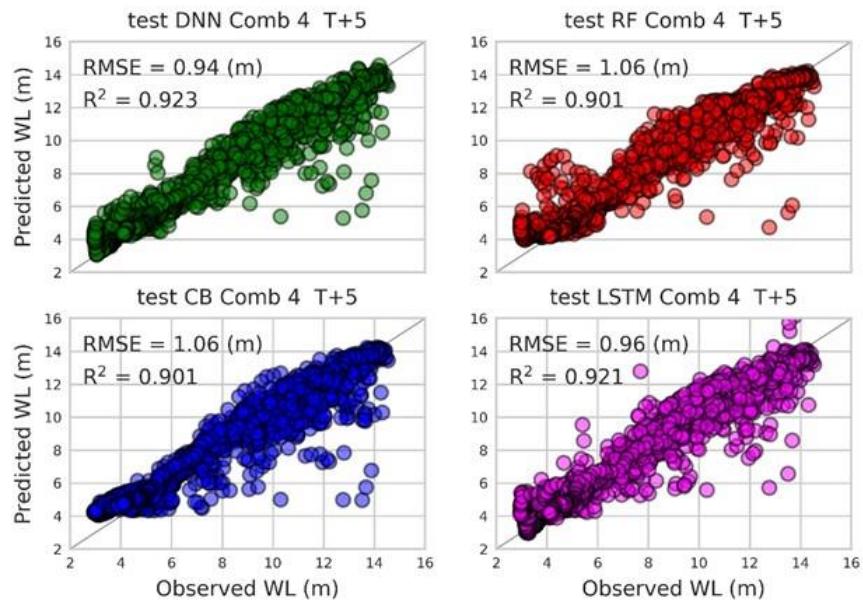
A. 36 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 3, Lead time 7 days)



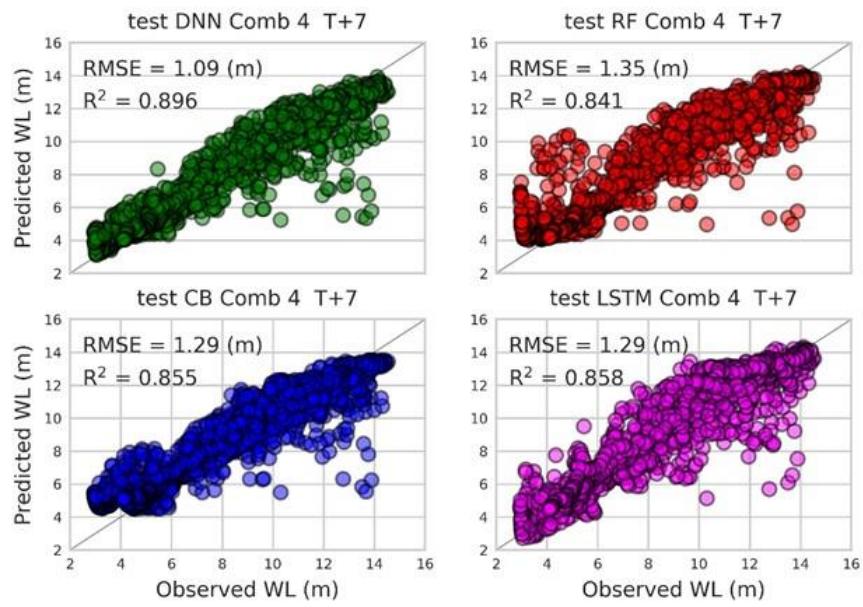
A. 37 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 4, Lead time 1 days)



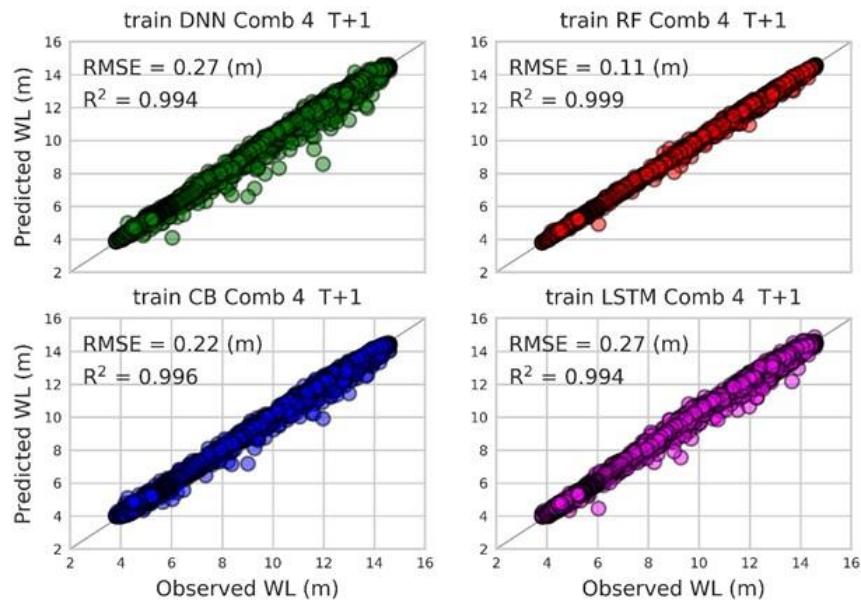
A. 38 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 4, Lead time 3 days)



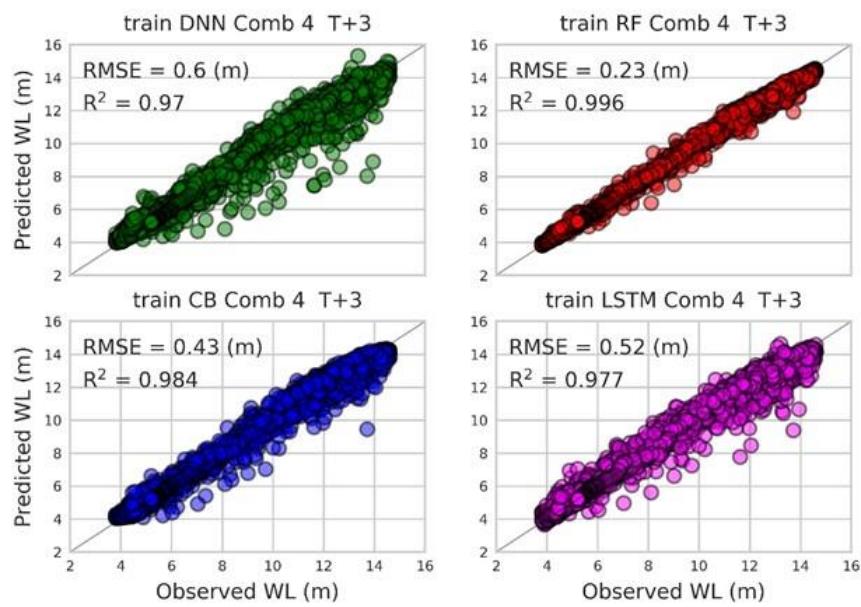
A. 39 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 4, Lead time 5 days)



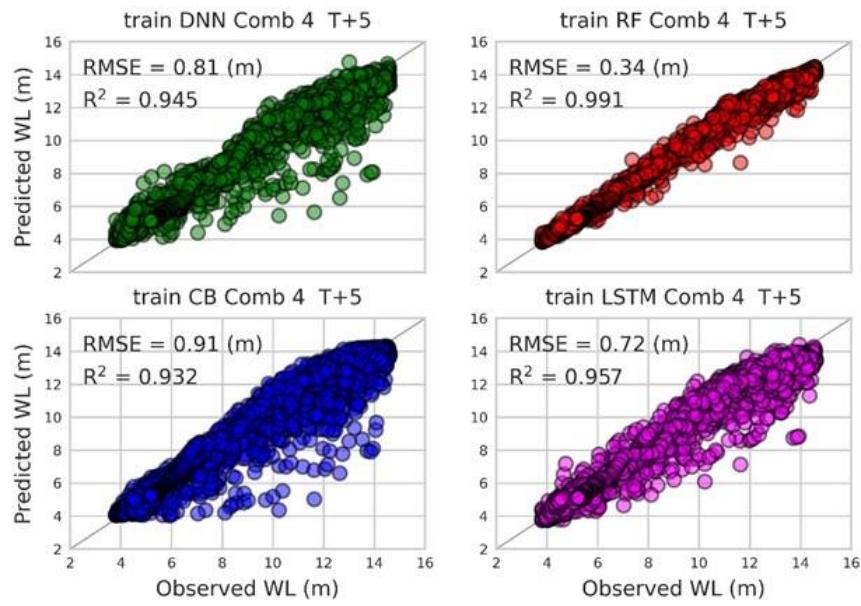
A. 40 Predicted WL(m) vs. Observed WL(m) for different models on test data. (Input combination 4, Lead time 7 days)



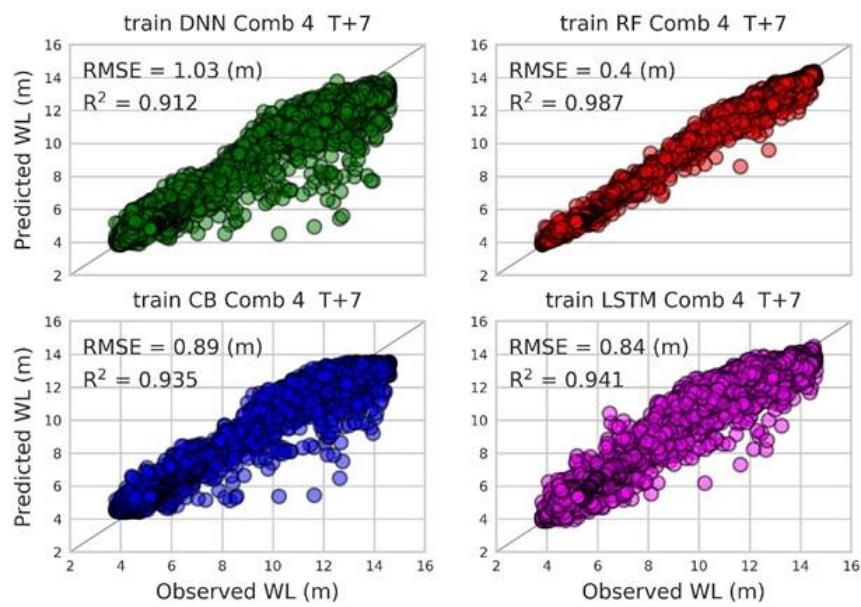
A. 41 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 4, Lead time 1 days)



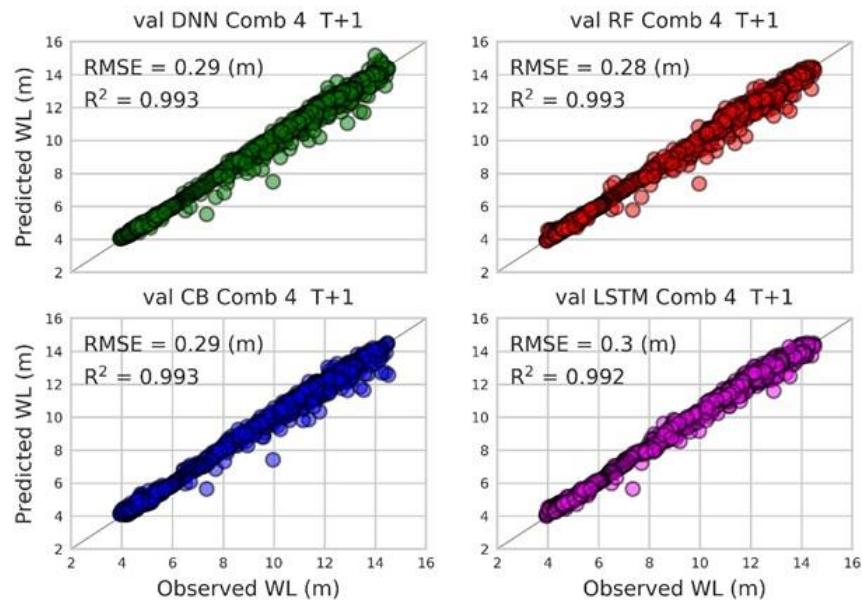
A. 42 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 4, Lead time 3 days)



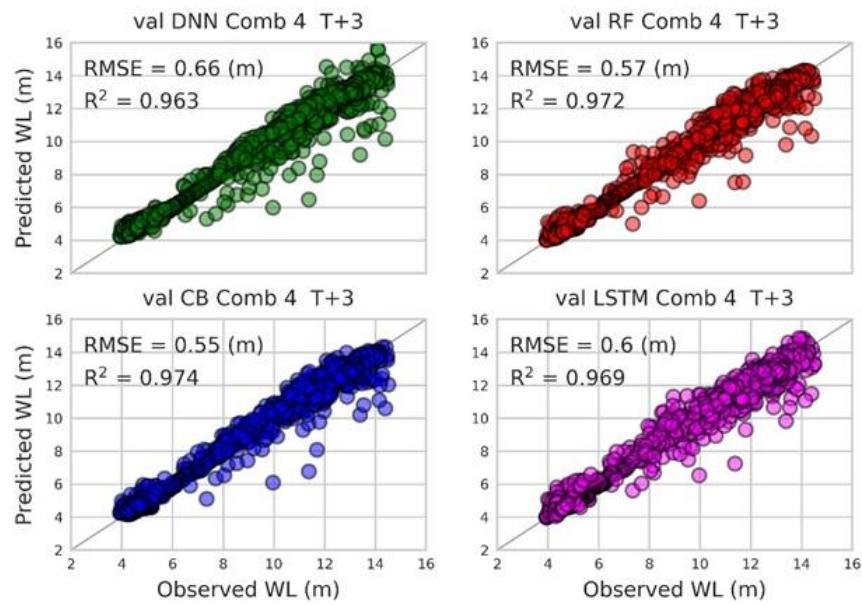
A. 43 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 4, Lead time 5 days)



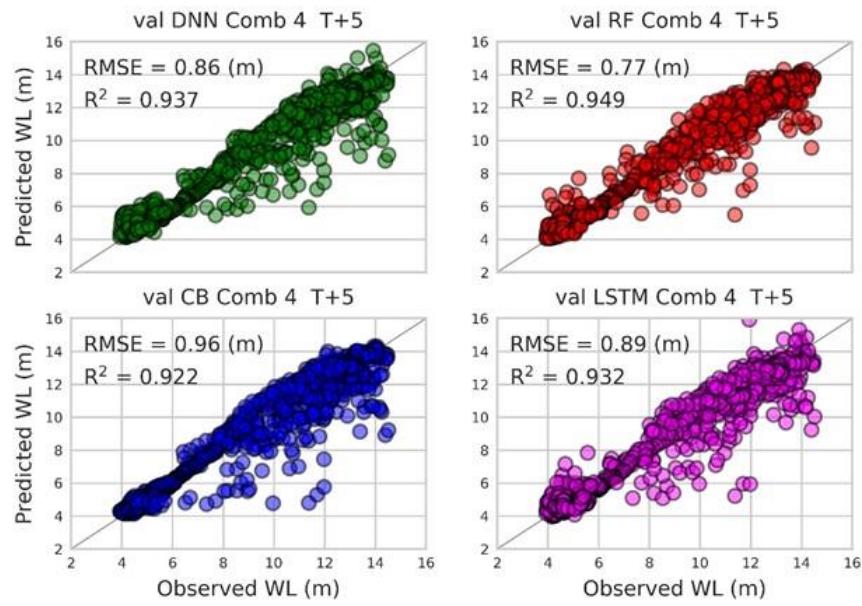
A. 44 Predicted WL(m) vs. Observed WL(m) for different models on train data. (Input combination 4, Lead time 7 days)



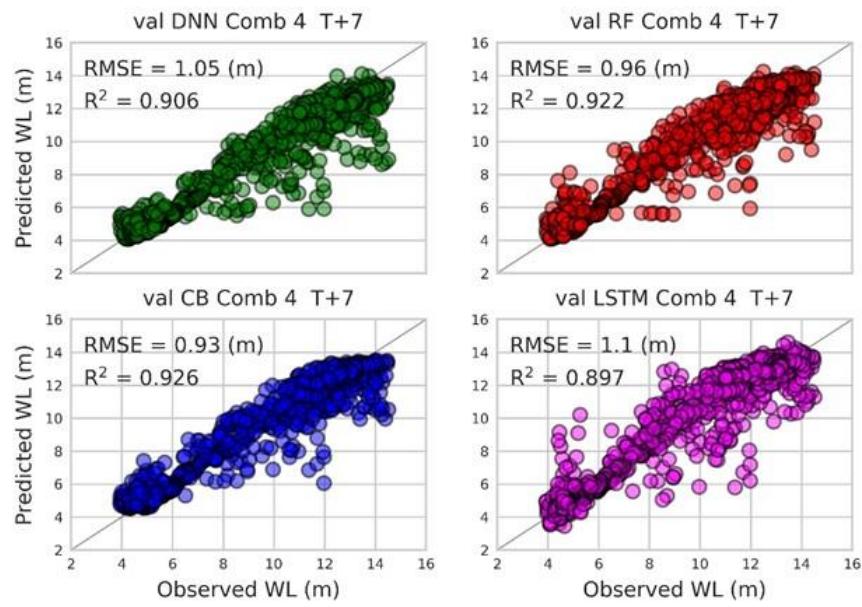
A. 45 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 4, Lead time 1 days)



A. 46 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 4, Lead time 3 days)

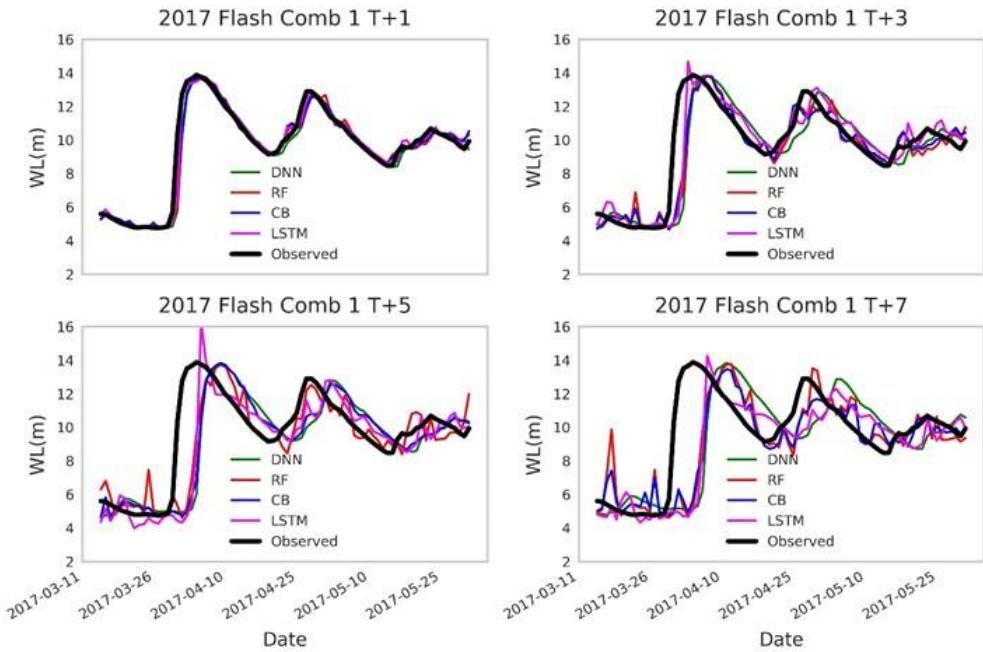


A. 47 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 4, Lead time 5 days)

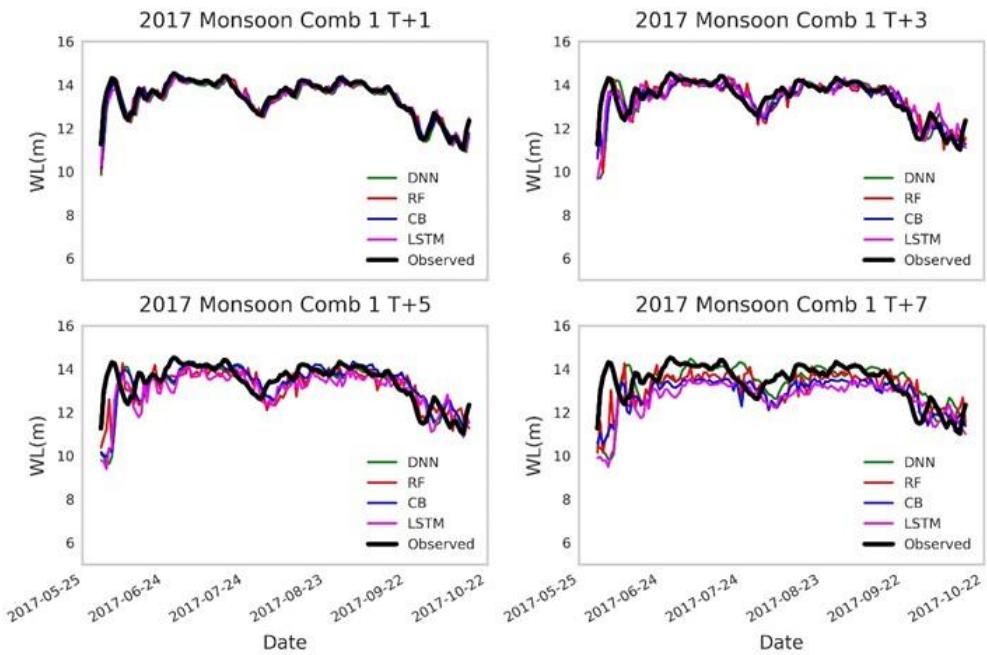


A. 48 Predicted WL(m) vs. Observed WL(m) for different models on validation data. (Input combination 4, Lead time 7 days)

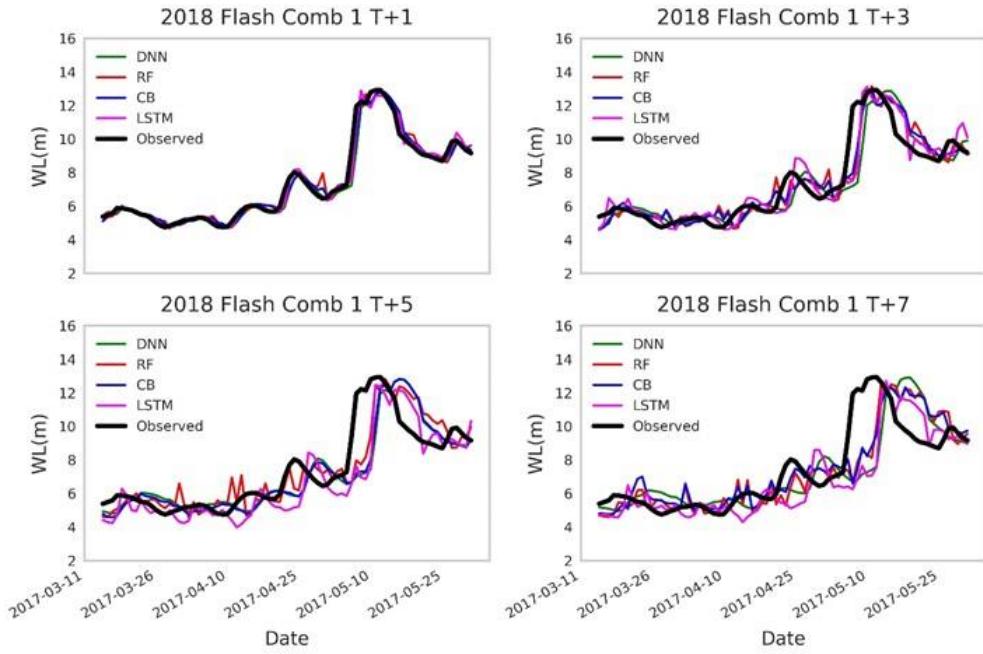
Appendix B



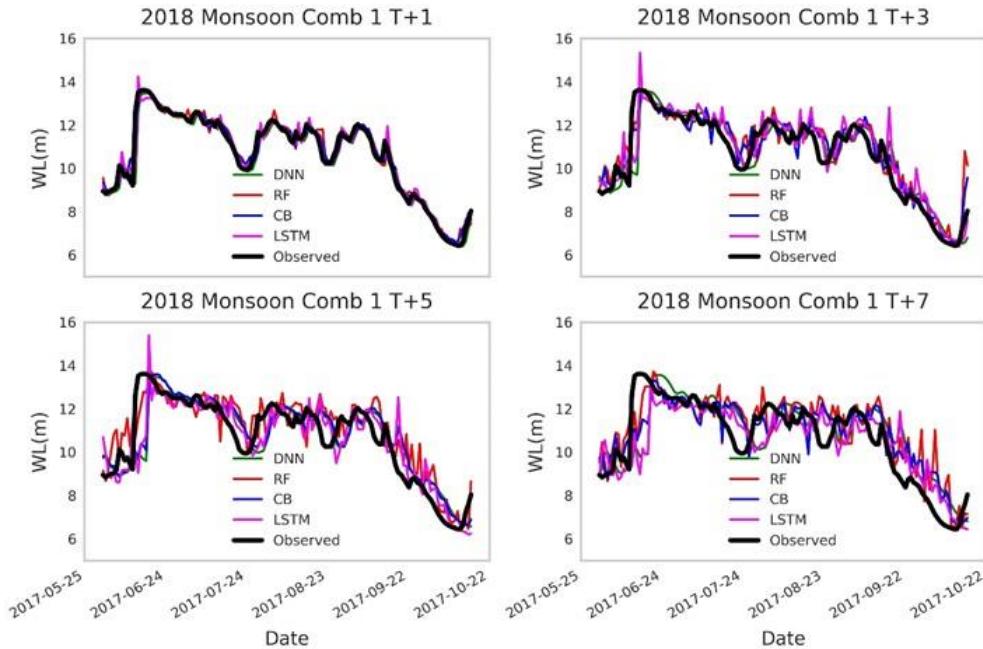
B. 1 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2017 Flash Flood



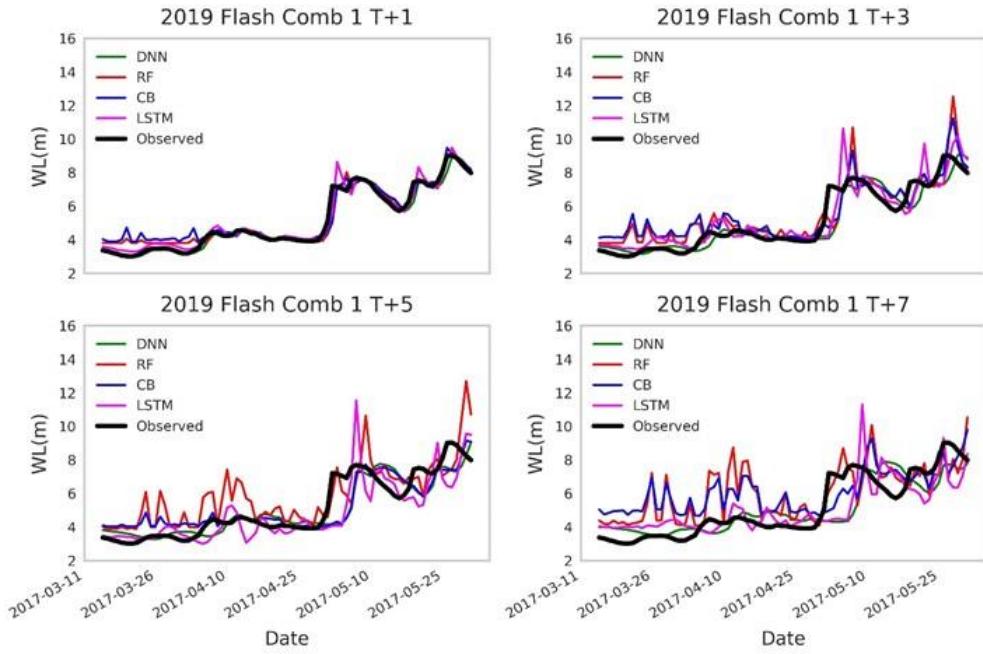
B. 2 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2017 Monsoon Flood



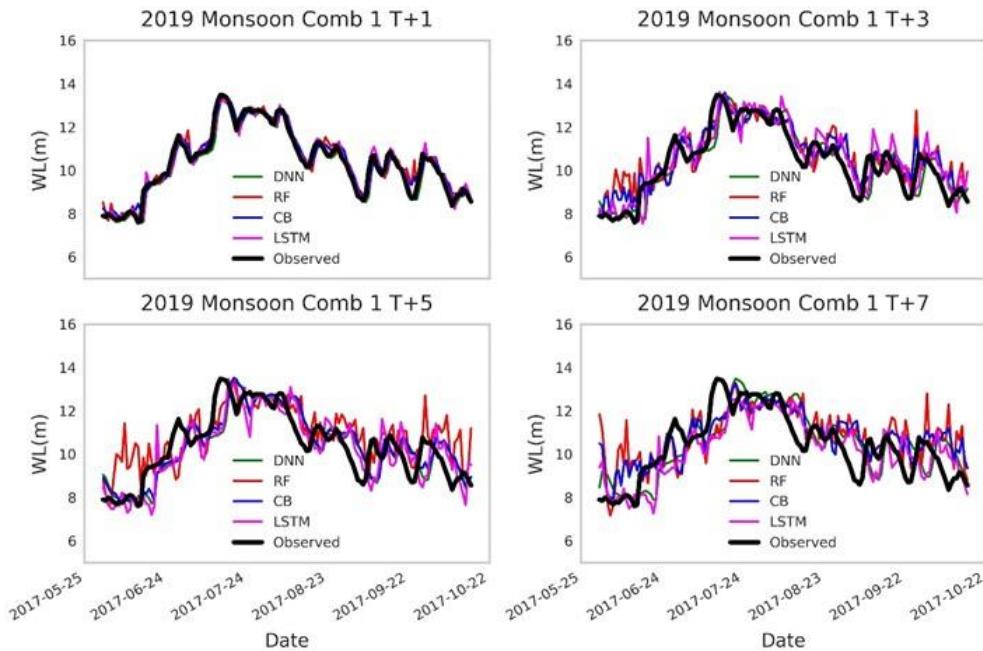
B. 3 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2018 Flash Flood



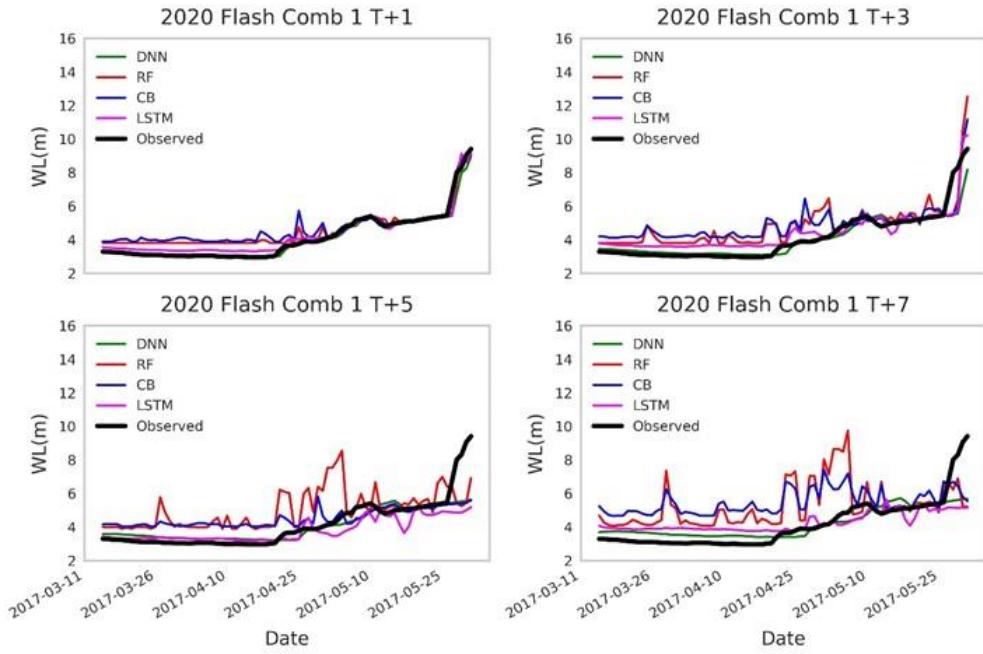
B. 4 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2018 Monsoon Flood



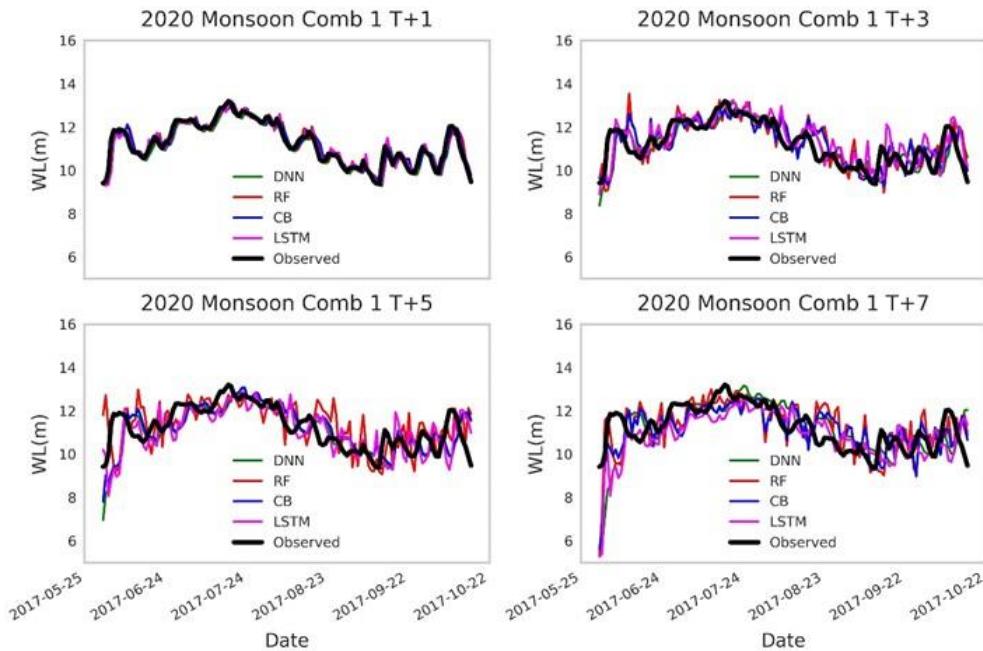
B. 5 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2019 Flash Flood



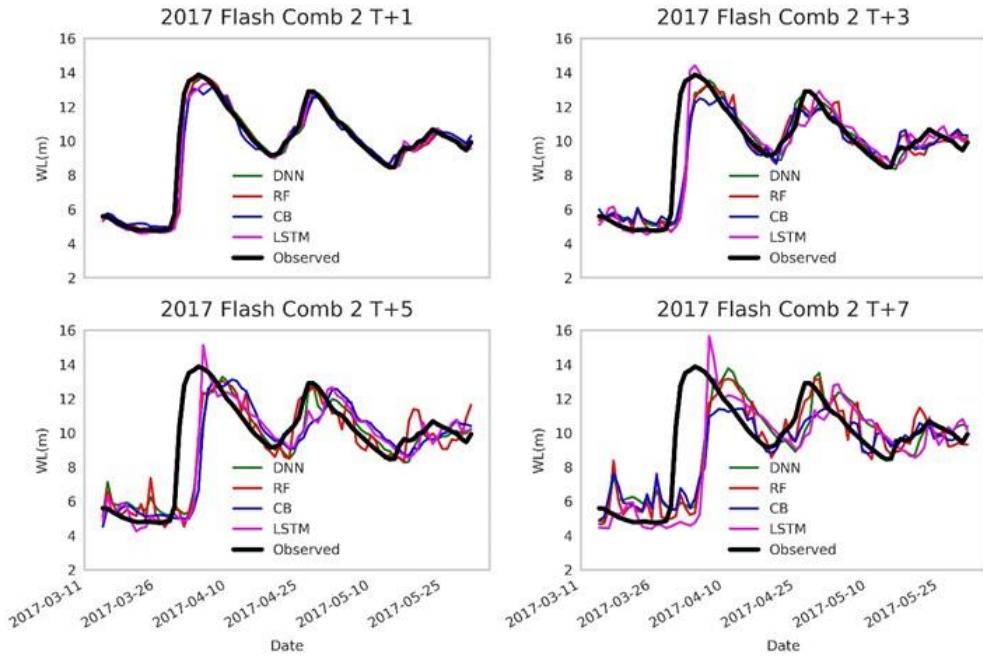
B. 6 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2019 Monsoon Flood



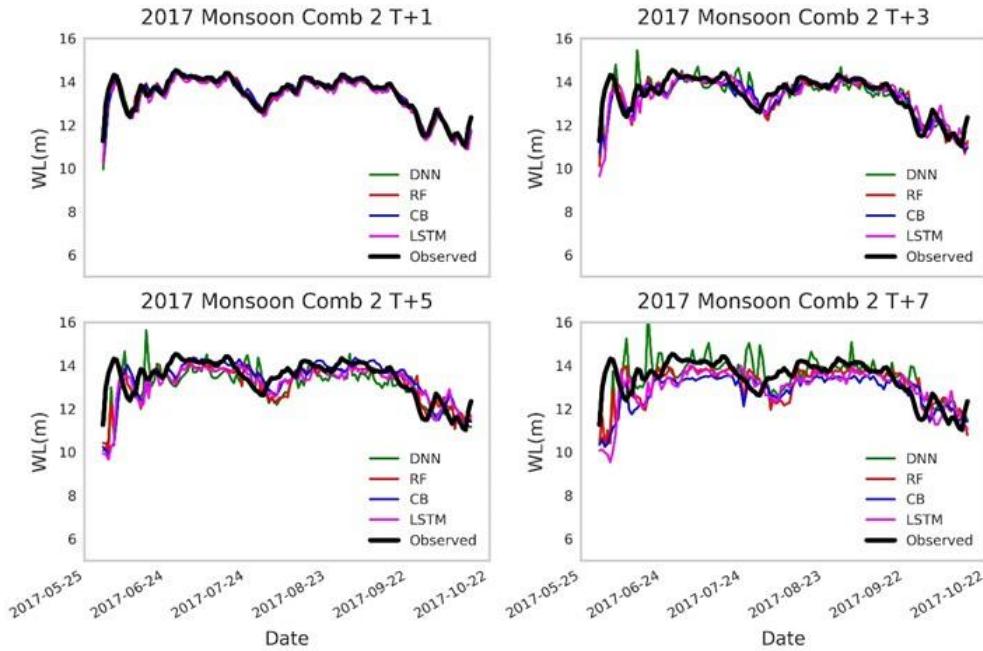
B. 7 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2020 Flash Flood



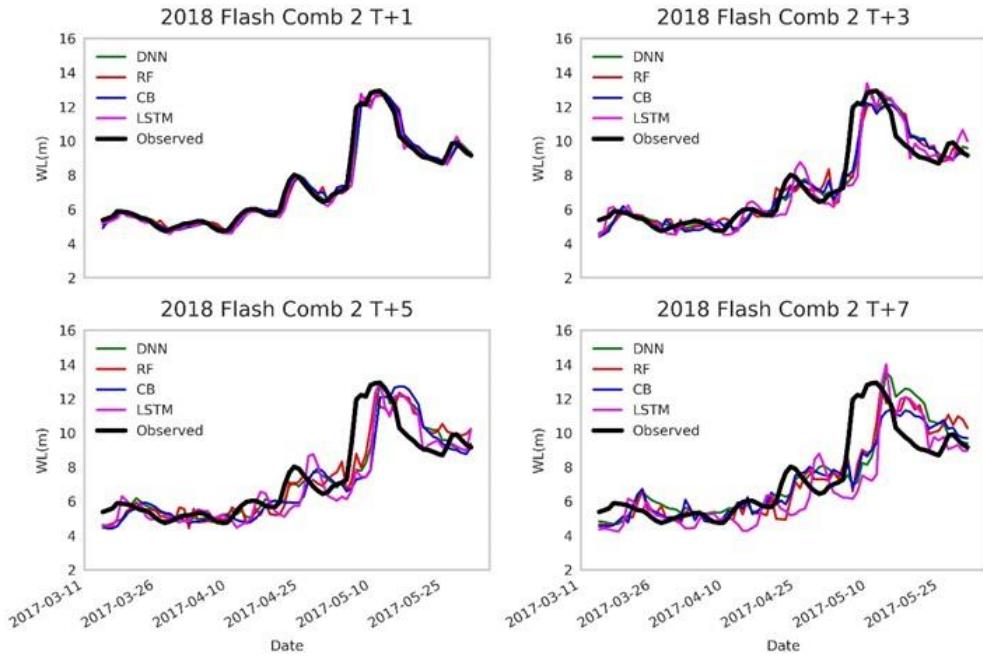
B. 8 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 1 on different time steps for 2020 Monsoon Flood



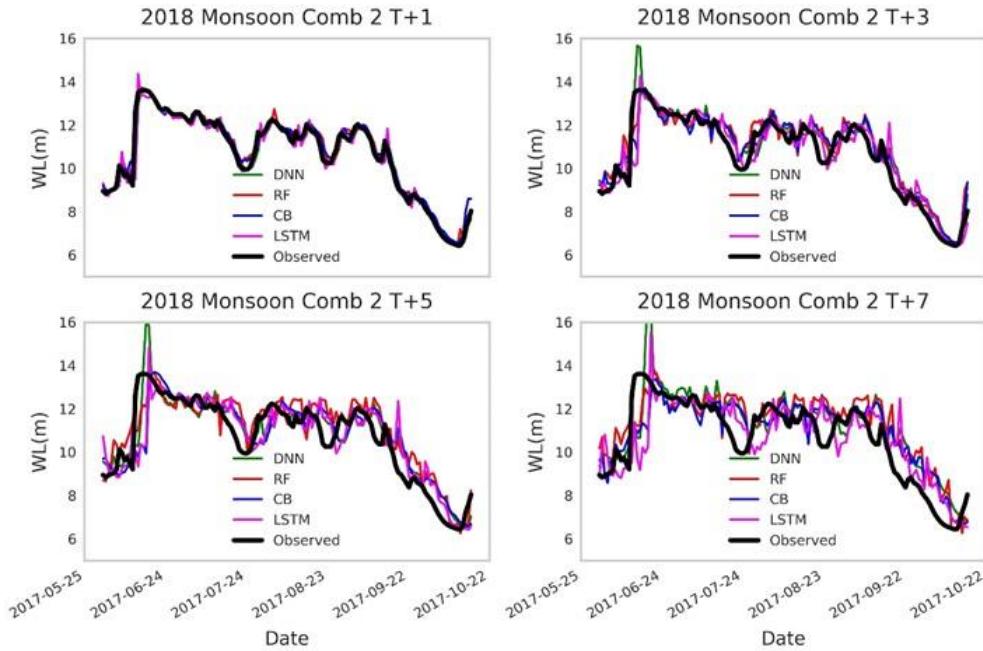
B. 9 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2017 Flash Flood



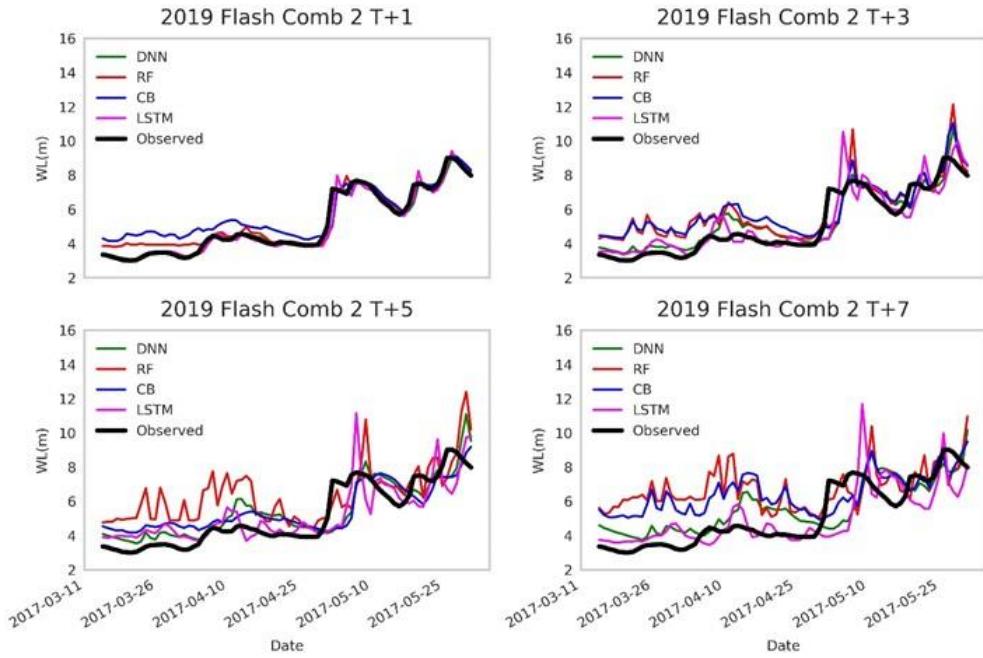
B. 10 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2017 Monsoon Flood



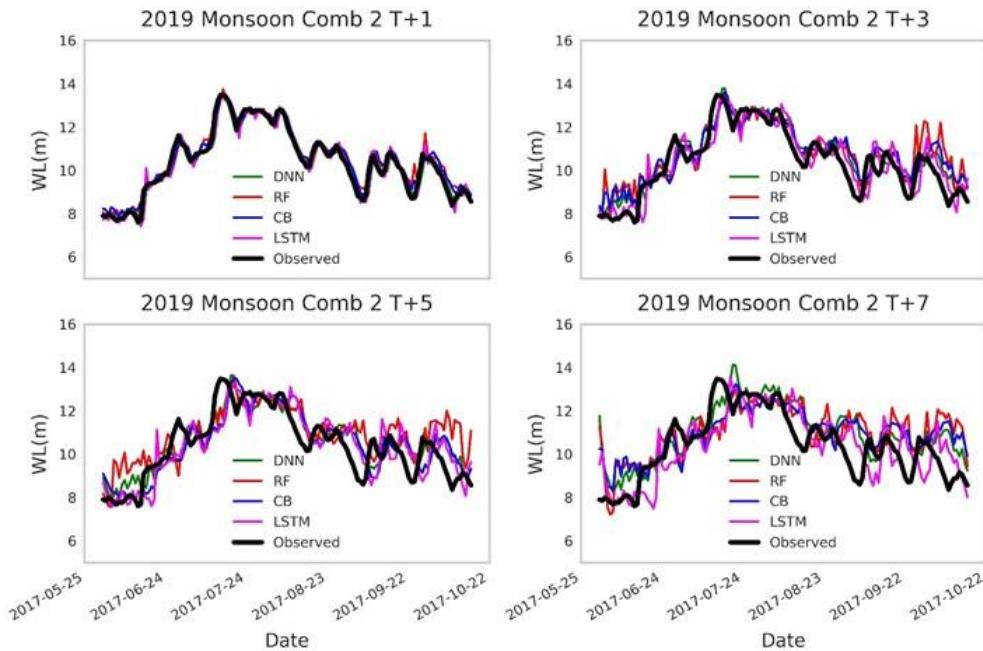
B. 11 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2018 Flash Flood



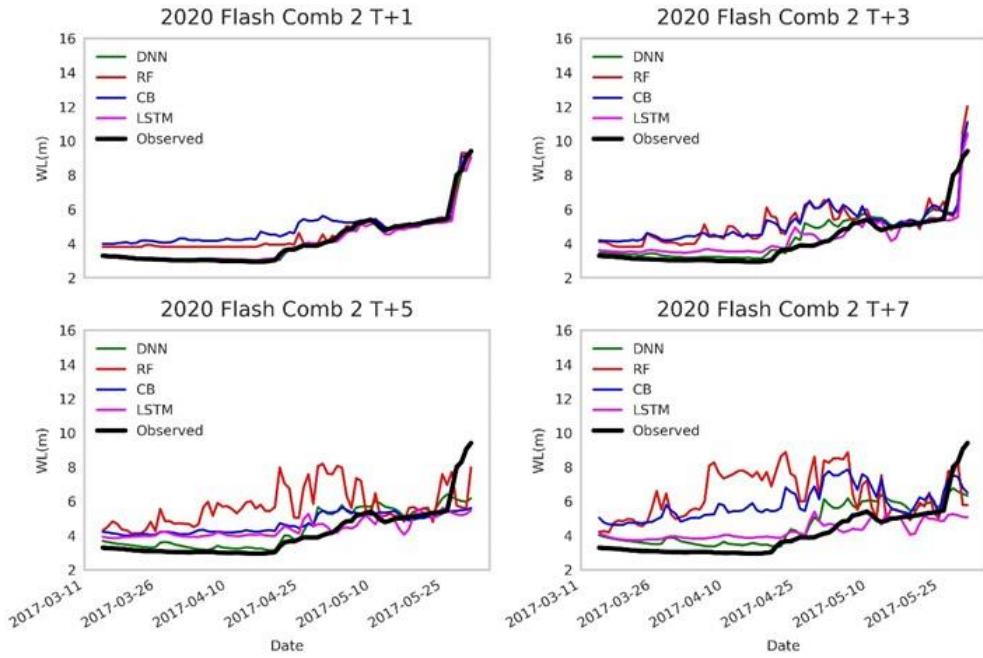
B. 12 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2018 Monsoon Flood



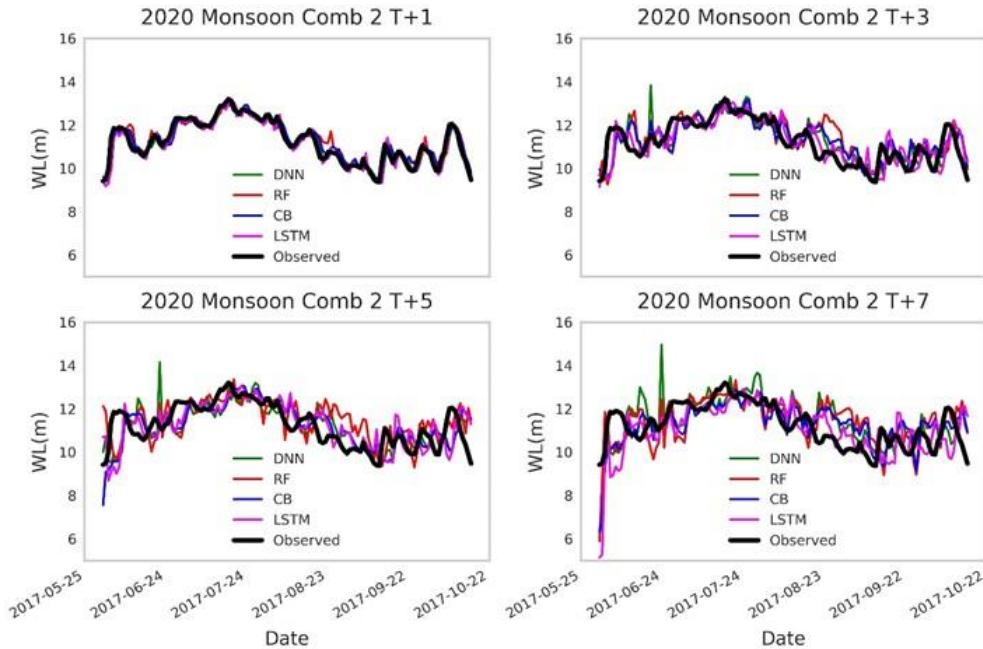
B. 13 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2019 Flash Flood



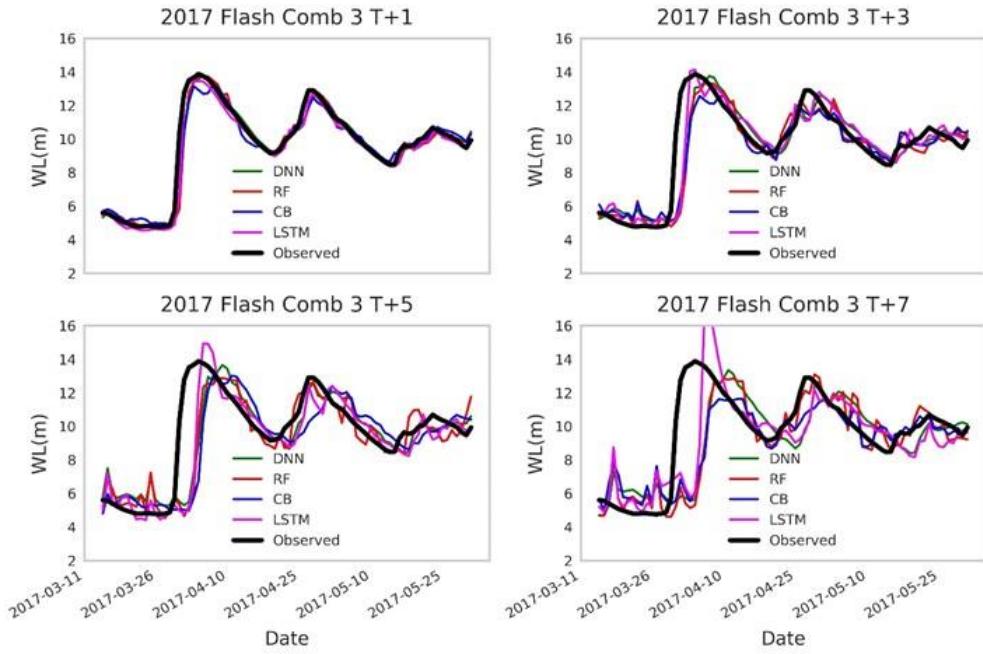
B. 14 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2019 Monsoon Flood



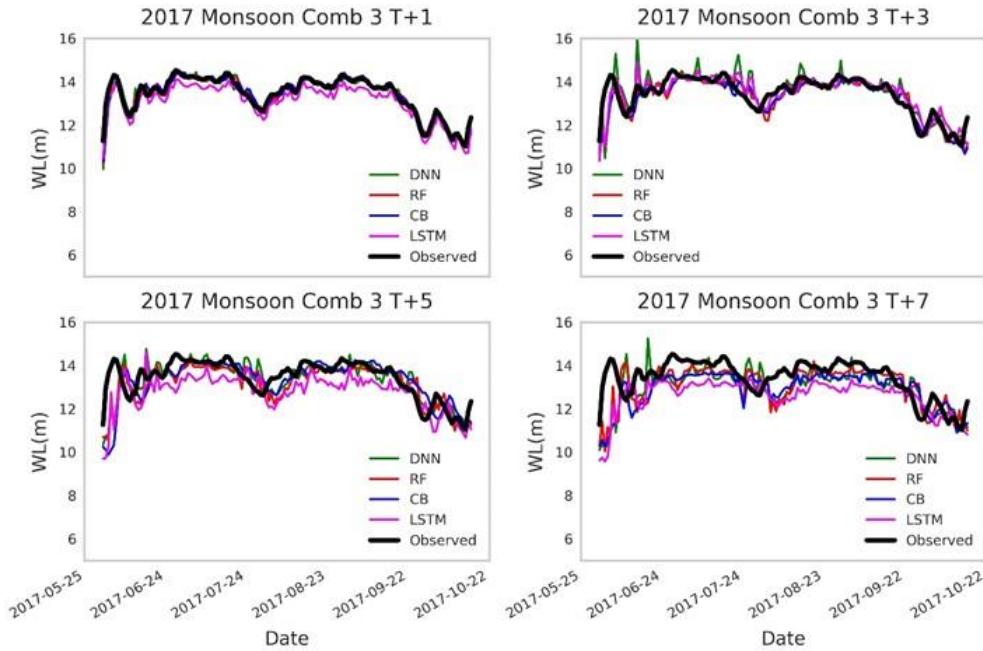
B. 15 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2020 Flash Flood



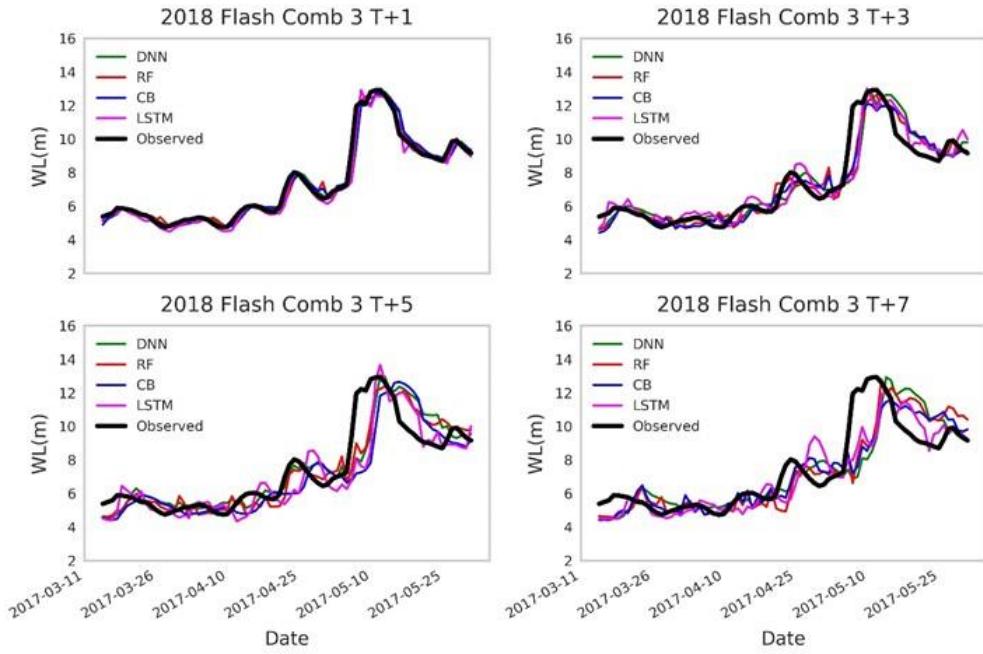
B. 16 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 2 on different time steps for 2020 Monsoon Flood



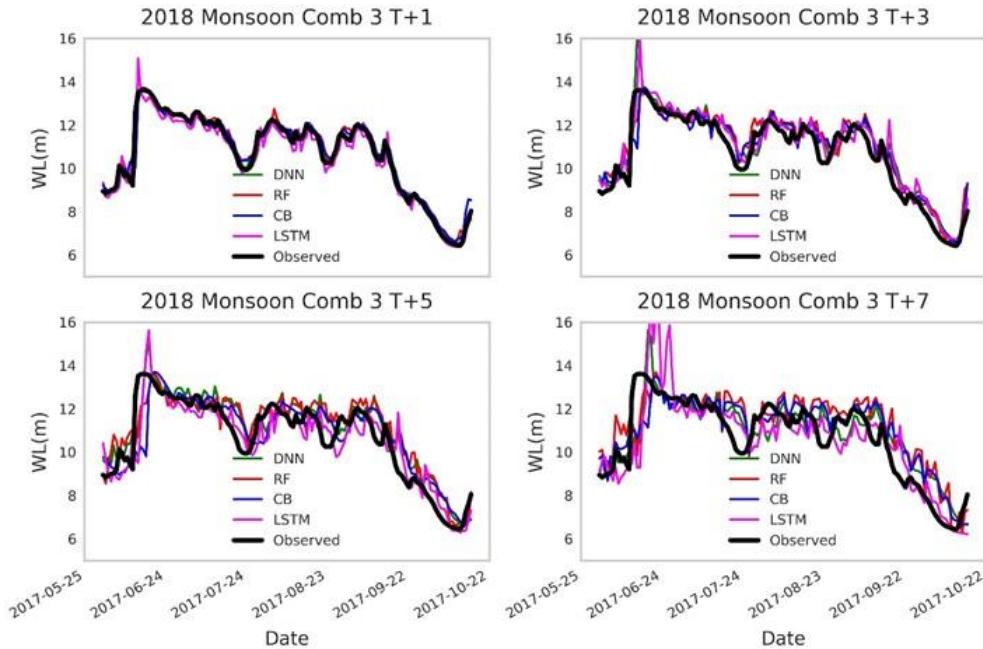
B. 17 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2017 Flash Flood



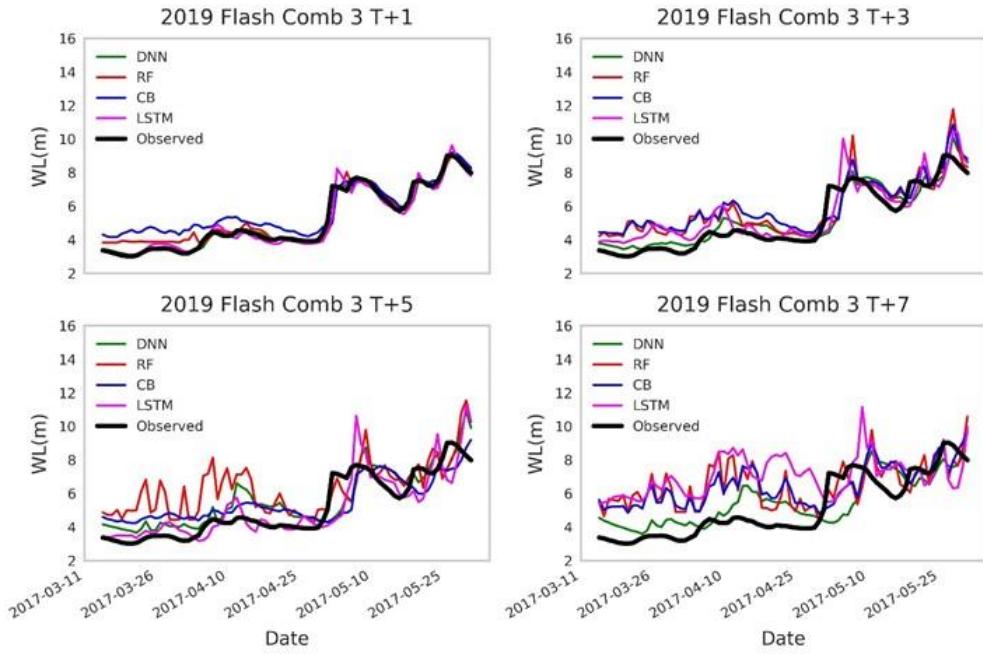
B. 18 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2017 Monsoon Flood



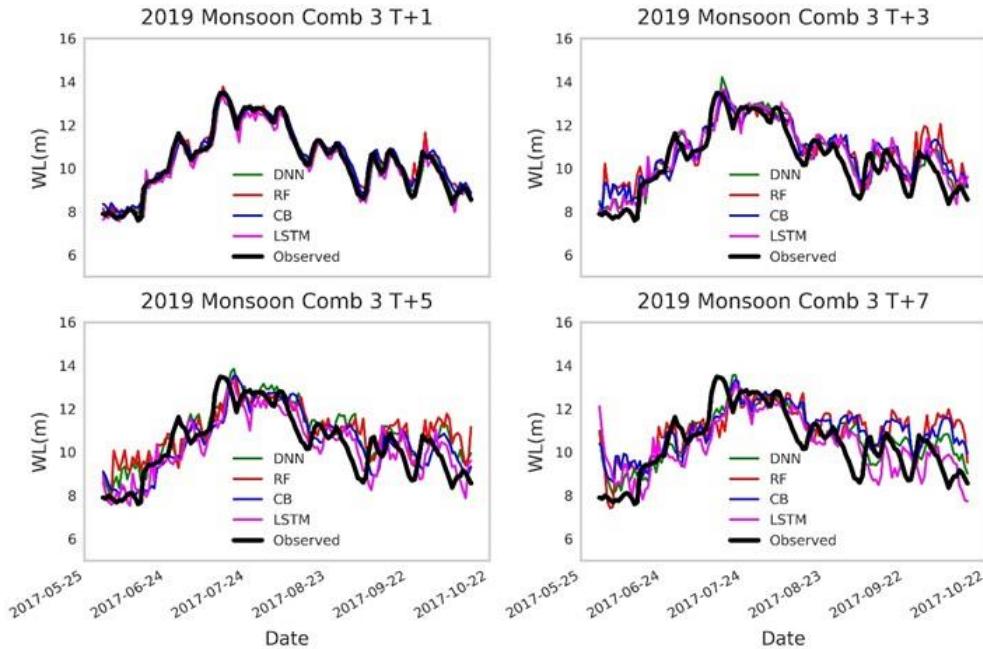
B. 19 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2018 Flash Flood



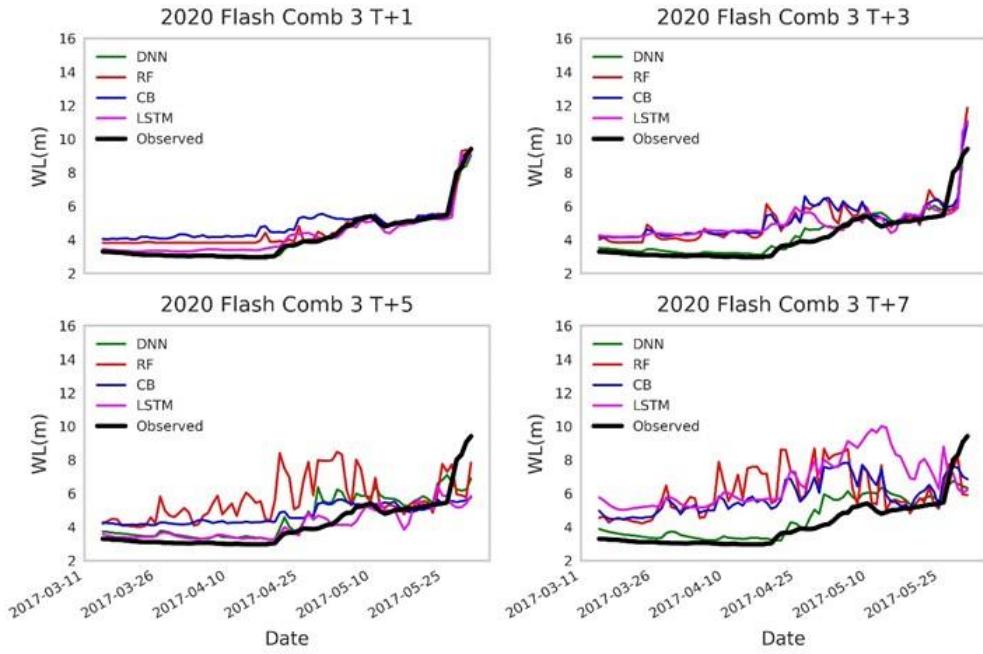
B. 20 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2018 Monsoon Flood



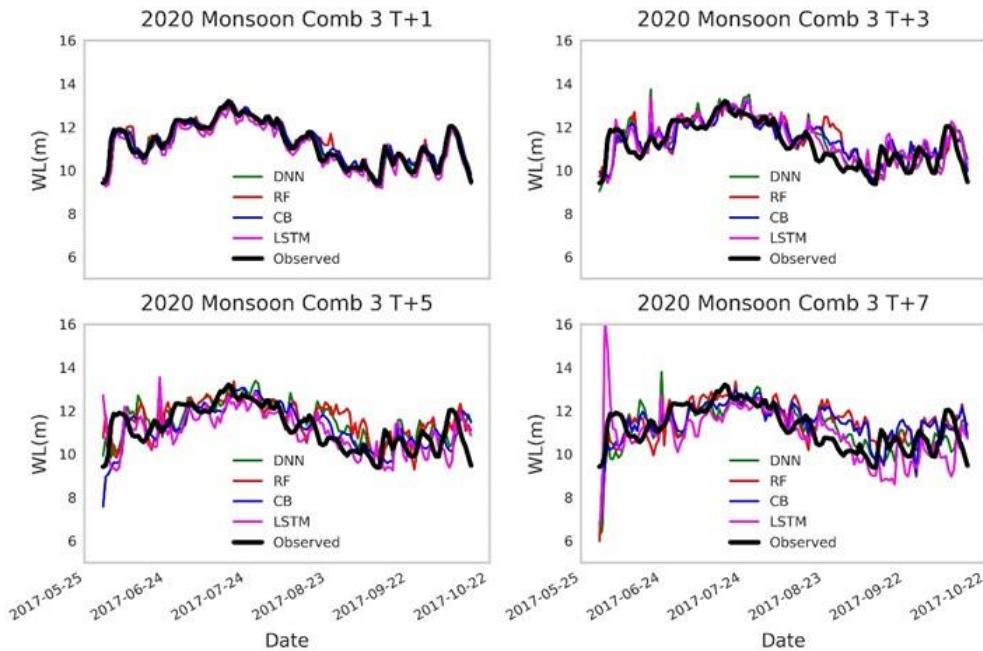
B. 21 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2019 Flash Flood



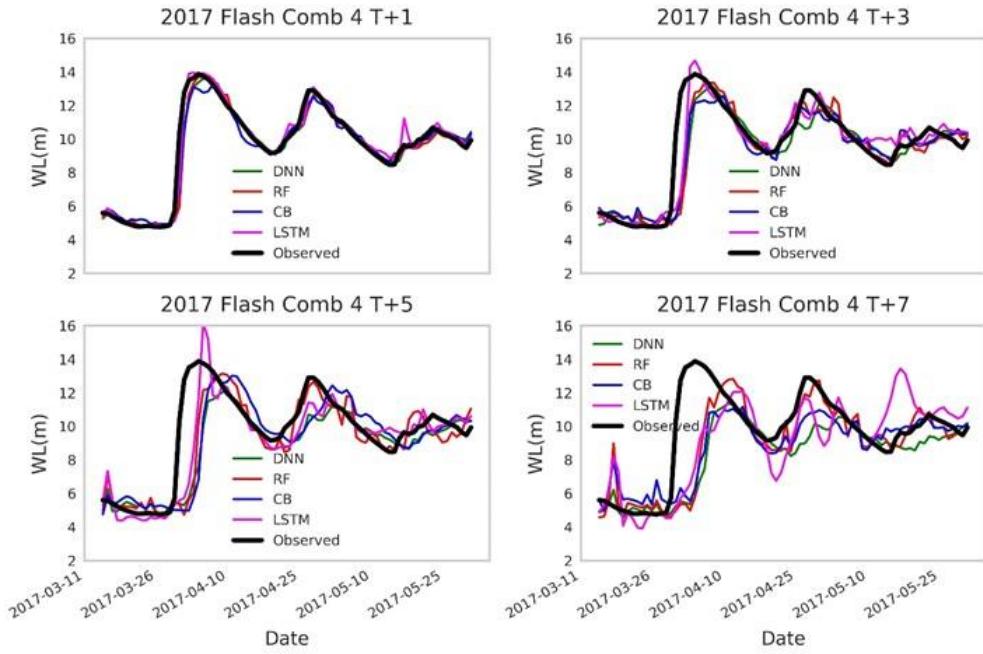
B. 22 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2019 Monsoon Flood



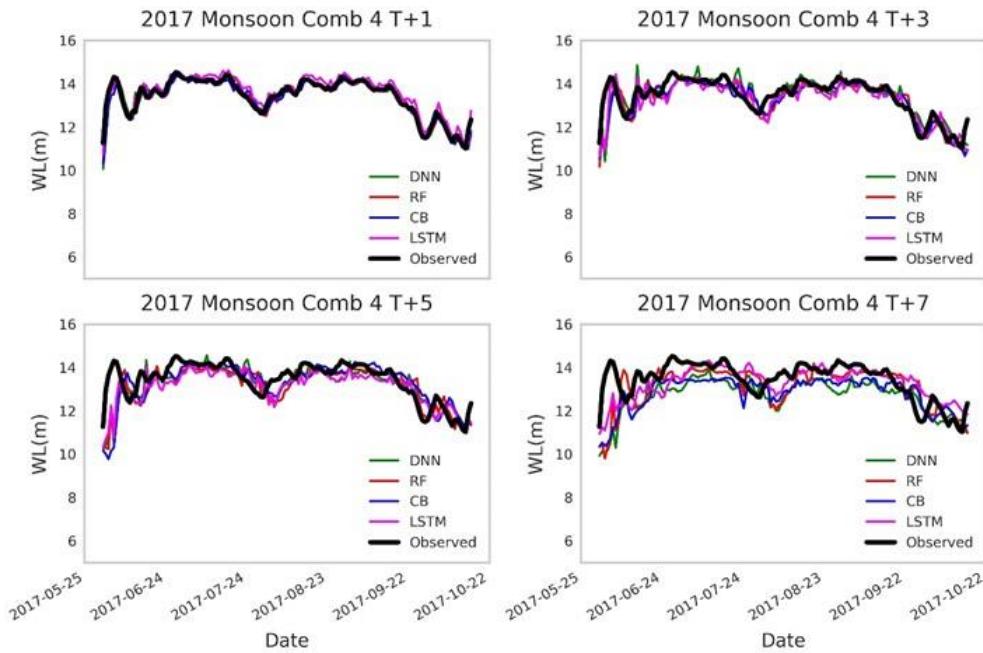
B. 23 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2020 Flash Flood



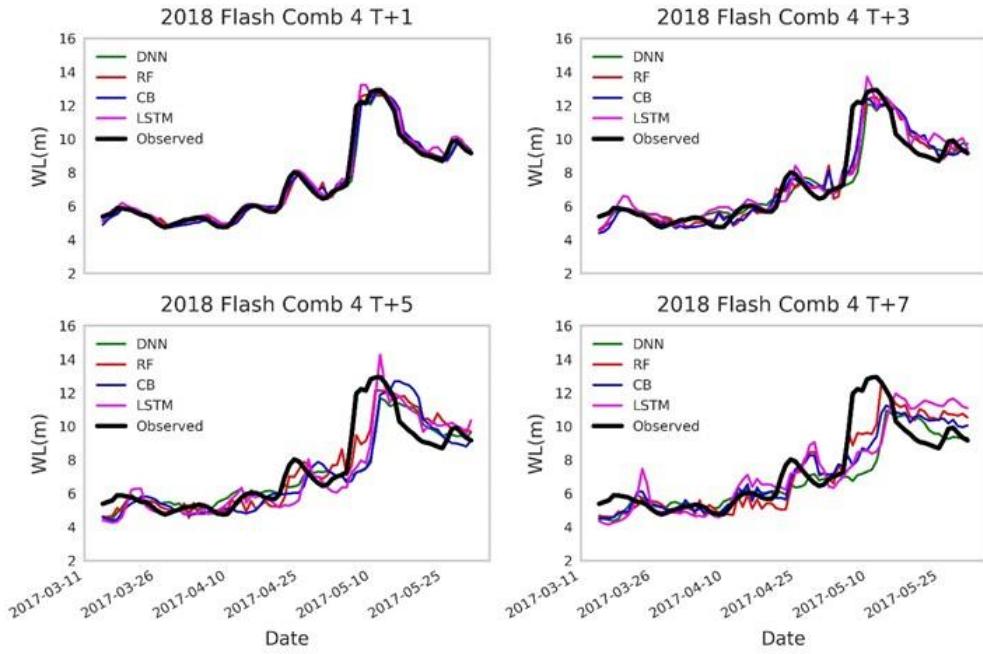
B. 24 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 3 on different time steps for 2020 Monsoon Flood



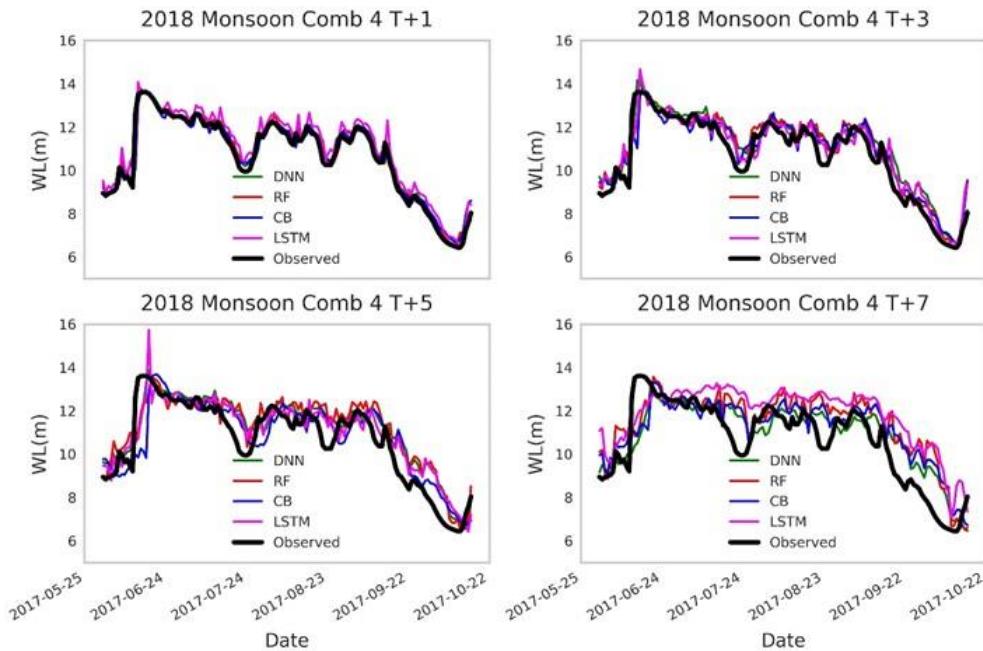
B. 25 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2017 Flash Flood



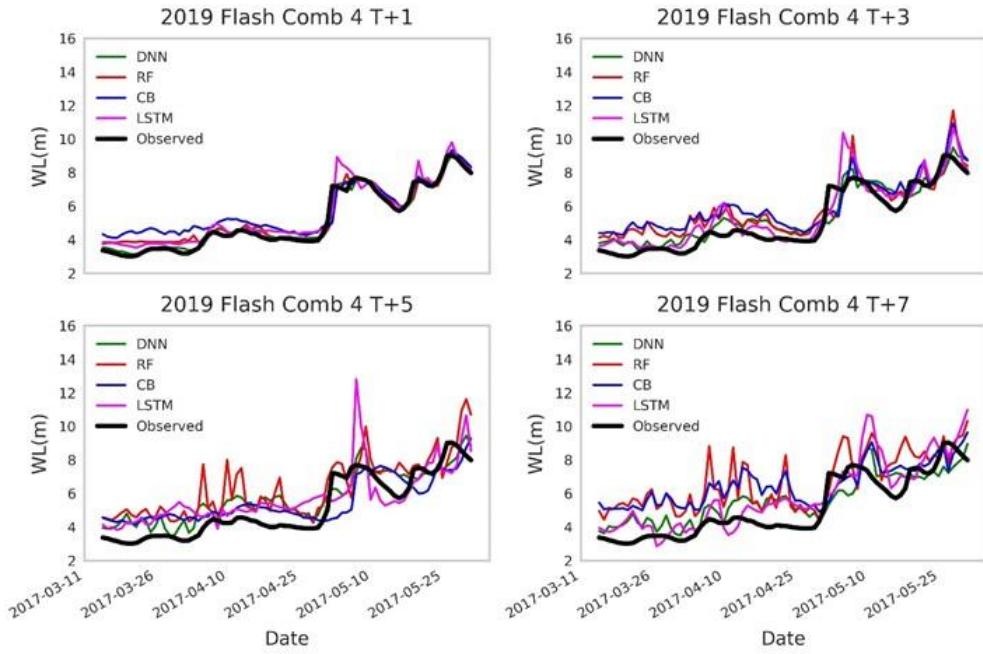
B. 26 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2017 Monsoon Flood



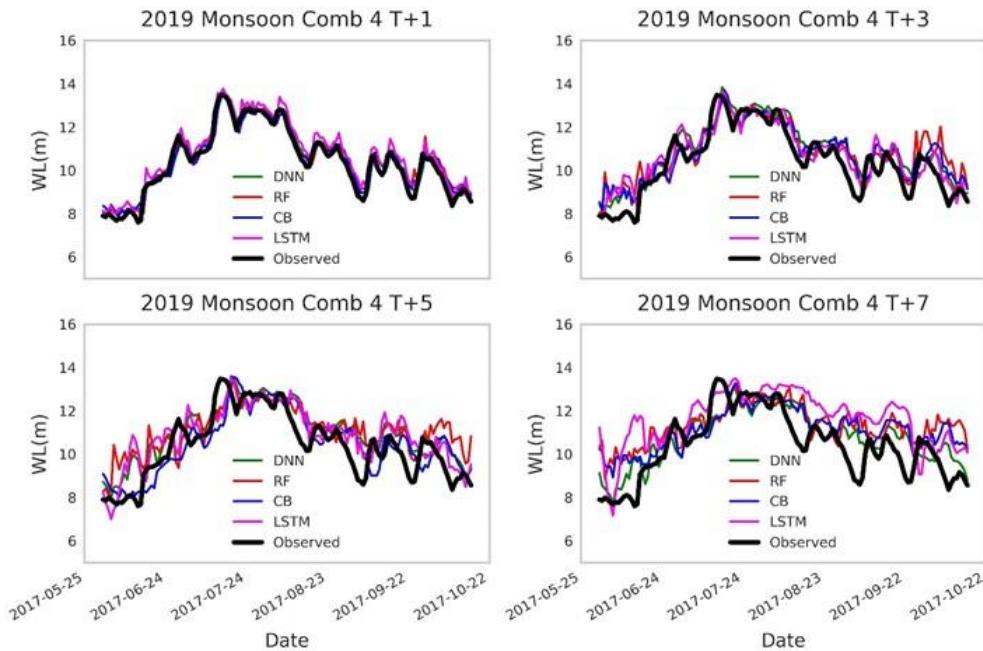
B. 27 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2018 Flash Flood



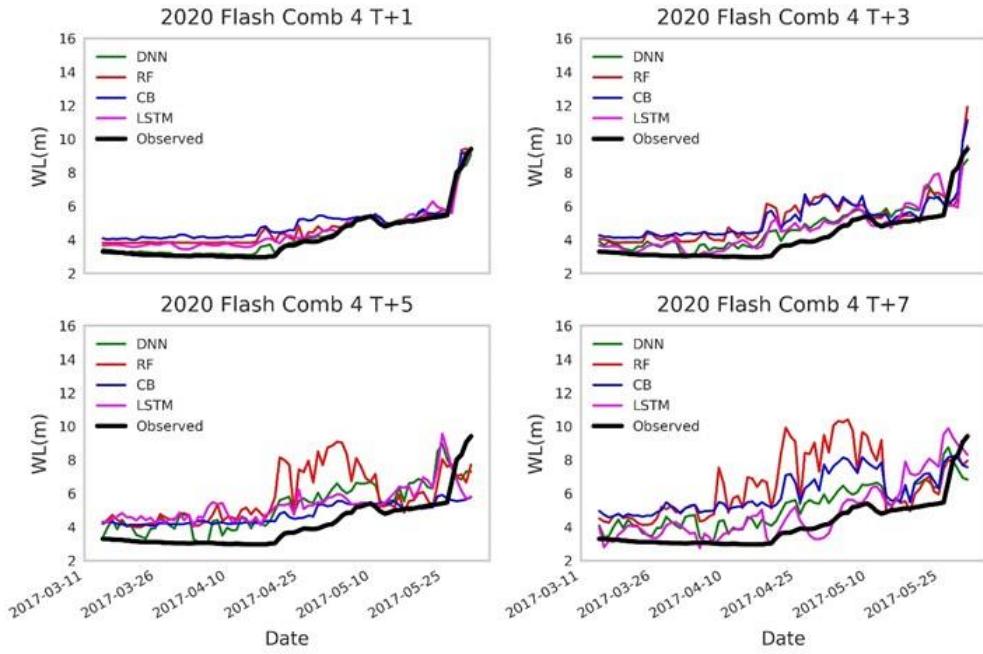
B. 28 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2018 Monsoon Flood



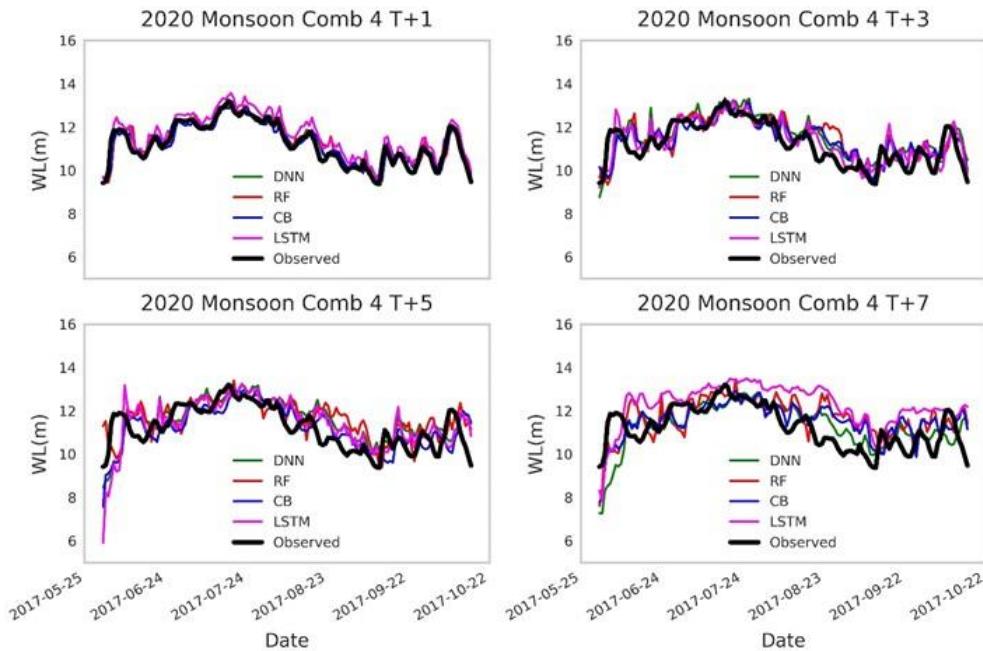
B. 29 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2019 Flash Flood



B. 30 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2019 Monsoon Flood



B. 31 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2020 Flash Flood



B. 32 Comparison of simulated WL(m) with actual WL(m) for different models with input combination 4 on different time steps for 2020 Monsoon Flood