APPLICABILITY OF NEURAL NETWORK MODELS FOR REAL-TIME FLOOD FORECASTING IN DRY ZONE AND WET ZONE RIVER BASINS, SRI LANKA

Cabila Subramaniyam

218071 R

Degree of Master of Science Engineering

Department of Civil Engineering

University of Moratuwa- Sri Lanka

October 2022

APPLICABILITY OF NEURAL NETWORK MODELS FOR REAL-TIME FLOOD FORECASTING IN DRY ZONE AND WET ZONE RIVER BASINS, SRI LANKA

Cabila Subramaniyam

218071 R

Thesis submitted in partial fulfilment of the requirements for the degree Master of Science

Department of Civil Engineering

University of Moratuwa- Sri Lanka

October 2022

Declaration

I declare that this is my own work, and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis / dissertation, in whole or in part in print, electronic, or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

UOM Verified Signature 2023-01-13

Signature: Date:

The supervisor should certify the thesis/dissertation with the following declaration.

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Prof. R. L. H. L. Rajapakse

2023-01-13

Signature of the Supervisor: UOM Verified Signature Date:

Applicability of Neural Network Models for Real-Time Flood Forecasting in Dry Zone and Wet Zone River Basins, Sri Lanka

ABSTRACT

Flood forecasting is a powerful tool for flood management and early warning, where the anticipated flow values are determined by incorporating basin attributes and climatic factors. In the field, data-driven models offer beneficial solutions compared to comprehensive physical and statistical tools; neural networks have evolved to perform flood forecasting without understanding the physical mechanism. However, forecasting efficiency and reliability are insufficient due to the augmentation of predictive span and improper data handling strategies. In addition, the poor interconnectivity of spatial-temporal resolution influences the accuracy of flood forecasting in a dry zone. Thus, the present study aimed to enhance the flood forecasting ability of neural network models for a 30-day horizon by learning the daily input series of climatic and physiographic factors of the catchment region. Further, the data manipulation strategies were adapted to enhance the learning capabilities. In addition, pre-trained models were developed based on the model performance in the wet zone basin to enhance the predictive quality in the dry zone basin.

The NN models were developed for the Kelani River flood forecasting, where significant flood events have frequently destroyed the socio-economic features of the basin. Besides, pre-trained models were tested on the Maduru Basin flood events, which have encountered inundation due to prolonged flood peaks. Thus, climatic and physiographic data were gathered for both basins and improved with hydrological and data science-based data manipulation strategies. On the other hand, the Box-Cox transformation was employed to redistribute the input series into a Gaussian state to enhance the learning ability of NN models.

Consecutive windows were proposed to consider 30-day daily input to forecast the next 30-day streamflow values while sampling. Thirteen (13) NN models were compiled, fitted, and tested on the Kelani Basin. In addition, grid analysis was adapted to rank the performance of models based on statistical tools, where bidirectional models explicated excellent quality in flood forecasting. Besides, uncertainty analysis was proposed to investigate the impacts of data handling and input combination on flood forecasting. Two hybrid models significantly expounded underperformance without box-cox transformation; none of the models illustrated excellent performance without box-cox transformation. Moreover, scaling/normalization severely influenced the model performance considerably for hybrid models. Besides, sensitivity analysis was applied to verify the applicability of model architecture on model performance. Unlike the types of optimizers, other sensitivity parameters revealed inconclusive results for model performance. None of the modified models delivered more excellent performance than the core models. Further, Bidirectional Gated Recurrent Unit (Bi-GRU), Bidirectional Long- and Short-Term Model (Bi-LSTM), and Attention Based Bi-LSTM (Att-BiLSTM) expressed 0.98, 0.95, and 0.97 for the wet zone flood forecasting, respectively, which were chosen as pre-trained models delivered a similar performance for the dry basin.

In future studies, the consecutive data batches must be determined according to the guiding parameters, such as global warming and climate change. Besides, the loss function should be replaced with other statistical terms to incorporate an optimizer, and autocorrelation must be adapted to control the error propagation. In addition, the core model must be trained for extended periods to effectively perform transfer learning on other basins.

Key Words: Box-Cox; Data science; Sensitivity analysis; Sliding window; Uncertainty analysis

Acknowledgement

I am genuinely pleased to express my sincere thanks to my supervisor, Prof. R. L. H. L. Rajapakse, Professor, Department of Civil Engineering, University of Moratuwa, for his keen interest and overwhelming tendency to assist his students. His meticulous scrutiny and scholarly advice have supported me in accomplishing the study profoundly.

A debt of gratitude is owed to Dr. P. K. C. De Silva and Dr. T. M. N. Wijayaratna, the examination panel members, for improving the quality of the research by assigning their valuable comments.

I owe a deep sense of gratitude to the Head of the Department, Prof. (Mrs) C. Jayasinghe, and the Department of Civil Engineering, University of Moratuwa, for offering me an opportunity to pursue an MSc degree. In addition, I profusely thank all staff members of the Department of Civil Engineering, University of Moratuwa for their cooperation throughout the study.

I especially thank my friend Mr. N. Kiruthihan, Associate Software Engineer at Calcey Tech, for sharing his valuable suggestions on online courses, tutorials, and reference materials regarding machine learning. Furthermore, I sincerely thank Mr. P. Loghi, Senior Software Engineer at Sysco Labs, for his valuable guidance on online courses. Finally, it is my privilege to thank my parents; without their tremendous understanding and encouragement; it would be impossible for me to complete my study.

Table of Content

List of Figures

List of Tables

List of Abbreviations

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ALSM	Attention Based CNN – LSTM
AMC	Antecedent Moisture Condition
ANN	Dense Artificial Neural Network
AMC	Antecedent Moisture Condition
ARIMA	Autoregressive Integrated Moving Average
Att Bi-LSTM	Attention -Based Bidirectional LSTM
BDLSTM	Bidirectional LSTM
Bi-GRU	Bidirectional GRU
Bi-LSTM	Bidirectional LSTM
BL	Baseline Model
BR	Bayesian Regularization
CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
CHIRPS	Climatic Hazards Group InfraRed Precipitation
CNN	Convolution Neural Network
CNTK	Cognitive Toolkit
CPU	Central Processing Unit
CTS-LSTM	Correlated Time Series - LSTM
DCNN	Dilated Casual CNN
DEM	Digital Elevation Model
DL	Deep Learning
DWT	Discrete WT
DWT-iNARX	Hybrid Model of Discrete Wavelet Transformation - Improved Nonlinear Autoregressive with Exogeneous Input Network
FCN	Fully Connected Neural Network
FDC	Flow Duration Curve
FDC-Q	FDC Behavioral Error
FNN	Feedforward Neural Network
GC-LSTM	Graph Convolution Embedded LSTM
GCN	Graph Convolution Neural Network
GEE	Google Earth Engine
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit (RNN)
IDE	Integrated Development Environment
IGBP	International Geosphere-Biosphere Program
IoT	Internet of Things
LeM	Levenberg-Marquart
LS-SVM	Least Squares Support Vector Machine Regression
LSTM	Long Short Term Memory (RNN)
LSTM-I	LSTM – Imputation
LULC	Land Use and Land Cover
MAE	Mean Absolute Error
MANN	Memory Augmented Neural Network
MCS	Monte Carlo Simulation
MLP	Multi-Layer Perceptron
MLR	Multi Linear Regression
MNDWI	Modified Normalized Difference Water Index
MRTPP	Multiple Relevant and Target variables Prediction Patterns

N A	Not Available
NARX	Nonlinear Autoregressive Network with Exogenous Input
NDBI	Normalized Difference Built-up Index
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
NN	Neural Network
NOAA	National Oceanic and Atmospheric Administration
PCA	Principal Component Analysis
R^2	Coefficient of Determination
ReLU	Rectified Linear Unit
R_{FDC}	FDC Behavioral Error
RM	Residual Max Error
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
S	Potential Maximum Retention
SAR	Synthetic Aperture Radar
SBU-LSTM	Stacked Bidirectional and Unidirectional LSTM
SCS CN	Soil Conservation Service Curve Number
SGD	Stochastic Gradient Descent
SRTM	Shuttle Radar Topography Mission
ST	Spatial – Temporal
SWIR	Shortwave Infrared
TCN	Temporal Convolutional Network
TCN-ED	TCN-Encoder Decoder
TOA	Top of Atmosphere
USDA	United States Department of Agriculture
WT	Wavelet Transformation

CHAPTER 1: INTRODUCTION

The background of the study focusing on flood forecasting, the problem statement, the objectives of the study, and the scope of the study are discussed in this section.

1.1 Background

Flood control and forecasting are the most predominant steps of powerful flood management. Flooding can lead to adverse impacts if development activities intercept the floodplain of a river system. The river overflows the bank and inundates the nearby areas leading to loss of life and property resulting from the negative behaviour of extreme flood conditions. The positive correlation between global warming and the frequent occurrence of extreme rainfall events resulting in floods has led to several research endeavours in flood studies. The intensification of the hydrological cycle simulates further concerns about the impacts of future flood events. A flood hydrograph is the chronological representation of the flow discharge due to the storm over the catchment. In addition, flood peaks are extracted from the hydrograph and utilized for the hydrological analysis, and annual peaks are employed to build the hydrologic series (Subramanya, 2017).

The rainfall-runoff relationship is a complex behaviour and is determined by various basin and event-specific factors. The most straightforward relationship is simulated by correlating annual runoff and rainfall values, where the linear regression coefficients validate the model fitness or goodness of fit. However, linear regression is only valid for small-range catchments; in addition, exponential or logarithmic mathematical transformation is a suitable representation for large catchments. On the other hand, synthetic hydrographs are famous for predicting future events with available rainfall data, incorporating soil moisture and antecedent rainfall data (Subramanya, 2017).

The most important forecasting achievement is preparing an adequately lengthy series without compromising model performance or accuracy. Previous endeavours in hydrology reveal that the hydrographs are a suitable representation to plot the catchment response for the rainfall data, which consists of surface runoff, interflow, and baseflow. Climatic factors and physiographic factors are the influencing factors in the development of hydrographs. Basin characteristics, infiltration characteristics, and

channel characteristics are included in the physiographic factors; simultaneously, evapotranspiration, storm characteristics, and initial losses are elements of climatic factors (Subramanya, 2017).

In past decades, forecasting models have been developed based on different principles incorporating computer technology. Besides, the process-driven models are determined based on the physical mechanisms, while the data-driven models are constructed based on machine learning. Complex mathematical formulae and a considerable amount of hydrological data are required to simulate the process-driven models. A deep understanding is crucial for the process-driven models for flood forecasting studies. On the other hand, the data-driven models are applied to develop statistical models between hydrological variables. However, the physical mechanism of hydrology is commonly not considered while performing data-driven modelling. In addition, unique advantages are identified for the model, such as solving numerical problems, generating non-linear functions, and simulating time series analysis. The Neural Network (NN) has recently been in demand for applying data-driven models (Xu et al., 2021).

Timely information gathering and real-time simulation support the essential aspects of forecasting; in addition, the NN architecture builds up with approximating a statistic non-linear mapping function. Multi-Layer Perceptron (MLP) is the most useful NN, consisting of input, hidden, and output layers. Besides, the activation function plays a significant role in generating outputs. Forecasting problems are classified into three categories such as short-term (days, weeks, months), medium-term (one or two years), and long-term (more than two years), where hydrological forecasting is one of the short- or medium-term problems. The future pattern of the event is predicted by identifying and extrapolating the historical data; therefore, statistical-based mathematical models are adequate for the analysis. Besides, deep learning (DL) is more reliable than shallow learning; thus, DL is famous for image classification, speech recognition, COVID-19 prognostic analysis, rainfall-runoff modelling, and streamflow prediction. The accumulated input data series is decomposed into several components, where each component involves in the timely prediction. Finally, the

outcomes of the forecasted series are added together to generate the final output series (Sha et al., 2021).

The NN models are used to adequately alert the risk of flood occurrence, which locates the peak flow rates with sufficient fine-temporal resolution (de la Fuente et al., 2019). However, the accuracy vanishes with the increment of predictive span and the combination of hydrological parameters. Therefore, the reliability of predictions is not appreciable in NN-based monthly flood forecasting models (Sha et al., 2021). Besides, NN models cannot simulate the pattern for the catchment region with frequent arid weather. Further, the spatial and temporal correlation is insufficient to simulate the complex non-linear pattern of hydrological data (Jimeno-Sáez et al., 2018).

Artificial Neural Network (ANN) is one of the standard Artificial Intelligence (AI) vastly used in forecasting, which is performed to predict the future without the acknowledgement of the physical characteristics of the basin. However, these models are unsuitable for dry weather (Jimeno-Sáez et al., 2018). A Recurrent Neural Network (RNN) is another AI typically employed for the sequential data for forecasting. Image recognition, internet of things, text translation, and stock prediction are performed with the Long-Short Term Memory (LSTM) models of RNN. In addition, affine transformation and ease of user memory are the two significant abilities of LSTM cells. Besides, the memory capacity supports predicting the event from the present and past states; nevertheless, the accuracy of the LSTM models is limited to a short range of forecasting spans. In addition, RNN requires a waiting time for the forward pass of previous steps. Convolutional Neural Network (CNN) is the next revolution of NN, mainly designed for fulfilling computer vision tasks, especially when it extracts high-level features from the grid topography.

Temporal Convolutional Network (TCN) is one of the variants of CNN able to produce the length of output as same as the input span. In addition, this is better than LSTM in stock trend prediction and recognition of sepsis. Recently, the rapid convergence of TCN models has built up effective flood forecasting models using rainfall data, evaporation data, and Normalized Difference Vegetation Index (NDVI) data (Xu et al., 2021). The performance of the hybrid CNN-LSTM is better than the stand-alone

models: CNN and LSTM. In addition, the input data processing gives a better solution than the original input data sets. Most importantly, non-periodic parameters show significant benefits for the data processing methods, where the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) is the tool applied for this task. The accuracy declines with the increment of predictive steps; however, the decaying of the data is restricted by using the CEEMDAN tool (Sha et al., 2021).

The physical mechanism of hydrology is non-linear; thus, simple prediction techniques never appreciate a favorable solution for forecasting. The false-positive warnings should be eliminated by enhancing the forecasting quality to respond to both tiny and extreme flow events (Jimeno-Sáez et al., 2018). Hydro-meteorological and geomorphological variables are employed in the data-driven models. Besides, LSTM models are suitable for mountainous and snow-dominated regions which is modified by incorporating the Data Integration tool to eliminate the baseflow bias in groundwater-dominated basins (Feng et al., 2020). The interdependency of time series analysis is impossible to carry out with an ordinary RNN. Therefore, Correlated Time series (CTS)– LSTM models count the interconnectivity of multiple time series (Wan et al., 2020). Nonlinear Autoregressive Network with Exogenous Input (NARX) is one of the dynamic RNNs, which is the best time-series predictive tool for seasonal components without significant computational losses (di Nunno & Granata, 2020).

1.2 Problem Statement

Flood forecasting and early warning systems requiring long-term hydrologic series are effectively simulated with neural network-based models; however, the efficiency and accuracy of the predictions are insufficient. The uncertainty occurs with the extension of the predictive span. Besides, the combination of input parameters influences the performance of the models, which eventually leads to catastrophic malfunctioning of NN models. Further, the poor interconnectivity of spatial and temporal resolution causes less accurate flood forecasting of dry zone basins.

1.3 Main Objective and Specific Objectives

Main Objective: To address the deficiencies and harness the strengths of the NN models developed for a 30-day forecasting horizon by employing available NN models in the literature and enhancing accuracy by handling the data manipulation strategies.

Specific Objectives:

- To manipulate the data and sampling to enhance better performance of NN models.
- To identify the best NN models by compiling the models on a wet zone river basin.
- To perform uncertainty analysis by handling the input data of the models.
- To perform sensitivity analysis by altering the architecture of the models.
- To evaluate the model performance in the dry zone river basin by applying transfer learning.

1.4 Significance of the Study

The accuracy of NN models is inadequate for monthly forecasting (30 days) with daily input series; thus, the reliability is insufficient to appreciate flood forecasting.

Expected outcomes:

- Extreme events must be considered with proper data manipulation and sampling techniques.
- The models should be compiled, fitted, and tested on Kelani River Basin (wet zone) data.
- The performance of models must be quantified using grid analysis.
- The uncertainty and sensitivity analysis must be performed to verify the applicability of input handling and the model architecture.
- Transfer learning must be applied to pre-train the models on Maduru River Basin (dry zone) data.

1.5 Scope of the Study

The climatic and physiographic factors of the catchment were gathered and utilized to develop thirteen Neural Network models. The Kelani River basin (wet zone) data were employed to assess the models. The pre-trained model performance is evaluated in the Maduru Oya basin (dry zone). Figure 1-1 illustrates the locations of selected basins on the climatic zone map.

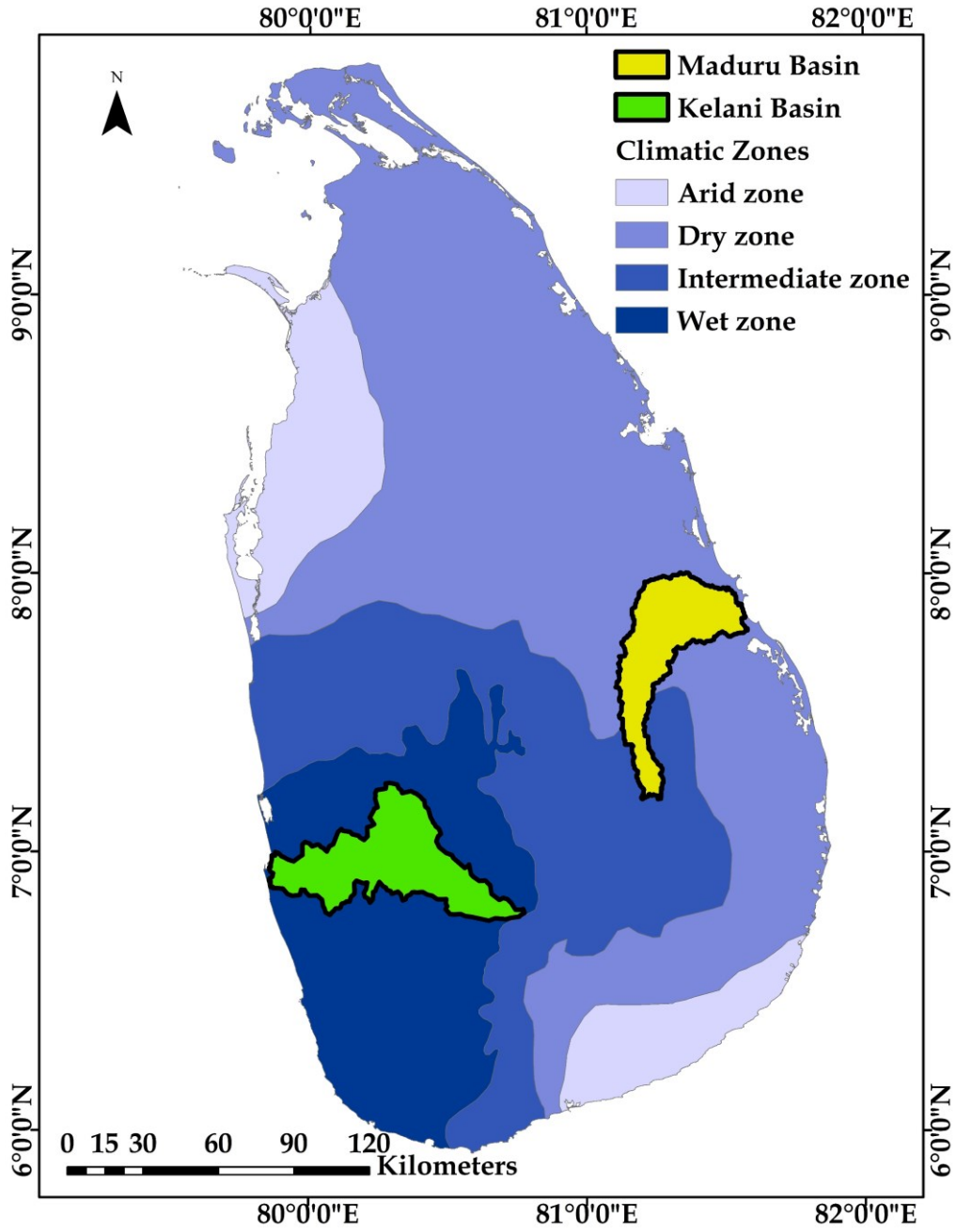


Figure 1-1 Climatic Zone Maps and Selected River Basins

CHAPTER 2: LITERATURE REVIEW

The section describes the previous research concepts, neural networks available in the literature, data processing, developing NN models for the present study, the selection of the best models, uncertainty analysis, sensitivity analysis, and transfer learning.

2.1 Previous Research Concepts in Forecasting using NN Models

Flood flow depends on soil properties, land usage, climate, river basin, snowfall, and geophysical elements. Forecasting models are based on physical, statistical, and computational intelligence or deep learning algorithms. Besides, comprehensive physical models are data-intensive and challenging to handle complex non-linear problems in flood forecasting. However, the ability has been enhanced by adopting advanced simulation and hybrid models. In addition, statistical models are associated with historical data to develop the pattern for the future flood, especially Multi Linear Regression (MLR), Autoregressive Integrated Moving Average (ARIMA), and Hybrid Least Squares Support Vector Machine Regression (LS-SVM) are the standard statistical models. However, the models lack to scale well with the increment in size and complexity of the data. Computational Intelligence, such as deep learning, handle the struggles with scale and complexity without acknowledging basic processes. Deep learning models require a minimal number of computational resources, and the expected performance is faster than others (Gude, Corns, & Long, 2020).

Since 1994, NN models have been believed to reflect more efficient solutions for flood forecasting than the traditional model. During the initialization of the forecasting work, the average rainfall data over the basin scale was used, and the simulation revealed an issue in neglecting sub-basin contribution. Certain portions of the basin have received rainfall; therefore, its response to rainfall depends on previous events. The time lag between rainfall at each rain gauge is examined by understanding the correlation between rainfall and water level. Therefore, it is determined by estimating the frequency distribution over the database event. On the other hand, calibration and validation are essential in the estimation, especially since the training set minimizes the error, while the testing set is important to avoid overfitting (Campolo, Andreussi, & Soldati, 1999).

An ANN is a mathematical structure utilized for computational purposes, where the network understands, learns, memorizes, and discloses the relationship among the data sets. In addition, the models consider the complex non-linear periodical relationship without explicit information on the physical characteristics of watersheds. According to the number of layers, it is classified as a single- and multi-layer network. In addition, it is divided into two categories according to the direction of passing information: feed-forward network and recurrent network (Jimeno-Sáez et al., 2018).

Algorithms and deep learning-related methodologies have been developed because of the advancement of water conservancy information based on the Internet of Things (IoT). The CNN are capable of complex feature extraction, where the gridding technique implements the spatial temporal interconnectivity???? of rainfall data. Besides, Digital Elevation Model (DEM), geographical features, and historical streamflow features are imported into the model. CNN model for flood forecasting proceeds with 24-hr and 36-hr leading times; mainly, the ability to extract features is the key reason to apply the model. Further, it is popular for computer vision because of its ability to parameter sharing. Besides, the Rectified Linear Units (ReLU) and dropouts improve the performance of the model. In addition, it is simulated with a smaller kernel than other NN models. A research paper on 'Convolutional Neural Networks for Forecasting Flood Process on the Internet of Things Enables Smart City' was organized based on the CNN forecasting tool.

Temporal Convolutional Neural Network (TCN) is currently popular in flood forecasting, which combines simple convolution and a 1D Fully Convolutional Network (FCN) that supports generating output data of the same length as input. The casual convolution ensures that future derivations do not affect the previous information. In addition, it provides a better solution than LSTM for flood forecasting. A research paper on 'Application of temporal convolutional network for flood forecasting' was constructed on a TCN-based flood forecasting model.

Researchers have paid attention to the RNNs and their variants, such as LSTM and GRU, which perform better than the standard RNNs. The models have the best ability to deal with high nonlinear interaction among complex hydrological parameters.

Besides, Principal Component Analysis (PCA) method is adopted to control dimensionality and redundancy within input datasets. RNN model constructed with multiple input data forecasts more accurate results than rainfall data alone. This method supports the classification of the original data into several comprehensive variables. The research paper on 'Daily Runoff Forecasting by Deep Recursive Neural Network' was prepared to elaborate on the PCA method adapted for flood forecasting. In addition, a research paper on 'An Integrated Approach for Weather Forecasting and Disaster Prediction Using Deep Learning Architecture Based on Memory Augmented Neural Networks (MANN's)' was developed to explain the MANN architecture as LSTM, GRU, and DNN.

Nonlinear Autoregressive with Exogenous Input Network (NARX) is suitable for the time series forecasting problems of seasonal components with more minor computational losses, which is a dynamic RNN model based on lagged input-output variables and prediction errors. On the other hand, wavelet transformation (WT) is developed based on Fourier analysis that simultaneously interprets both temporal and spectral information. The model adapted with Monte Carlo Simulation (MCS) quantifies the model uncertainty and performs probabilistic water level forecasting, and it achieves accurate predictions for short-term forecasting. A research paper on ‘Flood Forecasting Using an Improved NARX Network Based on Wavelet Analysis Coupled with Uncertainty Analysis by Monte Carlo Simulation: A Case Study of Taihu Basin, China’ thoroughly explained this hybrid model related to flood forecasting. Another research paper on ‘Groundwater Level Prediction in Apulia Region (Southern Italy) using NARX Neural Network’ was developed to address the performance of the NARX forecasting model.

The RNN is the best neural network to deal with sequence data by simulating chain-type structures and internal memory with loops. In addition, the vanishing gradient issue is controlled by the LSTM and GRU, which are variants of RNN. The system develops through the appreciated performance of the long-term dependencies of the model; however, the models fail to handle the missing data. Besides, the research paper on 'Stacked bidirectional and unidirectional LSTM recurrent neural network for

forecasting network-wide traffic state with missing values' was developed to eliminate the issues based on traffic forecasting.

Correlated Time Series – Oriented - LSTM (CTS-LSTM) is the best forecasting tool for analyzing its deficiencies compared to conventional LSTM. It captures the intra-sequence temporal and inter-sequence spatial dependencies of multiple series. RNN understands the temporal correlation among the neurons in the sequential data. However, the vanishing gradient causes issues in training the input data. Therefore, the model insufficiently handles the long-term dependencies in the sequential data. The correlated time series have their own temporal and spatial dependencies in-between each other. Besides, this unique feature makes forecasting more effective through the CTS-LSTM model. A research paper on 'CTS-LSTM: LSTM-Based Neural Networks for Correlated Time Series Prediction' on air quality prediction was constructed to explain the model structure.

Spatiotemporal characteristics should be considered to develop multi-time and multi-site forecasting models. Deep learning is a more powerful tool for developing various forecasting systems than the shallow learning mechanism. Therefore, it is famous for various fields, such as image recognition, speech identification, and the natural language process. Besides, LSTM is the most eligible modelling tool for forecasting. In addition, the hybrid models of LSTM give a unique solution to attain the best accuracy. CNN-LSTM is one of the excellent solutions for achieving the best accuracy in forecasting. Cluster analysis segregates the disordered data into different categories with high similarity to estimate the internal patterns to enhance the performance, where the generalization ability is achieved in this process to minimize the training period. The research paper on ‘multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering’ powerfully explained the CNN-LSTM and the cluster analysis based on air quality forecasting.

The input data severely dominate forecasting accuracy; further, calculation formulas and complicated processes require performing an LSTM-based neural network. In addition, the attention-based LSTM addresses, and copes with the dispersion in the standard LSTM model. The attention mechanism assigns weights to the features based

on the time steps. In contrast, the ordinary LSTM model has a finite ability to extract complex features and only depends on weights.

The hybrid model based on CNN and LSTM ignores the gravity of feature selection; therefore, this model requires much redundant information and strict variables. Attention-based long and short-term temporal neural network model is assembled with the conventional neural network, extended short-term neural network, and attention mechanism under the multiple relevant and target variables prediction pattern (MRTPP). In addition, the accuracy is enhanced by the MRTPP pattern, which is better than the typical input-output pattern. The research paper on 'Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern' elaborated on the performance excellence of the Attention Based CNN-LSTM models based on photovoltaic power forecasting.

2.2 Neural Network Forecasting Models Available in the Literature

The neural network models available in the literature are discussed in this section.

2.2.1 Ordinary Artificial Neural Networks (ANN)

The ANN is the multilayer perceptron, which consists of the input layer (input neurons), output layer (output neuron), and hidden layer.

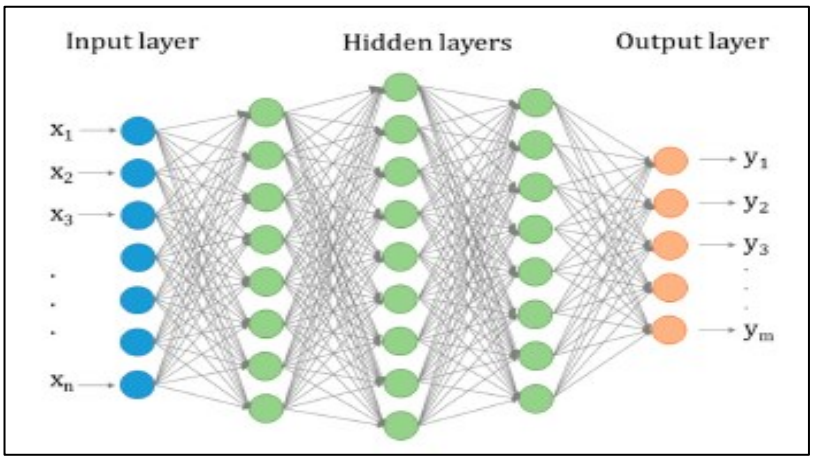


Figure 2-1 Model Structure of ANN, (de la Fuente et al., 2019)

The neuron receives a weighted sum from the neurons in the previous layer; thus, these are called feedforward neural networks because the last neuron generates the input for every neuron in the same direction.

In addition, the transfer function governs the activation of a neuron (de la Fuente et al., 2019). Figure 2-1 shows the configuration of the ANN model.

2.2.2 Convolution Neural Network (CNN)

The CNN extracts the repeating pattern from the time series by eliminating the noisy series, and the data set is input for learning weight and understanding bias. Dilated convolution solves the long-term dependence in time series, while 2D convolutional operation captures the rainfall spatial-temporal features, geographical aspects, and trends (Chen et al., 2021).

In the traditional model, rainfall data is the only input utilized for modeling. Besides, the rainfall range is calculated for the whole basin by taking the weighted summation of different stations. Therefore, the spatial distribution of the rainfall is difficult to attain through the system. In addition, topography details are collected from STRM, and the DEM matrix is employed to size the spatial distribution matrix using an average pooling operation. The first three layers of the CNN capture the complex features of rainfall spatial-temporal data and geographical aspects. Further, a dense layer concatenates the feature with the future trend (Chen et al., 2021).

2.2.3 Deep Recurrent Neural Network (RNN)

The LSTM consists of gate units and cell states where the memory cell and hidden state are updated on the input and output gates. Besides, the forget gate ignores irrelevant information. The GRU replaces the hidden blocks of ordinary RNNs with two logic gates. In addition, the update gate is employed to capture the long-term dependencies of the series. Simultaneously, the reset gate and the candidate hidden state are approached to capture the short-term dependencies of the series (Zhang et al., 2021). Figure 2-2 and Figure 2-3 illustrate the LSTM and GRU with gate units.

Differential Neural Network (DNN) is associated with the differentiable process of updating the memory rows. In addition, Stochastic Gradient Descent (SGD) is suitable

for learning algorithms. Besides, LSTM, GRU, and DNN are constructed based on the concept of MANN (Satwik & Sundram, 2021).

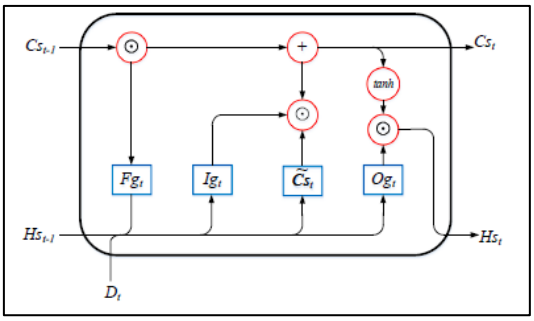


Figure 2-2 LSTM Model Structure with Forget Gate, Input Gate, Cell State and Output Gate, (Zhang et al., 2021)

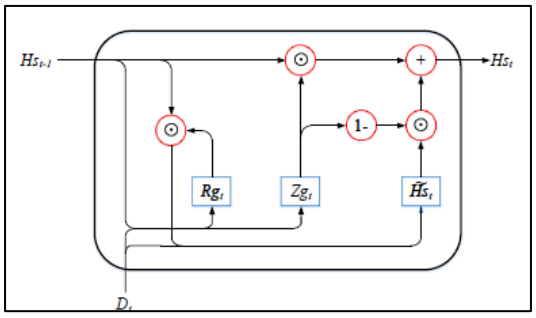


Figure 2-3 GRU Blocks with two Logic Gates, (Zhang et al., 2021)

Dropout is a crucial feature of deep RNN because of its high memory ability. Deep RNN overfits the results with the high-dimensional input. It is the regulation method to control the issues due to overfitting. Adaptation of the dropout technique controls the dependency of the output on the units of hidden layers. During the data pre-processing, the PCA method is applied to reduce the dimensionality of the input. Besides, data division is adopted to eliminate overfitting and test the predictive capabilities (Zhang et al., 2021).

2.2.4 Nonlinear Autoregressive with Exogenous Input Network (NARX)

The principle behind the feedforward neural network (FNN) is the basic concept of NARX, where an output of a neuron becomes an input for another neuron, flowing in a direction. In addition, Recurrent Neural Networks (RNN) deviate enormously from FNN, where the data flows in the forward and backward directions. Figure 2-4 explains the skeleton of the NARX model. The reduced number of input parameters is adequate to simulate the model. The time delay is evaluated with the cross-correlation function between input and output. In addition, Levenberg-Marquart (LeM) is widely employed in the model due to its fast and stable convergence. The LeM algorithm is the first layer to approximate the Hessian Matrix. Besides, Bayesian Regularization (BR) is the second algorithm that uses a Gauss-Newton approximation to control the probability of overfitting and the computational overhead. The LeM algorithm is generally faster, while the BR algorithm deals with complex problems (di Nunno & Granata, 2020).

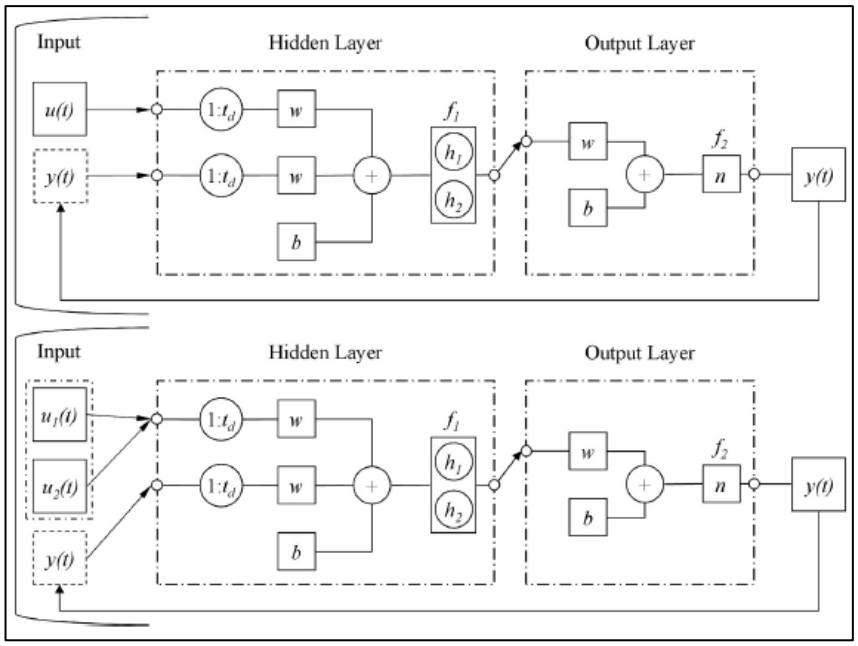


Figure 2-4 NARX Model Structure, (di Nunno & Granata, 2020)

2.2.5 Hybrid Model of Discrete Wavelet Transformation – Improved Nonlinear Autoregressive with Exogeneous Input Network (DWT-iNARX)

The NARX approximates all the nonlinear functions, and its memory provides a short path to optimize the forward- and backward- propagation of the signals. The behaviour destroys the long-term dependency of the model. Besides, the loss of time information with the domain causes a significant drawback to the WT model. The long-time intervals allow for low-frequency data, while the short-time intervals permit the requirement for high-frequency data. The transformation function is chosen according to the characteristic of a time interval. In addition, the time series is decomposed into subunits using WT. The multiple signals are used as input data sets for the ANN model (Jiang et al., 2021).

2.2.6 Stacked Bidirectional and Unidirectional LSTM Network (SBU-LSTM)

The model architecture of SBU-LSTM captures both forward and backward temporal dependencies with the Bidirectional LSTM structure. In addition, the data imputation mechanism is adapted with the model to infer the missing input data because missing data sets are one of the crucial reasons for the malfunctioning of forecasting models. There are two different models to accomplish imputation and prediction separately in

the model. However, the efficiency of the forecasting model is highly questionable through this separate system (Cui et al., 2020).

An ordinary LSTM unit does not adequately pass the information in the backward direction. Therefore, backward dependencies should be brought to the system by organizing the operation in the negative direction. The predictive performance is adequately improved in the temporal series perspective. Bidirectional LSTM already has the best ability to function as a backward and forward informative network. Besides, the number of layers, the size of model weights, and the spatial dimension of input data are the influencing factors in evaluating the performance of the prediction (Cui et al., 2020).

The model architecture of SBU-LSTM is developed as a combination of imputation and predictive tools. In addition, the skipping mechanism is the standard imputation method, and the dropout feature of RNN is applied to develop the skipping mechanism. In addition, regression, spectral analysis, EM algorithm, and matrix factorization are other popular imputation methods. The Bayesian temporal matrix factorization is the popular method to solve spatiotemporal data. Besides, state-of-the-art prediction accuracy is achieved through the matrix factorization method (Cui et al., 2020). The basic units of model architecture are represented in Figure 2-5.

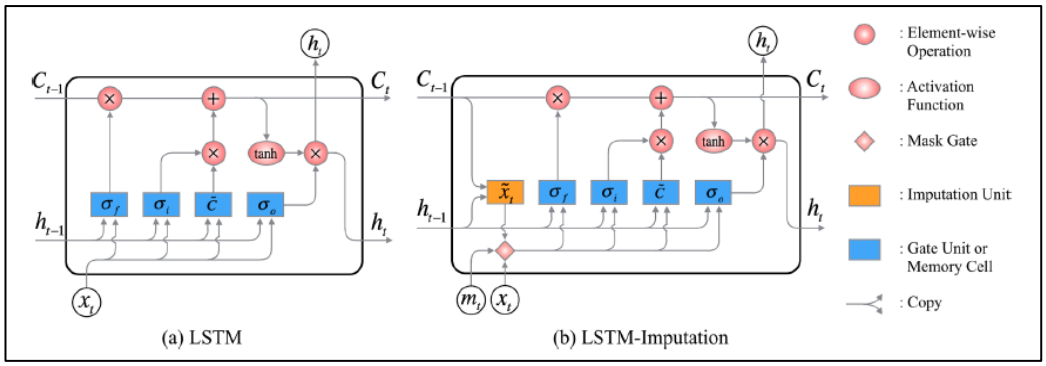


Figure 2-5 LSTM and LSTM-Imputation Units, (Cui et al., 2020)

Further, LSTMs significantly contribute to the sequence-based long-term dependency data analysis, consisting of cell states and three gate units (input, forget, and output). The current variable vector, the preceding output, and the initial cell state are the input for the LSTM. Besides, memory cells and input gate units are designed to learn the

long-term dependency data and allow the essential details to pass through. Forget gate makes an effective and scalable sequential data learning analysis (Cui et al., 2020).

LSTM-Imputation Unit is specially designed to perform prediction and imputation simultaneously. Commonly, predefined values, zeros, mean of historical observation, or last observation values are filled instead for the missing values; however, these approaches are biased in the training process. The separated arrangement of imputation (I) units and predictive tools affects forecasting efficiency. Therefore, the LSTM-I unit is developed by incorporating LSTM and imputation units to simulate both forecasting and imputation simultaneously. In addition, the missing data is identified as missing data from cell states and hidden states. The LSTM-I simulates forecasting and infers missing data through a single process (Cui et al., 2020).

Bidirectional LSTM (BDLSTM) deals with forwarding and backward direction with separate LSTM hidden layers, where it performs better than unidirectional LSTM. The model architecture is illustrated in Figure 2-6.

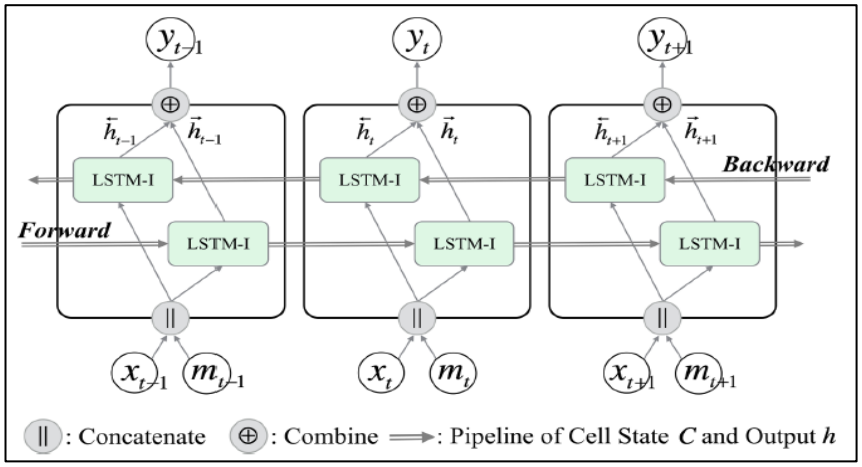


Figure 2-6 Bidirectional LSTM, (Cui et al., 2020)

The BDLSTM is designed to capture missing data with imputation units. Moreover, backward, and forward access to the architecture helps to infer the lost data efficiently. The structure of an un-fold BDLSTM-I layer combines the forward LSTM-I layer and backward LSTM-I layer (Cui et al., 2020).

Stacked bidirectional and unidirectional LSTM consists of several hidden layers of LSTM. The structure represents the sequential data. The stacked multi-layer

mechanism can feed the sequential hidden layers with the output of previously hidden layers. Multiple numbers of BDLSTM-I components are arranged inside the system. The SBU-LSTM consumes the sequential data sets. The first and foremost layer is designed as the BDLSTM-I layer, and this layer infers the missing data of the sequential data sets (Cui et al., 2020).

2.2.7 Correlated Time Series – Long-term Short-term Memory (CTS-LSTM)

The correlated time series processes with separate LSTM cells to avoid the loss of correlation. Therefore, an LSTM cell is applied to receive the correlated time series as an input. After that, the series of information is fused into a representation, and it is called fused representation. However, the characteristics of each time series are expressed entirely with the fused representation. Moreover, mutual correlation never attains through the analysis; consequently, each time series losses its unique features. In addition, extracting temporal features from each series distinguish is harder. Accordingly, the conventional LSTM model lacks to completely understand the spatial-temporal aspects of the time series data (Wan et al., 2020).

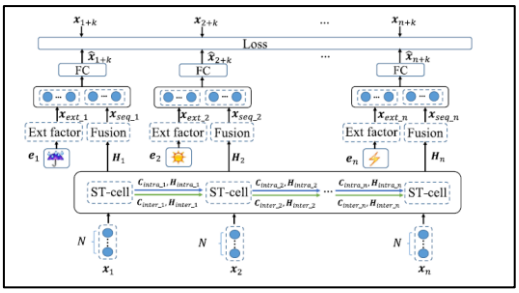


Figure 2-7 Structure of CTS-LSTM, (Wan et al., 2020)

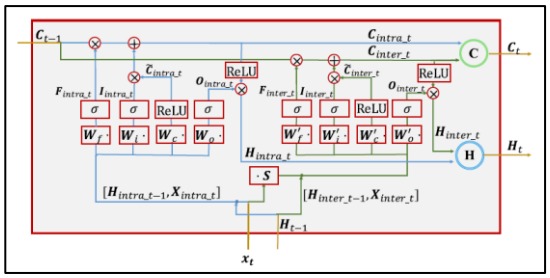


Figure 2-8 The Structure of ST Cell, (Wan et al., 2020)

The CTS-LSTM forecasting model addresses the issues with three critical units: Spatial-temporal cells (ST cells), Spatial-temporal fusion, and external factor module. The Spatial-temporal cell contains two channels to capture the intra-sequence temporal and inter-sequence spatial dependency. Intra-sequence cell state, Intra-sequence hidden state, and three gates are the significant units of the intra-sequence channel, and the unique feature of the intra-sequence dependencies of each series is updated using this channel. Another channel is applied to understand the pairwise influence between correlated series. Therefore, every series build up with the information on all the other series.

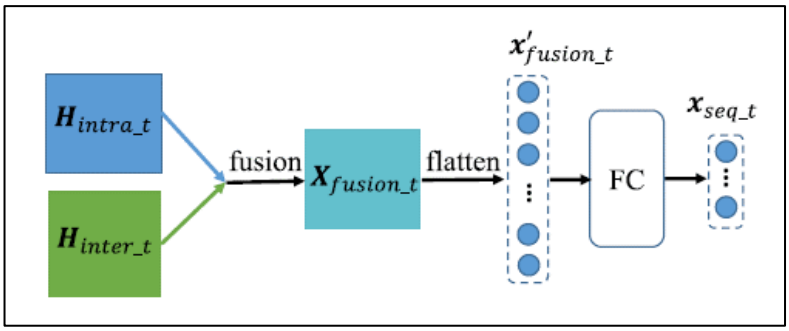


Figure 2-9 Structure of the Fusion Module, (Wan et al., 2020)

Inter-sequence cell state, inter-sequence hidden state, and three gates are the primary units of the channel. In addition, the spatial-temporal fusion consists of a module before the last ST cell layer to aggregate information in the two channels. The external factor module simulates the inherent time series pattern, regularities, and external factors from various domains (Wan et al., 2020). Figure 2-7, Figure 2-8 and Figure 2-9 illustrate the CTS-LSTM modules.

2.2.8 A hybrid model of Convolutional Neural Network – Long-term Short-term Memory (CNN-LSTM)

The CNN model effectively extracts the features of grid data, and the LSTM model excellently processes the time series analysis. Therefore, the error propagation is considerably low in these hybrid models. The input for the model should be arranged to face clustering analysis. The mean and standard values of the spatiotemporal data are involved in the clustering process. In addition, the spatial distribution is analyzed with Kriging Interpolation and Moran's I Statics. The Kriging mechanism is required to determine the semi-variogram model, which is selected based on accuracy, while Moran's I statistics explains the spatial correlation of variables. After the clustering analysis, the data are segregated into various classes; therefore, training time consumption is expected to be low (Yan et al., 2021).

Convolution and pooling layers are involved in the feature transformation and extraction process. Besides, the convolution layer extracts the features of input variables by kernels, where the layers connect the previous data with the current neuron. The multiple Kernels extract the features from the input matrix and provide

the feature maps. In addition, the pooling layers control the dimension of previous feature maps (Yan et al., 2021).

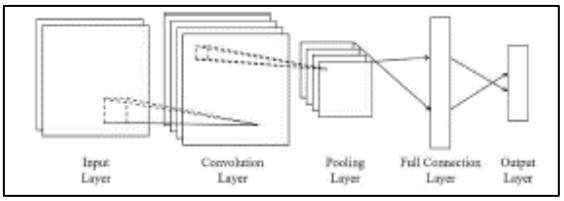


Figure 2-10 Structure of CNN, (Yan et al., 2021)

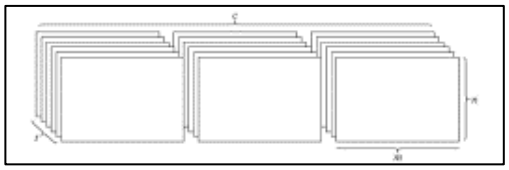


Figure 2-11 Input Dimension of CNN, (Yan et al., 2021)

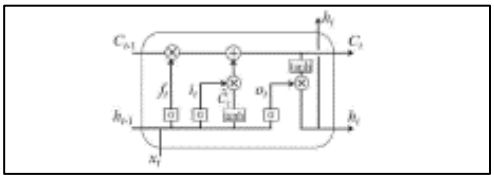


Figure 2-12 The Memory Cell of LSTM, (Yan et al., 2021)

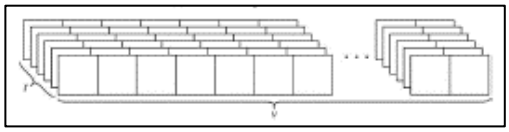


Figure 2-13 Input Dimension of LSTM, (Yan et al., 2021)

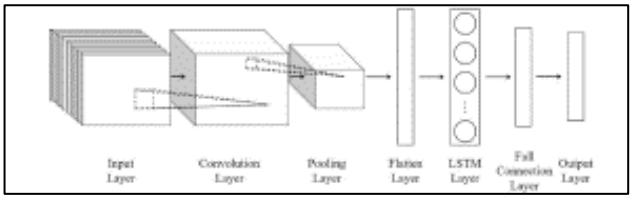


Figure 2-14 Structure of CNN-LSTM, (Yan et al., 2021)

The LSTM is the best version of RNN, and it has a recurring mechanism in the hidden layers. Besides, the input consists of the feature of current and past events, which correlates with the contextual information. The nodes of RNN forget with the increment of iterations, and due to that, the nodes start to ignore the previous information of the sequential data. In addition, the memory cells are applied to hold the historical data. The cell state and three gates support the LSTM model architecture for properly functioning the tasks. Besides, the cell state transmits sequential information, and gates are used to update and discard historical information. The hybrid model of CNN-LSTM represents the combined functional behaviour of CNN and LSTM (Yan et al., 2021). Figure 2-10 to Figure 2-14 expresses the model architecture of CNN-LSTM.

2.2.9 Attention-Based CNN-LSTM Neural Network Model (ALSM)

The historical data is commonly non-stationary, dynamic, and non-periodic; therefore, it is not practical for a traditional neural network. A unique model should be proposed

for forecasting by considering the strength of various neural network techniques. Therefore, CNN, which extracts the spatial features; LSTM, which draws the temporal features; and the Attention mechanism, which eliminates the flaws of distraction, are combined to develop the standard neural network model. Two modules based on LSTM and CNN consider the time characteristics to focus short-term and long-term periodicity of output. The model extracts the spatial and temporal correlation among relevant and target variables. The input-output of Multiple Relevant and Target Variables Prediction Patterns (MRTPP) absorbs the historical operation of the relevant and target variables. In addition, it converts those absorbed data as input variables to the deep learning model (Qu et al., 2021). Figure 2-15 illustrates the model architecture of ALSM.

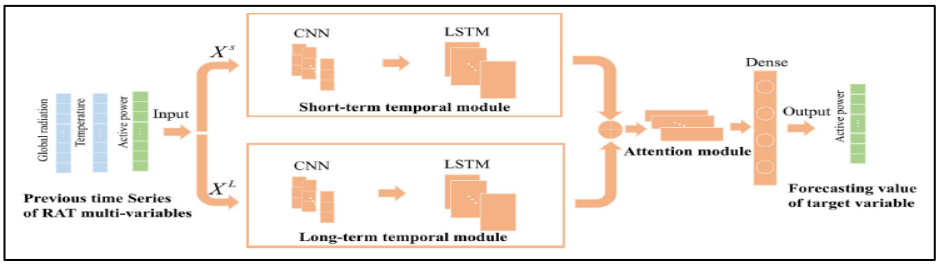


Figure 2-15 Model Architecture of ASLM, (Qu et al., 2021)

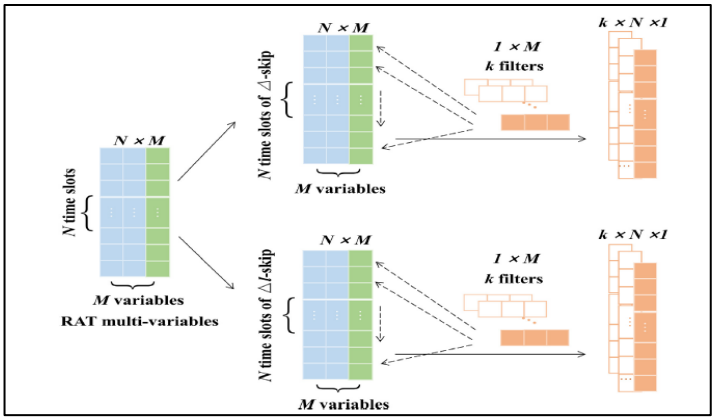


Figure 2-16 1D CNN Extracting Features of Short-Term and Long-Term Series Pattern, (Qu et al., 2021)

The short-term temporal module consumes the short-term time mode with CNN-LSTM cells to perform predictions, a long-term temporal module takes the long-time mode with the Recursive unit of LSTM to perform forecasting, and the attention module understands the interconnection among the different time-sequential modules

with its feature extraction capability are the crucial components of the ALSM model (Qu et al., 2021).

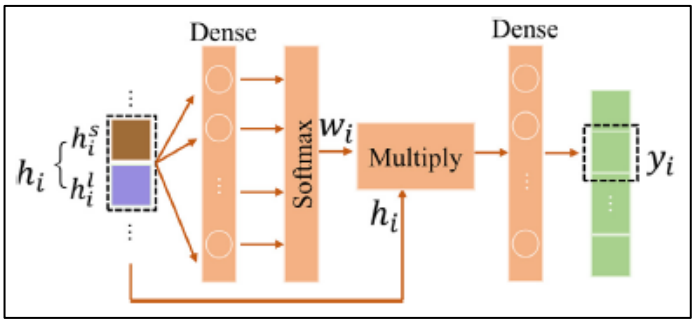


Figure 2-17 The Structure of An Attention Model, (Qu et al., 2021)

CNN is one of the FFNs; thus, the parameter-sharing properties control the number of parameters to be optimized. Therefore, efficiency and scalability are improved through the process. The convolution operation takes place in Euclidean space; therefore, time series prediction and image classification are the significant advantages of CNN. The 1D CNN is employed for time series forecasting through the convolution kernel to extract features of short- and long-term patterns (Qu et al., 2021). Figure 2-16 illustrates the fundamentals of CNN-LSTM.

The vanishing gradient and gradient explosion limit the learning capacity of the cyclic neural network. Besides, the relatively long-term memory capacity of the LSTM makes the usage broader in forecasting, and the attention mechanism comprises two complete connection layers: CNN and LSTM (Qu et al., 2021). Figure 2-17 illustrates the procedure.

2.2.10 Temporal Convolutional Network (TCN)

Temporal Convolutional Neural Network (TCN) is one of the convolutional neural networks, and the application is widely popular in forecasting. Two specific designs are used to develop TCN; firstly, 1D Fully Connected Neural Network (FCN) generates an output of the same length as the sequential input data. The output data is influenced by the present and past information of the sequential data sets. Simple convolution varies from standard and is identified with similar issues to traditional

CNN. The model should be arranged linearly with a stack of many layers to learn the long dependencies among data.

The TCN consists of a 1D expansion convolution layer to overcome the problem. The expansion convolution requires the interval sampling of input. Moreover, the dilation factor enhances exponentially with the depth of the network to utilize a more extensive adequate input history. Several TCN blocks are connected to increase the performance of the model. Besides, multi-step time series forecasting is effectively performed with the model. A moving window scheme of TCN is employed to create input and output pairs (Xu et al., 2021). Figure 2-18 explains the model architecture of the TCN unit.

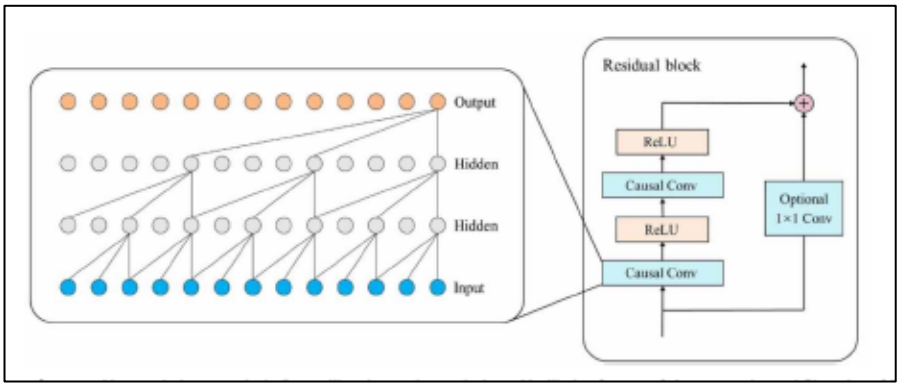


Figure 2-18 TCN Model Structure, (Xu et al., 2021)

2.3 Data Processing for the Development of the NN Model and Study Area

Data collection, data processing, and study area were discussed in the section.

2.3.1 Data Collection and Processing

The study area is divided into several sectors using Geographic and Hydrological features. Topographical data (DEM model, Land Use, Soil Type), Hydro-meteorological data (River stage, rain stage, radar rainfall), and River bathymetry (River cross-section, Manning's roughness) are considered as input (Hussain et al., 2021). In addition, the convergence speed is accelerated by scaling the input variables; therefore, the training process is simulated effectively. Daily runoff series is collected from USGS Hydrological Stations; simultaneously, Daily Meteorological Data is accrued from Weather Underground and NOAA. Besides, watershed geomorphology is extracted from SRTM with one arc-second resolution. Watershed area, mainstream

length, max/min/average elevation, and average slope are accrued as time series data from this portal (de la Fuente et al., 2019).

Further, data pre-processing includes data division and cleaning; the missing data and outlaying data are severe issues for accuracy deficiencies. Data cleaning is the method adopted to solve the problem, while data division is followed to eliminate overfitting and test the predictive capability. Nearly 80% of the data set is considered a training sample, while the balance is testing samples (Jiang et al., 2021).

2.3.2 Physiographic Factors

Basin shape influences the time consumption for the water to move from the remote location to the outlet. In addition, the overland flow is the predominant phase in the small catchment, while the channel flow phase is the significant one in the large catchment. The area of the catchment influences the peak discharge, while the slope of the basin governs the depletion of storage. Generally, the steep slope accelerates the depletion of storage; thus, the time base of the hydrograph is smaller. A higher drainage density leads to the quick disposal of overland flow. In addition, the vegetation cover controls the peak flow, retards the overland flow; the predominant impact is significant in a small catchment (less than 150 km^2) (Subramanya, 2017).

Catchment analysis is a denoted procedure to extract physiographic factors which significantly influence the hydrological processes. The physiographic factors are size, shape, slope, land use and cover, soil type, drainage network, altitude, and geology. However, storage factors are negligible for small catchments because drainage density governs the rate of water supply to the mainstream, which encourages the channel flow to reach the outlet (Vivekanandan, 2019).

Moreover, the shape of the catchment dominates the rise in flood flow over the catchment, where the fast convergence of the flood water is simulated with the semicircular catchment compared to the elongated catchment. Therefore, it takes the shortest time to attain peak flood. Besides, land slope generates a complex interconnection with the hydrograph shape because of its supremacy on infiltration, soil moisture content, and vegetation growth rate. The time of concentration is grievously affected by the land slope attributes of the catchment. In addition, the

overland flow regime is a crucial scenario for small catchments. The time relationship and the peak positions have a reverse behavior than large catchments where the channel flow is dominant (Vivekanandan, 2019).

2.3.3 Climatic Factors

The peak volumes of the runoff are proportional to the rainfall intensity for a given duration. For a given rainfall intensity, the storm - duration is directly proportional to the peak runoff volume—the swift movement of stormwater results in a quicker flow concentration at the outlet. Besides, rainfall intensity influences the small catchment equally as the large catchment. Elongated and narrow catchments behave more sensitively to the storm movement direction (Subramanya, 2017).

2.3.4 Study Area

Flood is the most frequent natural disaster in Sri Lanka due to its unique geographical, geomorphological, and climatological conditions; thus, the economic and social status is severely hit hard. Sri Lanka, as a tropical country, meets four main rainy seasons: First Inter-monsoon (March and April), Southwest monsoon (May to September), Second Inter-monsoon (October and November), and Northeast monsoon (December to February), most predominantly, 75% of rainfall is received during the Southwest and Northeast monsoon. The frequency of floods increased after 1925 and dramatically increased after 1989. Recently, Sri Lanka has undergone many flood circumstances every two to three years.

The Kelani River basin, located in the wet zone, is a highly vulnerable region to annual flooding, which recently experienced an extreme flood event in 2018 (Manawadu & Wijeratne, 2021). It is the fourth-longest river and the second-largest watershed that begins near the Adams Peak and Kirigalpotta region (central hills of Sri Lanka) and reaches the Colombo outfall. It drains nearly 2,300 km^2; location-wise and accommodates rich biodiversity and natural resources. The upper basin is mountainous, while the lower basin is flat. The average annual rainfall for the region is about 2,400 mm, and the peak discharge is nearly 800 – 1,500 m^3/s.

Moreover, the river water is the primary drinking water source for the Greater Colombo public. In addition, the industrial and business region depend on water

resources; therefore, the zone creates job opportunities for the community. In addition, the ecosystem contributes essential support to agriculture, mining, and urban development. The most pleasing landscapes inherently offer a massive development to the tourist industry. Tea, rubber, grass, and forest are the agricultural land used in the upper basin; similarly, the lower basin is highly urbanized (Kottagoda & Abeysingha, 2017).

The Maduru River begins from the Mahiyangana region and reaches the ocean at nearly 135 km, situated in the North Central and Eastern Provinces of Sri Lanka. In addition, it is included in the Accelerated Mahaweli Development Program and Mahaweli Multipurpose Scheme (Mahenthiran & Rajapakse, 2021). According to the climatological behaviour and the measure of annual rainfall, Sri Lanka is divided into Wet, Dry, and Intermediate zones. The wet zone receives 2,500 mm of annual mean rainfall, the intermediate zone receives 1,750 mm to 2,500 mm, and the dry zone receives less than 1,750 mm. Most predominately, the eastern dry zone contributes 25% of paddy production, where the Maduru Oya basin is one of the major river basins and the eighth longest river in Sri Lanka (Zuhail et al., 2007). The risk of overflowing was reported in 2014 in the Batticaloa district due to the prolonged heavy rainfall in the Maduru Oya (Withanage et al., 2018).

2.4 Time Series Forecasting Using NN Models

A neural network combines layers, input data, target data, loss functions, and optimizers; mainly, this is produced with the integration of layers. Besides, the input and the target data are inserted into the model, where the loss function is proposed to compare the predicted and targeted values. In addition, loss values are computed to acknowledge the optimizer, which updates the weights of input parameters. A layer is a data-processing tool that accepts input tensors to derive output tensors; further, this is proposed to represent the tensor format. The simplest vector data is stored as a 2D tensor of shape *(samples, features)*, which functions well in densely connected layers (Fully Connected Layers). Besides, sequence data is stored in the 3D tensor of shape *(samples, timesteps, features)*; mostly recurrent layers such as LSTM are preferable for these sets (Ketkar & Moolayil, 2021).

Developing DL models in Kera is a way of clipping compatible layers to produce supporting data transformation. Two-branch networks, multi-head networks, and inception blocks are the advanced version of a linear stack of layers that maps an input to an output. In-depth, each layer accepts a specific shape of the input tensor and returns a particular shape of the output tensor. Besides, the topology of a network is a hypothesis space; thus, choosing the exemplary network architecture is an art. The gradient descent is applied to learn and update the NN models, which initiates single scaler loss; in a multi-loss network, the loss values are combined as single scaler values (Ketkar & Moolayil, 2021).

Classification, regression, and sequence prediction are standard neural network-based models. Binary cross entropy is a suitable loss function for a two-class classification, while categorical cross entropy is the best for a many-class classification problem. In addition, the mean-squared error is an excellent loss (cost) function for regression, connectionist temporal classification, and sequence-learning problems (Ketkar & Moolayil, 2021). Figure 2-19 indicates the anatomy of the NN algorithm.

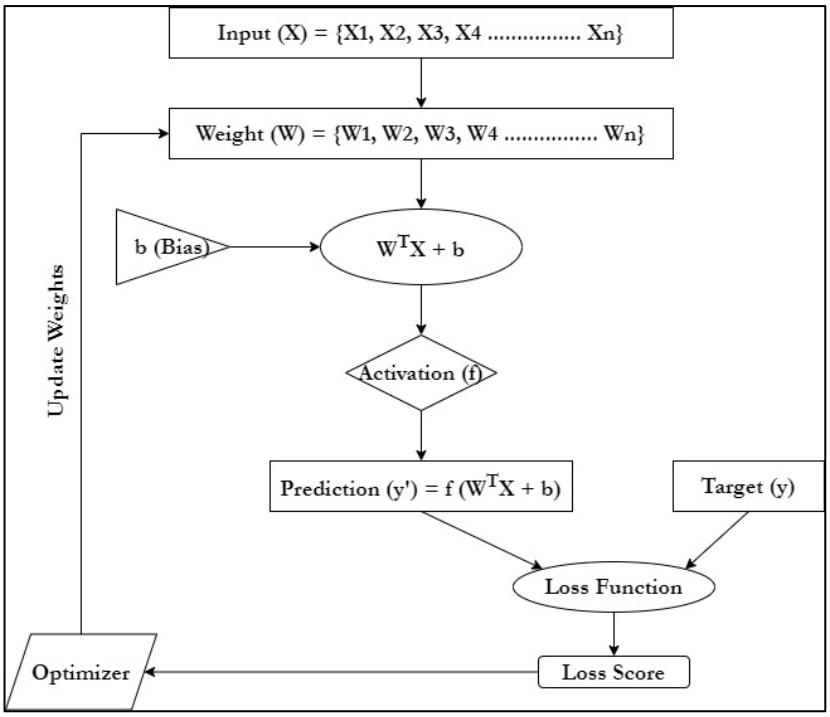


Figure 2-19 Anatomy of a Neural Network

Deep-Learning Framework for Python: Kera, specially developed for researchers, is the deep-learning framework for Python to define and train the models. In addition, it mainly targets fast experimentation and provides easy behaviour to prototype DL models. The framework has built-in support for convolutional networks, recurrent networks, and a combination. The script can be run on CPU or GPU; thus, the user-friendly API quickly supports a prototype of DL models.

The permissive MIT license of Kera makes it capable of freely simulating any commercial project; in addition, it is compatible with any python version. A specialized, well-optimized tensor library is the backbone of the framework; there are three existing backend engines of Kera: TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). Google has developed TensorFlow to simulate a piece of code with Kera without any special considerations. Moreover, it is commonly recommended for the user because it is the most widely adopted, scalable, and production-ready version (Ketkar & Moolayil, 2021).

Developing with Kera: The model configuration should be determined by choosing a loss function, an optimizer, and metrics monitoring, where the iteration is performed on the training data in the final stage. Sequential class and Functional API are the two different methods of defining the models. In addition, sequential class is suitable for linear stacks of layers, which is the typical architecture in Data Science (Ketkar & Moolayil, 2021).

Jupyter Notebook: Preferred way to run Deep-Learning models: Jupyter notebook is editable using the browser. The notebook supports to execution of Python code with text-editing capabilities for annotating. In addition, the platform allows for the breakdown of the long coding into smaller portions, which leads to executing coding independently. Data science and machine-learning communities use this app to accomplish this purpose (Ketkar & Moolayil, 2021).

Regression Problems: Two-classification or binary-classification problems, multi-classification problems deal with positive-negative targets, unlike regression problems. In addition, regression problems are other types of machine-learning problems that predict a sequential value instead of a discrete label, such as

meteorological and weather prediction problems. The mean absolute error (MAE) is proposed in the monitoring metrics. Preprocessing is performed by scaling each feature in different ranges to control accuracy deficiencies (Ketkar & Moolayil, 2021).

Preparing the Data: A box-cox transformation is a one-dimensional conversion technique with a parameter called λ, which enhances the Gaussian or normality of data. However, the applicability is not viable for all the cases (Blum et al., 2022). Eq 2-1 and Eq 2-2 illustrate the formulas.

$$\text{For } \lambda \neq 0,\ y_i = \frac{y_i^{\lambda}-1}{\lambda} \qquad \textit{Eq 2-1}$$

$$\text{For } \lambda = 0,\ y_i = \ln y_i \qquad \textit{Eq 2-2}$$

Feeding the neural network with highly deviating-data ranges must create a problematic situation in the state of accuracy. In addition, data preprocessing is essential to the machine learning model (Ketkar & Moolayil, 2021). Besides, feature-wise normalization is adapted to handle heterogeneous data, which means they are settled as 0 with a unit standard deviation by subtracting the mean from the features and dividing by the standard deviation. The Numpy tool powerfully supports normalizing the features.

Feature selection is a task to eliminate non-informative attributes, and feature engineering is an appropriate way to discretize continuous features to enhance performance. Besides, feature scaling is the process of controlling computational losses (Géron, 2019).

Data Sampling: It mainly deals with Lookback, steps, and delays. Lookback is the observation period to be considered while developing the model architecture. Besides, steps are considered a period for sampling. In addition, the delay is the target period to be predicted. Sampling is essential to control the redundant information of the input. Besides, the python generator is a flexible tool for sampling, which can assemble the recent data to generate batches and targets. Therefore, the time steps of N samples and (N+1) samples have reasonably similar time steps. Alternatively, the samples can be moving sets on the original data, which is the most reliable strategy for hydrological

data. The generator function is constructed based on data, Lookback, delay, index details for validation and test, shuffle (Changing the chronological order is prohibited in forecasting problems), batch size, and steps. Moreover, the function can develop training, testing, and validation sets as varying temporal segments (Ketkar & Moolayil, 2021).

Building the Network: Time series is a sequence of one or more values per time step, and univariate time series consists of a single value per time. Simultaneously, multivariate time series is known as multiple variables represented at a time. The function of the batch-size argument ([batch size, time steps, dimensionality]) is involved in time series generation, which returns the sum of two sine waves of fixed amplitudes but random frequencies plus a small amount of noise. The dimensionality expresses one for univariate time series, while this value is more than one for multivariate time series. In addition, a flat list of features is required to perform the forecasting. The Mean Square Error (MSE) loss and the Adam optimizer are compiled and fitted to the training set for significant epochs; ultimately, the validation set evaluates the model (Géron, 2019).

Two hidden layers, each with 64 units, are adequate to avoid overfitting an NN model with the most straightforward model configuration. Besides, a linear layer is generated with a single unit without an activation function. These layers play a significant role in scalar regression where the predicted values fall between 0 and 1; most predominantly, the last layer is proposed with a sigmoid function. MSE loss function – Mean Squared Error computes the level of coinciding nature of predictions and targets. MAE is a monitoring metric that verifies the training process (Ketkar & Moolayil, 2021).

Window Generator: This technique facilitates a dataset to call consecutive windows where the batch and number of windows must be fed as a reference. Input size, target size, and shift values should be determined to create consecutive windows without shuffling the time series. If it defines as batch(x), then x consecutive windows should be considered for the batch. The first batch contains windows 1 to x, and the second

consists of x+1 to 2x. Besides, each batch can develop a training, validation, and testing set (Hassan & Hassan, 2021).

Baseline Model (BM): Commonly, common-sense baseline models are constructed to ensure the worth of investing further complexity in the model architecture. These are cheap and straightforward machine learning models. The fully connected layers are proposed for the BM, and the input data must be flattened at the top layers of the models before constructing a trainable model.

Linear Model (LM): The straightforward trainable model is proposed with the linear conversion between input and output, where is no interaction between independent prediction steps. No activation function is considered for the model; in addition, mean squared error and mean absolute error is the most common loss functions. Therefore, these models can be categorized as no-memory models. The input features proceed with the model independently, without input states. The model is fed with the entire sequence; consequently, the single data point is derived throughout its performance.

2.4.1 Standard Models

ANN, CNN, RNN: LSTM & GRU, Bi-LSTM, Bi-GRU, and NARX were the standard models.

ANN: It is a stack of dense layers used for the input features without specific features; moreover, it is famous for categorical data. Typically, the number of units in the stack of dense layers equals the number of values forecasted in the regression problem (Ketkar & Moolayil, 2021).

RNN-LSTM and RNN-GRU: An internal model supports building from past information and constantly updating future information. Keeping the memory of previous data is crucial to maintaining a fluid representation in the sequential data prediction. Thus, the internal looping ability of RNN discrete the forecasting capability. The input is encoded into a 2D tensor with the size of *(time steps, input features)*, which elaborates as each time step of looping, the model considers its current state at *t* and input at *t* of shape *(input feature)* and associates those into the output at *t*. Keras framework has allocations for other versions of LSTM and GRU. Hochreiter and Schmidhuber developed the LSTM algorithm in 1997 to overcome the issues like

vanishing gradient. LSTM introduces a way to carry information across the time steps. Therefore, it saves data from the last portion and controls vanishing later signals. Figure 2-22 and Figure 2-23 illustrate it. Recurrent dropout is the built-in feature of the Keras framework, implemented to fight overfitting issues. Stacking recurrent layers increase the suggestive potential of the model.

Chung et al. introduced the GRU modules of RNN in 2014. The function of GRU is like the role of LSTM, and its streamlined ability is cheaper to run. However, the symbolic power is comparably lower for GRU. Overfitting is identified with the visualization of training and validation curves. The training and validation losses begin to separate after the significant epochs. The dropout feature helps to break happenstance correlations in the training data by introducing this argument before the recurrent layer obstructs learning rather than assisting with regularization. Yarin Gal introduced the appropriate method for handling dropout units in 2015. Randomly proposing dropout features disrupt the signal and propagate error (Ketkar & Moolayil, 2021). Figure 2-20 illustrates the configuration of the RNN model.

Bi-LSTM & Bi-GRU: Common RNNs are order-dependent or time-dependent, which operate the time steps of the input sequence in order; shuffling the timesteps changes the representation. Learning both sides of the sequence is impossible with the chronologically ordered RNN. Bidirectional RNNs offer an ideal solution for learning from chronological and anti-chronological directions to merge the terms (Ketkar & Moolayil, 2021). Figure 2-21 illustrates the configuration of the Bidirectional RNNs.

CNN: The convolution operation extracts feature from local input patches and represents modularity. 1D ConvNets perform faster with less computational cost than RNN in simple tasks; in addition, 1D ConvNets with dilated kernels offer a better solution. 1D ConvNets understands the pattern in a sequence locally, and input transformation is performed on every patch where 1D patches (subsequences) extract from the input and deliver the peak output. The layers mentioned above are 2D pooling features to prevent the length of 1D input data (Ketkar & Moolayil, 2021). Figure 2-24 illustrates the configuration of the CNN model.

NARX: The model is famous for various non-linear dynamic systems. The forecasted value is determined from the present and past values and the previous values of the target. This parallel architecture is suitable for multi-step-ahead prediction problems (Boussaada et al., 2018). Addressing the converging speed and local minima is a significant achievement in adapting suitable gradient descent for NARX. The loss gradient updates the parameters to accomplish the tasks (Ketkar & Moolayil, 2021). Figure 2-25 illustrates the configuration of the NARX model.

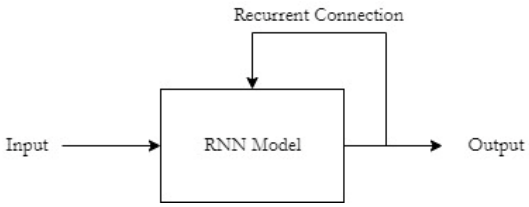


Figure 2-20 Configuration of RNN Model

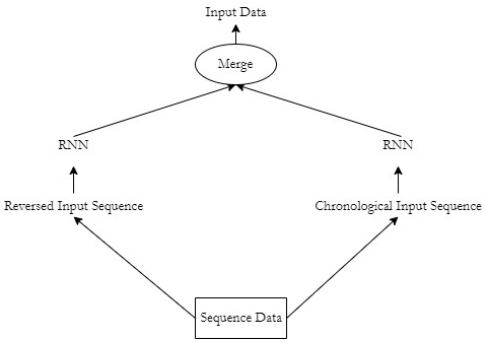


Figure 2-21 Configuration of Bidirectional RNN Model

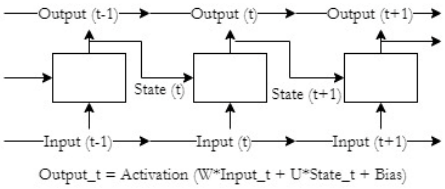


Figure 2-22 Function of RNN Model: Flowchart

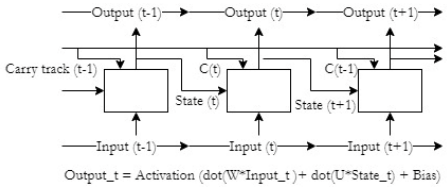


Figure 2-23 Function flowchart: LSTM

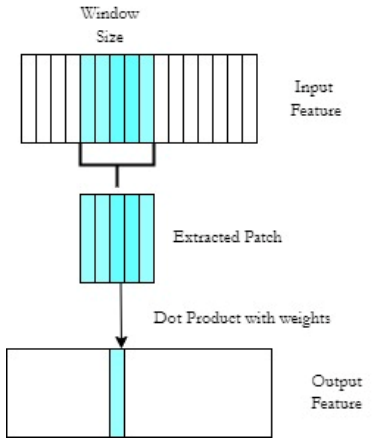


Figure 2-24 Configuration of CNN Model

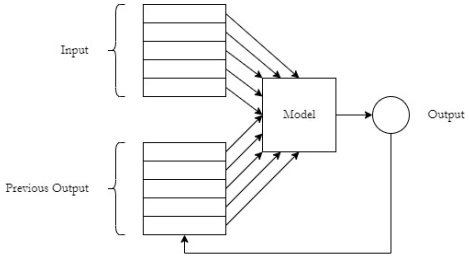


Figure 2-25 Configuration of NARX Model

2.4.2 Hybrid Models

CNN-LSTM, SBU-LSTM, DCNN, GC-LSTM, TCN, and Att-BiLSTM were the hybrid models.

CNN-LSTM: Temporal features are learnt by LSTM, BiLSTM, and GRU, while CNN, CNN-LSTM, and ConvLSTM capture temporal and spatial features. CNN-LSTM integrates standalone models; both CNN and LSTM capture the temporal and spatial patterns separately. Foremost, the processing begins with CNN, and it understands the spatial features of the input set. Eventually, the gathered patterns are converted into one-dimensional features to feed the LSTM unit (Shen & Lin, 2020). Figure 2-26 illustrates the configuration of CNN-LSTM.

Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network (SBU-LSTM): Deep LSTM with hidden layers supports processing with sequential data sets. The architecture generally obeys the feed-forward theories where the previous unit of stacked LSTM feeds the following LSTM unit. BiLSTM captures the backward and forward dependencies of the data. The feature understands the temporal and spatial dependencies. Therefore, BiLSTM is the perfect choice for the initial portion, and the LSTM is placed at the bottom layer to learn the forward dependencies. The comprehensive learning process is enhanced by adding LSTM/BDLSTM layers in the middle of the architecture (Cui et al., 2020). Figure 2-27 illustrates the configuration of SBU-LSTM.

Dilated Casual CNN Model (DCNN): The WaveNet Concept is the basic concept of the model architecture, which is developed by incorporating a dilated (and casual) convolution and a 1 X 1 convolution. Therefore, the model consists of dilated, casual, and residual connections. The estimation is restricted to previous values due to the sparsely connected network, and the gathered information passes across multiple layers (Börjesson & Singull, 2020). Further, the concept introduces maximizing the joint probabilities of the series; thus, it encourages considering exogenous input. Significantly, the model architecture does not work with softmax activation, unlike other WaveNet models (Börjesson & Singull, 2020). Figure 2-28 illustrates the configuration of DCNN.

Graphical Convolution Network – LSTM (GC-LSTM): Graphical convolution (GC) unit extracts the interconnection features of the data set, and the LSTM unit is applied to perform with the matrix data set, which is created using Boolean adjacency of input features. GC converts the spatial data into a concise form for further analysis. Besides, the feature expression efficiency improves the simulation of the model. In addition, LSTM is applied to solve the issues related to long-term and short-term dependencies where the data dependence of samples is sorted out by the adaption (Yu, 2021). Figure 2-29 illustrates the configuration of GC-LSTM.

Temporal Convolution Network (TCN): The 1D convolution network is the basic module of the TCN Encoder Decoder (TCN-ED), where the simple convolution of TCN influences the outputs with present and past input data. Besides, dilated convolutions support the receptive fields with less computation cost. In addition, TCN produces residual connections to combine the previous input and the results of the convolutions. Firstly, the encoder module is fed with information encoded into context value. Secondly, the decoder module decodes the context value into the output sequence. Finally, the Encoder-Decoder architecture is introduced to compress the input features into a definite-length vector to make the tensor flow more stable (Shen & Lin, 2020). Figure 2-30 illustrates the configuration of TCN.

Attention -Based Bidirectional LSTM (Att-BiLSTM): Input series data is applied to the bidirectional LSTM layer to learn the input dependency features, which layer acts like an encoder structure for the input series. Thus, it understands the weight dependence of the input series in both forward and backward directions. After encoding the input, a new series is received and inserted into the attention mechanism. Finally, an additional layer is allocated at the bottom level to decode the series as output (Lee et al., 2022). Figure 2-31 illustrates the configuration of Att-BiLSTM.

2.5 Identification of the Best Models

The *compile ()* method specifies the loss function and the optimizer, which argument proceeds with training and evaluation. Besides, the *fit ()* function is proposed to pass the input features, targeted classes, and epochs for training the models. The loss and monitoring metrics are provided for each epoch to indicate the forecasting quality. In

addition, overfitting is grasped by understanding loss growth on training and validation sets; additionally, the early stopping argument of the callbacks module (*tf.keras.callbacks.EarlyStopping*) supports avoiding overfitting. The command helps detect the position where the validation loss is no longer improving. Further, EarlyStopping callbacks interrupt the training iteration once a monitoring metric fails to show further improvement for a fixed epoch (Géron, 2019).

The analysis is estimated by evaluating the performance using four statistical tools: Residual Mean Maximum (RM), Nash-Sutcliffe Efficiency (NSE) / Coefficient Determination (R^2), Mean Absolute Error (MAE), and Root Mean Square (RMSE) (Jiang et al., 2021). NSE is the standard application to explain the linearity of observed and predicted values, and R^2 similarly indicates the collinearity between predicted and observed data. RM describes the mean tendency of predicted and observed values to understand the deviation. Finally, RMSE and MAE measure the performance of NN models directly (Jimeno-Sáez et al., 2018). Table 2-1 illustrates the statistical tools to measure the performance of forecasting models.

In regression neural network analysis, Mean Squared Error and Mean Absolute Error play a significant role in comparing observed and predicted values. Because these two are fed to the compile and fit procedure of the NN model.

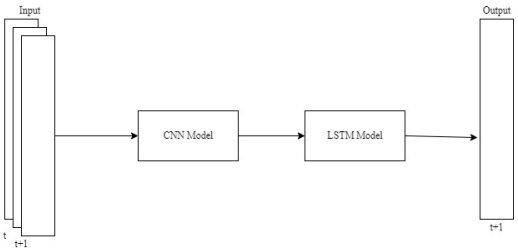


Figure 2-26 Configuration of CNN-LSTM Model

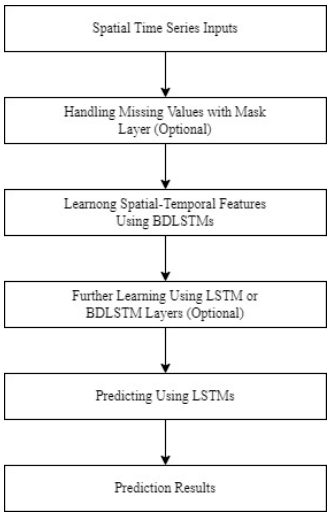


Figure 2-27 Configuration of SBU-LSTM Model

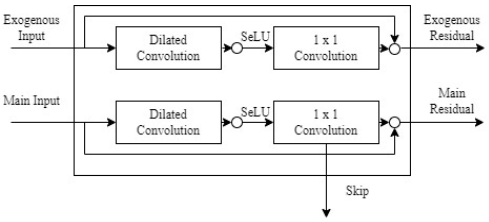


Figure 2-28 Configuration of DCNN Model

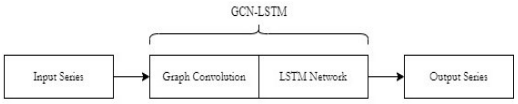


Figure 2-29 Configuration of GC-LSTM Model

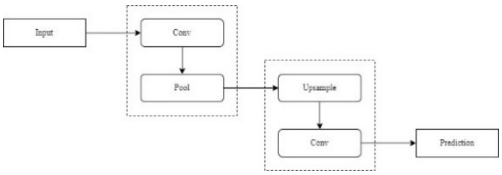


Figure 2-30 Configuration of TCN Model

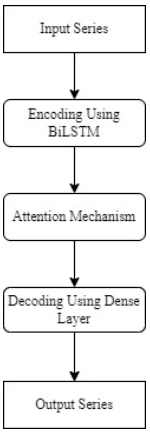


Figure 2-31 Configuration of Att- Bi LSTM Model

The flow duration curve is a cumulative frequency curve expressing the percentage of time specified flow rate equalled or exceeded during a given period. Without understanding the sequence of occurrence, FDC incorporates into a single graph to explain the flow characteristics of the stream, and Figure 2-32 illustrates the components of FDC.

Table 2-1 Statistical Model for Evaluation of Performance, Jimeno-Sáez et al., 2018

Equations	Range
$R^2 = NSE = 1 - \frac{\sum_{i=1}^{n}(O_i - E_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$	[0, 1]
$RM = \max \lvert(O_i - E_i)\rvert$	[0, ∞)
$MAE = \frac{\sum_{i=1}^{n}\lvert(O_i - E_i)\rvert}{n}$	[0, ∞)
$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(O_i - E_i)^2}{n}}$	[0, ∞)

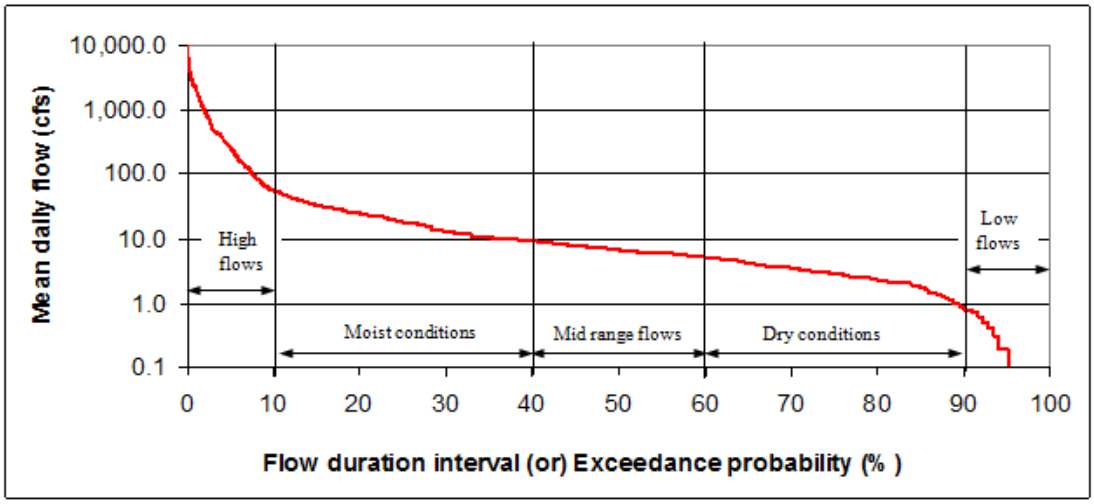


Figure 2-32 Flow Duration Curve and Flow Characteristics

The high flow range of FDC explains the dynamic response of the catchment; most significantly, it contains information on catchment characteristics. Therefore, evaluation points (EPs) are chosen from the FDC to compare observed and predicted values. Two different practices are available for EP selection: discharge method and volume method. The discharge method mainly targets the high flow range; in contrast, the volume method is appropriate for water balance problems which give attention to both high and low flows. Therefore, the volume method is suitable for comparing actual and forecasted FDCs (Westerberg et al., 2011).

Step 1: The area under the FDC is divided into 20 equal classes/volume increments of 5% (Rescaling the total volume into small volume portions).

Step 2: The observed flow rate is placed into each interval, and the upper and lower boundaries of the strips are noted.

Step 3: The scaled scores of the predicted values are calculated using the triangular weighting function.

Step 4: Performance is measured using Eq 2-3.

$$\text{For } \lambda \neq 0,\ y_i^{\lambda} = \frac{y_i^{\lambda}-1}{\lambda} \qquad \textit{Eq 2-3}$$

Figure 2-33 graphically illustrates the steps. Hydrological data generally consists of noise and missing information. It deviates due to the complex and non-linear nature of hydrological processes. Significantly, it affects the forecasting abilities of data-driven models. The performance of models is expressed by comparing the model performance using statistical models (Hassan & Hassan, 2021).

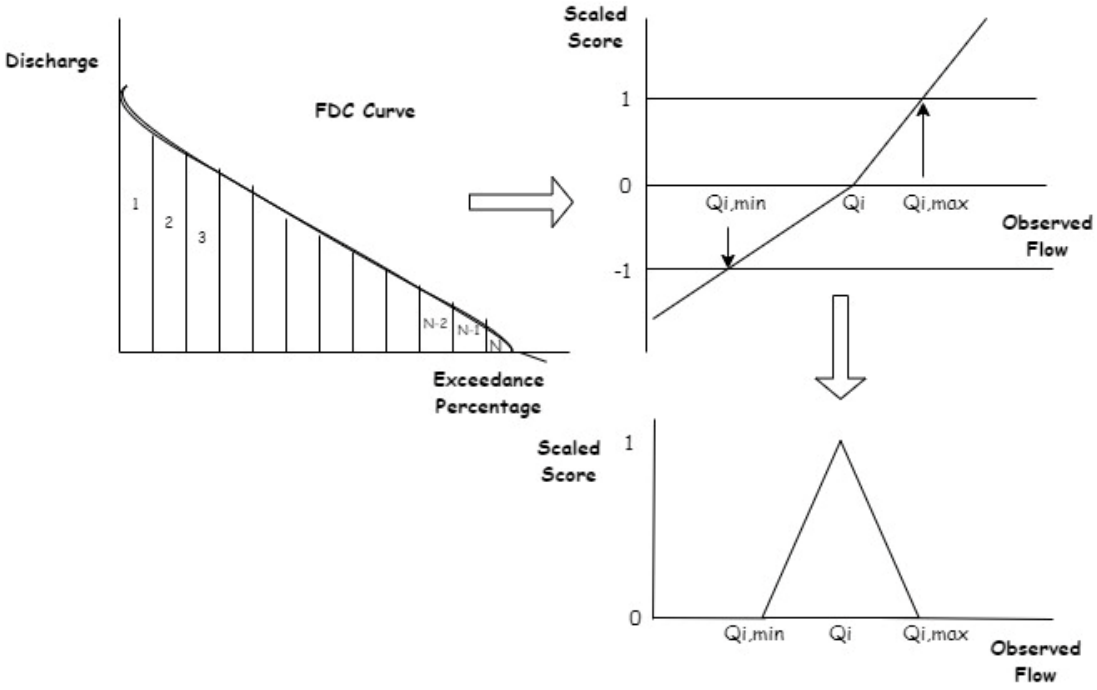


Figure 2-33 Behavioral Error Estimation

2.6 Uncertainty Analysis

The uncertainty analysis is a perfect tool to investigate the reliability of NN forecasting models, where the analysis is conducted using statistical measures. In addition, the uncertainty measures of the models are expressed as the fraction of core model performance. Besides, data-driven forecasting models deal with various uncertainty sources, such as the importance of data handling and input combinations (Shamshirband et al., 2019). Table 2-2 illustrates the uncertainty parameters and description.

Table 2- 2 Uncertainty Analysis and Parameters

Parameters	Characteristics	Description
Input Handling	Without Box-Cox Transformation	It determines the influences of extremities location and computational losses.
	Without Box-Cox Transformation and Normalization	
Input Distribution	Rainfall only	It illustrates the importance of input parameters considered for the study.
	Rainfall & NDVI	
	Rainfall & Evaporation	
	Rainfall & Relative Humidity	

2.7 Sensitivity Analysis

A sensitivity analysis among artificial neural networks provides an opportunity to assess the importance of model architecture that contributes to the functionality of NN.

Table 2-3 Sensitivity Analysis and Parameters

Sensitivity Parameters	Model Characteristics	Description
Learning Rate	0.001	It influences the time length of convergence. For instance, a high rate contributes to fast convergence, while a low rate provides slow convergence.
	0.1	
	0.01	
	0.0001	
Optimizers	Adam	It combines the momentum and RMSProp, which monitors an exponentially declining mean of past squared gradients.
	Nadam	Adam optimization and Nesterove trick produce this optimizer. Thus, it is faster converging than Adam.
	SGD	It is a momentum optimizer that considers the previous gradients at each iteration.
	RMSProp	It fixes the issue related to fast convergence without reaching a global optimum.
Input	Climatic & Physiographic	Dimensionality reduction is the way to control the complexity of forecasting.
	Climatic only	
	Physiographic only	
Sensitivity Parameters	**Model Characteristics**	**Description**
Lead Time Span	30 Days	Lead time and span length are significant in assessing the model performance.
	15 Days	
	45 Days	

In addition, it can be expressed based on the primary model quality (Song et al., 2020); therefore, the R^2 values represent the performance (Mrzygłód et al., 2020). The architecture of the models (learning rates, optimizers), input parameters, and lead time play a significant role in a sensitivity analysis. Table 2-3 illustrates the parameters and characteristics.

2.8 Transfer Learning Technique

Reusing the pre-trained layer is a practical way to avoid rebuilding the NN models from scratch, which can be proceeded with transfer learning techniques. Generally, transfer learning offers favorable results for the inputs with similar features to core models that should express an accuracy greater than 90%. The bottom layer of the existing model must be restored with a new layer suitable for a new task. An exact number of reusable layers must be determined based on the freezing and unfreezing layers. In addition, the weights of reusable layers can be kept constant by freezing those layers (Géron, 2019). Figure 2-34 visualizes the theory behind the transfer learning technique.

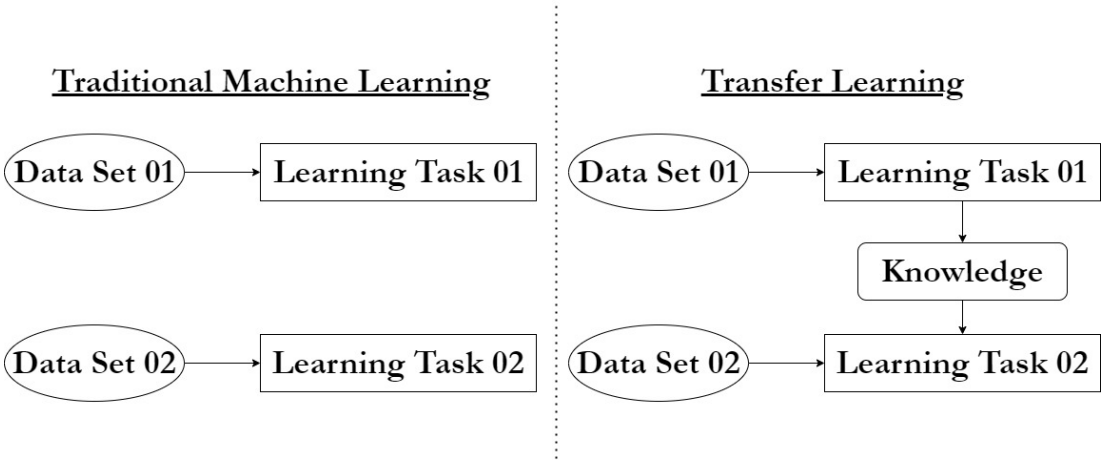


Figure 2- 34 Theory of Transfer Learning

CHAPTER 3: METHODOLOGY

The methodology flowchart, study area, data checking, developing NN models, the identification of the best models, uncertainty analysis, sensitivity analysis, and pre-trained models are discussed in this section.

3.1 Methodology Flowchart

Daily rainfall and evaporation data were collected from the Meteorological Department of Sri Lanka, while streamflow data were gathered from the Irrigation Department of Sri Lanka. Wind speed, temperature, relative humidity, and solar irradiance were acquired from NASA's Power Data Access Viewer website, which are the influencing factors of transpiration. The soil wetness index was derived to represent the initial losses, like infiltration losses, gathered from the same source. Hydrological cleansing was applied to the streamflow and rainfall, while the missing values in the evaporation data were filled with data science techniques. Box-cox transformation was adapted to bring gaussian distribution on input data sets. Besides, scaling and normalization were applied in the final stages to control computational losses.

Thirteen NN models, including standard and hybrid models, were developed with Python Jupyter notebook to compile, fit, and test the Kelani Basin data. First, grid analysis was developed using statistical tools such as the determination of coefficient (R^2), root mean square error (RMSE), mean absolute error (MAE), and flow duration curve error (FDC-Q). Next, uncertainty analysis was proposed to investigate the input characteristics, such as input combination and input-handling techniques. Besides, sensitivity analysis was applied to verify the model architecture, such as types of optimizers, lengths of the horizon, input parameters, and learning rates. Finally, the NN models with R^2 of more than 0.9 were implemented to develop pre-trained models on the Maduru Oya Basin. Figure 3-1 expresses the methodology flowchart adapted for the study.

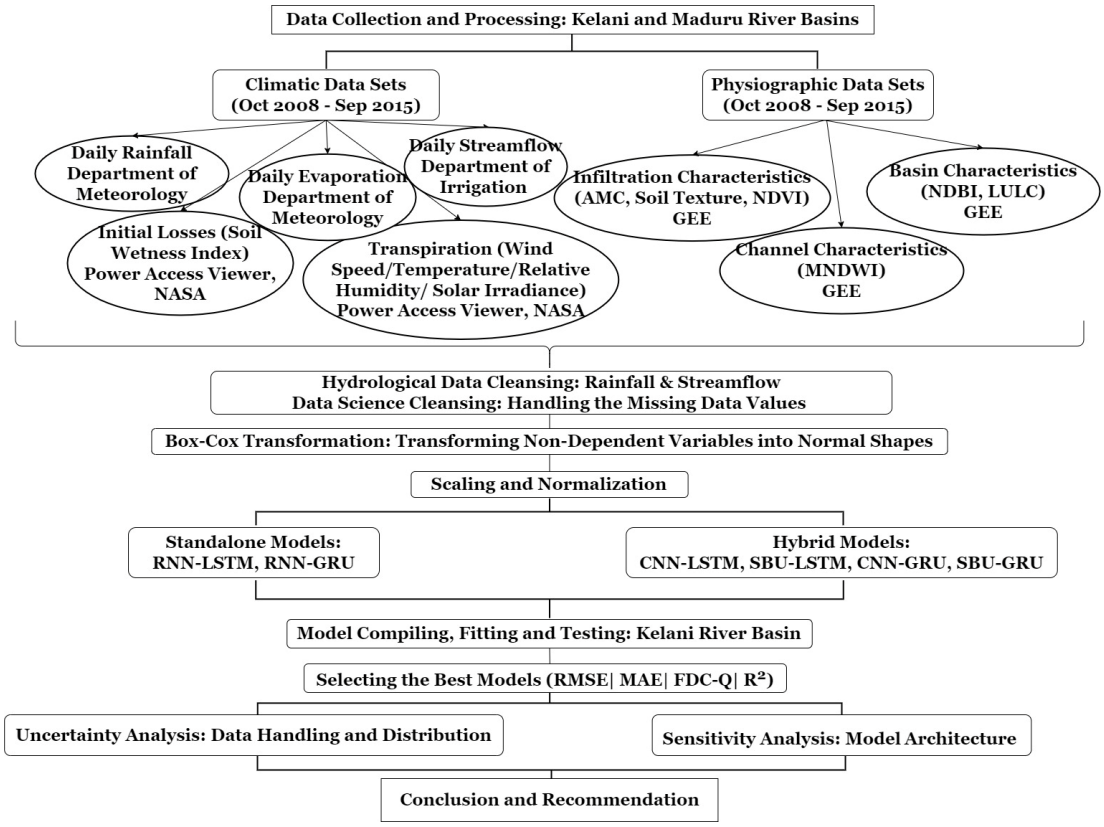


Figure 3-1 Methodology Flowchart

3.2 Study Area

For the present study, Kelani River Basin and Maduru River Basin were selected to develop flood forecasting models.

Kelani River Basin: The entire catchment is situated in the wet zone of Sri Lanka, where the average rainfall varies from 5,700 mm in the upper stream to 2,300 mm in the lower stream. Consequently, the lower basin is frequently subjected to extreme flood conditions; the minor floods have not led to severe damages because of the retardation structures. Figure 3-2 illustrates the Kelani River Basin and its river network. The confluence point is identified at Hanwella; therefore, the streamflow station was chosen as Hanwella for flood forecasting (Hettiarachchi, 2020).

Maduru Oya Basin: The catchment is in the dry zone of Sri Lanka, which receives 3,060 million m^3 of rainfall annually for a 1,541 km^2 catchment area (Mahenthiran & Rajapakse, 2021). Natural vegetation and cultivated lands are the effective land use and land cover patterns of the basin, which is severely affected by heavy inundation

due to a prolonged peak flood (Withanage et al., 2018). Figure 3-3 illustrates the Maduru Oya basin and river network.

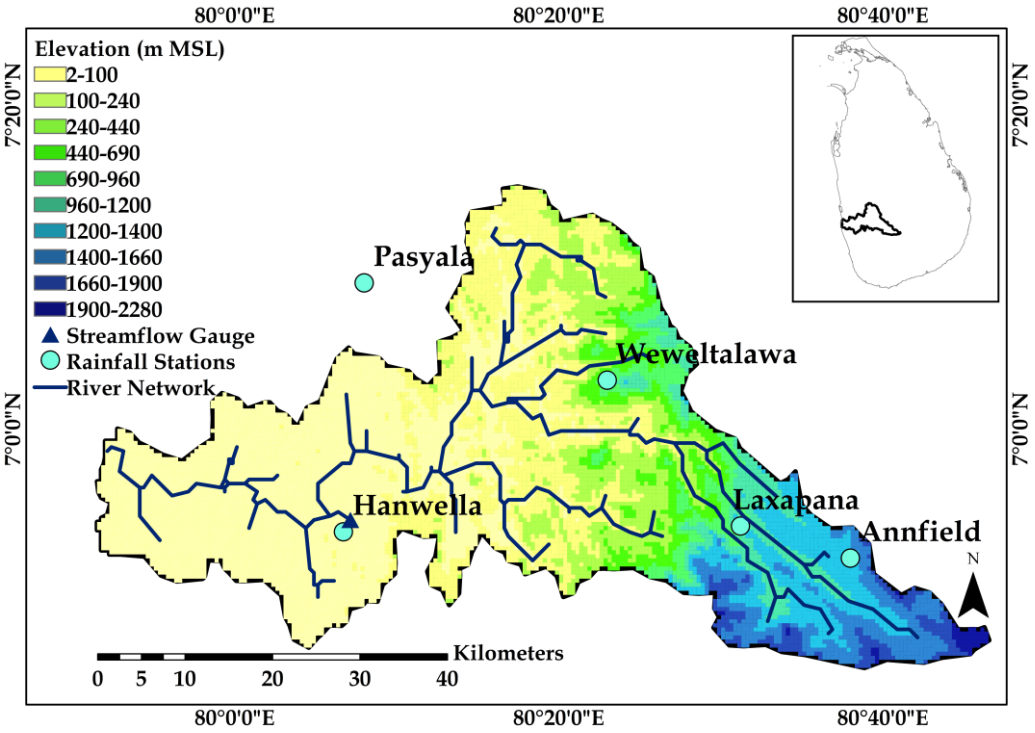


Figure 3-2 Location Map (Kelani River Basin) with DEM

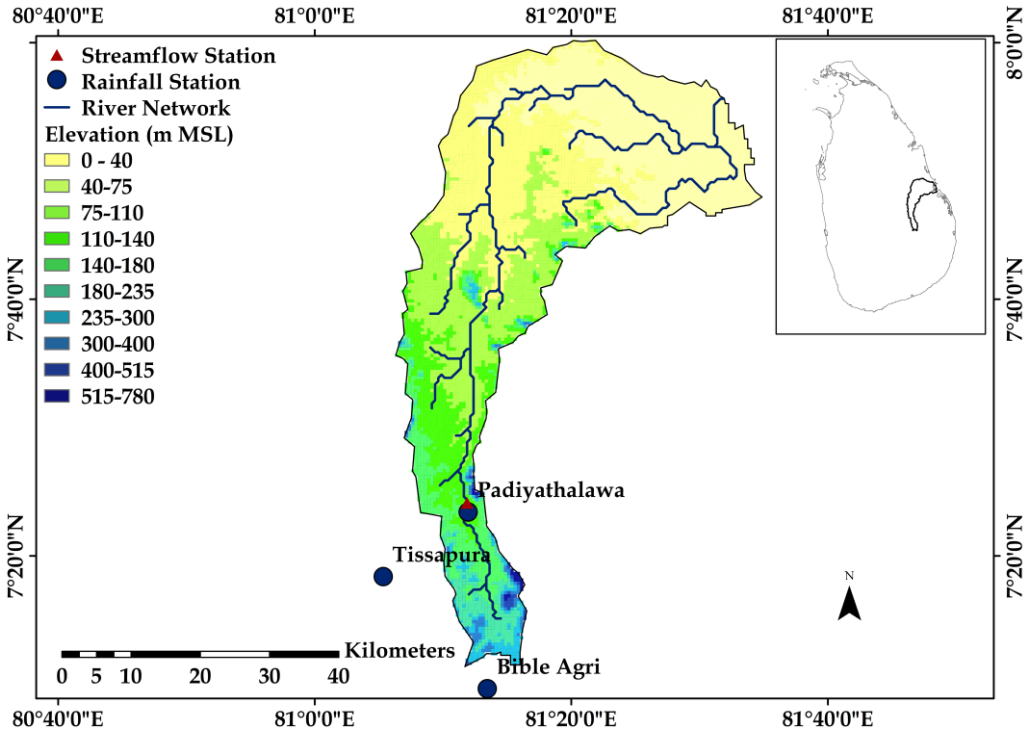


Figure 3-3 Location Map (Maduru Oya) with DEM

3.3 Data and Data Checking

Climatic and physiographic data checking was conducted in the section.

3.3.1 Data Source and Resolution: Climatic Data

Daily streamflow data of the Hanwella and Padiyathalawa river gauges were collected from the Department of Irrigation; In contrast, daily rainfall data for every three stations of Kelani and Maduru Oya were collected from the Meteorological Department, Sri Lanka. In addition, the evaporation data sets were obtained from the Meteorological Department, Sri Lanka. Unfortunately, no sources are available to collect the transpiration data set directly. Therefore, the most impactful factors: wind speed, temperature, relative humidity, and solar irradiance, were gathered to handle the absence of transpiration data. Like, transpiration data, initial loss data was another vulnerable data set to acquire straight away from the literature. Therefore, the soil wetness index was gathered to represent this parameter. Table 3-1 shows the data types and those resolutions. Thiessen rainfall and evaporation stations are tabulated in Table 3-2 for Kelani and Maduru.

Table 3-1 Data Type and Data Resolution

Data Type	Temporal Resolution	Data Period	Data Source
Rainfall	Daily	Oct 2008 – Sep 2015	Department of Meteorology
Streamflow	Daily	Oct 2008 – Sep 2015	Department of Irrigation
Evaporation	Daily	Oct 2008 – Sep 2015	Department of Meteorology
Transpiration (Wind Speed, Temperature, Relative Humidity, Solar Irradiance)	Daily	Oct 2008 – Sep 2015	Power Data Access Viewer Website, NASA
Initial Losses (Soil Wetness Index)	Daily	Oct 2008 – Sep 2015	Power Data Access Viewer Website, NASA

Visual Data Checking: Rainfall-Runoff

Visual data checking was carried out to identify the abnormalities of the hydrological series, which is elaborated by plotting the response of streamflow and rainfall for stations to verify the patterns. The visual checking is attached in Appendix A and Appendix D. Besides, the abnormalities in the data set were considerably low according to the visual checking.

Table 3-2 Comparing the Area of Stations with WMO Standards

Kelani River Basin			
No	**Rainfall Stations**	**Area**	**WMO Standards, (Subramanya, 2008)**
1	Hanwella	357 km^2	600 km^2 – 900 km^2
2	Pasyala	084 km^2	600 km^2 – 900 km^2
3	Weweltalawa	083 km^2	600 km^2 – 900 km^2
No	**Evaporation Station**	**No of Stations**	**WMO Standards**
1	Colombo	01	A Station for every 50, 000 km^2
Maduru River Basin			
No	**Rainfall Stations**	**Co-ordinates**	**WMO Standards, (Subramanya, 2008)**
1	Padiyathalawa	093 km^2	600 km^2 – 900 km^2
2	Tissapura	014 km^2	600 km^2 – 900 km^2
3	Bible Agri	069 km^2	600 km^2 – 900 km^2
No	**Evaporation Station**	**No of Stations**	**WMO Standards**
1	Girandurukotte	01	A Station for every 50, 000 km^2

Annual Water Balance: Rainfall-Runoff

Table 3-3 illustrates the annual runoff coefficient values; Figure 3-4 and Figure 3-6 shows the graphical representation of the table. Simultaneously, Figure 3-5 and Figure 3-7 express the annual rainfall and streamflow patterns.

Table 3-3 Annual Runoff Coefficient and Annual Evaporation

Kelani River Basin				
Year	**Annual Rainfall (mm/year)**	**Annual Streamflow (mm/year)**	**Annual Evaporation (mm/year)**	**Annual Runoff Coefficient**
2008/2009	3,777.9	1,265.4	1,278.3	0.335
2009/2010	4,043.8	1,661.0	1,205.6	0.411
2010/2011	4,579.9	1,960.1	1,171.1	0.428
2011/2012	2,516.2	630.9	1,269.8	0.251
2012/2013	4,648.6	2,133.2	1,207.2	0.459
2013/2014	3,367.9	1,154.0	1,317.9	0.343
2014/2015	3,379.3	1,523.1	1,209.8	0.451
Average	3,759.1	1,475.4	1,246.4	0.400
Madhuru River Basin				
Year	**Annual Rainfall (mm/year)**	**Annual Streamflow (mm/year)**	**Annual Evaporation (mm/year)**	**Annual Runoff Coefficient**
2008/2009	3,386.78	443.12	390.67	0.131
2009/2010	4,467.43	844.39	744.44	0.189
2010/2011	7,636.30	5,017.89	4,423.94	0.657
2011/2012	3,087.71	1,586.46	1,398.68	0.514
2012/2013	7,065.16	2,729.86	802.25	0.386
2013/2014	1,674.08	336.99	99.03	0.201
2014/2015	7,679.95	3,439.39	1,010.76	0.448
Average	4,999.63	2,056.87	1,267.11	0.360

The runoff coefficients of Kelani Basin vary from 0.34 to 0.46, where the water year 2011/12 shows a considerable deviation. In contrast, the values of Maduru Basin fall from 0.19 to 0.66, where the water year 2010/11 illustrates a significant variation.

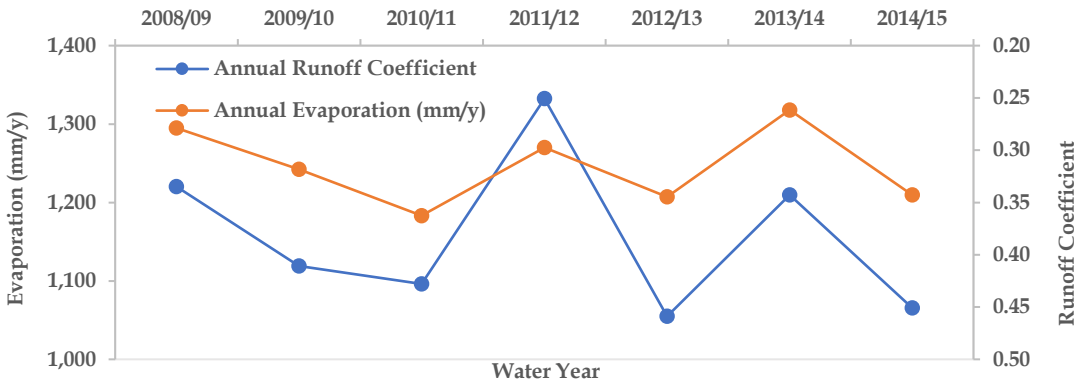


Figure 3-4 Annual Water Balance – Kelani

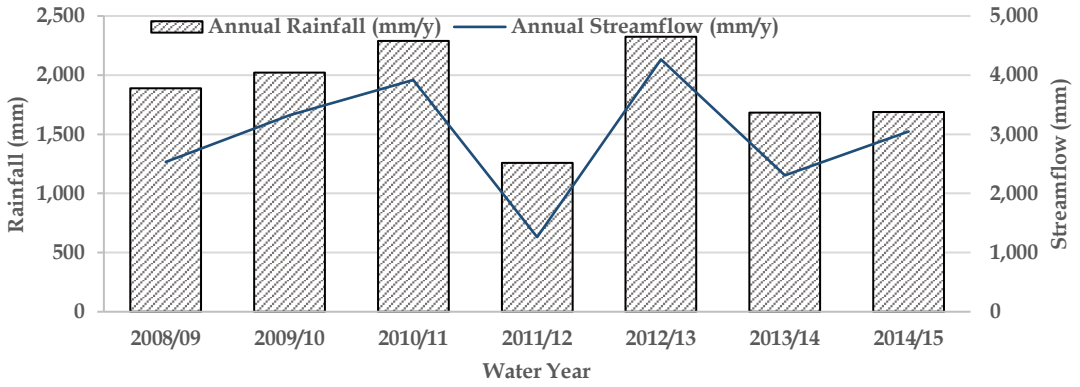


Figure 3-5 Annual Rainfall and Annual Streamflow - Kelani

Due to missing values in the rainfall and evaporation, the data expressed disagreements at a few points. However, those values were filled using the data-cleaning process of machine learning.

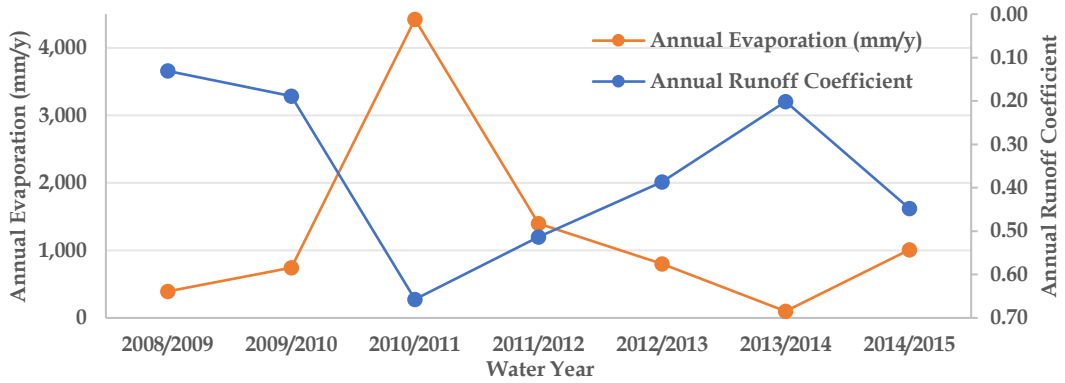


Figure 3-6 Annual Water Balance – Maduru Oya

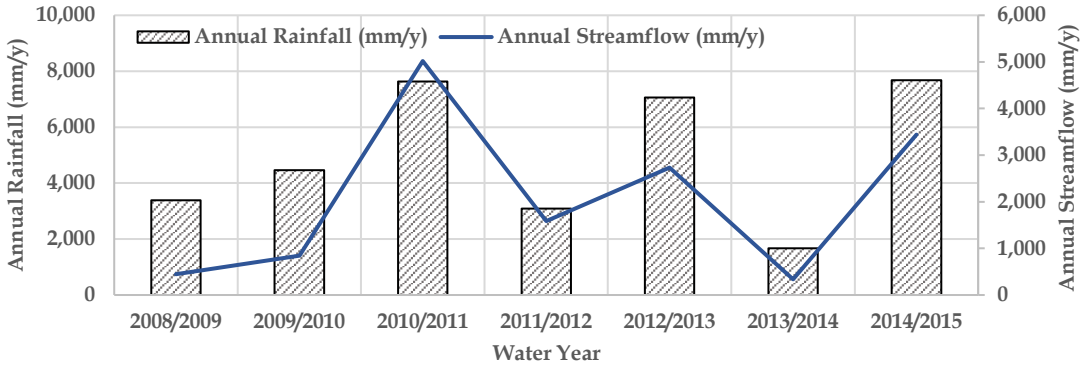


Figure 3-7 Annual Rainfall and Annual Streamflow - Maduru Oya

Thiessen Average Rainfall

The average rainfall was calculated by multiplying the rainfall values and the respective Thiessen weights. Table 3-4 illustrates the Thiessen weights of Kelani and Maduru rainfall stations.

Table 3-4 Thiessen Weights of Stations

Kelani River Basin			
No	**Rainfall Stations**	**Co-ordinates**	**Thiessen Weights**
1	Hanwella	80.117º E, 6.883º N	0.19
2	Pasyala	80.133º E, 7.150º N	0.10
3	Weweltalawa	80.383º E, 7.050º N	0.38
4	Laxapana	80.520º E, 6.900º N	0.20
5	Annfield	80.633º E, 6.867º N	0.13
No	**Streamflow Station**	**Co-ordinates**	
1	Hanwella	80.117º E, 6.883º N	
No	**Evaporation Station**	**Co-ordinates**	
1	Colombo	79.861º E, 6.927º N	
Maduru River Basin			
No	**Rainfall Stations**	**Co-ordinates**	**Thiessen Weights**
1	Padiyathalawa	81.175º E, 7.396º N	0.53
2	Tissapura	81.089º E, 7.457º N	0.08
3	Bible Agri	81.220º E, 7.150º N	0.39
No	**Streamflow Station**	**Co-ordinates**	
1	Padiyathalawa	81.175º E, 7.396º N	
No	**Evaporation Station**	**Co-ordinates**	
1	Girandurukotte	81.018º E, 7.463º N	

Visual data-checking strategies were applied again on streamflow, and Thiessen averaged rainfall to visualize the overall response. The graphs are attached in

Appendix B and Appendix E. Figure 3-8 and Figure 3-9 illustrate the Thiessen polygons of the Kelani and Maduru River Basins.

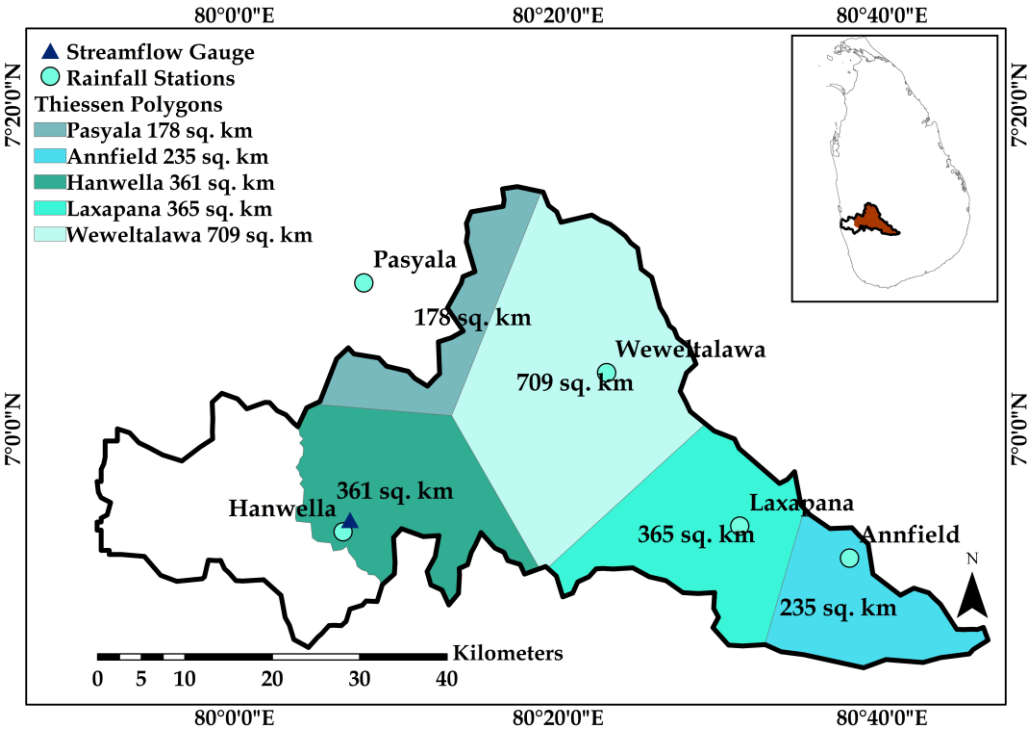


Figure 3-8 Thiessen Polygon of Kelani River Basin

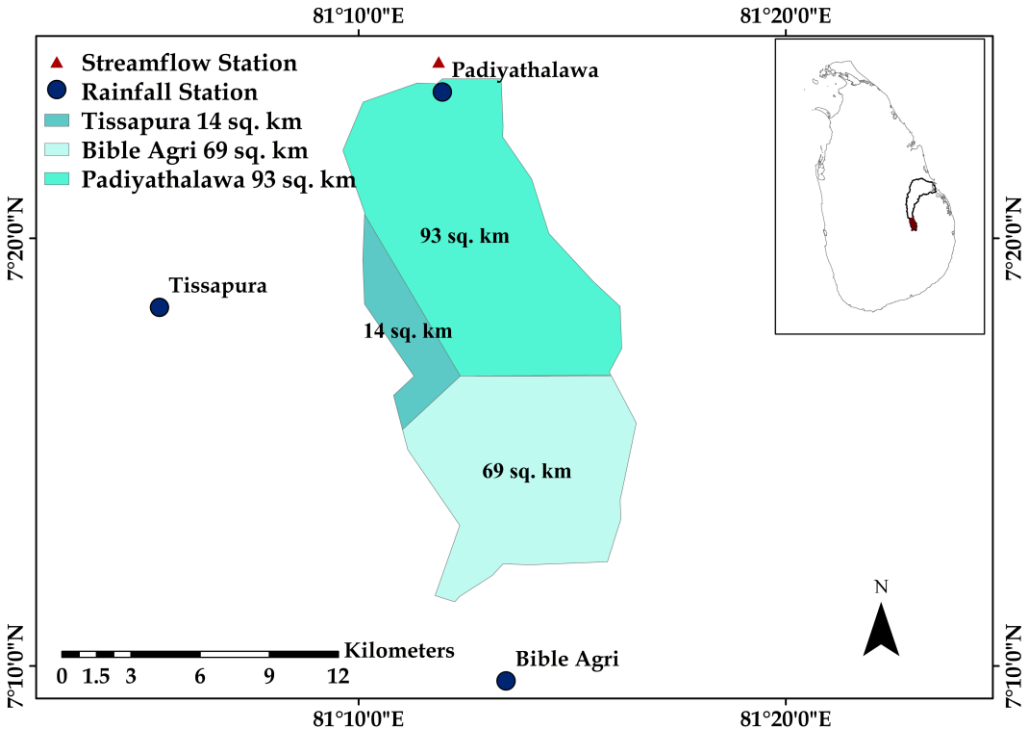


Figure 3-9 Thiessen Polygon of Maduru Oya Basin

Single Mass Curve: Rainfall

The missing data destroys the continuity of the rainfall record due to the damage to the rainfall gauges. Therefore, the perioding trends are considered for testing the impact of missing values. In addition, the closet station patching method is the appropriate hydrological imputation method using a single mass curve, where the slope values of each station were computed to fill in the missing values (Subramanya, 2017). Figure 3-10 and Figure 3-11 shows the single mass curves of Kelani and Maduru Oya, respectively.

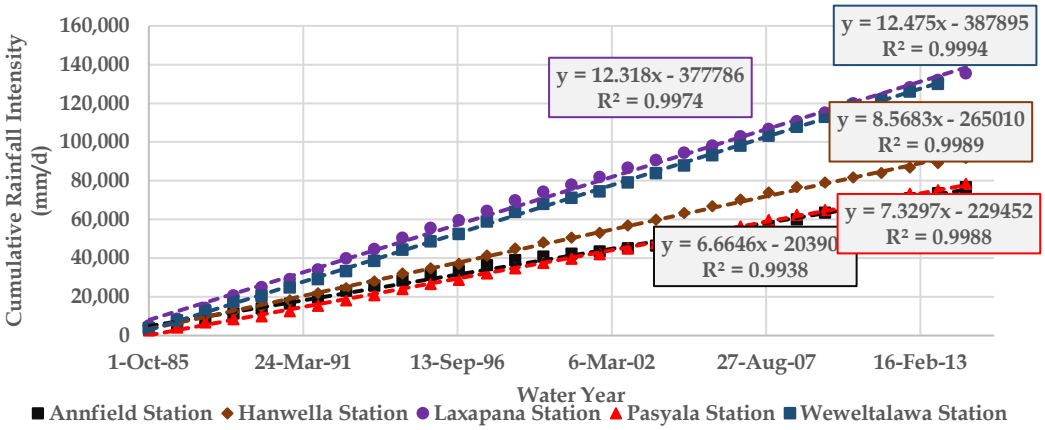


Figure 3-10 Single Mass Curve – Kelani

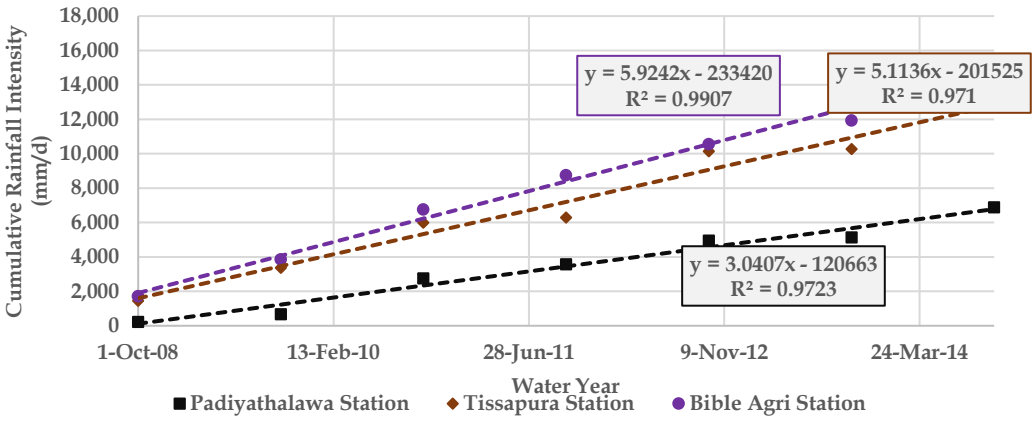


Figure 3-11 Single Mass Curve - Maduru Oya

Double Mass Curve: Rainfall

The hydrological behaviour is compared with the other station by plotting a double mass curve, where inconsistency in the trend induces due to shifting gauges,

environmental circumstances, and observational errors. Besides, the perfect linearity of the double mass curve indicates the best relationship (Subramanya, 2017). Figure 3-12 and Figure 3-13 represent the double mass curves of Kelani and Maduru Oya, respectively.

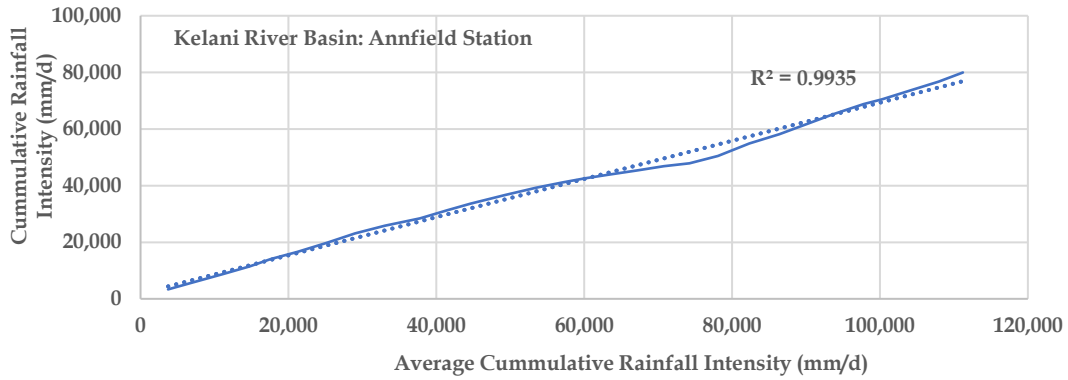


Figure 3-12 Double Mass Curve of Stations - Kelani

$R^2 = 0.9935$, $R = 0.996$

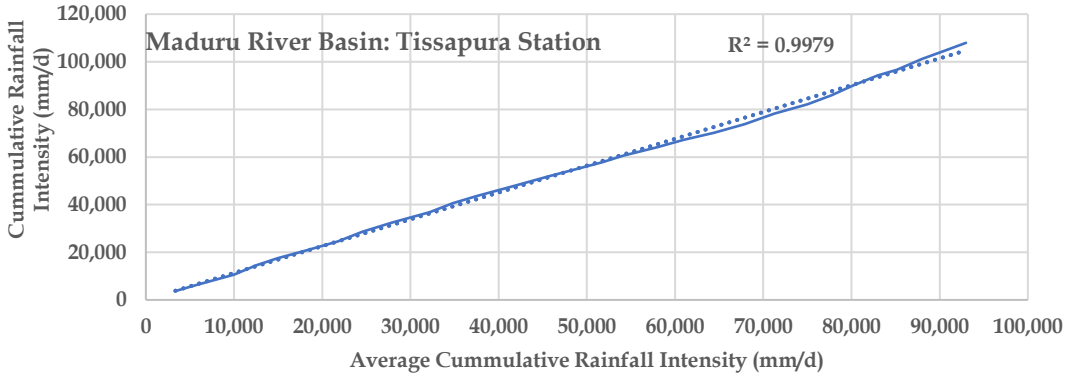


Figure 3-13 Double Mass Curve of Stations - Maduru Oya

$R^2 = 0.9979$, $R = 0.9989$

The perfect linearity was expressed on the double mass curves of the stations. The graphs are attached in Appendix C and Appendix F.

Data Checking: NASA Power Access Data (Transpiration & Initial Losses)

The previous research on evaluating the quality of NASA POWER reanalysis product was concluded with the following findings, which illustrates in Table 3-5.

Table 3-5 Comparing the quality of NASA POWER ACCESS data.

Data Type	Quality of Data
Temperature	Identified with high accuracy
Solar Irradiance	Identified with high accuracy
Wind Speed	Required local bias correction
Relative Humidity	Required local bias correction
Soil wetness Index	Required local bias correction

However, the source is a beneficial source of data for the regions identified with missing or unavailable ground weather stations (Rodrigues & Braga, 2021).

3.3.2 Data Source and Resolution: Physiographic Data

Google Earth Engine (GEE) and Data Collection: GEE is a portal that allows the user to perform geospatial analysis. A considerable amount of physiographic data is available on the GEE cloud, which appreciates its cloud computing features (Nashwan et al., 2019).

Runoff Coefficient: Dynamic Land Use Land Cover (MCD12Q1.006 Modis LULC), rainfall (Climatic Hazards Group InfraRed Precipitation (CHIRPS)), Global Soil Data (Open Land Map), and Soil Conservation Service Curve Number (SCS CN) model are required to derive the model. Eq 3-1 to Eq 3-8 represent the computation steps of the SCS CN model. The CN method is a perfect tool to develop a rainfall-runoff model for small or medium catchments. .

MODIS Landcover type (MCD12Q1.006) provides land-cover maps at yearly temporal and 0.050 spatial resolution in geographic latitude/longitude projection from 2001 to the present. In addition, the layers categorized in the International Geosphere-Biosphere Program (IGBP) classification scheme provide labels for Landsat-based land cover. Table 3-6 illustrates the MODIS land cover details.

Further, Open Land Map soil texture class (USDA system) provides soil texture classes for six different depth conditions at 250 m resolution; significantly, soil texture details at 0 cm depth are imposed for formal analysis. Table 3-7 illustrates the soil type details of the USDA system.

Filter functions were involved in shortlisting the data sets from the image collections. In addition, the soil texture map was converted to a four-class hydrological soil using the ternary operator in the GEE platform, which was added as a second band to the downscaled LULC MODIS data. CN(II) was prepared by considering 17 LULC classes and four soil texture classes. Besides, CN(I) and CN(III) were estimated by CN(II), which was considered along with AMC to create S images. Daily AMC images were gathered using rainfall data to receive the sum of the previous five days. In addition, surface runoff values were estimated using the SCS CN method (Jain et al., 2021). Figure 3-14 and Figure 3-15 illustrate the runoff coefficient values.

$P = I_a + F + Q$ ..... *Eq 3-1*

$\frac{Q}{P - I_a} = \frac{F}{S}$ ..... *Eq 3-2*

$I_a = \lambda S$ ..... *Eq 3-3*

$Q = \frac{(P - I_a)^2}{P - I_a + S}$ for P >/= I_a , Q = 0 for P </= I_a ..... *Eq 3-4*

$S = 25.4\left(\frac{1000}{CN} - 10\right)$ ..... *Eq 3-5*

$CN(I) = \frac{CN(II)}{2.281 + 0.0128CN(II)}$ ..... *Eq 3-6*

$CN(III) = \frac{CN(II)}{0.427 + 0.00573CN(II)}$ ..... *Eq 3-7*

$\frac{Q}{P} = \frac{\left(1 - \frac{\lambda S}{P}\right)^2}{\left(1 + \frac{(1-\lambda)S}{P}\right)}$, $\frac{Q}{P}$ is used to represent the $\frac{S}{P}$ ..... *Eq 3-8*

Where, P: Daily Rainfall, I_a: Initial Abstraction, F: Actual Retention, λ: Initial Abstraction Coefficient, Q: Direct Runoff, S: Potential Maximum Retention, AMC: Antecedent Moisture Condition

CN(I): AMC < 13 mm

CN (II): 13 mm =/< AMC < 28 mm

CN(III): AMC >/= 28 mm

Table 3-6 MODIS Landcover type (MCD12Q1.006)

Class Name	Description	Group
Evergreen Need leaf Forest	Evergreen conifer trees (canopy >2m); Tree Region >60%	1
Evergreen Broadleaf Forest	Evergreen broadleaf and palmate trees (canopy >2m); Tree Region >60%.	2
Deciduous Need leaf Forest	Deciduous needle leaf (larch) trees (canopy >2m); Tree Region >60%.	3
Deciduous Broadleaf Forest	Deciduous broadleaf trees (canopy >2m); Tree Region >60%.	4
Mixed Forest	Neither deciduous nor evergreen (40-60% of each) trees (canopy >2m); Tree Region >60%.	5
Closed Shrublands	Woody perennials (1-2m height) Region >60%	6
Open Shrublands	Woody perennials (1-2m in height) Region 10%-60%	7
Woody Savannas	Tree Region 30%-60% (canopy >2m).	8
Savannas	Tree Region 10%-30% (canopy >2m).	9
Grasslands	Herbaceous annuals (<2m).	10
Permanent Wetlands	Permanently inundated lands with 30-60% water region and >10% vegetated region	11
Croplands	Cultivated cropland (>60%).	12
Urban and Built-up lands	Impervious surface (building materials, asphalt, and vehicles) (>30%)	13
Cropland/Natural Vegetation	Mosaics of small-scale cultivation 40-60% (natural tree, shrub, or herbaceous vegetation)	14
Permanent Snow and Ice	Snow and ice (>60%) for at least ten months.	15
Barren	Non-vegetated barren (>60%) [sand, rock, soil areas with less than 10% vegetation]	16
Water Bodies	Permanent water bodies (>60%)	17
Unclassified	Missing inputs.	0

Table 3-7 Open Land Map soil texture class (USDA system)

Values	Soil type	Group
1	Clay	1
2	Silty clay	2
3	Sandy clay	
4	Clay loam	
5	Silty clay loam	3
6	Sandy clay loam	
7	Loam	
8	Silty loam	
9	Sandy loam	
10	Silt	
11	Loamy sand	4
12	Sand	

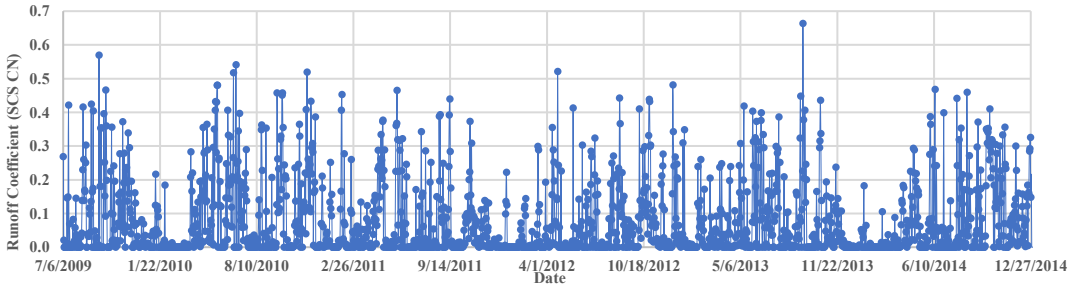


Figure 3-14 Rainfall - Runoff Coefficient 2008 – 2015 (Kelani Basin)

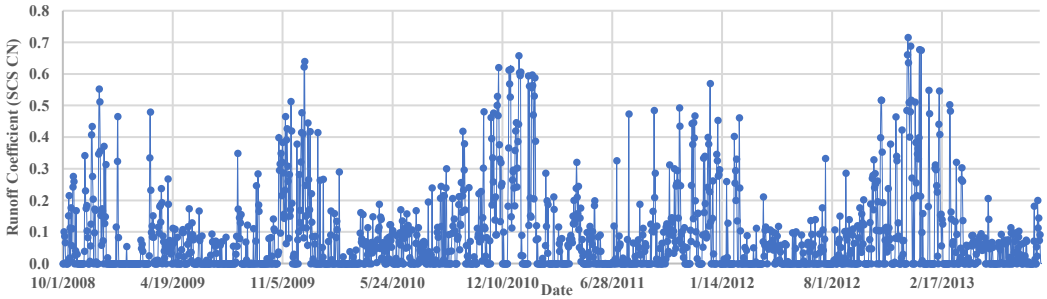


Figure 3-15 Rainfall - Runoff Coefficient 2008 – 2015 (Maduru Oya Basin)

Normalized Difference Vegetation Index (NDVI) & Normalized Difference Water Index (NDWI) - Normalized Difference Built-up Index (NDBI), Modified

Normalized Difference Water Index (MNDWI): The time series were collected using Landsat 7 and 8 top atmosphere reflectance using a specific data processing method to extract the NDVI, NDWI, and MNDWI indices. Landsat 7 collection 2 Tier 1 calibrated top–of–atmosphere (TOA) reflectance is applied to extract with 30 m resolution bands. Table 3.8 represents the band type involved in the computation of indices. Eq 3.9 to Eq 3.12 illustrate the method to calculate the indices. Figure 3-16 to Figure 3-21 illustrate series data.

Table 3-8 Landsat Band type and description

Band Name	Description
B3	Green
B4	Red
B5	Near Infrared (NIR)
B6	Shortwave infrared (SWIR)

NDVI values indicate negative values for the regions with no vegetation cover, like bare land, dessert, glaciers, or water bodies, while it provides positive values for regions with vegetation cover.

$$NDVI = \frac{(NIR-RED)}{(NIR+RED)} \quad \text{Eq 3-9}$$

{NIR and RED: The reflectance in bands 5 and 4 of Landsat 7}

NDWI values represent the water content of plant sensitiveness to the urban land, which is essential to calculate the water content status of plants using remote sensing.

$$NDWI = \frac{(NIR-SWIR)}{(NIR+SWIR)} \quad \text{Eq 3-10}$$

{NIR and SWIR: The reflectance in bands 5 and 6 of Landsat 7}

NDBI values are applied to emphasize built-up regions of a watershed.

$$NDBI = \frac{(SWIR-NIR)}{(NIR+SWIR)} = -NDWI \quad \text{Eq 3-11}$$

MNDWI values express the enhancement of open water features by diminishing built-up features (Ashok et al., 2021).

$$MNDWI = \frac{(Green-SWIR)}{(Green+SWIR)} \quad \text{Eq 3-12}$$

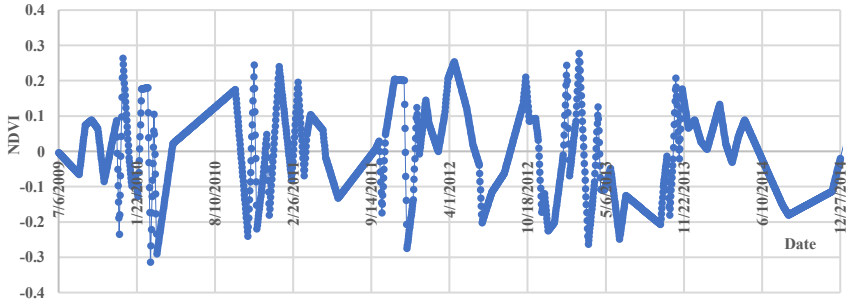


Figure 3-16 NDVI time series data (2008 – 2015) – Kelani

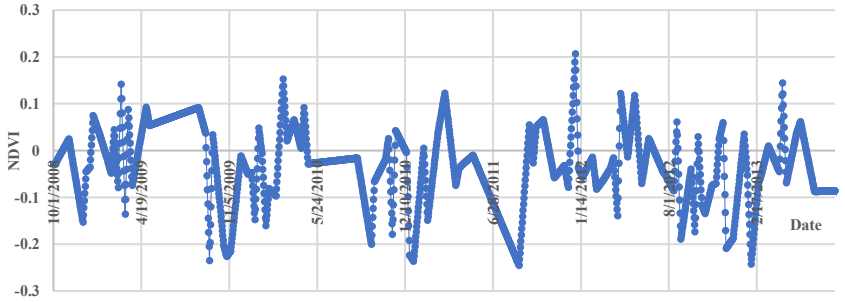


Figure 3-17 NDVI time series data (2008 – 2015) - Maduru

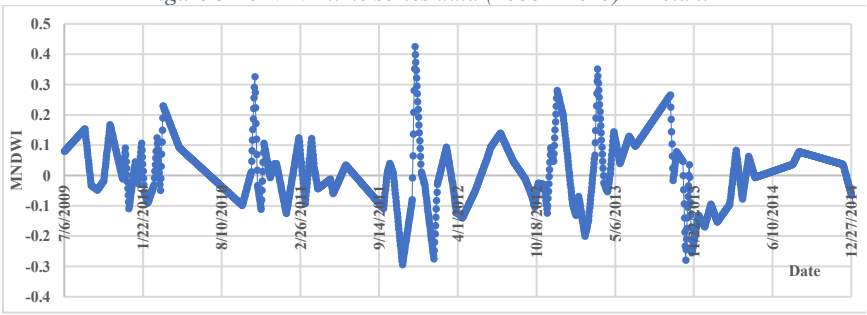


Figure 3-18 MNDWI time series data (2008 – 2015) – Kelani

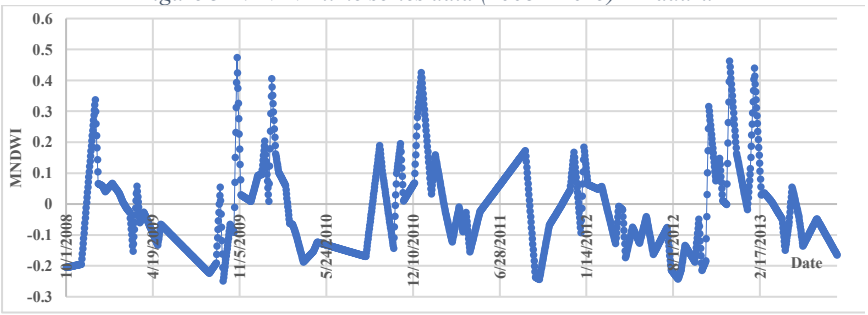


Figure 3-19 MNDWI time series data (2008 – 2015) - Maduru

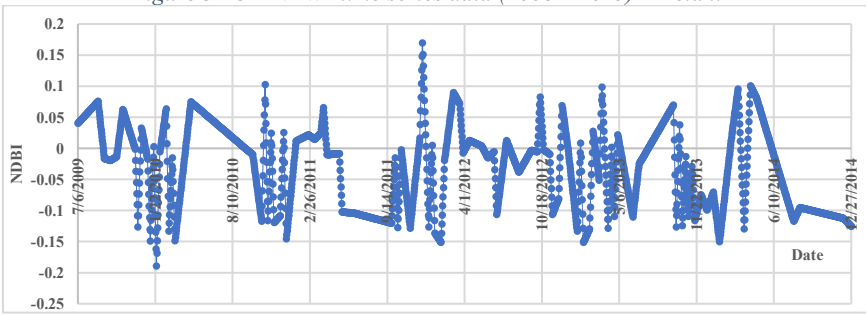


Figure 3-20 NDBI time series data (2008 – 2015) – Kelani

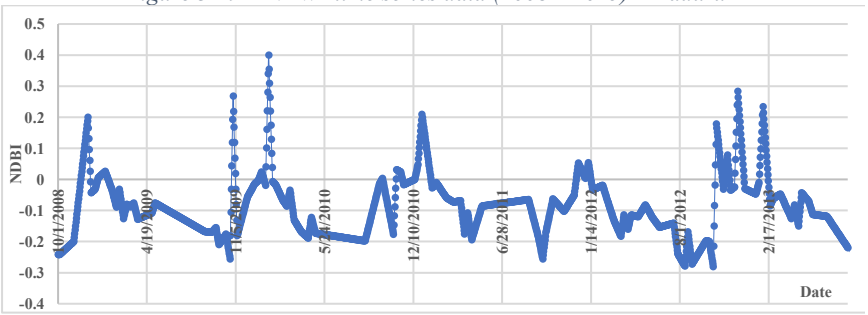


Figure 3-21 NDBI time series data (2008 – 2015) - Maduru

Table 3-9 Google Earth Engine Links

Physiographic Factors	Basin	Google Earth Engine
Runoff Coefficient	Kelani	https://code.earthengine.google.com/e31cc4eec31dc738d2c631fa7c476ec6
	Maduru	https://code.earthengine.google.com/4bcc3b53a0a7b7ba53da728e1105faea
NDVI	Kelani	https://code.earthengine.google.com/2d964ae418fe3e3ff411ba0489e39f39
	Maduru	https://code.earthengine.google.com/d45aabb27a9caf0bfec03af824c4288d
NDWI / NDBI	Kelani	https://code.earthengine.google.com/2766cb636c34d420c9edbc7d3fc28d42
	Maduru	https://code.earthengine.google.com/74730f57e55de9a56204f765e335e488)
MNDWI	Kelani	https://code.earthengine.google.com/12cb7d0d74f2ad7e48f89e8339602cd6
	Maduru	https://code.earthengine.google.com/fd99907482adeb499f8f44b05fbbec23

Table 3-9 represents the Java Script used for data collection.

Date Checking: Physiographic Data

Radar data are calibrated and orthorectified by data preprocessing. Besides, In Sentinel-1 data collection, various instrument configurations, resolutions, and band combinations are involved; the heterogeneity problem affects the data extraction process. Therefore, morphological filters and speckle noise elimination are significant aspects of data calibration. The backscatter coefficient is developed to represent the backscattering area (radar cross-section) per unit of ground area, which is severely influenced by terrain-physical aspects, geometry, and electromagnetic behaviors. Filter preprocessing was employed during the extraction of physiographic data using GEE.

Data Preprocessing using Data Science

The hydrological data filling was employed for data imputation, and data science techniques were applied for the other input data sets. The ‘Not Available (N A)’ locations were delignated and replaced with the appropriate ways by grouping the

column details. As a result, the evaporation data were identified with 'N A' distributions; however, the percentile was considerably low. In addition, inappropriate values such as -0.05/-99.9 were replaced entirely with the 'N A' category to maintain uniformity throughout the data.

Inappropriate data types, such as text instead of numerical types, were converted into float values to encourage data cleaning. By employing data science cleansing, the unknown rows were filled using interpolation techniques based on timely information. Besides, the column data distribution was inspected using descriptive statistical analysis. Box-Cox transformation was incorporated to develop the Gaussian distribution (normal distribution) on the input parameters. Further, normalization was proposed for scaling techniques which control the computational losses.

3.4 Developing Neural Network Forecasting Models

After the Data Processing, the data set was invited as data frame structures to the Jupyter Notebook (or Google Colab). The data were divided into training, validation, and testing sets by 70%, 20%, and 10%, respectively. Moreover, random shuffling was avoided entirely during the dividing process to ensure the chopping of consecutive samples and deliver more realistic results for validation and testing. Scaling, such as normalization, is crucial to avoid the high computational losses in the NN model. The mean and standard deviation of training data sets were used to rescale the data between 0 and 1.

The window generator task was implemented to develop the required data sets for the analysis, which argument was designed to receive the input width (the number of time events should be considered from the previous data set), label width (the number of time events should be predicted in the future), shift (the number of steps to be considered in the sampling process) and label columns (the column that is relevant to the prediction scenario). The argument was proposed with thirty days of previous events to predict thirty days in the future, and the step was taken as one day. In addition, the entire climatic and physiographic data columns were chosen as input to the model to forecast the daily streamflow events. Figure 3-22 illustrates the sliding window sampling.

Then, the split window function was developed to define the input and target(/label) tensors where the outermost index indicates the batch dimension, the middle represents the time or space (width/height), and the innermost indices are the features. The window splits into two sets: a batch of 30-time step 16 features and a batch of 30-time step 1 feature label. The Data frame is converted to a *tf.data.Dataset* of (*input_window, label_window*) pairs, and the window supports iterating training, validation, and testing sets over batches.

Common-sense Baseline Models (BM) are cheap and straightforward machine learning models to ensure the worth of investing further complexity in the model architecture. It was developed to compare with the later, more complicated models. Figure 3-23 illustrates the configuration of BM. However, the effectiveness of the Baseline model declines by increasing the number of predictions. The Linear Model (LM) - the simplest trainable model is proposed with the linear transformation between input and output, where is no interaction between independent prediction steps. Besides, no activation function is considered in the model architecture. Therefore, these are labelled as no-memory models. Figure 3-24 shows the configuration of LM.

3.4.1 Basic Forecasting Models

Artificial Neural Network (ANN): Its architecture is like the linear model but intensely performed more strongly. Single-step models are not capable of understanding the correlation between input features. The model needs access to multiple time steps to address the issues. In addition, the model accepts the multiple time steps as input to deliver a single output, which is executed on the input window of the exact shape of the model. A dense layer with a ReLu function was proposed to build the model. The output layer was proposed as a dense layer with a single unit. Figure 3-25 illustrates the configuration of ANN.

CNN Model: The model architecture is like Dense ANN; here, the dense layer arguments were replaced with *tf.keras.layers.Conv1D*. Reshape argument is not essential for CNN because the model keeps the time axis in the output. A convolution layer with a kernel size of three was proposed with a ReLu activation function. A dense

layer with a ReLu function was attached additionally to the model, and the last layer was proposed as a dense layer. Figure 3-26 illustrates the configuration of CNN.

RNN Model (LSTM and GRU): A *tf.keras.layer.LSTM* was applied to build up the model. Return sequence argument was paid special attention for two main reasons: by addressing with false, the layer returns the output of the final step and offers the model to warm up its internal state before making a prediction, and by addressing true, it produces the output for each input. Therefore, it is instrumental in stacking the RNN layer and simultaneously training the model on multiple time steps. In addition, the architecture was proposed with *return_sequences* to generate the NN. Like the LSTM model, the GRU model was developed using a *tf.keras.layer.GRU.* Figure 3-27 illustrates the configuration of RNN.

Bidirectional LSTM and GRU: A *tf.keras.model.Bidirectional LSTM* was added to prepare the forecasting model where the return sequence argument was fixed as True. By replacing the LSTM with GRU, the Bidirectional GRU was developed on the window to develop the forecasting model. The architecture was proposed with bidirectional LSTM and GRU layers with *return_sequences* to initiate the performance. Figure 3-28 illustrates the configuration of the Bidirectional Model.

NARX Model: A *tf.keras.Model* was used to develop the NARX unit from the basic principles. The model was constructed to consider the present inputs and the previously predicted values. NARX layer, dense layer with Tanh activation and dense layer with linear activation were involved in developing the model architecture. Figure 3-29 illustrates the configuration of NARX.

3.4.2 Hybrid NN Models

CNN-LSTM: Data set was passed through the units; after extracting the sequential features, those features were inserted into the Fully Connected Dense Layers (FCN). A Conv1D layer with a kernel size of five was proposed as the first layer. Then, LSTM layers with *return_sequences* were introduced, dense layers with ReLu activation followed the previous layer, and the dense layer was proposed as the last layer. Figure 3-30 illustrates the configuration of CNN-LSTM.

SBU-LSTM Model: A bidirectional layer with *lstm_cells* with *return_state* was introduced to develop the model architecture. Deep LSTM NN added a fully connected deep neural network to boost the model. According to the preference, the BiLSTM/LSTM was introduced in the model architecture for better accuracy. Figure 3-31 illustrates the configuration of SBU-LSTM.

Dilated Casual CNN Model (DCNN): For time series analysis, the filter was one-dimensional, and an activation function was selected as ReLu to avoid vanishing gradient issues. A dropout factor of 0.2 was introduced to prevent overfitting issues of the model. At last, a dense layer was introduced to the model architecture. Figure 3-32 illustrates the configuration of DCNN.

Graphical Convolution – LSTM (GC-LSTM): Sliding window techniques and the model architecture were adjusted before running the NN. The graph convolution layer was organized to build the adjacency matrix, while LSTM was arranged to understand the input series nature. Both layers were combined to generate the output series. The GC was incorporated into the model architecture without activation function. The LSTM with the ReLu activation function was additionally introduced to the architecture. The last unit was proposed as a dense layer. Figure 3-33 illustrates the configuration of GC-LSTM.

Temporal Convolution Network (TCN): Casual padding and kernel size of two were adapted to the model architecture. The dilation rate began at one and reached 128. A dense layer with a ReLu activation function was introduced with a 0.2 dropout rate to control overfitting. Besides, the Tanh activation was arranged at last. The encoder and decoder module handled the input before sending it through the model architecture. Figure 3-34 illustrates the configuration of TCN.

Table 3-10 Jupyter Notebook and Details

Title	Jupyter Notebook
Kelani Basin	
Climatic Data	https://colab.research.google.com/drive/1x8RH6bsSXdYroHdkHOcWRKS08lhhwozz?usp=sharing
Physiographic Data	https://colab.research.google.com/drive/1IZZm_7n7itJMai1UhDO7OU4trHd87cyI?usp=sharing
Data Cleaning	https://colab.research.google.com/drive/1hxerL4OjxpQiDfdJwa9M-fAFXUcS14Ud?usp=sharing
Maduru Basin	
Climatic Data	https://colab.research.google.com/drive/1SbdVFlR18Te9wg-kKFHuw5yejtos6sOZ?usp=sharing
Physiographic Data	https://colab.research.google.com/drive/1hRoYXK6uZ0ybNcoEBJbWyk0bhxS1KbE_?usp=sharing
Data Cleaning	https://colab.research.google.com/drive/1Vil6uxFh2LeZaxCLG6SmPwdAd1Mc1Rex?usp=sharing
Box-Cox	https://colab.research.google.com/drive/16SNyt4ZaudP_ni7XUVoVTKco3YM70mOh?usp=sharing
Kelani Basin	
NN Models	https://colab.research.google.com/drive/1APN-XaGY7LEppD6HMWX5KzAfzCYly-de?usp=sharing
Sensitivity	https://colab.research.google.com/drive/1kgsoNHA20bYLh4OOrMcmJlayOnsNULmB?usp=sharing
Sensitivity-Parameter	https://colab.research.google.com/drive/11gInbTFZMvLZLUQr8G-787A-8gBqiFlM?usp=sharing
Sensitivity-Time Span	https://colab.research.google.com/drive/1C-6whhdkIVeQNI5WWr5Ie0iPv4ZaU-mY?usp=sharing
Sensitivity-Learning Rate	https://colab.research.google.com/drive/13rlb564q8MWWg4C94Y2AK4AXjz49Lw2r?usp=sharing
Maduru Basin	
NN Models	https://colab.research.google.com/drive/1hJWnCUS7HiJD8eEhMBPGhQD766m0Kq6a?usp=sharing

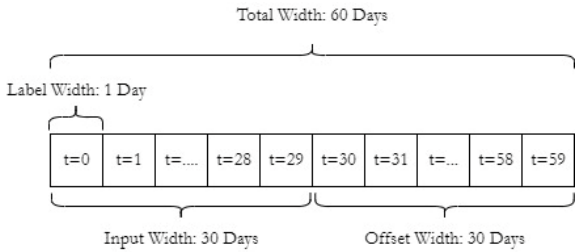


Figure 3-22 Configuration of Window Sampling

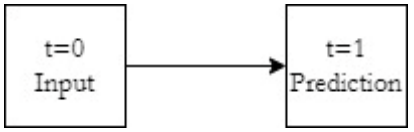


Figure 3-23 Configuration of Baseline Model

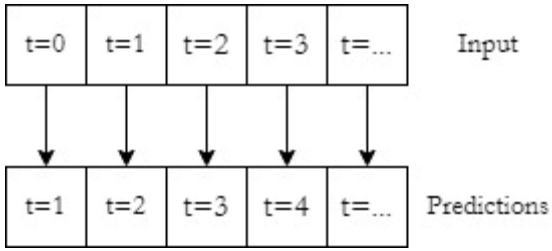


Figure 3-24 Configuration of Linear Model

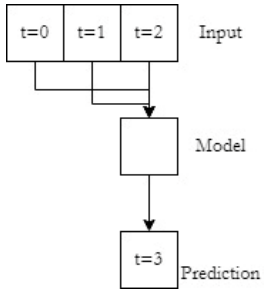


Figure 3-25 Configuration of ANN Model

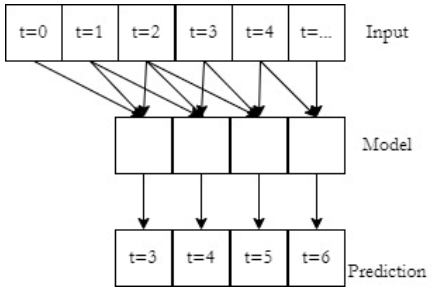


Figure 3-26 Configuration of CNN Model

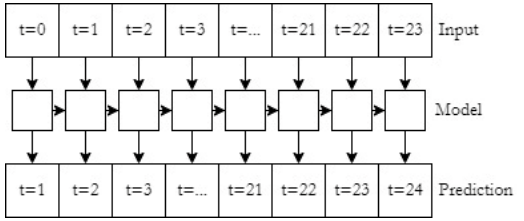


Figure 3-27 Configuration of RNN Model

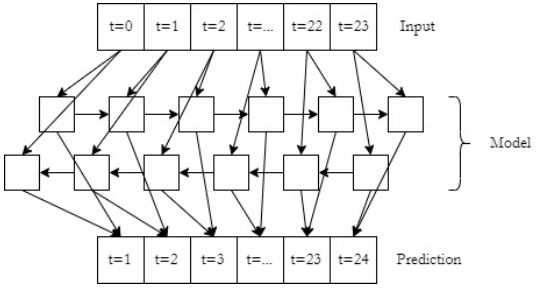


Figure 3-28 Configuration of Bidirectional Model

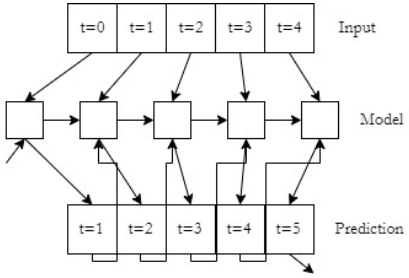


Figure 3-29 Configuration of NARX Model

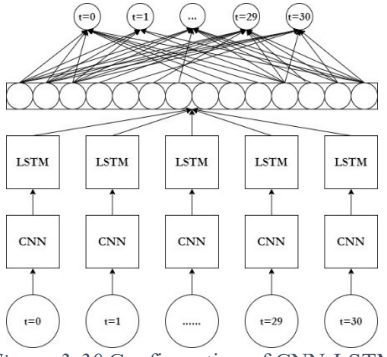


Figure 3-30 Configuration of CNN-LSTM

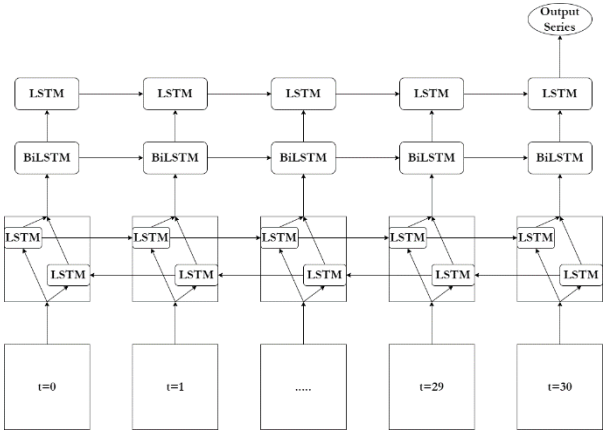


Figure 3-31 Configuration of SBU-LSTM

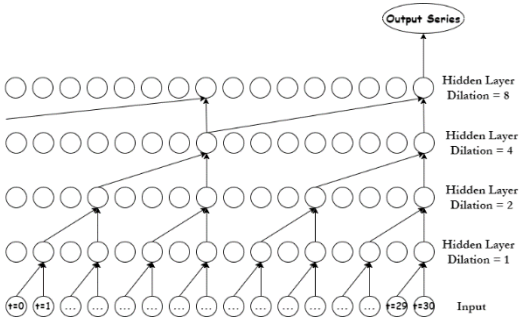


Figure 3-32 Configuration of DCNN

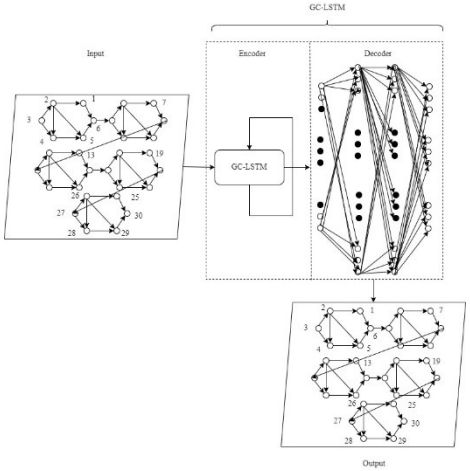


Figure 3-33 Configuration of GC-LSTM

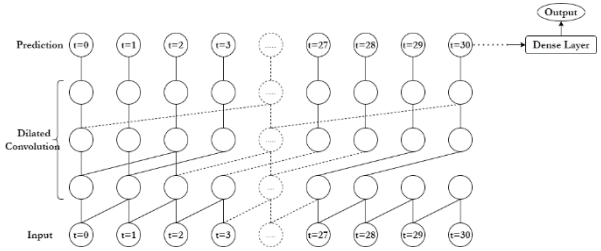


Figure 3-34 Configuration of TCN

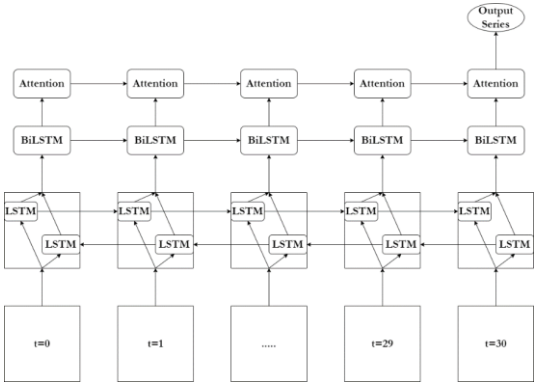


Figure 3-35 Configuration of Att-BiLSTM

Attention-Based Bidirectional LSTM: The data was passed through the encoder (BiLSTM) and decoder (Dense layer) modules. BiLSTM and the attention layer with return sequence, which was generated with SoftMax and Tanh activation functions, deal with generating the model architecture. A 0.5 dropout rate was considered, and the output layer was proposed with the dense module. Figure 3-35 illustrates the configuration of Att-BiLSTM.

All the models were compiled and fitted with Adam optimizer, RMSE loss function, and MAE monitoring metrics. Table 3-10 illustrates the Jupyter Notebook for the studies.

3.5 Selecting the Best NN Models

Analytic Hierarchy Process (AHP) analysis was proposed to identify the weights of decision-making factors such as R^2, RE, R_{FDC}, MAE, and RMSE. First, grid analysis was employed to select the best models where higher rank values were assigned for the worst performance in each criterion. Then, the overall weight for each model was computed by multiplying weights and ranks.

Table 3-11 Importance of Factors

Factors	Reason for Influence on Decision-Making
R^2	It is used to express the fitness of both observed and forecasted values, which is the appropriate statistical tool for the performance. Therefore, it was assigned a higher value.
RE	It measures the scattered deviation of predicted values from the observed values. Therefore, it was assigned the second highest value.
R_{FDC}	This tool expresses the behavioural error of observed and forecasted FDC. FDC greatly explicates the catchment characteristics. Therefore, it was assigned a fair value.
MAE	It was the monitoring metrics of the NN models. Therefore, it was assigned a low value.
RMSE	It was the loss function considered for the model compiling and fitting. Therefore, it was considered the lowest value.

Table 3-10 describes the values of decision-making factors, and Table 3-11 illustrates the AHP analysis. Grid analysis was employed to select the best models where higher rank values were assigned for the worst performance in each criterion. In contrast,

lower rank values were proposed for the best performance. The overall weight for each model was computed by multiplying weights and ranks.

Table 3-12 AHP Analysis and Weights

	R^2	RE	R_{FDC}	MAE	RMSE	Weights
R^2	1	2	3	4	5	0.44
RE	1/2	1	3/2	2	5/2	0.22
R_{FDC}	1/3	2/3	1	4/3	5/3	0.15
MAE	1/4	2/4	3/4	1	5/4	0.11
RMSE	1/5	2/5	3/5	4/5	1	0.09
Total	2.28	4.57	6.85	9.13	11.42	1.00

Grid analysis was performed with the derived weights of decision-making factors. In addition, rank values based on the statistical measure were considered as a, b, c, d, and e in the column of R^2, RE, R--FDC, MAE and RMSE, respectively. Table 3-12 expresses the AHP analysis to quantify the weights. Table 3-13 illustrates the sample calculation for Grid Analysis.

3.6 Uncertainty Analysis

Uncertainty analysis of data-driven forecasting models expresses the reliability of anticipating values based on the input parameters (Shamshirband et al., 2019). The performance is expressed as the performance of the core model. Table 3-14 expresses the uncertainty parameters and their importance.

3.7 Sensitivity Analysis

A sensitivity analysis among artificial neural networks provides an opportunity to assess the importance of impactful factors that contribute to the functionality of neural networks. It can be expressed based on the primary model quality (Song et al., 2020).

The architecture of the models (learning rates, optimizers), input parameters, and lead time play a significant role in a sensitivity analysis. The performance of models was checked with the base model performance as in equation Eq 3-13 (R^2: R-Sq).

Table 3-15 illustrates the sensitivity parameters and their importance.

$$Performance\ Factor = {}^{E_i}/_{E_b} \ldots\ldots\ldots\ldots \qquad \textbf{Eq 3-13}$$

Table 3-13 Grid Analysis (Sample Sheet)

	R^2	RE	R_{FDC}	MAE	RMSE	Weight * Rank
Weights	0.44	0.22	0.15	0.11	0.09	1
BL						
LM						
ANN	a	b	c	d	e	0.44 a + 0.22 b + 0.15 c + 0.11 d + 0.09 e
CNN						
LSTM						
GRU						
NARX						
Bi-LSTM						
Bi-GRU						
CNN-LSTM						
SBU-LSTM						
DCNN						
GC-LSTM						
TCN						
Att-BiLSTM						

Table 3- 14 Uncertainty Parameters and Description

Parameters	Characteristics	Description
Input Handling	Without Box-Cox Transformation	It determines the influences of extremities location and computational losses.
	Without Box-Cox Transformation and Normalization	
Input Distribution	Rainfall only	It illustrates the importance of input parameters considered for the study.
	Rainfall & NDVI	
	Rainfall & Evaporation	
	Rainfall & Relative Humidity	

Table 3-15 Sensitivity Parameters and Description

Parameters	Characteristics	Description
Learning Rate	0.001	The learning rate influences the time length of convergence. For instance, a high rate contributes to fast convergence, while a low rate provides slow convergence.
	0.1	
	0.01	
	0.0001	
Optimizers	Adam	It combines the momentum and RMSProp, which keeps track of an exponentially decaying average of past squared gradients.
	Nadam	Adam optimization and Nesterove trick produce this optimizer. Thus, it is faster converging than Adam.
	SGD	It is a momentum optimizer that considers the previous gradients at each iteration.
	RMSProp	It fixes the issue related to fast convergence without reaching a global optimum.
Input	Climatic & Physiographic	Dimensionality reduction is the way to control the complexity of forecasting.
	Climatic only	
	Physiographic only	
Lead Time Span	30 Days	Lead time and span length are significant in assessing the model performance.
	15 Days	
	45 Days	

3.8 Evaluating the NN Model Performance (Wet Zone) on Dry Zone Basin

Reusing the pre-trained layer is a practical way to avoid rebuilding the NN models from scratch, which can be proceeded with transfer learning techniques. Generally, it offers favorable results for the inputs with similar features to core models that should express an accuracy greater than 90%. The output layer of the existing model must be replaced with a new layer suitable for a new task. An exact number of reusable layers must be determined based on the freezing and unfreezing layers, which change the state of weights (Géron, 2019). Pre-training models were developed based on the models identified with R^2 of more than 90%. The models fixed the output layer as a dense layer with one unit, and the uppermost layers were locked to control gradient descent effects. Those four statistical tools were employed to explain the performance of the newly developed models. Figure 3-36 illustrates the model architecture of the pre-training model.

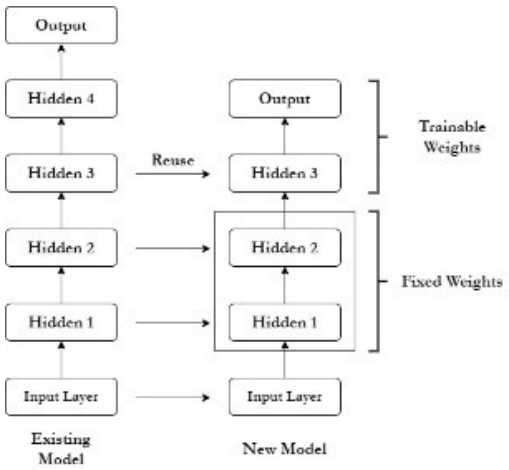


Figure 3-36 Skeleton of Transfer Learning

CHAPTER 4: RESULTS AND DISCUSSION

The results of model performance, the best model selection, uncertainty analysis, sensitivity analysis, and comparison with existing models are discussed in this section. Finally, the hydrographs and FDCs were plotted for 2015 to visually represent the actual and forecasted values from Figure 4-6 to Figure 4-31.

4.1 Performance of NN Models

RMSE and MAE measured the performance of NN models; most significantly, these two were the controlling parameters of NN models. TCN and GC-LSTM expressed the worst performance in forecasting; however, bidirectional models illustrated outstanding performance. Figure 4-1 and Figure 4-2 illustrate the performance.

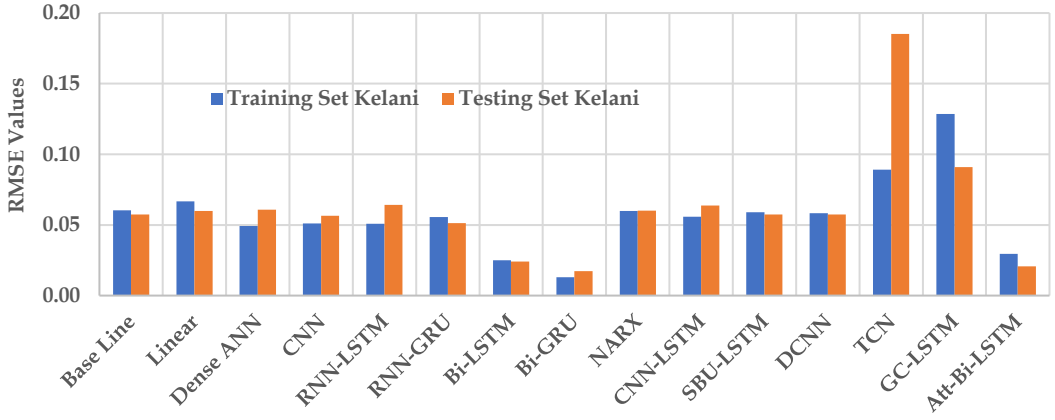


Figure 4-1 RMSE Values - NN Models

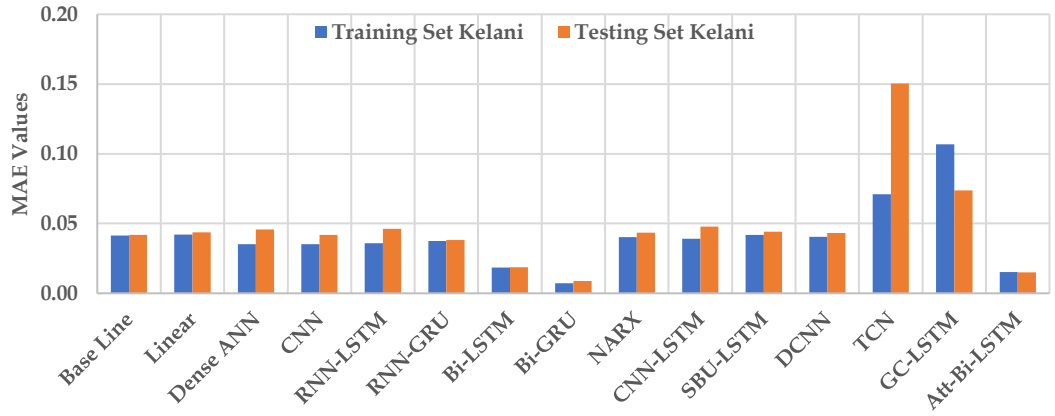


Figure 4-2 MAE Values - NN Models

The R^2 and FDC-Q values were computed to quantify the model performance. Bi-GRU, Att-BiLSTM, and Bi-GRU expressed 0.98, 0.97, and 0.95, respectively. In addition, Bi-GRU and Att-BiLSTM forecasted lower and higher values with similar accuracy; nevertheless, Bi-LSTM lacked to predict the higher values than lower values. The FDC-Q values were observed as 0.99 for all three models. Figure 4-3 and Figure 4-4 illustrate the R^2 and FDC-Q values, respectively.

RNN-GRU, CNN, and RNN-LSTM expressed 0.81, 0.79, and 0.78, respectively. Both RNN-GRU and CNN failed to predict the higher values than the lower values. However, RNN-LSTM forecasted both higher and lower values with adequate accuracy. The FDC-Q values were 1, 0.98, and 0.98 for RNN-GRU, CNN, and RNN-LSTM, respectively.

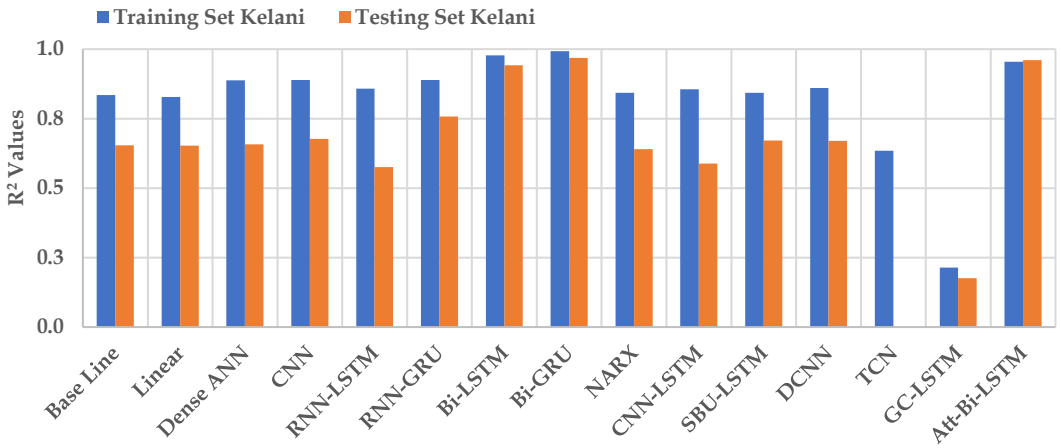


Figure 4-3 R^2 Values - NN Models

ANN and NARX indicated 0.78, while SBU-LSTM expressed 0.77 for R^2. ANN imperfectly forecasted the higher streamflow. NARX correctly predicted the lower values; nevertheless, the prediction quality seems acceptable for the higher values. SBU-LSTM was obtained with an irregular pattern for predictive capability on lower and higher values; nevertheless, the deviation was low. The FDC-Q values revealed 0.99, 0.97, and 1 for ANN, NARX, and SBU-LSTM, respectively.

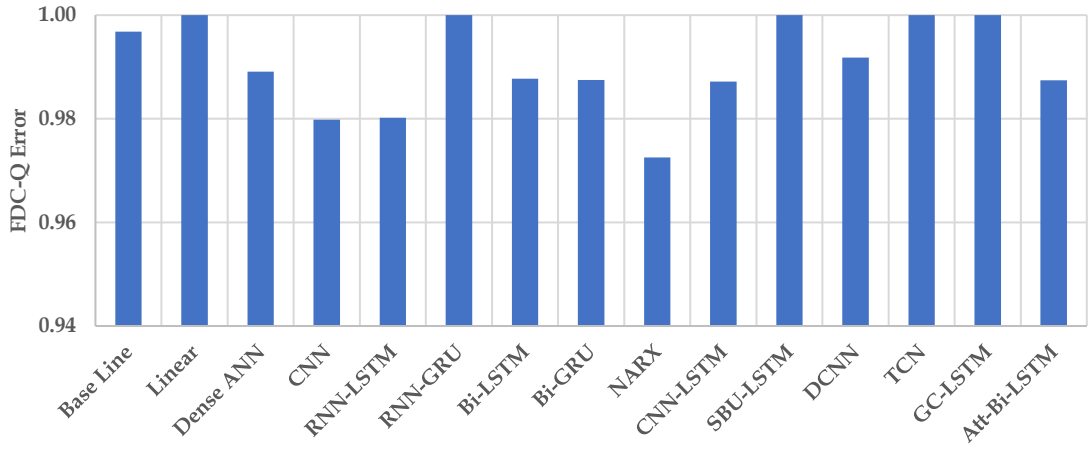


Figure 4-4 FDC-Q Values - NN Models

DCNN, CNN-LSTM, TCN, and GC-LSTM explicated 0.77, 0.75, 0.53, and 0 for R^2, respectively. According to the FDCs, DCNN excellently predicted both lower and higher values. CNN-LSTM almost perfectly predicted the lower and higher values; nevertheless, it could not forecast the intermediate values. TCN expressed the worst performance on predicting higher and lower values. In addition, GC-LSTM expressed repulsive performance compared to other models in forecasting streamflow values. DCNN and CNN-LSTM were identified with 0.99 for FDC-Q, while TCN and GC-LSTM expounded 1.

4.2 Selection of the Best Models

Bi-GRU, Bi-LSTM, Att-Bi-LSTM, RNN-GRU, ANN, CNN, DCNN, and SBU-LSTM were explicated excellent performance than both BM and LM models. On the other hand, RNN-LSTM, CNN-LSTM, and NARX expounded an intermediate level of performance, while TCN and GC-LSTM were appalling poor performance for forecasting. The deviation between forecasted and actual values was lower for the models: Bi-LSTM, Bi-GRU and Att-Bi-LSTM; nevertheless, the FDC-Q values, which elaborate the behavioral errors, revealed that TCN and GC-LSTM were excellent models. On the other hand, RNN-GRU and SBU-LSTM explicated the best results for FDC-Q and moderate values for RM. Figure 4-5 illustrates the RM values of NN Models.

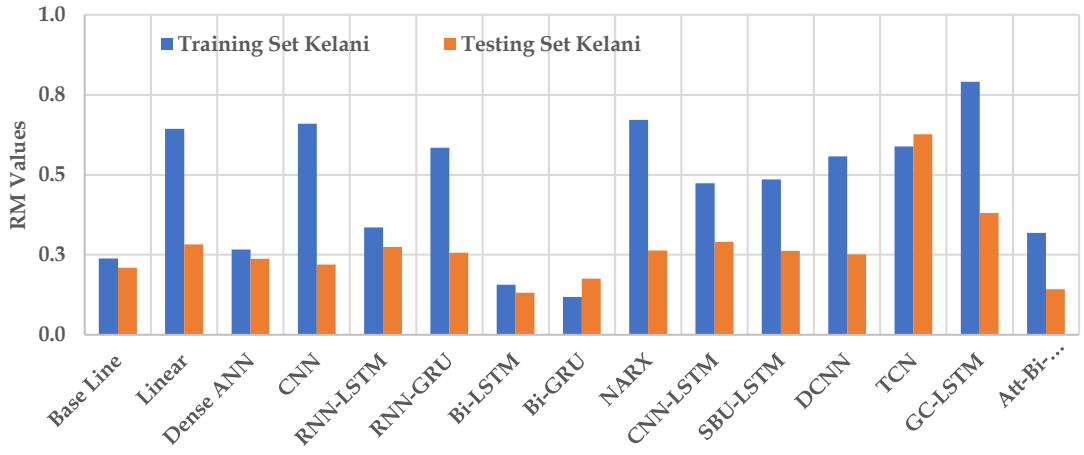


Figure 4-5 RM Values - NN Models

Except for GC-LSTM and TCN, other models delivered an excellent performance for the loss function and monitoring metrics which directly represents the performance of NN models. Overall, Bi-GRU, Bi-LSTM, and Att-Bi-LSTM were the top best models, while GC-LSTM and TCN were identified as the low worst models. RNN-LSTM, CNN-LSTM, and NARX moderately performed according to the grid analysis. On the other hand, other models expressed excellent behaviour for forecasting. Table 4-1 illustrates the grid analysis and the ranks of the models.

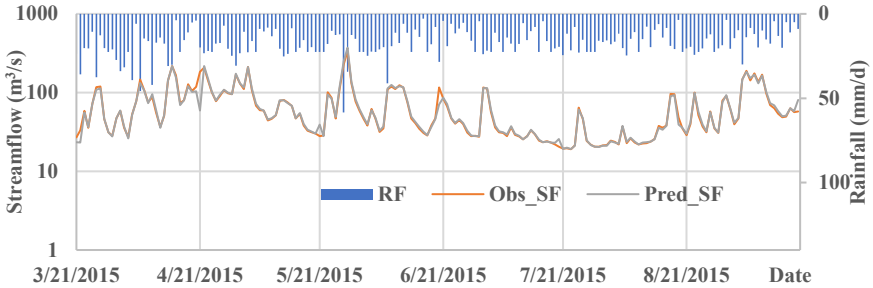


Figure 4-6 Hydrograph for Kelani (Bi-GRU)

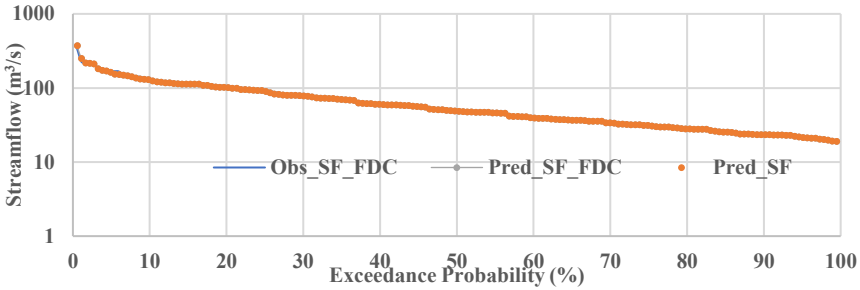


Figure 4-7 FDC for Kelani (Bi-GRU)

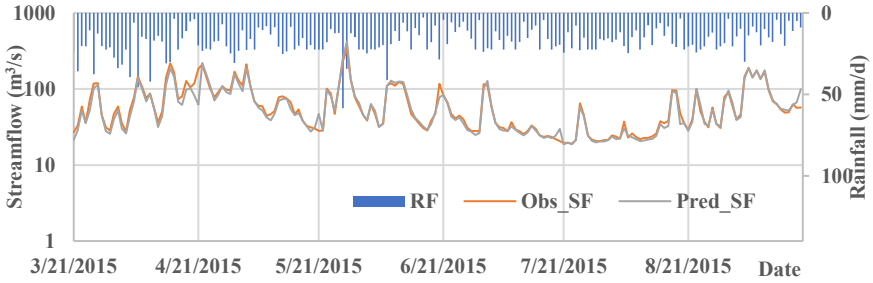


Figure 4-8 Hydrograph for Kelani (Att-Bi-LSTM)

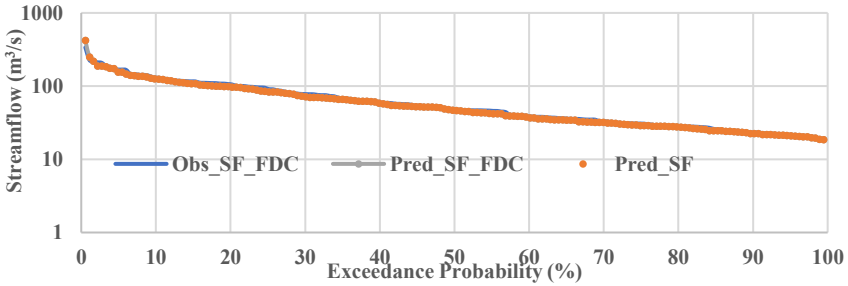


Figure 4-9 FDC for Kelani (Att-Bi-LSTM)

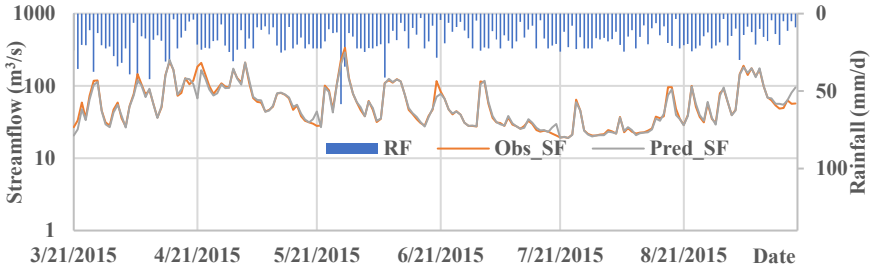


Figure 4-10 Hydrograph for Kelani (Bi-LSTM)

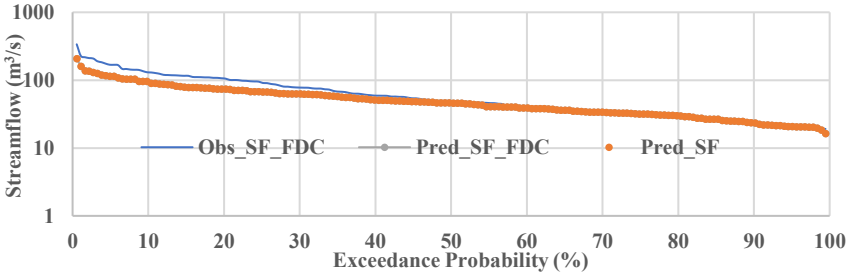


Figure 4-11 FDC for Kelani (Bi-LSTM)

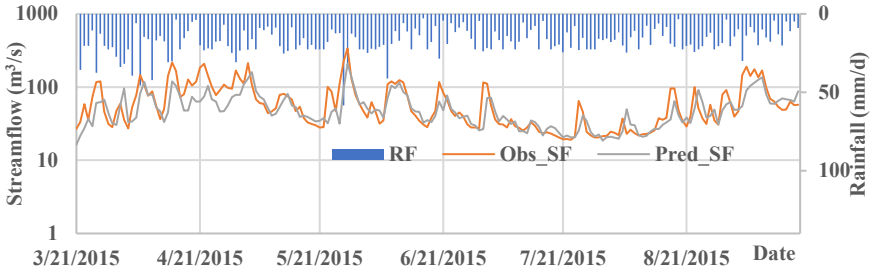


Figure 4-12 Hydrograph for Kelani (RNN-GRU)

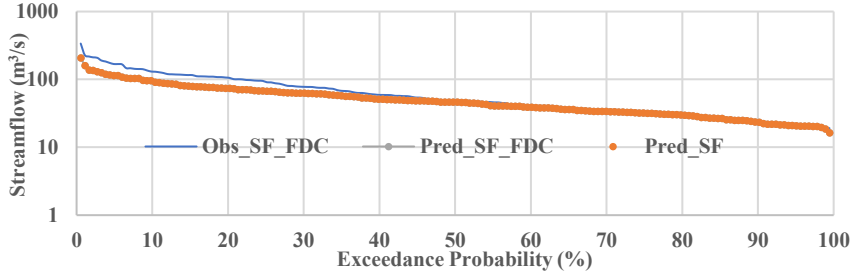


Figure 4-13 FDC for Kelani (RNN-GRU)

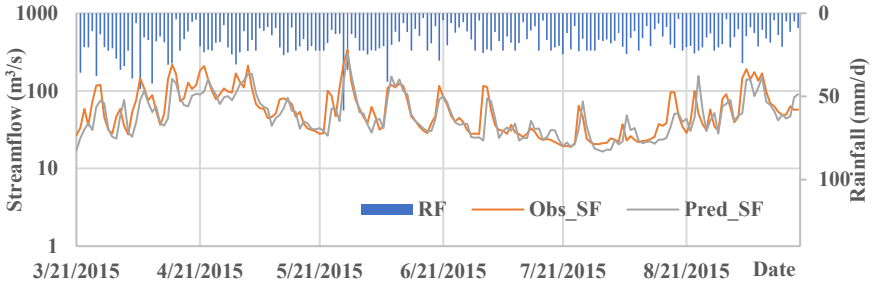


Figure 4-14 Hydrograph for Kelani (CNN)

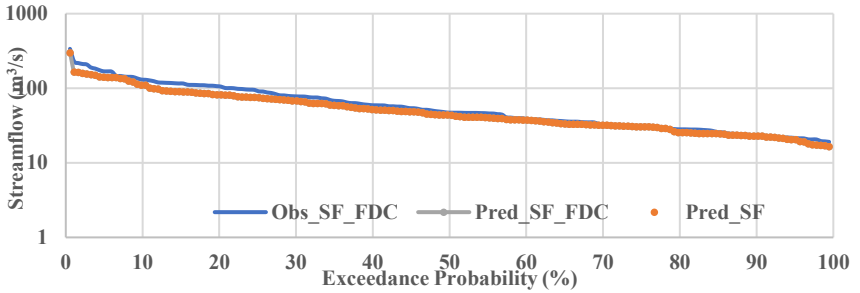


Figure 4-15 FDC for Kelani (CNN)

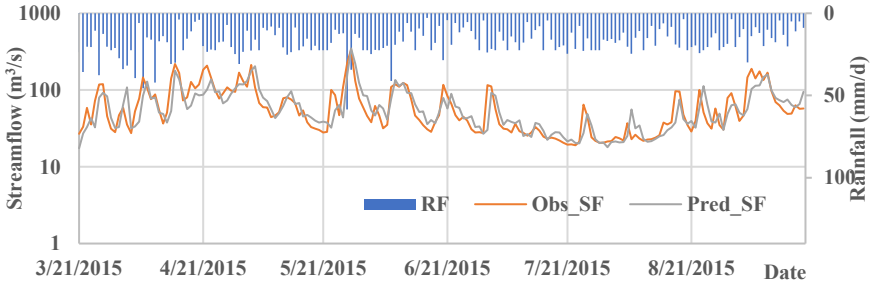


Figure 4-16 Hydrograph for Kelani (RNN-LSTM)

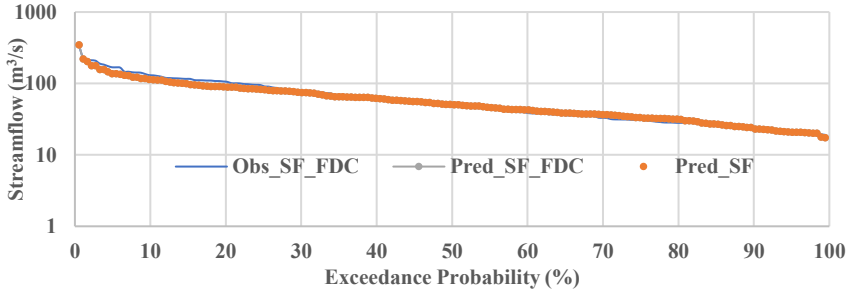


Figure 4-17 FDC for Kelani (RNN-LSTM)

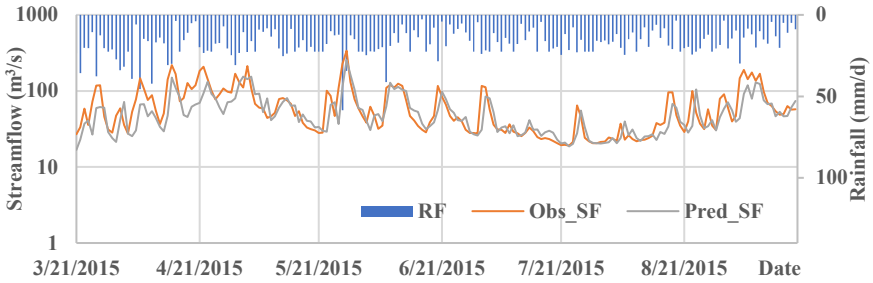


Figure 4-18 Hydrograph for Kelani (ANN)

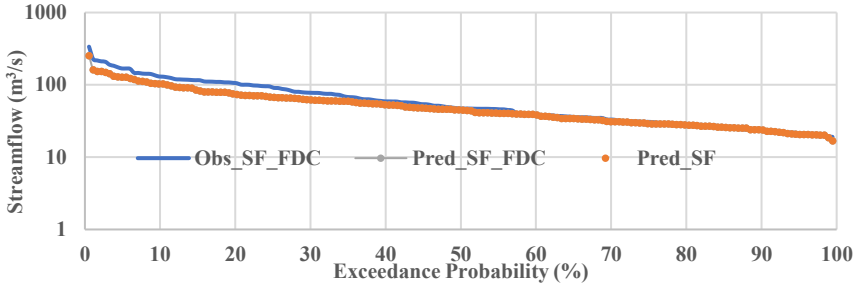


Figure 4-19 FDC for Kelani (ANN)

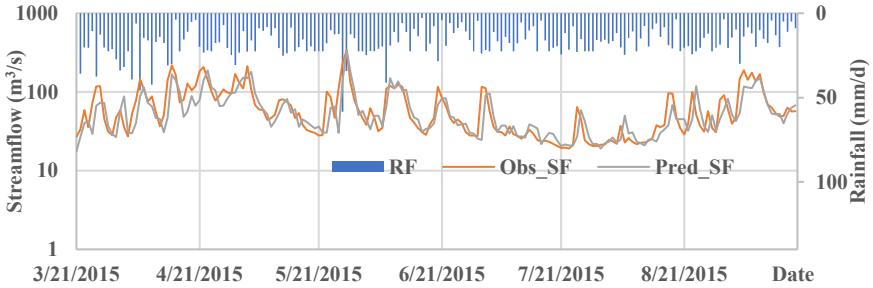


Figure 4-20 Hydrograph for Kelani (NARX)

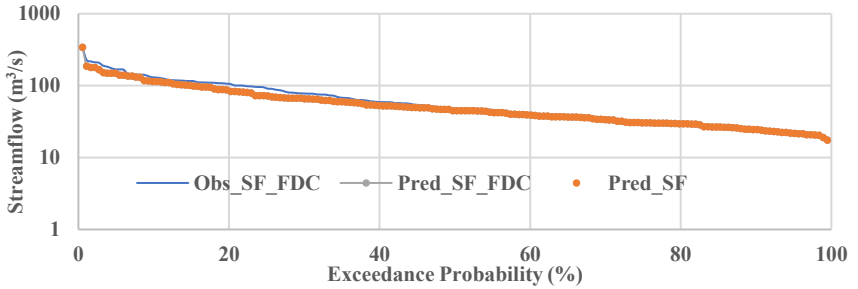


Figure 4-21 FDC for Kelani (NARX)

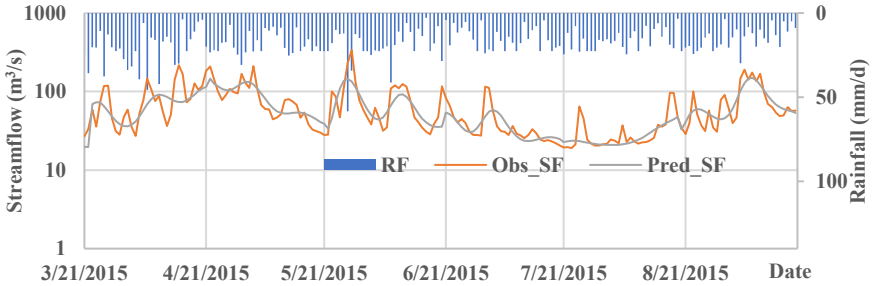


Figure 4-22 Hydrograph for Kelani (SBU-LSTM)

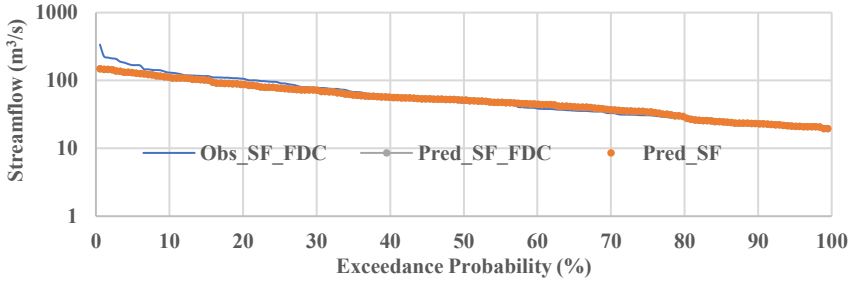


Figure 4-23 FDC for Kelani (SBU-LSTM)

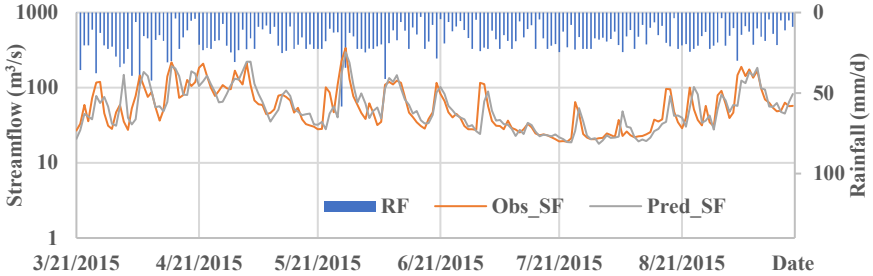


Figure 4-24 Hydrograph for Kelani (DCNN)

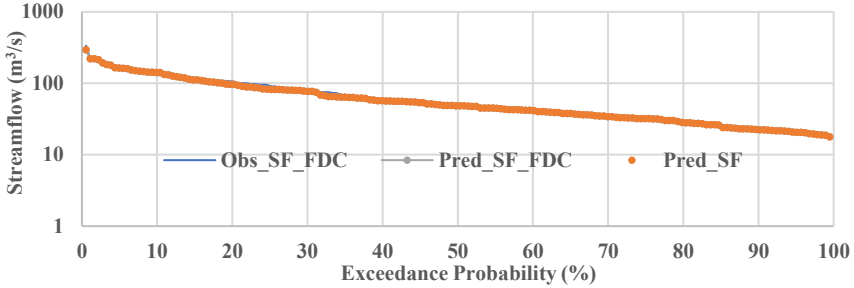


Figure 4-25 FDC for Kelani (DCNN)

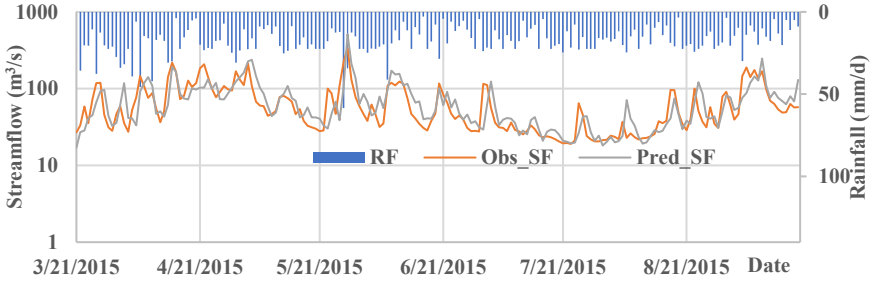


Figure 4-26 Hydrograph for Kelani (CNN-LSTM)

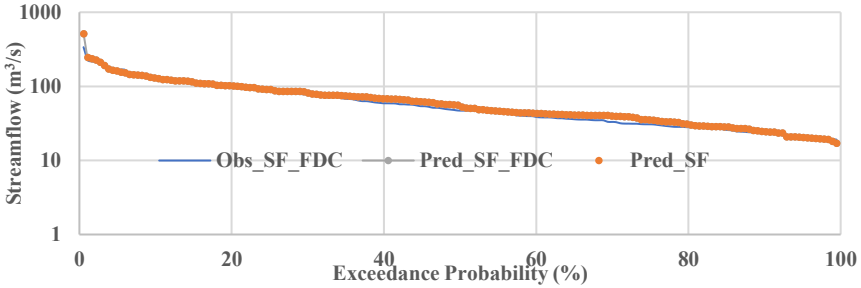


Figure 4-27 FDC for Kelani (CNN-LSTM)

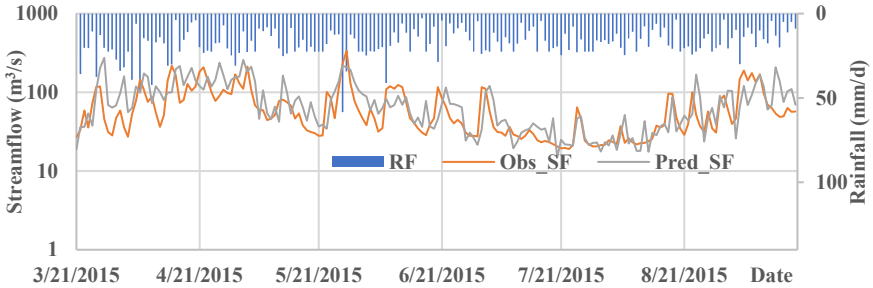


Figure 4-28 Hydrograph for Kelani (TCN)

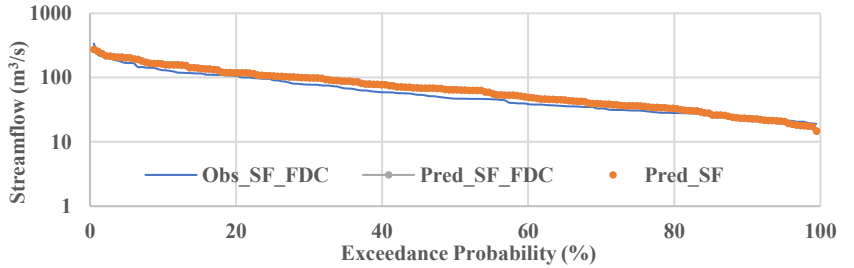


Figure 4-29 FDC for Kelani (TCN)

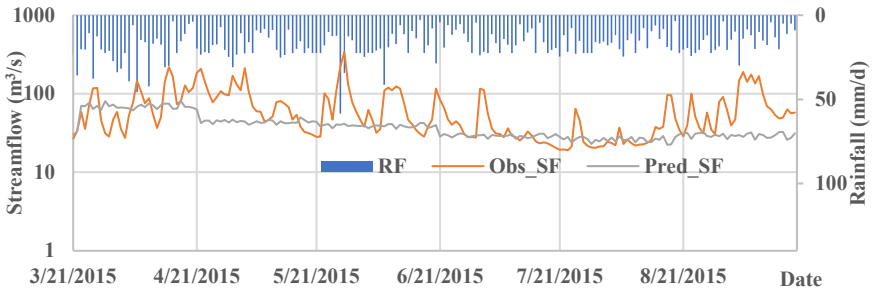


Figure 4-30 Hydrograph for Kelani (GC-LSTM)

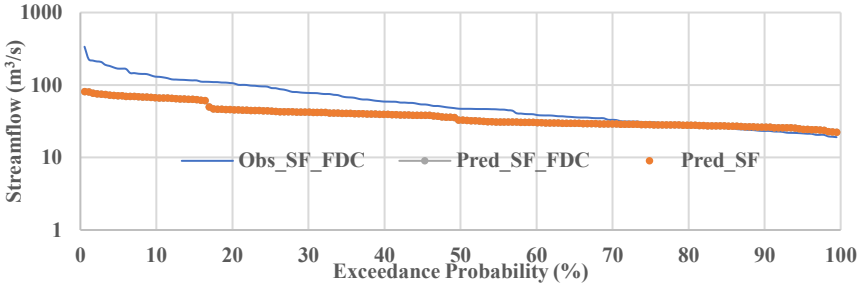


Figure 4-31 FDC for Kelani (GC-LSTM)

Table 4-1 Grid Analysis for Uncertainty Analysis

Weights	0.4		0.2		0.1		0.1		0.1		Weighted Rank	Ranking the Models
NN	R^2	Ranks	RM	Ranks	FDC-Q	Ranks	MAE	Ranks	RMSE	Ranks		
BL	0.74	10.5	0.22	3.5	1.00	6.0	0.04	8.0	0.06	9.5	7.9	9.0
LM	0.74	11.5	0.46	12.0	1.00	1.0	0.04	11.0	0.06	11.0	10.0	10.0
ANN	0.77	7.0	0.25	5.0	0.99	8.0	0.04	7.5	0.05	7.5	6.8	5.0
CNN	0.78	4.5	0.44	9.0	0.98	14.0	0.04	4.5	0.05	5.5	7.0	6.0
LSTM	0.72	10.5	0.30	8.5	0.98	13.0	0.04	9.0	0.06	9.0	10.1	11.0
GRU	0.82	4.5	0.42	9.0	1.00	1.0	0.04	5.5	0.05	5.5	5.2	4.0
Bi-LSTM	0.96	2.5	0.14	1.5	0.99	9.0	0.02	3.0	0.02	2.5	3.3	2.0
Bi-GRU	0.98	1.0	0.15	2.0	0.99	10.0	0.01	1.0	0.02	1.0	2.5	1.0
NARX	0.74	10.5	0.47	12.0	0.97	15.0	0.04	8.5	0.06	10.5	11.3	13.0
CNN-LSTM	0.72	10.5	0.38	10.0	0.99	12.0	0.04	10.5	0.06	10.0	10.6	12.0
SBU-LSTM	0.76	8.5	0.37	8.5	1.00	1.0	0.04	11.0	0.06	8.0	7.6	8.0
DCNN	0.77	7.0	0.40	8.0	0.99	7.0	0.04	8.5	0.06	8.0	7.5	7.0
TCN	0.32	14.5	0.61	13.0	1.00	1.0	0.11	14.5	0.14	14.5	12.2	14.0
GC-LSTM	0.19	14.5	0.59	14.5	1.00	1.0	0.09	14.5	0.11	14.5	12.5	15.0
Att-BiLSTM	0.96	2.5	0.23	3.5	0.99	11.0	0.02	2.0	0.03	2.5	3.9	3.0
***Red: Worst models *Green: Good Models *Blue: Excellent models *Yellow: Indicators**												

4.3 Uncertainty Analysis

None of the models explicated excellent performance than the models with box-cox transformation; however, SBU-LSTM and GC-LSTM expounded underperformance without box-cox transformation. Normalization severely influenced the model performance based on the analysis; except for CNN-LSTM, the performance of hybrid models significantly deteriorated without scaling the data. The modelling performed with rainfall data improved the performance of GC-LSTM; nevertheless, except for TCN, other models explicated similar performance to the core models. TCN and GC-LSTM expounded the worst performance for the data set with rainfall and NDVI. TCN expressed worst performance for the data combination considered for the analysis than the core models. Figure 4-32 expresses the performance of NN models based on uncertainty analysis.

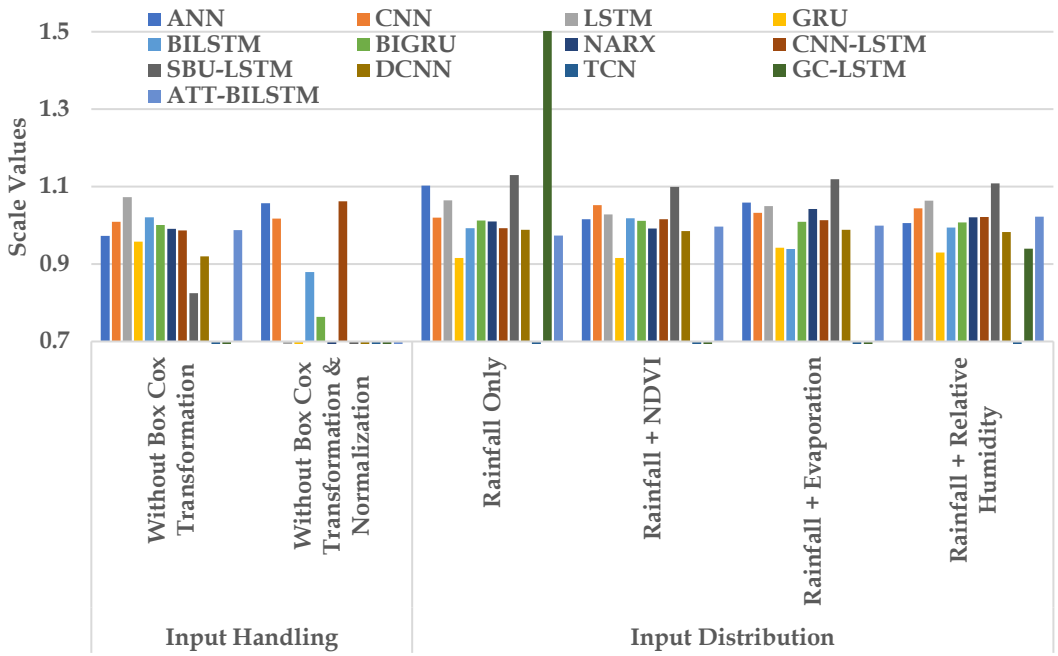


Figure 4-32 Uncertainty Analysis and the Scaled Values

4.4 Sensitivity Analysis

The SBU-LSTM expounded outstanding performance and was slightly higher for both semi-monthly and monthly spans, which was better in weekly spans than other models. In addition, GC-LSTM performed exceptionally for the seme-monthly span. LSTM

derived great results for both input parameters, such as climatic and physiographic data, where LSTM was one of the best models developed for climatic data sets. In addition, no models were identified with considerably more excellent performance than the core models. LSTM was observed with perfect results for both learning rates: 0.01 and 0.0001, which were the most incredible models for a 0.0001 learning rate. Simultaneously, SBU-LSTM was identified with noticeable results for a 0.01 learning rate. Nadam entirely improved the model performance compared to SGD, while RMSProp considerably supported to gain more performance than RNN-based models and TCN. Unlike the optimizers, other sensitivity parameters explicated inconclusive results on model performance. Figure 4-33 graphically expresses the sensitivity analysis.

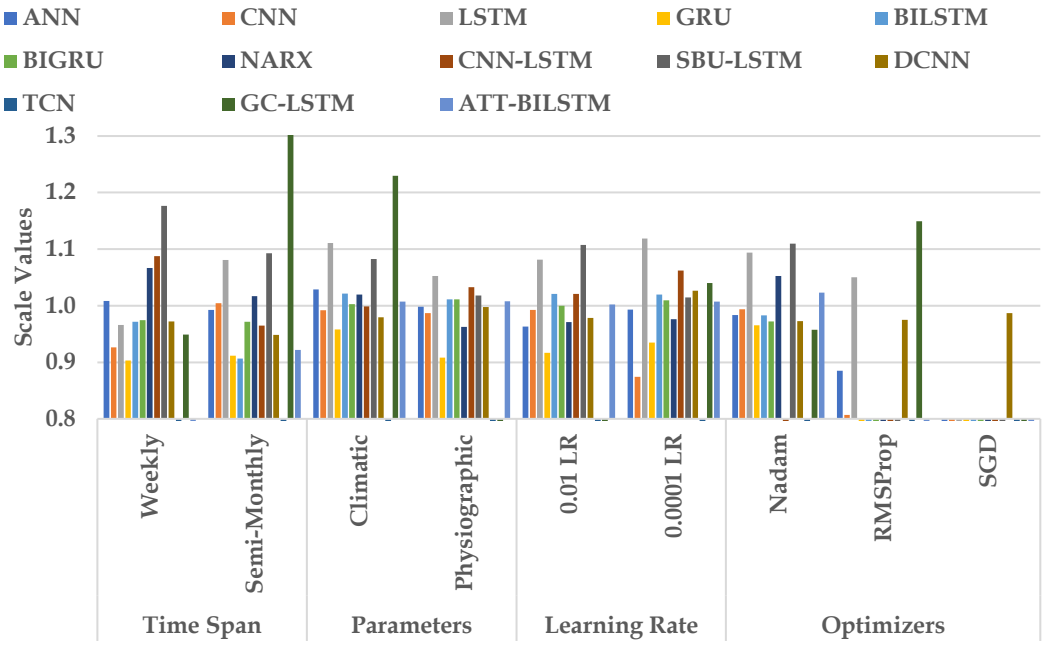


Figure 4-33 Sensitivity Analysis and the Scaled Values

4.5 Evaluating the Wet Zone Models on Dry Zone

The Bi-GRU, Bi-LSTM, and Att-Bi-LSTM were developed for the dry zone basin, and models were identified with greater ability in the wet zone basin. Newly prepared models represented quite similar performance to the core models. The R^2 values expressed 0.97, 0.98, and 0.94 for Bi-GRU, Bi-LSTM, and Att-Bi-LSTM,

respectively. Figure 4-34 to Figure 4-36 illustrates the hydrographs and FDCs of the Maduru River Basin for 2010/2011.

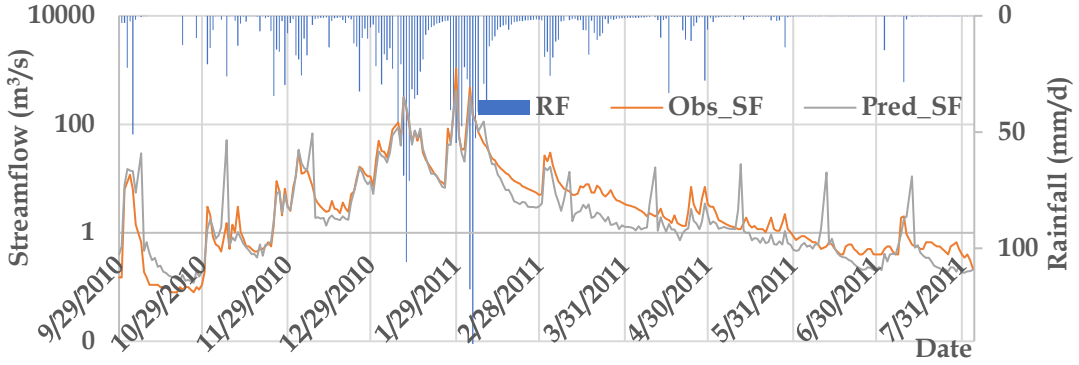


Figure 4-34 Hydrograph for Att-Bi-LSTM

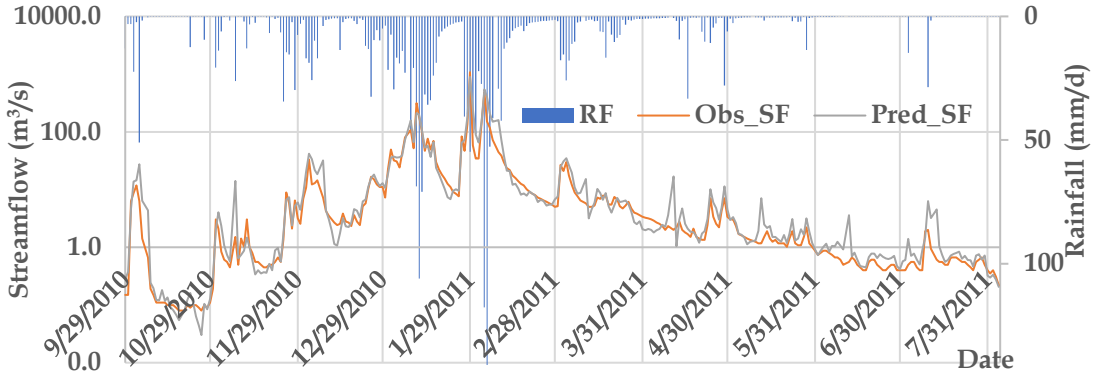


Figure 4-35 Hydrograph for Bi-LSTM

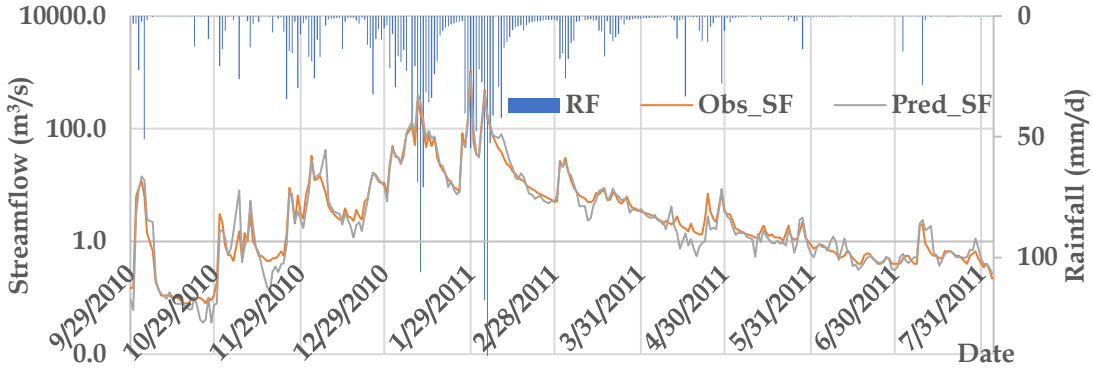


Figure 4-36 Hydrograph for Bi-GRU

4.6 Comparing the Present Study Model Performance with Existing Models

The present study was compared with the models gathered from the most recent literature; the NSE values were also derived from comparing with the R^2. However, the forecasting horizon and the input parameters varied from the models available in

the literature and the present study; this is a significant limitation. Figure 4-37 illustrates the comparison of R^2 values. Besides, Table 4-2 shows the models available in the literature.

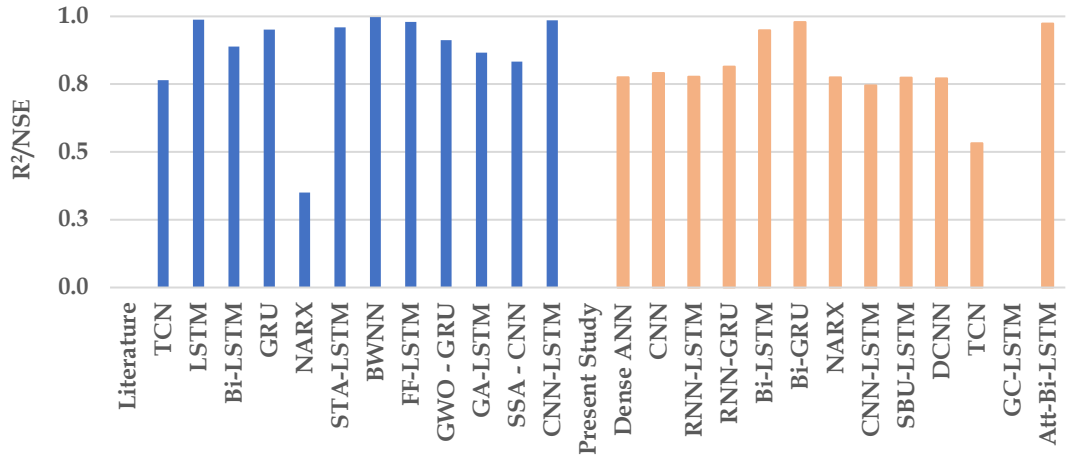


Figure 4-37 Comparison of NN Models (Blue: Models available in the literature, Orange: Models in the study)

Bi-LSTM, Bi-GRU, and Att-Bi-LSTM expressed a competitive manner with the models available in the literature, where the forecasting horizon of the available models differed from the present study. RNN-GRU explicated a performance like GA-LSTM, SSA-CNN, and Bi-LSTM available in the literature. Dense ANN, CNN, RNN-LSTM, NARX, CNN-LSTM, SBU-LSTM, and DCNN expressed a similar performance to the TCN model available in the literature. However, the TCN model in the present study could have performed better than other models. In contrast, the NARX model of the present study was well-performed than the NARX available in the literature. GC-LSTM expounded worse performance than both models of present and previous studies.

Table 4-2 NN Models in the Literature

NN Models	Linearity	Characteristics of the Analysis	Research Paper
TCN	0.76	Forecasting Horizon: 12 hrs Data (Hourly): Rainfall (19), Discharge, NDVI Input Series: 1971 – 2013	(Xu et al., 2021)
LSTM	0.99	Forecasting Horizon: 2 Days Data (Daily): Streamflow Input Series: 1995 – 2014	(Le et al., 2021)

NN Models	Linearity	Characteristics of the Analysis	Research Paper
Bi-LSTM	0.89	Forecasting Horizon: 30 min Data (Hourly): Water Level, Rainfall Input Series: 2013 - 2020	(Won et al., 2022)
GRU	0.95	Forecasting Horizon: 1 Day Data (5 – 10 Days): Daily Runoff, Rainfall Input Series: 2007 - 2014	(Wang et al., 2020)
NARX	0.35	Forecasting Horizon: 2 Months Data (Monthly): River Flow, Weighted Rainfall, Average Evaporation Input Series: 1998 - 2016	(Hayder et al., 2022)
Stacked LSTM (STA-LSTM)	0.96	Forecasting Horizon: 6 hrs Data (12 hrs): Hourly Flowrate, Rainfall Input Series: 1981 - 2007	(Ding et al., 2020)
Biased Wavelet Neural Network (BWNN)	1.00	Forecasting Horizon: 3 Days Data (7 Days): Daily Streamflow Input Series: 1931 – 2015	(Saraiva et al., 2021)
Feed Forward Neural Network (FF-LSTM)	0.98	Forecasting Horizon: 1 hr Data: (1-10 hrs) Daily Streamflow, (1-7 hrs) Daily Rainfall Input Series: 1980 - 2016	(Lin et al., 2021)
Grey Wolf Algorithm Based GRU (GWO-GRU)	0.91	Forecasting Horizon: 1 Day Data: Daily Flowrate Input Series: 2000 - 2009	(Kilinc & Yurtsever, 2022)
Genetic Algorithm (GA-LSTM)	0.87	Forecasting Horizon: 1 Day Data: Daily Flowrate Input Series: 2000 - 2009	(Kilinc & Haznedar, 2022)
Semantic Self-Attention CNN (SSA-CNN)	0.83	Forecasting Horizon: 1 Day Data: Antecedent Streamflow, Precipitation, Relative Humidity Input Series: 1996 - 2016	(Apaydin et al., 2021)
CNN-LSTM	0.99	Forecasting Horizon: 1 Month Data (Hourly): Streamflow Input Series: 1996 - 2016	(Ghimire et al., 2021)

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

The conclusions and limitations of the present study and recommendations for future studies are discussed in this section.

5.1 Conclusions

The Box-Cox transformation was applied to control the anomalies of the input data sets, which generated Gaussian distribution to handle the extremities. Normalization was employed in the final data set to minimize the computational losses. In addition, the sliding window concept was adapted to slice the time series data into the required horizons. Except for TCN and GC-LSTM, the other eleven NN models expressed excellent performance for 30-day forecasting; mainly, all three bidirectional models, including the attention-based model, delivered an extraordinary performance to forecast streamflow values. On the other hand, Feedforward models such as ANN, SBU-LSTM, and NARX explicated underperformance more than bidirectional and other RNN models. CNN models such as standard CNN and DCNN expounded better performance than the LSTM model; nevertheless, those performances were almost equal to the SBU-LSTM model. CNN-LSTM was the only hybrid model that expressed a performance like the LSTM model; however, the performance was observed as poor compared to other RNN variants. Grid analysis perfectly notified the bidirectional model as the best among other NN models. Bi-GRU, Att-Bi-LSTM, RNN-LSTM, NARX, and DCNN followed the defined behaviour of FDC; however, TCN and GC-LSTM expounded an outstanding result for behavioural FDC-Q. In contrast, TCN and GC-LSTM were identified with disappointing results for other categories of grid analysis.

The influence of Normalization was considerably higher than the box-cox transformation based on the uncertainty analysis. The DCNN model achieved noticeable results for all the sensitivity parameters, including for the SGD optimizer. Newly developed pre-trained models for dry zone basins interpreted outstanding results for forecasting. Thirteen NN models were compared with the models available in the previous studies, and the present study expressed a remarkable achievement even with the 30-days forecasting horizon. Especially, bidirectional models such as

BiGRU, Att-BiLSTM, and BiGRU revealed outstanding values for R^2 as 0.98, 0.97, and 0.95, respectively.

5.2 Limitations

Satellite data platforms lend a helping hand to gathering poorly accessible data for the NN model preparation. Input series were collected from various satellite products besides rainfall intensity and streamflow data. GEE was primarily implemented to sort out the lack of data issues. Generally, it provides excellent access to remote sensing products through the cloud platform. However, the platform needs missing data handling tools (Shelestov et al., 2017). Since the excellency of NN models effectively handles the missing data by imposing timely interpolation strategies.

The cloud system supports a limited number of training and validation samples in case of large-scale data collection. Commonly, the high-resolution platform does not support SAR data. The derived data sets using GEE must combine with monitoring and management decisions to improve modeling quality (Zhao et al., 2021). Climatic data sets such as rainfall and streamflow were cleansed with hydrological terms to enhance the data quality. Nevertheless, other data sets were not manipulated with hydrological cleaning strategies.

Literature has not defined any systematic scheme to address the input handling for NN models and those architecture characteristics. In addition, there needs to be adequate clarity on neglecting trend and seasonal components. There is neither empirical nor theoretical methodology to choose the suitable NN models among the alternatives (Andrea Sánchez-Sánchez et al., 2020). This research used grid analysis and AHP analysis to identify the best models.

5.3 Recommendations

Solar irradiance influences global warming, one of the factors considered for NN model preparation. In most parts of the world, global warming severely influences flood magnitudes and frequencies. Recent studies denote that the 50-year return period of river flow increases from 2% to 2.4% at 1.5ºC and 5.4 % at 4ºC of global warming

(He et al., 2022). Batches for sliding window assignment must be regarded as based on the global warming pattern, which is expected to reveal a reliable input handling.

The loss function is adapted to find the optimum error and update the weights and bias. The present study is proposed using a statistical tool, RMSE, to develop a relationship between target and predicted values. Forecasting problems are expected to coincide predicted with observed data sets. Therefore, the applicability of other statistical tools must be checked as loss functions. The gradient descent and the loss function must be assembled from the fundamental strategies to achieve this task. On the other hand, FDC curves and the relevant theory can be implemented to reconstruct the errors between observed and forecasted values. The viability of this application must be tested on the extended studies in the above area.

Autocorrelation, which means the inter-dependency of errors, is a popular term in time series data sets. The temporal nature of the data generally supports enabling the autocorrelation ability on error distribution (Sun et al., 2021). Therefore, the autocorrelation ability should be verified, and the newly developed error must be adapted to control the scattering of forecasted values.

The core model must be prepared for extended periods to facilitate addressing transfer learning effectively on other basins. In addition, the model must be evaluated on another wet zone basin to ensure its authenticity.

REFERENCE

Andrea Sánchez-Sánchez, P., Rafael García-González, J., & Haidy Perez Coronell, L. (2020). Encountered Problems of Time Series with Neural Networks: Models and Architectures. In *Recent Trends in Artificial Neural Networks - from Training to Prediction* (p. 13). IntechOpen. https://doi.org/10.5772/intechopen.88901

Apaydin, H., Taghi Sattari, M., Falsafian, K., & Prasad, R. (2021). Artificial intelligence modelling integrated with Singular Spectral analysis and Seasonal-Trend decomposition using Loess approaches for streamflow predictions. *Journal of Hydrology*, *600*(March), 1–16. https://doi.org/10.1016/j.jhydrol.2021.126506

Ashok, A., Rani, H. P., & Jayakumar, K. V. (2021). Monitoring of dynamic wetland changes using NDVI and NDWI based landsat imagery. *Remote Sensing Applications: Society and Environment*, *23*(May), 100547. https://doi.org/10.1016/j.rsase.2021.100547

Blum, L., Elgendi, M., & Menon, C. (2022). Impact of Box-Cox Transformation on Machine-Learning Algorithms. *Frontiers in Artificial Intelligence*, *5*(April), 1–16. https://doi.org/10.3389/frai.2022.877569

Börjesson, L., & Singull, M. (2020). Forecasting Financial Time Series through Causal and Dilated Convolutional Neural Networks. *Entropy*, *22*(10), 1094. https://doi.org/10.3390/e22101094

Boussaada, Z., Curea, O., Remaci, A., Camblong, H., & Mrabet Bellaaj, N. (2018). A Nonlinear Autoregressive Exogenous (NARX) Neural Network Model for the Prediction of the Daily Direct Solar Radiation. *Energies*, *11*(3), 620. https://doi.org/10.3390/en11030620

Chen, C., Hui, Q., Xie, W., Wan, S., Zhou, Y., & Pei, Q. (2021). Convolutional Neural Networks for forecasting flood process in Internet-of-Things enabled smart city. *Computer Networks*, *186*, 107744. https://doi.org/10.1016/j.comnet.2020.107744

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, *118*(March 2019), 102674. https://doi.org/10.1016/j.trc.2020.102674

de la Fuente, A., Meruane, V., & Meruane, C. (2019). Hydrological Early Warning System Based on a Deep Learning Runoff Model Coupled with a Meteorological Forecast. *Water*, *11*(9), 1808. https://doi.org/10.3390/w11091808

di Nunno, F., & Granata, F. (2020). Groundwater level prediction in Apulia region (Southern Italy) using NARX neural network. *Environmental Research*, *190*(July), 1–17. https://doi.org/10.1016/j.envres.2020.110062

Ding, Y., Zhu, Y., Feng, J., Zhang, P., & Cheng, Z. (2020). Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing*, *403*, 348–359. https://doi.org/10.1016/j.neucom.2020.04.110

Feng, D., Fang, K., & Shen, C. (2020). Enhancing Streamflow Forecast and Extracting Insights Using Long-Short Term Memory Networks With Data Integration at Continental Scales. *Water Resources Research*, *56*(9), 1–45. https://doi.org/10.1029/2019WR026793

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn , Keras & TensorFlow* (2nd Editio). O'Reilly Media. http://oreilly.com/catalog/errata.csp?isbn=9781492032649

Ghimire, S., Yaseen, Z. M., Farooque, A. A., Deo, R. C., Zhang, J., & Tao, X. (2021). Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. *Scientific Reports*, *11*(1), 1–26. https://doi.org/10.1038/s41598-021-96751-4

Hassan, M., & Hassan, I. (2021). Improving Artificial Neural Network Based Streamflow Forecasting Models through Data Preprocessing. *KSCE Journal of Civil Engineering*, *25*(9), 3583–3595. https://doi.org/10.1007/s12205-021-1859-y

Hayder, G., Iwan Solihin, M., & Najwa, M. R. N. (2022). Multi-step-ahead prediction of river flow using NARX neural networks and deep learning LSTM. *H2Open Journal*, *5*(1), 43–60. https://doi.org/10.2166/h2oj.2022.134

He, Y., Manful, D., Warren, R., Forstenhäusler, N., Osborn, T. J., Price, J., Jenkins, R., Wallace, C., & Yamazaki, D. (2022). Quantification of impacts between 1.5 and 4 °C of global warming on flooding risks in six countries. *Climatic Change*, *170*(1–2), 15. https://doi.org/10.1007/s10584-021-03289-5

Hettiarachchi, P. (2020). Hydrological Report on the Kelani River Flood in May 2016. *Research Gate*, 1–12. https://www.researchgate.net/publication/342865359%0AHydrological

Hussain, F., Wu, R.-S., & Wang, J.-X. (2021). Comparative study of very short-term flood forecasting using physics-based numerical model and data-driven prediction model. *Natural Hazards*, *107*(1), 249–284. https://doi.org/10.1007/s11069-021-04582-3

Jain, S., Jaiswal, R. K., Lohani, A. K., & Galkate, R. (2021). Development of Cloud-Based Rainfall–Run-Off Model Using Google Earth Engine. *Current Science*, *121*(11), 1433. https://doi.org/10.18520/cs/v121/i11/1433-1440

Jiang, F., Dong, Z., Wang, Z., Zhu, Y., Liu, M., Luo, Y., & Zhang, T. (2021). Flood forecasting using an improved NARX network based on wavelet analysis coupled with uncertainty analysis by Monte Carlo simulations: a case study of Taihu Basin, China. *Journal of Water and Climate Change*, *12*(6), 2674–2696. https://doi.org/10.2166/wcc.2021.019

Jimeno-Sáez, P., Senent-Aparicio, J., Pérez-Sánchez, J., & Pulido-Velazquez, D. (2018). A Comparison of SWAT and ANN Models for Daily Runoff Simulation in Different Climatic Zones of Peninsular Spain. *Water*, *10*(2), 192. https://doi.org/10.3390/w10020192

Ketkar, N., & Moolayil, J. (2021). Deep Learning with Python. In *Deep Learning with Python*. Apress. https://doi.org/10.1007/978-1-4842-5364-9

Kilinc, H. C., & Haznedar, B. (2022). A Hybrid Model for Streamflow Forecasting in the Basin of Euphrates. *Water*, *14*(1), 80. https://doi.org/10.3390/w14010080

Kilinc, H. C., & Yurtsever, A. (2022). Short-Term Streamflow Forecasting Using Hybrid Deep Learning Model Based on Grey Wolf Algorithm for Hydrological Time Series. *Sustainability*, *14*(6), 1–20. https://doi.org/10.3390/su14063352

Kottagoda, S., & Abeysingha, N. (2017). Morphometric analysis of watersheds in Kelani river basin for soil and water conservation. *Journal of the National Science Foundation of Sri Lanka*, *45*(3), 273. https://doi.org/10.4038/jnsfsr.v45i3.8192

Le, X.-H., Nguyen, D.-H., Jung, S., Yeon, M., & Lee, G. (2021). Comparison of Deep Learning Techniques for River Streamflow Forecasting. *IEEE Access*, *9*, 71805–71820. https://doi.org/10.1109/ACCESS.2021.3077703

Lee, M.-C., Chang, J.-W., Yeh, S.-C., Chia, T.-L., Liao, J.-S., & Chen, X.-M. (2022). Applying attention-based BiLSTM and technical indicators in the design and performance analysis of stock trading strategies. *Neural Computing and Applications*, *34*(16), 13267–13279. https://doi.org/10.1007/s00521-021-06828-4

Lin, Y., Wang, D., Wang, G., Qiu, J., Long, K., Du, Y., Xie, H., Wei, Z., Shangguan, W., & Dai, Y. (2021). A hybrid deep learning algorithm and its application to streamflow prediction. *Journal of Hydrology*, *601*(July), 1–10. https://doi.org/10.1016/j.jhydrol.2021.126636

Mahenthiran, B., & Rajapakse, L. (2021). Water Resources Availability and Low Flow Discharge Analysis of Two Selected River Basins in the Dry Zone Under Changing Climate Conditions. *2021 Moratuwa Engineering Research Conference (MERCon)*, 504–509. https://doi.org/10.1109/MERCon52712.2021.9525654

Manawadu, L., & Wijeratne, V. P. I. S. (2021). Anthropogenic drivers and impacts of urban flooding- A case study in Lower Kelani River Basin, Colombo Sri Lanka. *International Journal of Disaster Risk Reduction*, *57*(January), 102076. https://doi.org/10.1016/j.ijdrr.2021.102076

Mrzygłód, B., Hawryluk, M., Janik, M., & Olejarczyk-Wożeńska, I. (2020). Sensitivity analysis of the artificial neural networks in a system for durability prediction of forging tools to forgings made of C45 steel. *The International Journal of Advanced Manufacturing Technology*, *109*(5–6), 1385–1395. https://doi.org/10.1007/s00170-020-05641-y

Nashwan, M. S., Shahid, S., & Wang, X. (2019). Uncertainty in Estimated Trends Using Gridded Rainfall Data: A Case Study of Bangladesh. *Water*, *11*(2), 349. https://doi.org/10.3390/w11020349

Qu, J., Qian, Z., & Pei, Y. (2021). Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern. *Energy*, *232*, 120996. https://doi.org/10.1016/j.energy.2021.120996

Rodrigues, G. C., & Braga, R. P. (2021). Evaluation of NASA POWER Reanalysis Products to Estimate Daily Weather Variables in a Hot Summer Mediterranean Climate. *Agronomy*, *11*(6), 1207. https://doi.org/10.3390/agronomy11061207

Saraiva, S. V., Carvalho, F. de O., Santos, C. A. G., Barreto, L. C., & Freire, P. K. de M. M. (2021). Daily streamflow forecasting in Sobradinho Reservoir using machine learning models coupled with wavelet transform and bootstrapping. *Applied Soft Computing*, *102*, 1–11. https://doi.org/10.1016/j.asoc.2021.107081

Satwik, P. M., & Sundram, M. (2021). An integrated approach for weather forecasting and disaster prediction using deep learning architecture based on memory Augmented Neural Network's (MANN's). *Materials Today: Proceedings*, *xxxx*, 1–6. https://doi.org/10.1016/j.matpr.2021.01.705

Sha, J., Li, X., Zhang, M., & Wang, Z.-L. (2021). Comparison of Forecasting Models for Real-Time Monitoring of Water Quality Parameters Based on Hybrid Deep Learning Neural Networks. *Water*, *13*(11), 1547. https://doi.org/10.3390/w13111547

Shamshirband, S., Jafari Nodoushan, E., Adolf, J. E., Abdul Manaf, A., Mosavi, A., & Chau, K. (2019). Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters. *Engineering Applications of Computational Fluid Mechanics*, *13*(1), 91–101. https://doi.org/10.1080/19942060.2018.1553742

Shelestov, A., Lavreniuk, M., Kussul, N., Novikov, A., & Skakun, S. (2017). Exploring Google Earth Engine Platform for Big Data Processing: Classification of Multi-Temporal Satellite Imagery for Crop Mapping. *Frontiers in Earth Science*, *5*(February), 1–10. https://doi.org/10.3389/feart.2017.00017

Shen, H., & Lin, J. (2020). Investigation of crowdshipping delivery trip production with real-world data. *Transportation Research Part E: Logistics and Transportation Review*, *143*(August), 102106. https://doi.org/10.1016/j.tre.2020.102106

Song, T., Ding, W., Liu, H., Wu, J., Zhou, H., & Chu, J. (2020). Uncertainty Quantification in Machine Learning Modeling for Multi-Step Time Series Forecasting: Example of Recurrent Neural Networks in Discharge Simulations. *Water*, *12*(3), 912. https://doi.org/10.3390/w12030912

Subramanya, K. (2017). Engineering Hydrology. In *African, American* (Third Edit). Zed Books Ltd. https://doi.org/10.5040/9781350218178.0013

Sun, F. K., Lang, C. I., & Boning, D. S. (2021). Adjusting for Autocorrelated Errors in Neural Networks for Time Series. *Advances in Neural Information Processing Systems*, *36*(NeurIPS), 1–21. https://doi.org/https://doi.org/10.48550/arXiv.2101.12578

Vivekanandan, N. (2019). Use of Catchment Physiographic Factors in Selection of Design Storm and its Effect on Floods Estimated for Ungauged Catchments. *Civil Engineering Research Journal*, *9*(2), 67–75. https://doi.org/10.19080/CERJ.2019.09.555759

Wan, H., Guo, S., Yin, K., Liang, X., & Lin, Y. (2020). CTS-LSTM: LSTM-based neural networks for correlatedtime series prediction. *Knowledge-Based Systems*, *191*(xxxx), 105239. https://doi.org/10.1016/j.knosys.2019.105239

Wang, Q., Liu, Y., Yue, Q., Zheng, Y., Yao, X., & Yu, J. (2020). Impact of Input Filtering and Architecture Selection Strategies on GRU Runoff Forecasting: A Case Study in the Wei River Basin, Shaanxi, China. *Water*, *12*(12), 1–20. https://doi.org/10.3390/w12123532

Westerberg, I. K., Guerrero, J.-L., Younger, P. M., Beven, K. J., Seibert, J., Halldin, S., Freer, J. E., & Xu, C.-Y. (2011). Calibration of hydrological models using flow-duration curves. *Hydrology and Earth System Sciences*, *15*(7), 2205–2227. https://doi.org/10.5194/hess-15-2205-2011

Withanage, N. S., Dayawansa, D. K., Silva, R. P. D., & Rathnayake, C. W. M. R. (2018). Assessment of Morphological Characteristics of Maduru Oya Assessment of Morphological Characteristics of Maduru Oya. *39th Asian Conference on Remote Sensing: Remote Sensing Enabling Prosperity, ACRS 2018 , Malaysia*, *November*, 1–10. https://www.researchgate.net/publication/329164617%0AASSESSMENT

Won, Y.-M., Lee, J.-H., Moon, H.-T., & Moon, Y.-I. (2022). Development and Application of an Urban Flood Forecasting and Warning Process to Reduce Urban Flood Damage: A Case Study of Dorim River Basin, Seoul. *Water*, *14*(2), 187. https://doi.org/10.3390/w14020187

Xu, Y., Hu, C., Wu, Q., Li, Z., Jian, S., & Chen, Y. (2021). Application of temporal convolutional network for flood forecasting. *Hydrology Research*, *52*(6), 1455–1468. https://doi.org/10.2166/nh.2021.021

Yan, R., Liao, J., Yang, J., Sun, W., Nong, M., & Li, F. (2021). Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. *Expert Systems with Applications*, *169*(December 2020), 114513. https://doi.org/10.1016/j.eswa.2020.114513

Yu, J. (2021). A New Way of Airline Traffic Prediction Based on GCN-LSTM. *Frontiers in Neurorobotics*, *15*(December), 1–8. https://doi.org/10.3389/fnbot.2021.661037

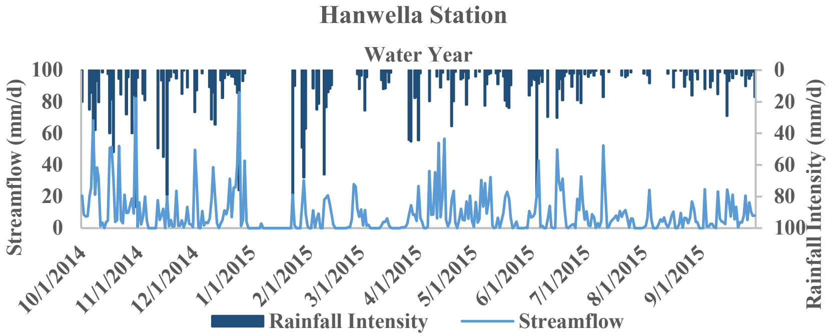
Zhang, J., Chen, X., Khan, A., Zhang, Y., Kuang, X., Liang, X., Taccari, M. L., & Nuttall, J. (2021). Daily runoff forecasting by deep recursive neural network. *Journal of Hydrology*, *596*(December 2020), 126067. https://doi.org/10.1016/j.jhydrol.2021.126067

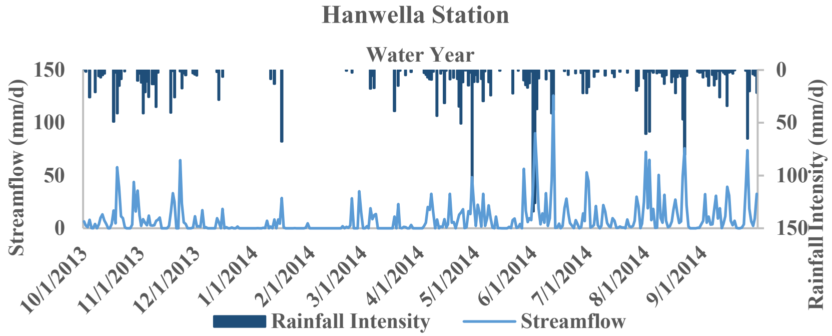
Zhao, Q., Yu, L., Li, X., Peng, D., Zhang, Y., & Gong, P. (2021). Progress and Trends in the Application of Google Earth and Google Earth Engine. *Remote Sensing*, *13*(18), 3778. https://doi.org/10.3390/rs13183778

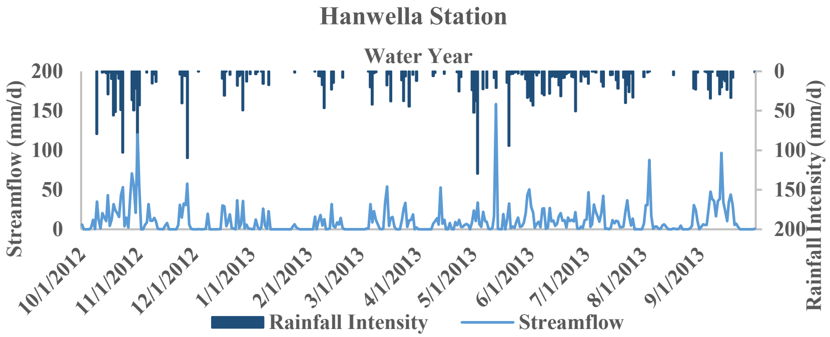
Zuhail, M., Rasmy, M., Koike, T., & Hitoshi, L. (2007). *Development of Integrated Water Resources Management Plan for Eastern Dry Zone in Sri Lanka: The Case of Gal Oya*. 1–6. https://www.pwri.go.jp/icharm/training/master/img/2019/synopses/08_Zuhail_synopsis.pdf

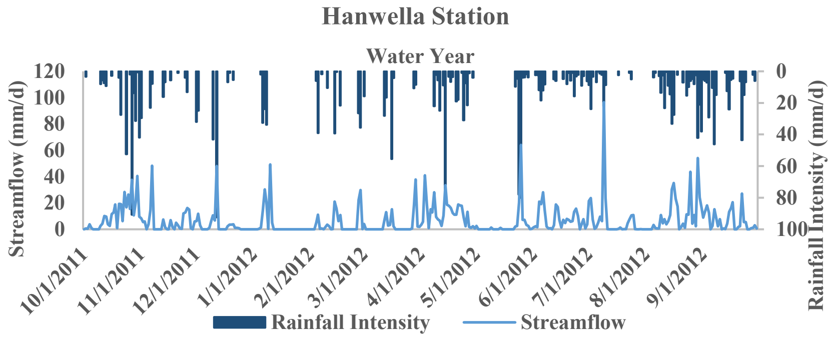
APPENDICES

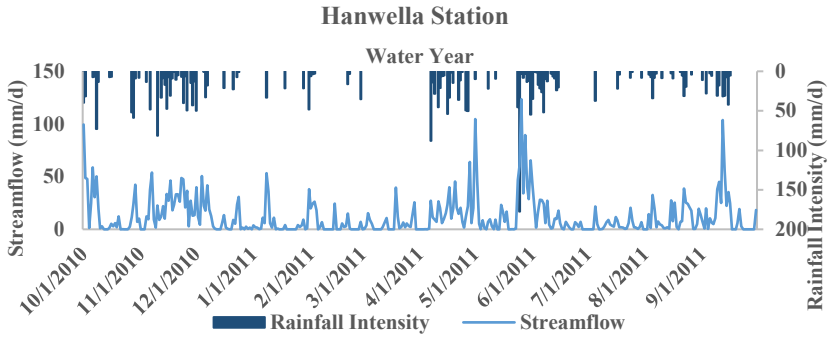
Appendix A: Visual Checking of Streamflow and Rainfall of each station for Water Year (2008-2015) – Kelani River Basin

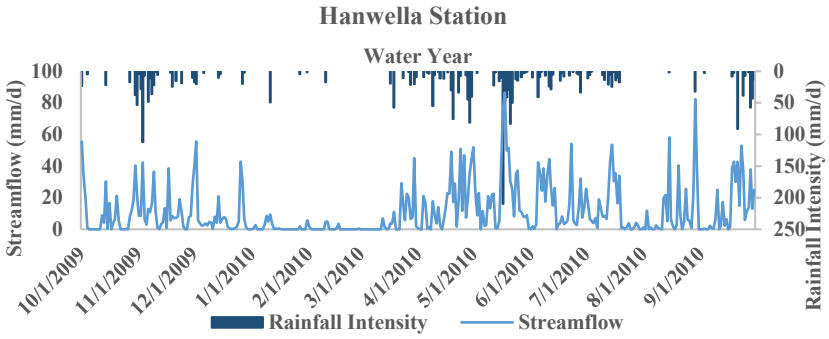


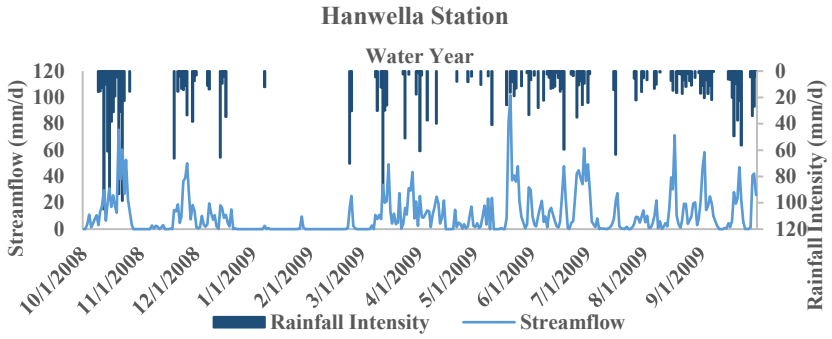


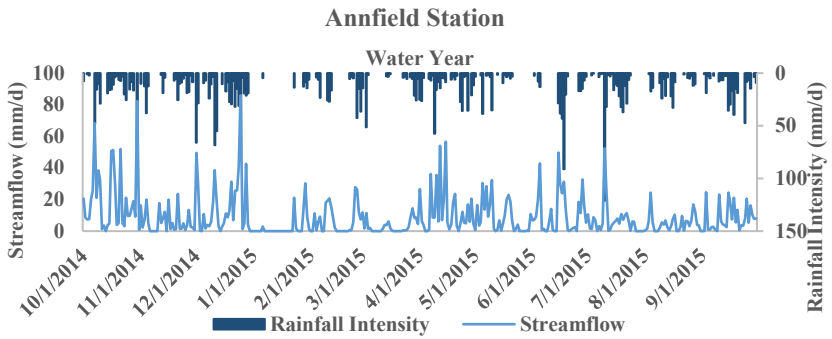


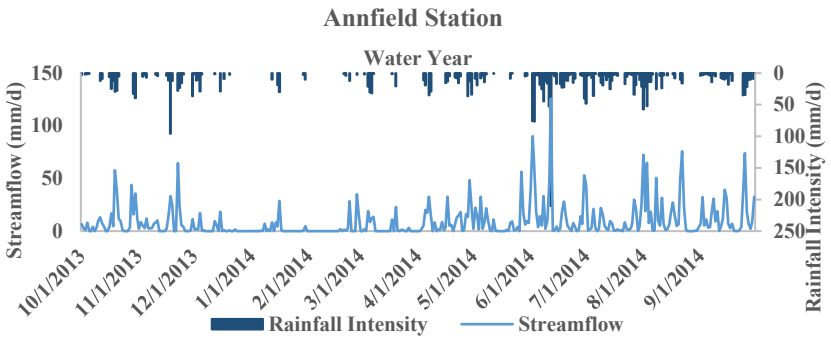


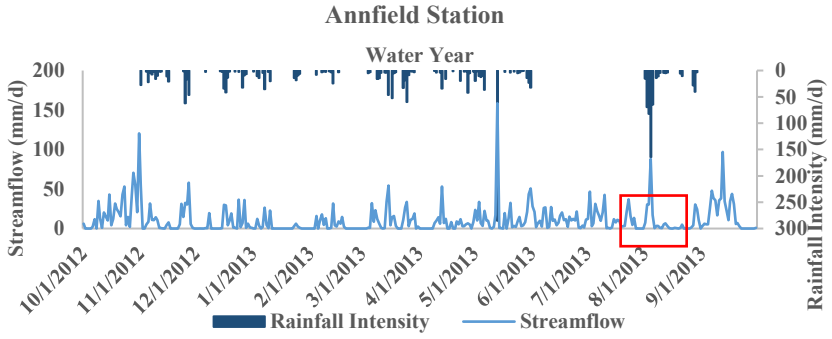

Hanwella Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
200
10/1/2010
11/1/2010
12/1/2010
1/1/2011
2/1/2011
3/1/2011
4/1/2011
5/1/2011
6/1/2011
7/1/2011
8/1/2011
9/1/2011
Rainfall Intensity
Streamflow

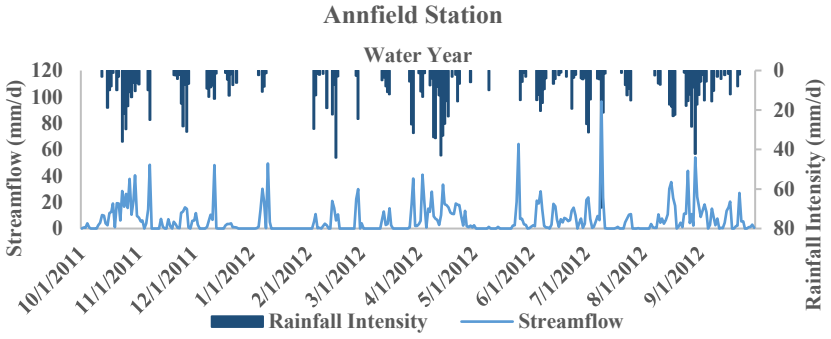

Hanwella Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
200
250
10/1/2009
11/1/2009
12/1/2009
1/1/2010
2/1/2010
3/1/2010
4/1/2010
5/1/2010
6/1/2010
7/1/2010
8/1/2010
9/1/2010
Rainfall Intensity
Streamflow

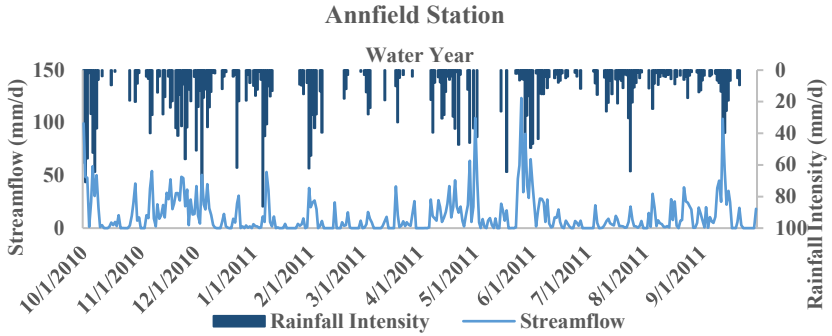

Hanwella Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
20
40
60
80
100
120
10/1/2008
11/1/2008
12/1/2008
1/1/2009
2/1/2009
3/1/2009
4/1/2009
5/1/2009
6/1/2009
7/1/2009
8/1/2009
9/1/2009
Rainfall Intensity
Streamflow

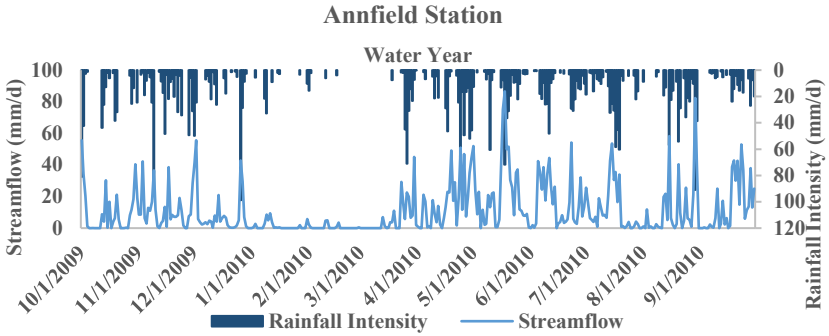

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
10/1/2014
11/1/2014
12/1/2014
1/1/2015
2/1/2015
3/1/2015
4/1/2015
5/1/2015
6/1/2015
7/1/2015
8/1/2015
9/1/2015
Rainfall Intensity
Streamflow

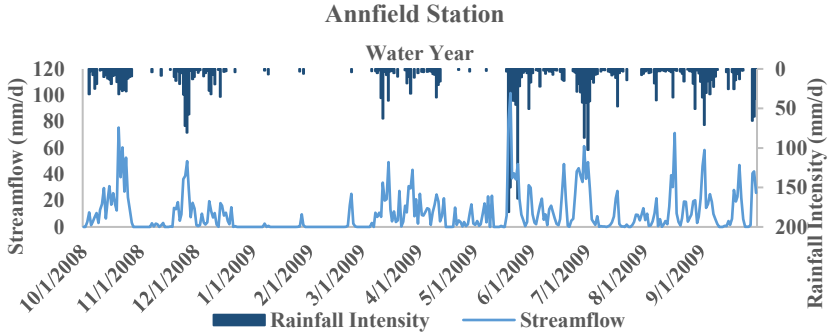

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
200
250
10/1/2013
11/1/2013
12/1/2013
1/1/2014
2/1/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014
Rainfall Intensity
Streamflow

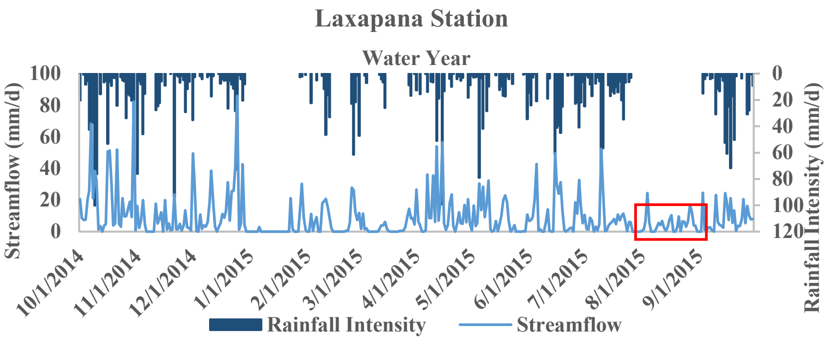

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
200
150
100
50
0
0
50
100
150
200
250
300
10/1/2012
11/1/2012
12/1/2012
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013
Rainfall Intensity
Streamflow

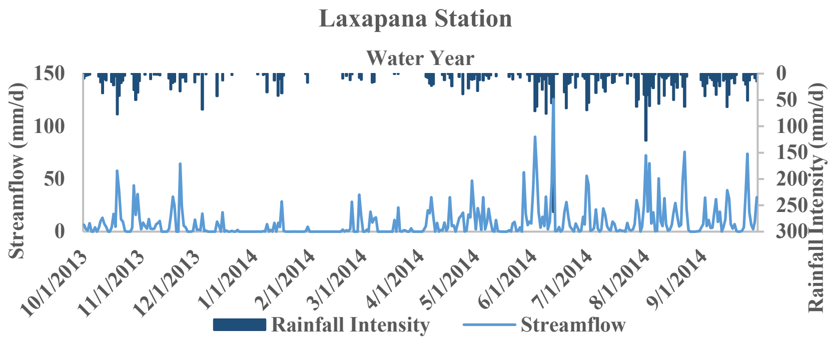

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
20
40
60
80
10/1/2011
11/1/2011
12/1/2011
1/1/2012
2/1/2012
3/1/2012
4/1/2012
5/1/2012
6/1/2012
7/1/2012
8/1/2012
9/1/2012
Rainfall Intensity
Streamflow

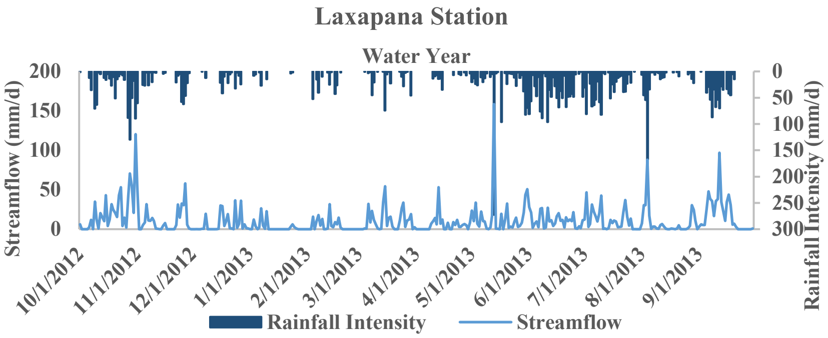

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
20
40
60
80
100
10/1/2010
11/1/2010
12/1/2010
1/1/2011
2/1/2011
3/1/2011
4/1/2011
5/1/2011
6/1/2011
7/1/2011
8/1/2011
9/1/2011
Rainfall Intensity
Streamflow

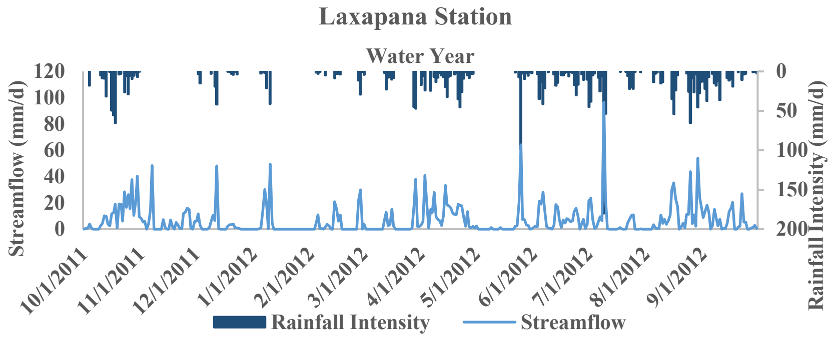

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
20
40
60
80
100
120
10/1/2009
11/1/2009
12/1/2009
1/1/2010
2/1/2010
3/1/2010
4/1/2010
5/1/2010
6/1/2010
7/1/2010
8/1/2010
9/1/2010
Rainfall Intensity
Streamflow

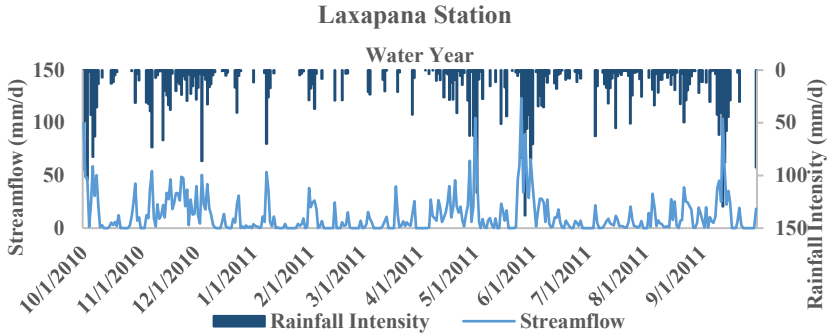

Annfield Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
50
100
150
200
10/1/2008
11/1/2008
12/1/2008
1/1/2009
2/1/2009
3/1/2009
4/1/2009
5/1/2009
6/1/2009
7/1/2009
8/1/2009
9/1/2009
Rainfall Intensity
Streamflow

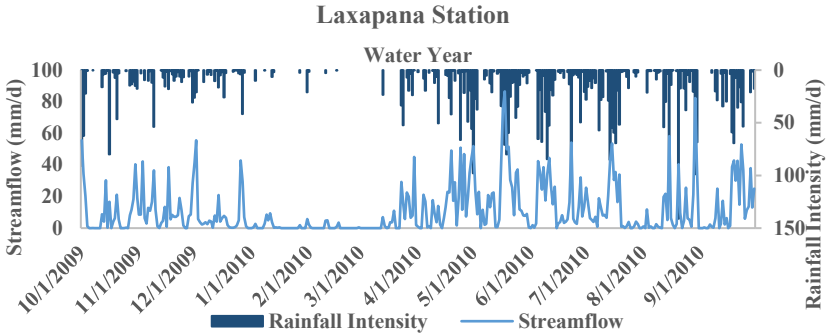

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
20
40
60
80
100
120
10/1/2014
11/1/2014
12/1/2014
1/1/2015
2/1/2015
3/1/2015
4/1/2015
5/1/2015
6/1/2015
7/1/2015
8/1/2015
9/1/2015
Rainfall Intensity
Streamflow

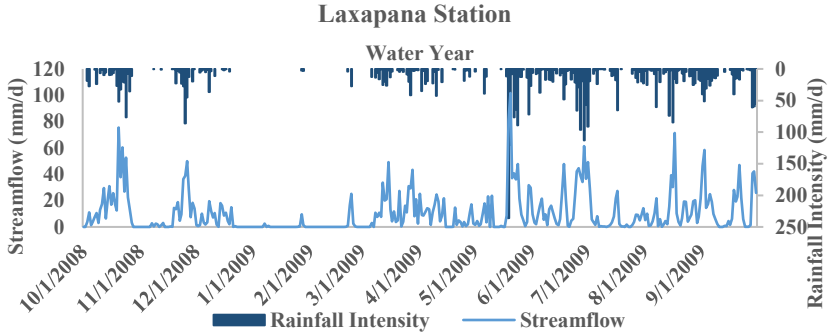

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
200
250
300
10/1/2013
11/1/2013
12/1/2013
1/1/2014
2/1/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014
Rainfall Intensity
Streamflow

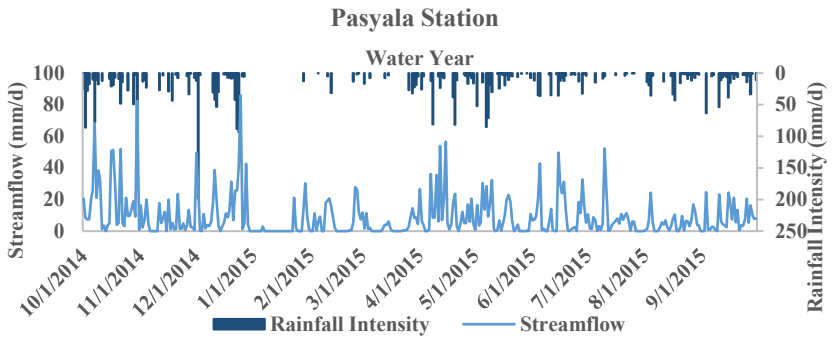

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
200
150
100
50
0
0
50
100
150
200
250
300
10/1/2012
11/1/2012
12/1/2012
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013
Rainfall Intensity
Streamflow

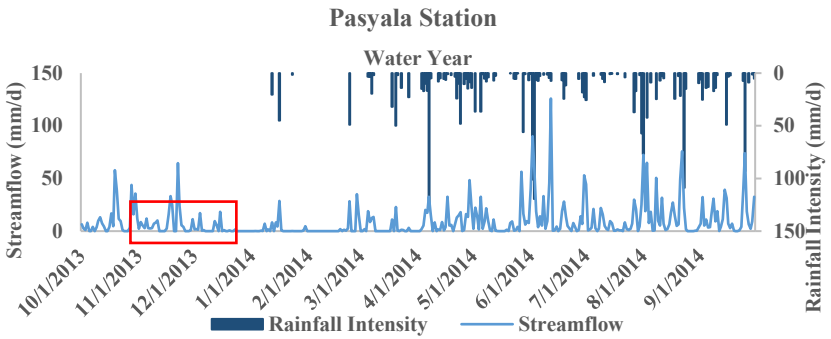

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
50
100
150
200
10/1/2011
11/1/2011
12/1/2011
1/1/2012
2/1/2012
3/1/2012
4/1/2012
5/1/2012
6/1/2012
7/1/2012
8/1/2012
9/1/2012
Rainfall Intensity
Streamflow

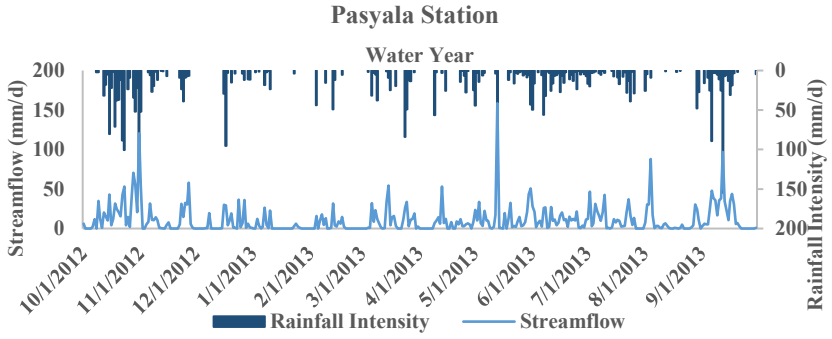

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
10/1/2010
11/1/2010
12/1/2010
1/1/2011
2/1/2011
3/1/2011
4/1/2011
5/1/2011
6/1/2011
7/1/2011
8/1/2011
9/1/2011
Rainfall Intensity
Streamflow

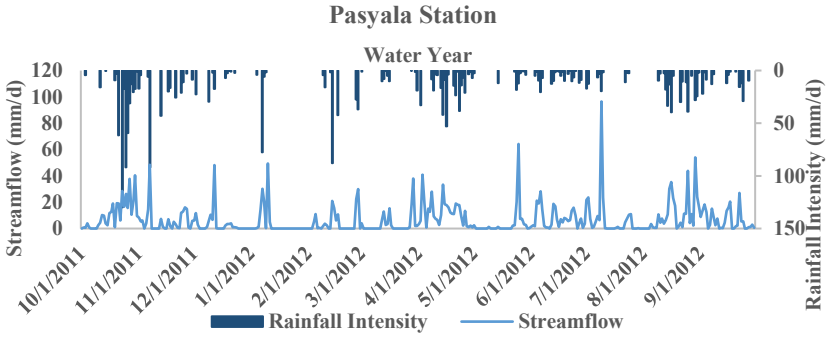

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
10/1/2009
11/1/2009
12/1/2009
1/1/2010
2/1/2010
3/1/2010
4/1/2010
5/1/2010
6/1/2010
7/1/2010
8/1/2010
9/1/2010
Rainfall Intensity
Streamflow

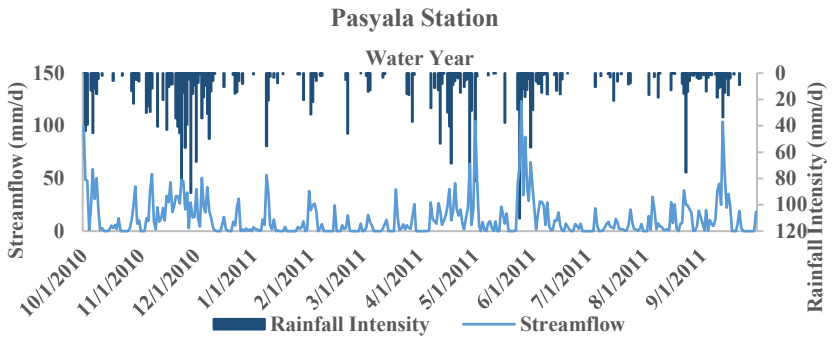

Laxapana Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
50
100
150
200
250
10/1/2008
11/1/2008
12/1/2008
1/1/2009
2/1/2009
3/1/2009
4/1/2009
5/1/2009
6/1/2009
7/1/2009
8/1/2009
9/1/2009
Rainfall Intensity
Streamflow

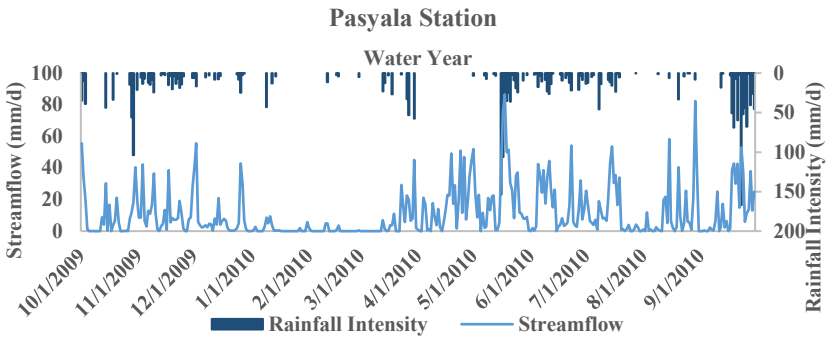

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
200
250
10/1/2014
11/1/2014
12/1/2014
1/1/2015
2/1/2015
3/1/2015
4/1/2015
5/1/2015
6/1/2015
7/1/2015
8/1/2015
9/1/2015
Rainfall Intensity
Streamflow

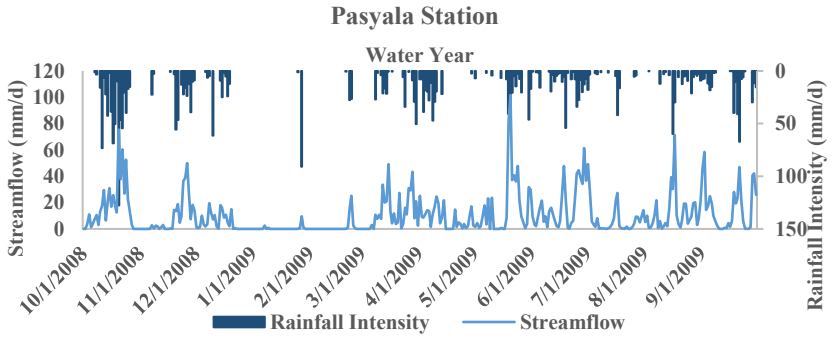

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
10/1/2013
11/1/2013
12/1/2013
1/1/2014
2/1/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014
Rainfall Intensity
Streamflow

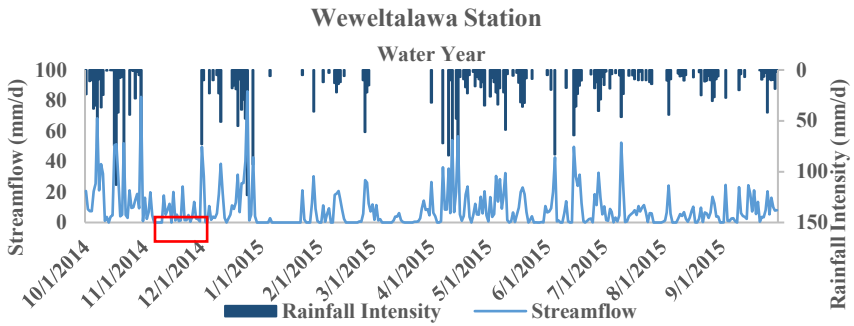

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
200
150
100
50
0
0
50
100
150
200
10/1/2012
11/1/2012
12/1/2012
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013
Rainfall Intensity
Streamflow

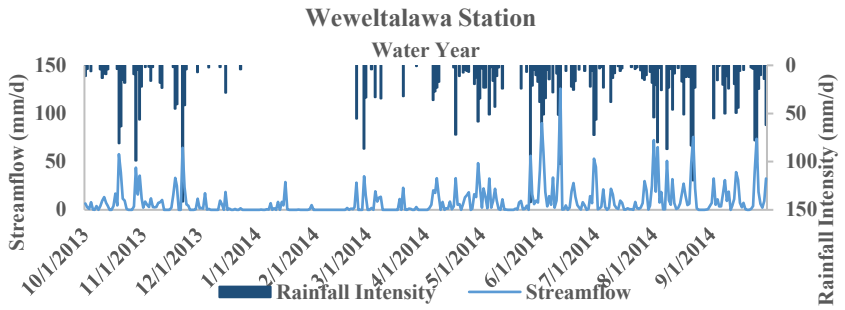

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
50
100
150
10/1/2011
11/1/2011
12/1/2011
1/1/2012
2/1/2012
3/1/2012
4/1/2012
5/1/2012
6/1/2012
7/1/2012
8/1/2012
9/1/2012
Rainfall Intensity
Streamflow

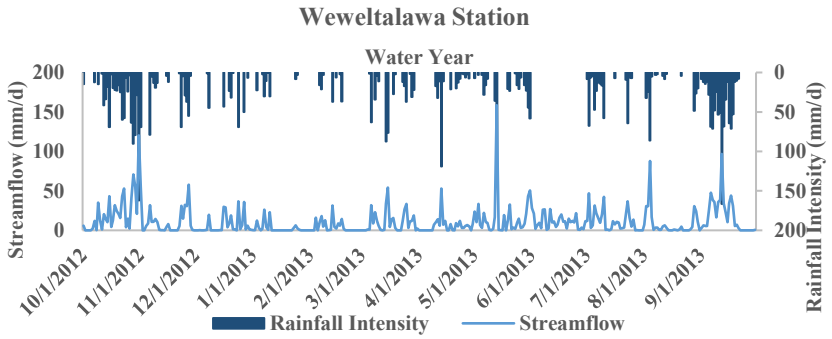

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
20
40
60
80
100
120
10/1/2010
11/1/2010
12/1/2010
1/1/2011
2/1/2011
3/1/2011
4/1/2011
5/1/2011
6/1/2011
7/1/2011
8/1/2011
9/1/2011
Rainfall Intensity
Streamflow

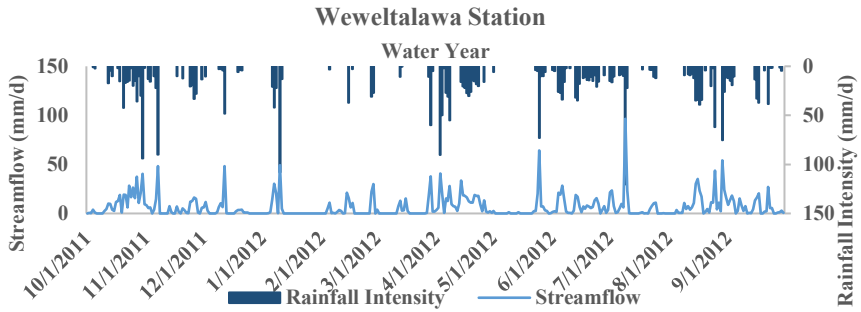

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
200
10/1/2009
11/1/2009
12/1/2009
1/1/2010
2/1/2010
3/1/2010
4/1/2010
5/1/2010
6/1/2010
7/1/2010
8/1/2010
9/1/2010
Rainfall Intensity
Streamflow

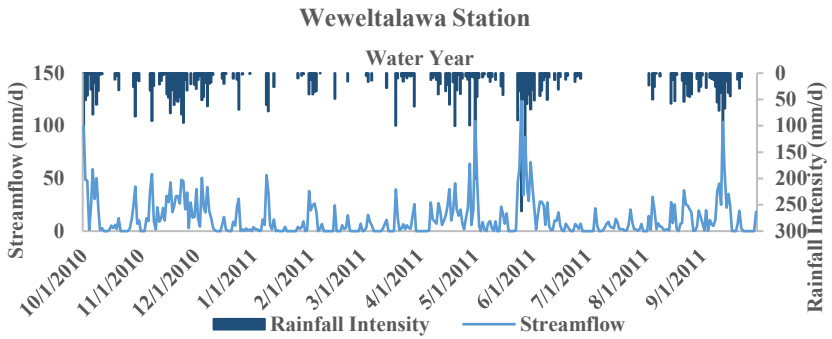

Pasyala Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
120
100
80
60
40
20
0
0
50
100
150
10/1/2008
11/1/2008
12/1/2008
1/1/2009
2/1/2009
3/1/2009
4/1/2009
5/1/2009
6/1/2009
7/1/2009
8/1/2009
9/1/2009
Rainfall Intensity
Streamflow

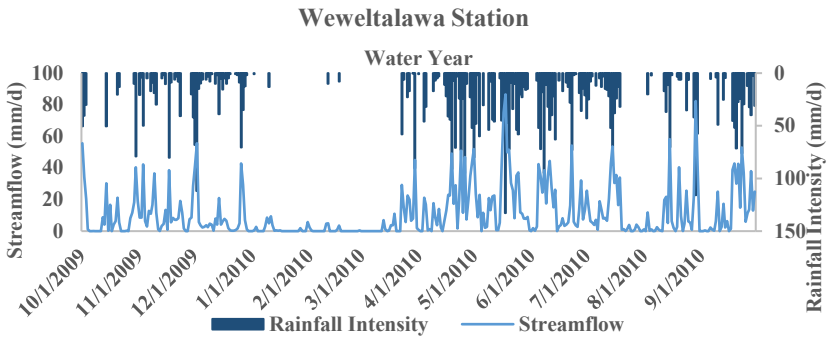

Weweltalawa Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
100
80
60
40
20
0
0
50
100
150
10/1/2014
11/1/2014
12/1/2014
1/1/2015
2/1/2015
3/1/2015
4/1/2015
5/1/2015
6/1/2015
7/1/2015
8/1/2015
9/1/2015
Rainfall Intensity
Streamflow

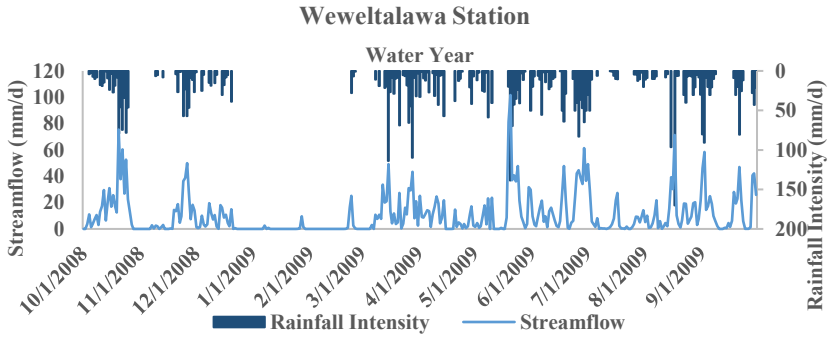

Weweltalawa Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
10/1/2013
11/1/2013
12/1/2013
1/1/2014
2/1/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014
Rainfall Intensity
Streamflow


Weweltalawa Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
200
150
100
50
0
0
50
100
150
200
10/1/2012
11/1/2012
12/1/2012
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013
Rainfall Intensity
Streamflow


Weweltalawa Station
Water Year
Streamflow (mm/d)
Rainfall Intensity (mm/d)
150
100
50
0
0
50
100
150
10/1/2011
11/1/2011
12/1/2011
1/1/2012
2/1/2012
3/1/2012
4/1/2012
5/1/2012
6/1/2012
7/1/2012
8/1/2012
9/1/2012
Rainfall Intensity
Streamflow

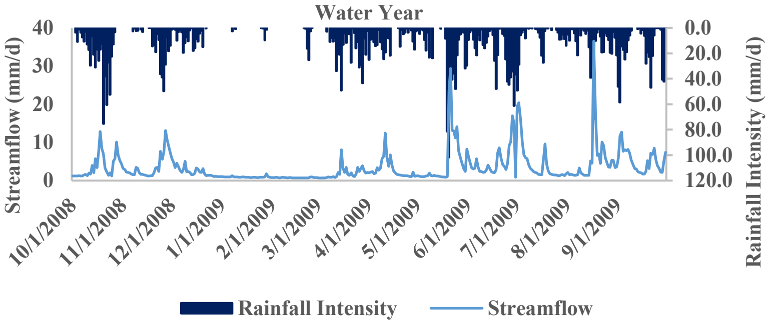


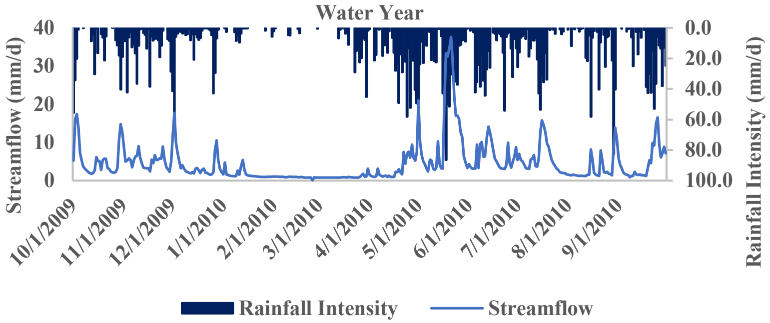


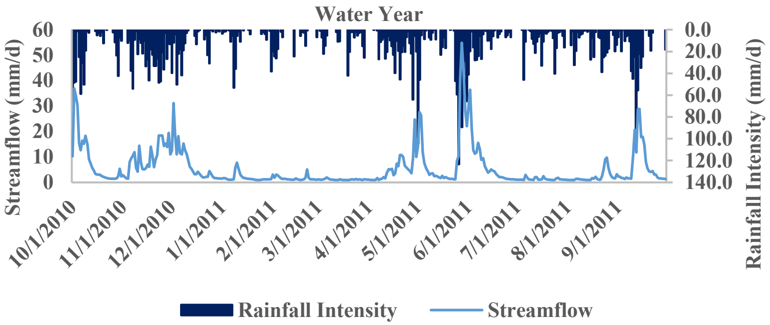


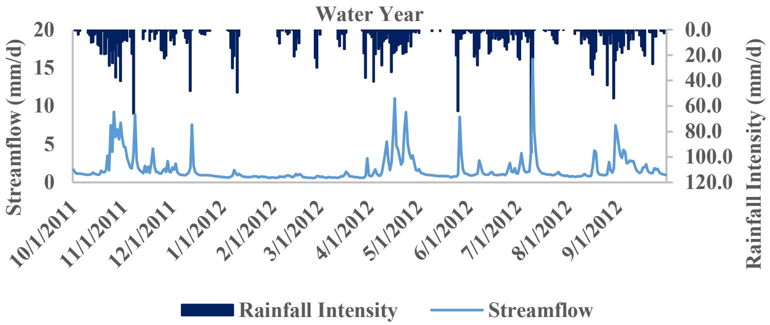
*According to the visual checking, the abnormalities and inconsistencies of the hydrological data are negligible. It was indicated with a red color box.

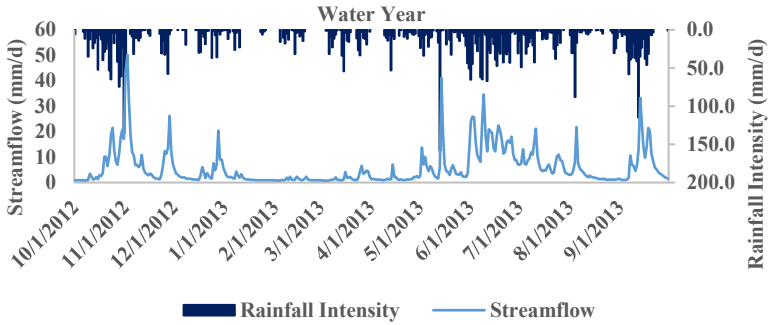
Appendix B: Visual Checking of Streamflow and Thiessen Averaged Rainfall for Water Year (2008-2015) – Kelani River Basin

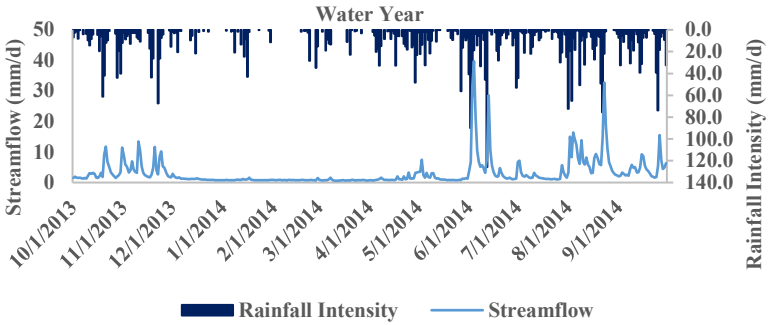


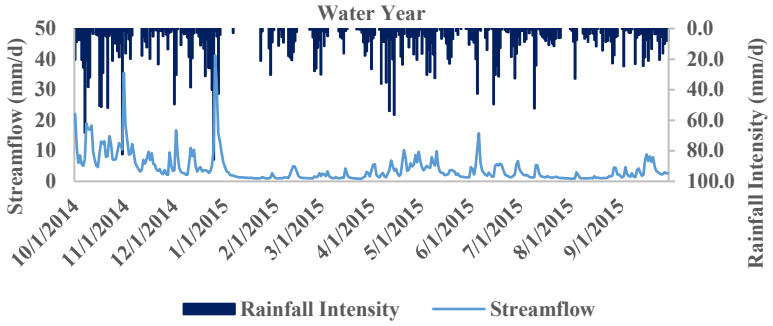






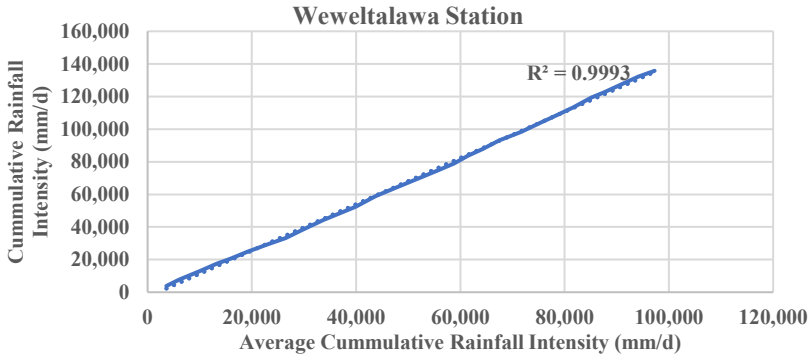


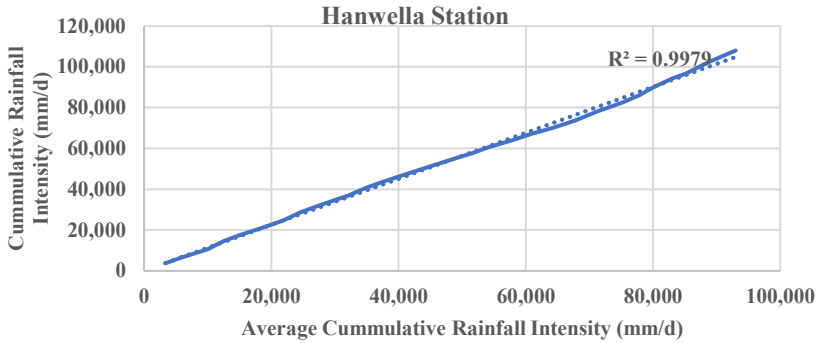




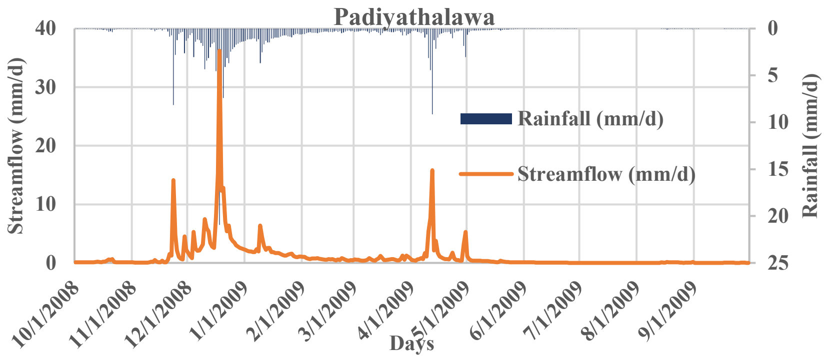
* The response of average rainfall and streamflow shows a good visual appearance.

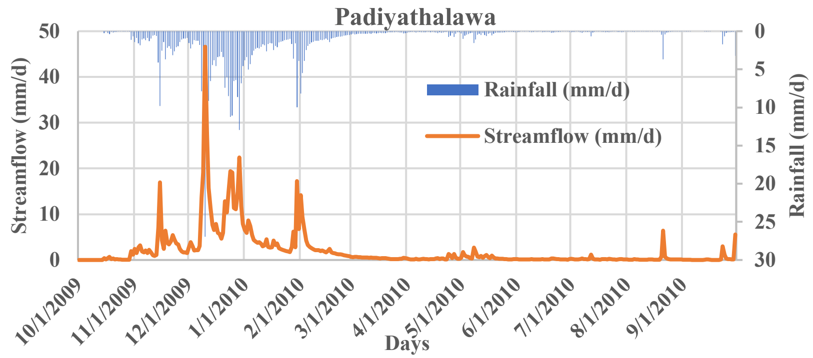
Appendix C: Double Mass Curve for the Rainfall Stations – Kelani River Basin

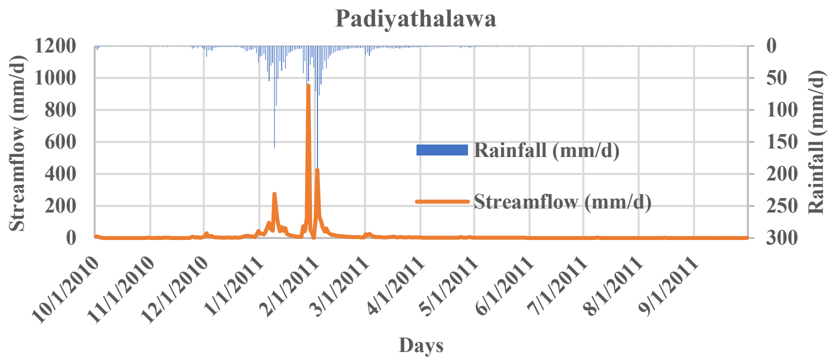


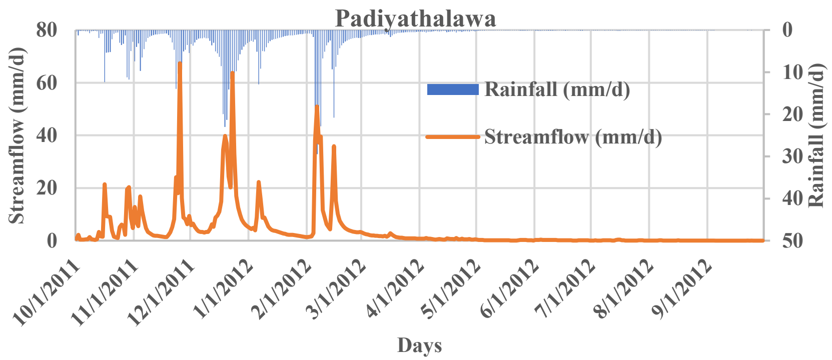


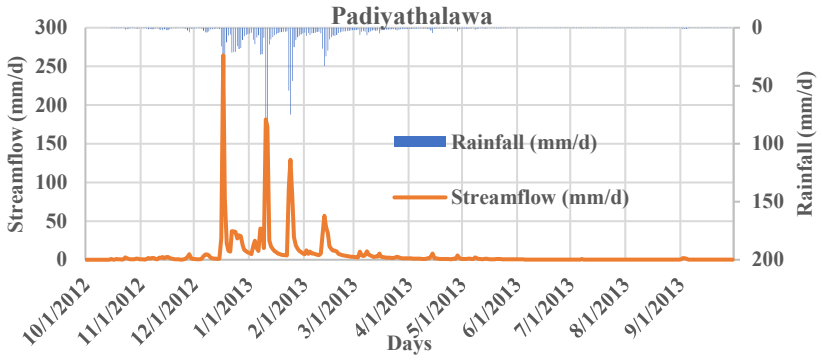
Appendix D: Visual Checking of Streamflow and Rainfall of each station for Water Year (2008-2015) – Maduru Oya Basin

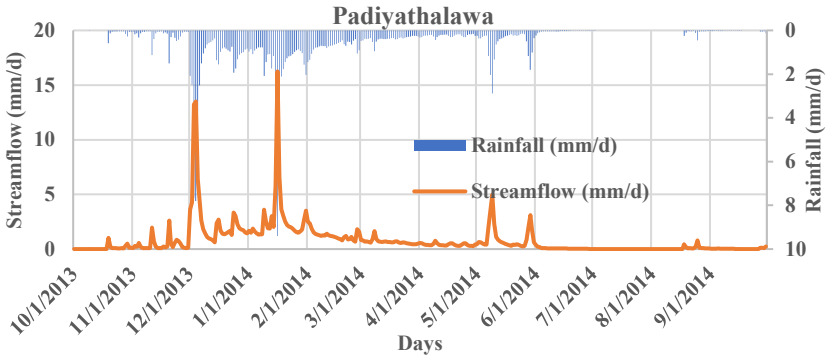


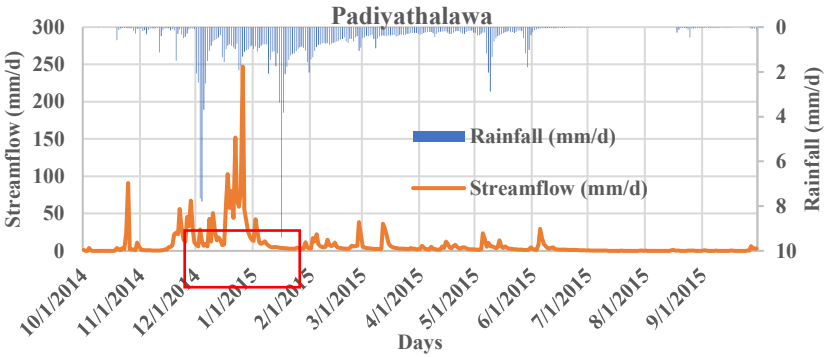


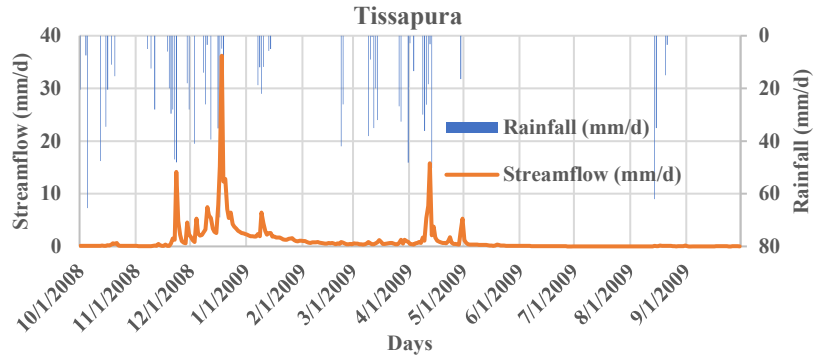


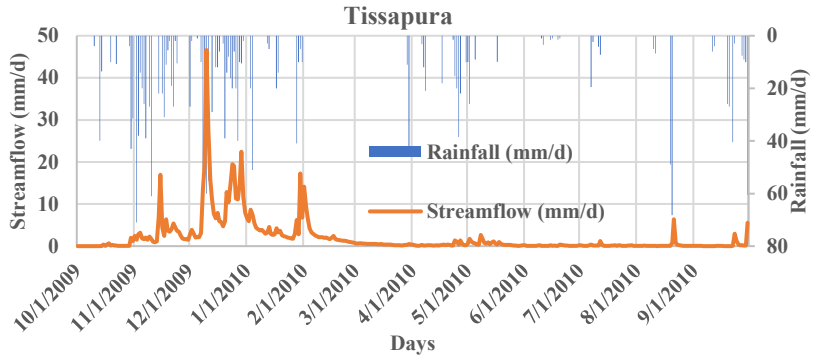


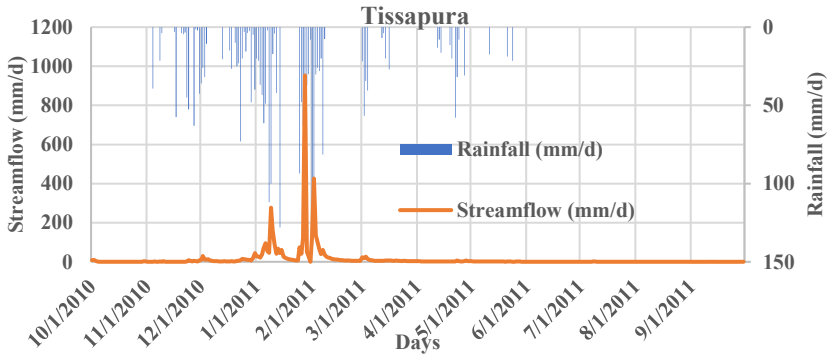

Padiyathalawa
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
300
250
200
150
100
50
0
0
50
100
150
200
10/1/2012
11/1/2012
12/1/2012
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013

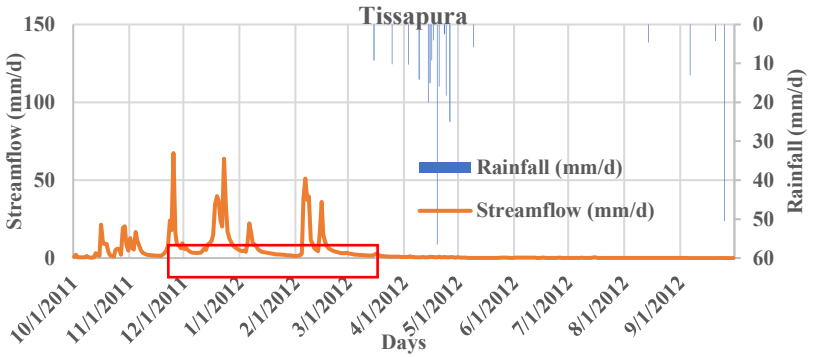

Padiyathalawa
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
20
15
10
5
0
0
2
4
6
8
10
10/1/2013
11/1/2013
12/1/2013
1/1/2014
2/1/2014
3/1/2014
4/1/2014
5/1/2014
6/1/2014
7/1/2014
8/1/2014
9/1/2014

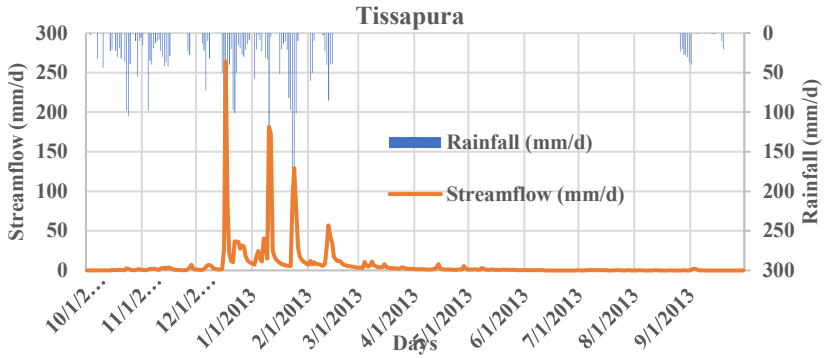

Padiyathalawa
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
300
250
200
150
100
50
0
0
2
4
6
8
10
10/1/2014
11/1/2014
12/1/2014
1/1/2015
2/1/2015
3/1/2015
4/1/2015
5/1/2015
6/1/2015
7/1/2015
8/1/2015
9/1/2015

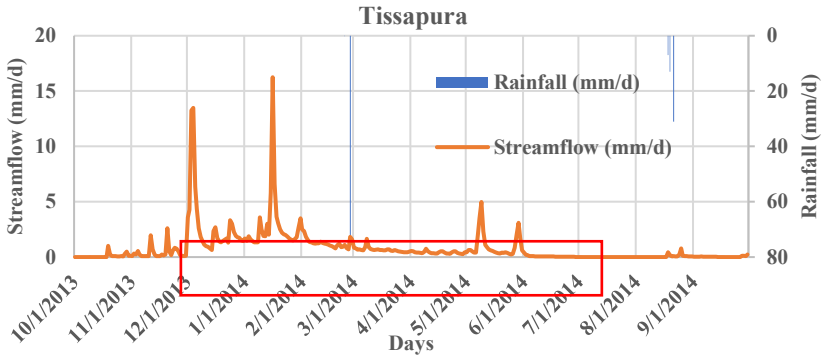

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
40
30
20
10
0
0
20
40
60
80
10/1/2008
11/1/2008
12/1/2008
1/1/2009
2/1/2009
3/1/2009
4/1/2009
5/1/2009
6/1/2009
7/1/2009
8/1/2009
9/1/2009

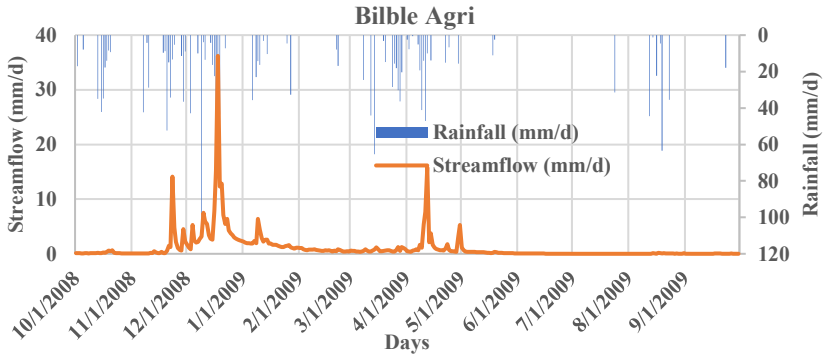

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
50
40
30
20
10
0
0
20
40
60
80
10/1/2009
11/1/2009
12/1/2009
1/1/2010
2/1/2010
3/1/2010
4/1/2010
5/1/2010
6/1/2010
7/1/2010
8/1/2010
9/1/2010

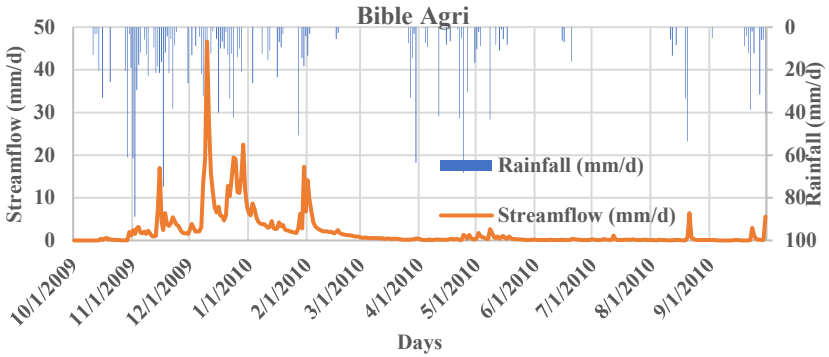

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
1200
1000
800
600
400
200
0
0
50
100
150
10/1/2010
11/1/2010
12/1/2010
1/1/2011
2/1/2011
3/1/2011
4/1/2011
5/1/2011
6/1/2011
7/1/2011
8/1/2011
9/1/2011

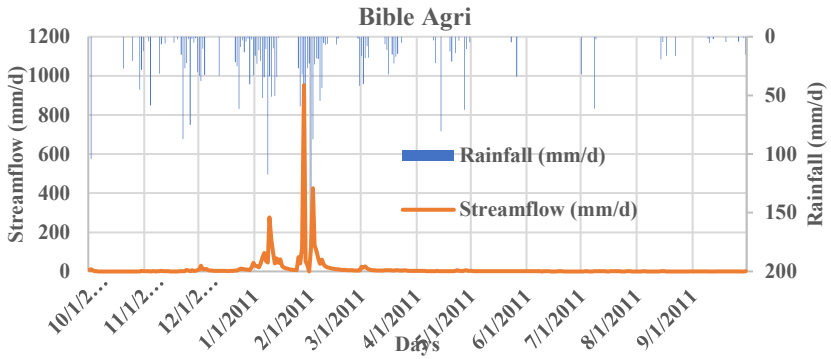

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days
150
100
50
0
0
10
20
30
40
50
60
10/1/2011
11/1/2011
12/1/2011
1/1/2012
2/1/2012
3/1/2012
4/1/2012
5/1/2012
6/1/2012
7/1/2012
8/1/2012
9/1/2012

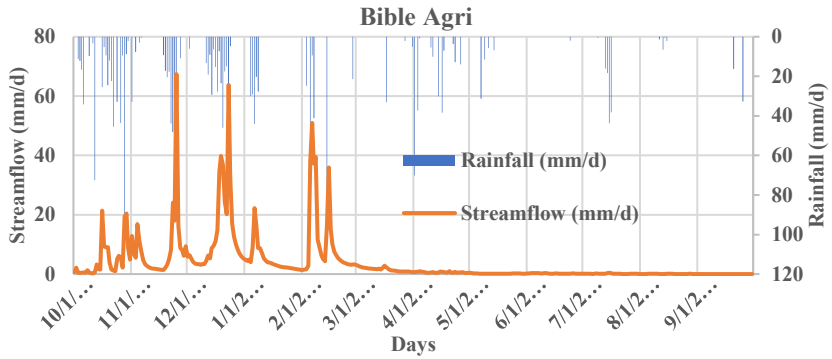

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days

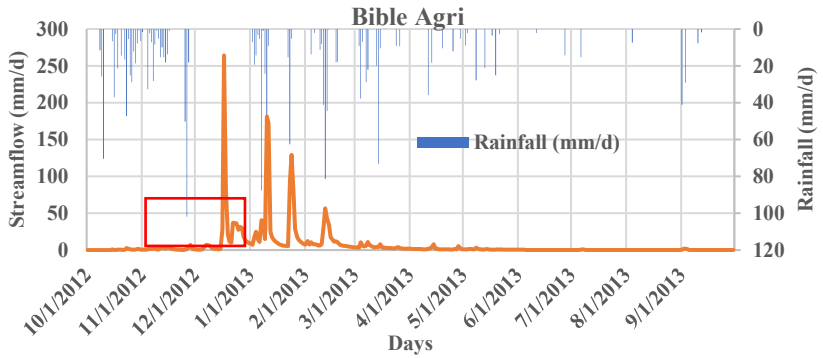

Tissapura
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days

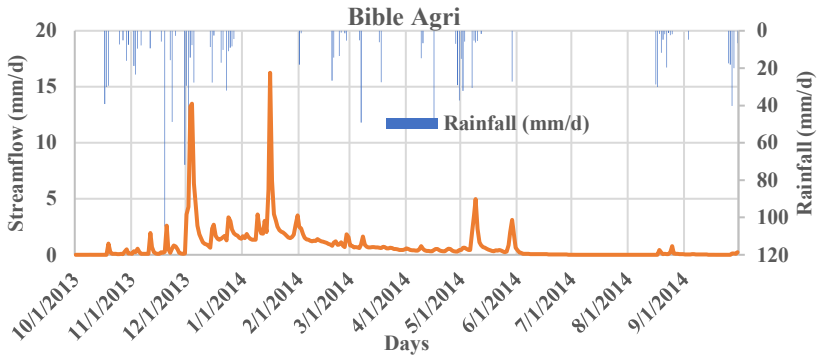

Bilble Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days

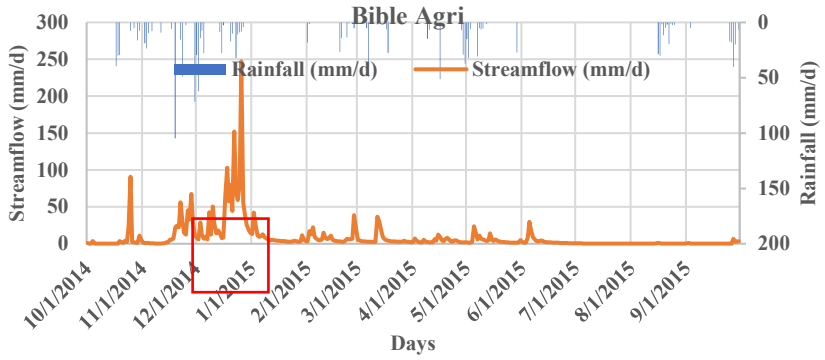

Bible Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days


Bible Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days


Bible Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Streamflow (mm/d)
Days

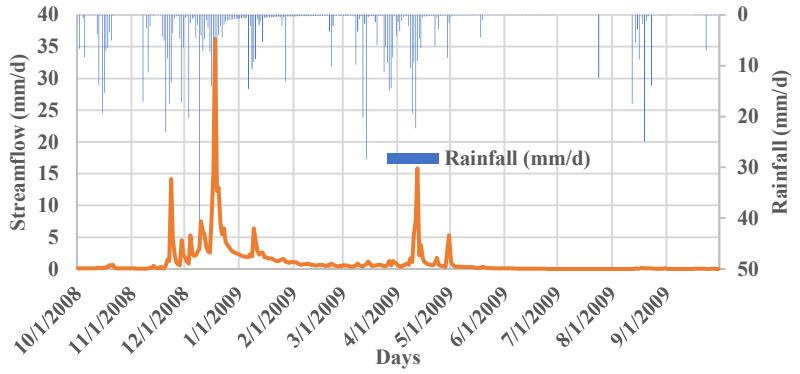

Bible Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Days

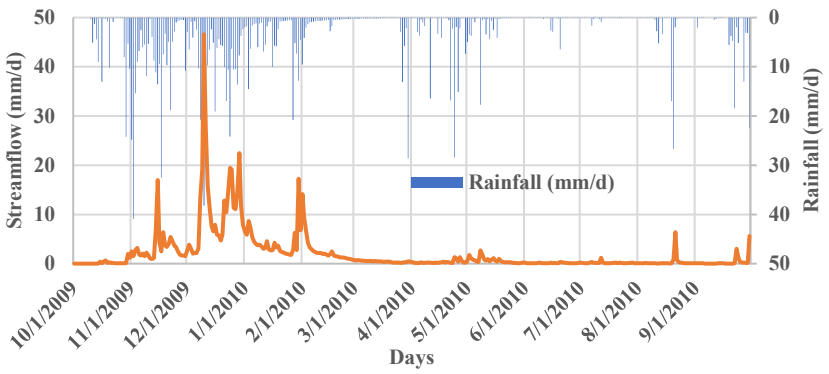

Bible Agri
Streamflow (mm/d)
Rainfall (mm/d)
Rainfall (mm/d)
Days

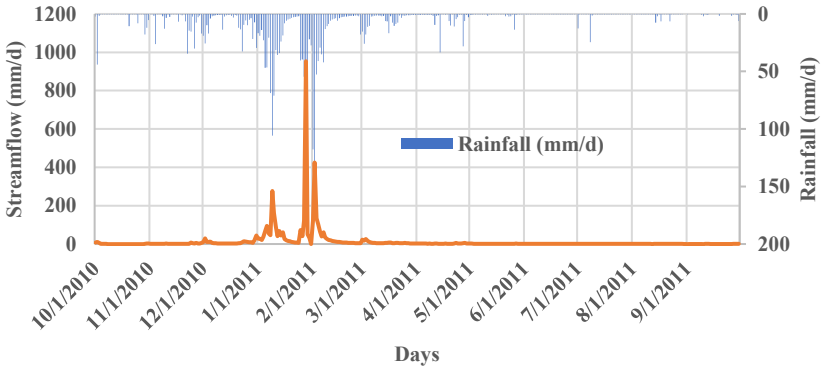


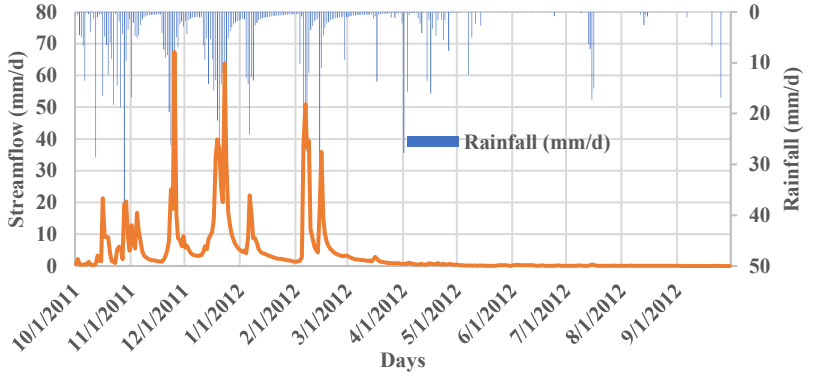
*According to the visual checking, the abnormalities and inconsistencies of the hydrological data are negligible. It was indicated with a red color box.

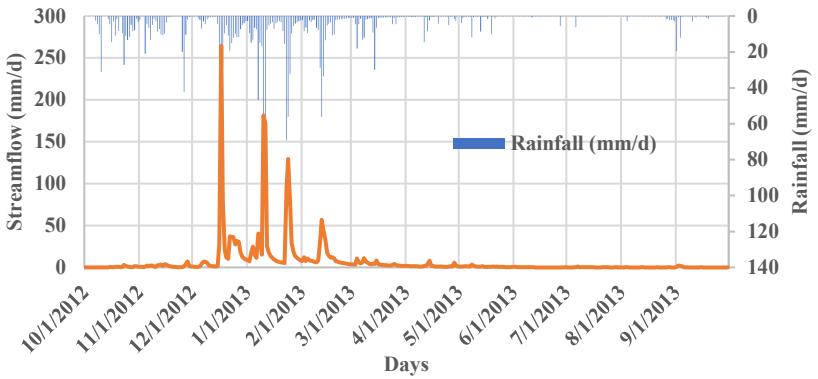
Appendix E: Visual Checking of Streamflow and Thiessen Averaged Rainfall for Water Year (2008-2015) – Maduru River Basin

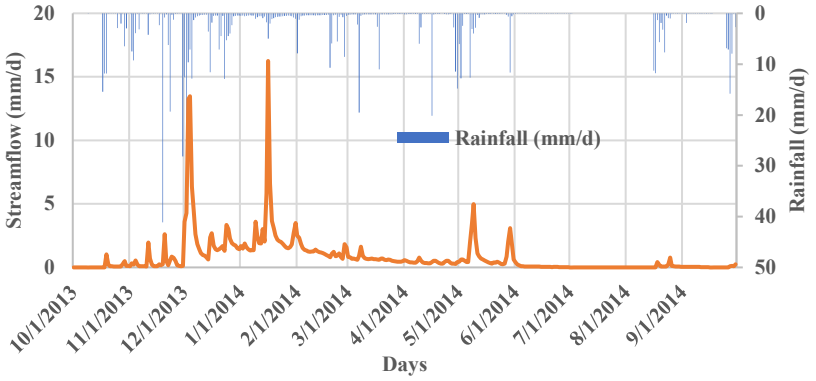


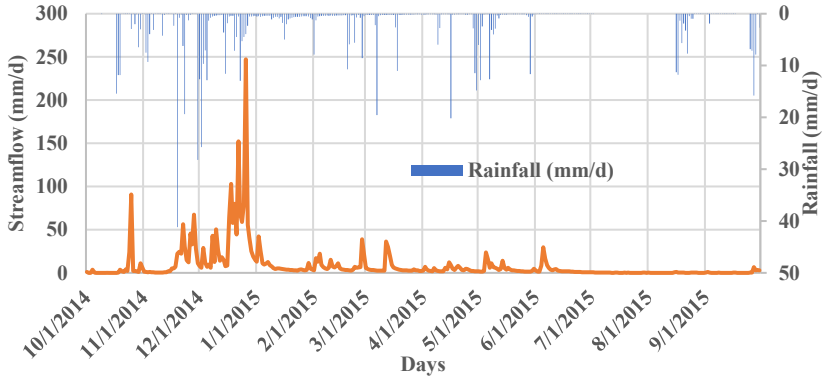












* The response of average rainfall and streamflow shows a good visual appearance.

Appendix F: Double Mass Curve for the Rainfall Stations – Maduru River Basin