Proceedings of the 7th International Conference on Climate Change, Vol. 7, Issue 1, 2023, pp. 56-74
Copyright © 2023 Author(s)
ISSN 2513-258X online
DOI: https://doi.org/10.17501/2513258X.2023.7105

# VARIANTS OF RECURRENT NEURAL NETWORK MODELS FOR REAL-TIME FLOOD FORECASTING IN KELANI RIVER BASIN, SRI LANKA

Subramaniyam C* and Rajapakse RLHL

*Department of Civil Engineering, Faculty of Engineering, University of Moratuwa, Sri Lanka*

**Abstract:** The rapid advancement in computer technology has supported flood forecasting, especially neural networks (NN), an application of data-driven models. However, prediction reliability is compromised due to the data manipulation strategies and the length of the predictive horizon, especially the one-month horizon, which is ample for pre-flood management. Therefore, six (06) variants of recurrent neural networks (RNN) such as Long- and Short-Term Model (LSTM), Gated Recurrent Unit (GRU), Stacked Bidirectional and Unidirectional LSTM (SBU-LSTM), SBU-GRU, Convolution Neural Network LSTM (CNN-LSTM) and CNN-GRU, were developed for the Kelani River Basin to validate their applicability in encouraging the accuracy of monthly flood forecasting by adapting a proper data manipulation technique. Initially, climatic, and physiographic factors of the basin, where the social and economic values are grievously interrupted by frequent floods, were gathered for the study. Then, the hydrological and data science cleansing strategies were adapted to enhance the quality of the data. Besides, a Box-Cox transfomation was implemented to redistribute the hydrological data into a Gaussian form to remove the significant deviation between higher and lower values. Next, grid analysis was conducted using statistical tools to quantify the performance, while the influence of data handling and model architecture was examined using uncertainty and sensitivity analysis. LSTM, GRU, SBU-LSTM, SBU-GRU, CNN-LSTM, and CNN-GRU expressed nearly 81%, 81%, 83%, 83%, 76%, and 62%, respectively, for the coefficient of determination (R2) which measures how well the forecasted values fit with the actual values. SBU-LSTM and SBU-GRU interpreted similar behavior to LSTM and GRU; however, the pattern was different in CNN-LSTM and CNN-GRU. Specifically, simple variants LSTM and GRU provided satisfactory results for the uncertainty and sensitivity analysis categories.

**Keywords:** box-cox, data science, gated recurrent unit, long- and short- term model, statistical tools

## Introduction

Flooding can lead to adverse impacts if development activities intercept the floodplain of a river system; further, the positive correlation between global warming and the frequent occurrence of extreme rainfall resulting in floods has led to several research endeavours. In addition, flood forecasting is elaborated with a hydrograph that illustrates the chronological behaviour of streamflow patterns. Commonly, climatic and physiographic factors of the catchment influence the hydrographs

*Corresponding Authors' Email: *09vikkigod@gmail.com*

in locating the peak flow magnitude and duration. Besides, the most significant achievement in flood forecasting is developing lengthy series with adequate accuracy (Subramanya, 2017).

In the previous days, comprehensive physical models played a significant role in flood forecasting; however, the data-intensive manner and poor quality in handling non-linear problems have deteriorated the sustainability of those models. After, statistical models became popular in the field for developing a future pattern from their associative nature on historical data; nevertheless, these models face struggles to scale the complex data sets. Finally, flood forecasting studies have developed their interest in computer intelligence, where the performance is fast with complex data sets without acknowledging basic processes (Gude, Corns, & Long, 2020).

Forecasting has been developed based on various principles, specifically adapting computer technology; for instance, process-driven and data-driven models are the improved versions of traditional models. Besides, the physical mechanism is crucial for process-driven models, while data-driven models are developed based on machine learning techniques. Even though in recent days, data-driven models have been chosen over process-driven models because the models can handle complex numerical analysis without acknowledging the primary mechanism. Moreover, machine learning is one of the popular applications of data-driven models, suitable for generating non-linear functions to develop relations among hydrological variables. Especially neural network (NN) models and deep machine learning have been in high demand for the application (Xu et al., 2021). In addition, NN models are built with timely information where the deep learning framework supports timely prediction on decomposed components and combines the outcomes to generate the final series (Sha et al., 2021).

According to the NN model performance, highly deviating data enables problematic situations in high accuracy status. Thus, data preprocessing is considered one of the essential tasks before inserting the data for modelling. Besides, feature-wise normalization is proposed to handle the heterogeneous data by re-distributing and re-scaling (Ketkar & Moolayil, 2021). Data cleansing and handling are the most predominant processes supporting neural network models to learn the input easily and quickly with fewer computational losses. In previous studies, principal component analysis was proposed to reduce dimensionality while handling data (Chen et al., 2021). Moreover, Bayesian Regulation strategies were applied to control the probability of complex problems (Di Nunno & Granata, 2020). Thus, the combination of hydrological parameters influences the performance of the NN model, especially without proper data handling strategies (Sha et al., 2021).

On the other hand, NN models adequately address the flood risk by locating the peak flow rates with sufficient temporal resolution (de la Fuente et al., 2019). Timely information gathering and real-time simulation can be expressed through the perfect flood forecasting models. Fundamentally, NN model architecture consists of input, hidden, and output layers; further, the activation function simulates the output from the input based on a complex nonlinear mapping function. Besides, hydrological forecasting problems are classified as short- and medium-term problems. However, the reliability of forecasting declines with the increment of the forecasting horizon (Sha et al., 2021).

The application of recurrent neural networks (RNN) is famous for sequential data types; significantly, image recognition, text translation, and stock prediction have been successfully implemented with RNN. Besides, affine transformation and ease of user memory are the significant features of the RNN models, which allow the model to predict future events based on present and past information. However, the interdependencies of time series data require learning the interconnectivity of multiple time scenarios. Therefore, hybrid models are introduced based on RNN variants to enhance the forecasting quality (Wan et al., 2020). Researchers have paid more interest to RNN and its variants, such as LSTM and GRU, which effectively deal with nonlinear interaction among hydrological models. In addition, multiple input parameters handle the variants and accurately forecast the flood occurrence than the input with rainfall data alone (Zhang et al., 2021). Therefore, the performance of standalone and hybrid RNN models varies based on their learning abilities.

Flood is a frequent natural disaster in Sri Lanka due to its climatological and geographical conditions; thus, flood forecasting is essential to mitigate economic and social vulnerabilities. Further, the frequency increased after 1925 and dramatically affected after 1989. Recently, the country has undergone flood circumstances every two to three years. The Kelani basin is in a wet zone, severely desecrated by the 2018 flood (Manawadu & Wijeratne, 2021). The river begins near the Adams Peak and Kirigalpotta region and reaches the Colombo outfall, nearly draining a 2,300 km$^2$ region accommodating rich biodiversity and natural resources (Kottagoda & Abeysingha, 2017). Therefore, the present study considered Kelani Basin for flood forecasting.

The present study targets to develop a monthly forecasting horizon based on 30-day previous input parameters. In addition, proper data manipulation strategies and sampling techniques are applied to encourage the learning abilities of RNN models. Consequently, the deficiencies and strengths of the standalone and hybrid models are investigated based on the performance in forecasting Kelani River streamflow. Besides, Long - Short Term Model (LSTM) and Gated Recurrent Unit (GRU) are the standalone models; at the same time Convolution (CNN)–LSTM, CNN–GRU, Stacked Bidirectional and Unidirectional based LSTM (SBU-LSTM), and SBU-GRU are the hybrid models considered for the study.

## Material and Methods

Daily rainfall and evaporation data were gathered from the Meteorological Department of Sri Lanka, and daily streamflow data were collected from the Irrigation Department of Sri Lanka. On the other hand, wind speed, temperature, relative humidity, and solar irradiance were derived from NASA's Power Data Access Viewer Website to represent the impactful climatic factor transpiration. The initial losses were represented using the soil wetness index, which was gathered from the same source. Hydrological cleansing methods were employed for rainfall data based on the streamflow-rainfall interaction. In addition, evaporation data was improved by applying data science techniques. The input sets were re-distributed as gaussian using box-cox transformation. Besides, scaling and normalization were implemented to control the computational losses.

RNN-variants such as LSTM and GRU were involved in developing six NN models, including standalone and hybrid models. Python coding proceeded with Jupyter Notebook to compile, fit and test the Kelani Basin data. Besides, grid analysis was conducted with statistical tools such as determination of coefficient (R$^2$), root mean square error (RMSE), mean absolute error (MAE), and flow duration curve error (FDC-Q) to rank the performance of models. In addition, uncertainty and sensitivity analysis were employed to verify the influence of model architecture, such as types of optimizers, the learning rate, and the input handling and sampling size.

A methodology flow chart is attached here in Figure 1 to represent the steps visually.

### *Data Collection and Study Area*

Kelani River Basin is the primary drinking water source of Grater Colombo; in addition, the basin region accommodates industrial and business spots (Kottagoda & Abeysingha, 2017). Figure 2 illustrates the Kelani Basin location and river network. The catchment region is identified with geographical (physiographical) and hydrological (climatological) features (Hussain et al., 2021). Hanwelle river gauge was chosen to collect the streamflow values for seven years from the 2008 water year, while daily rainfall data were gathered for three stations: Hanwella, Pasyala, and Weweltalawa. Besides, daily evapotranspiration data were assembled for Colombo station for the same period. Other climatic factors such as wind speed, temperature, relative humidity, solar irradiance, and soil wetness index were gathered for Colombo with similar temporal resolution.

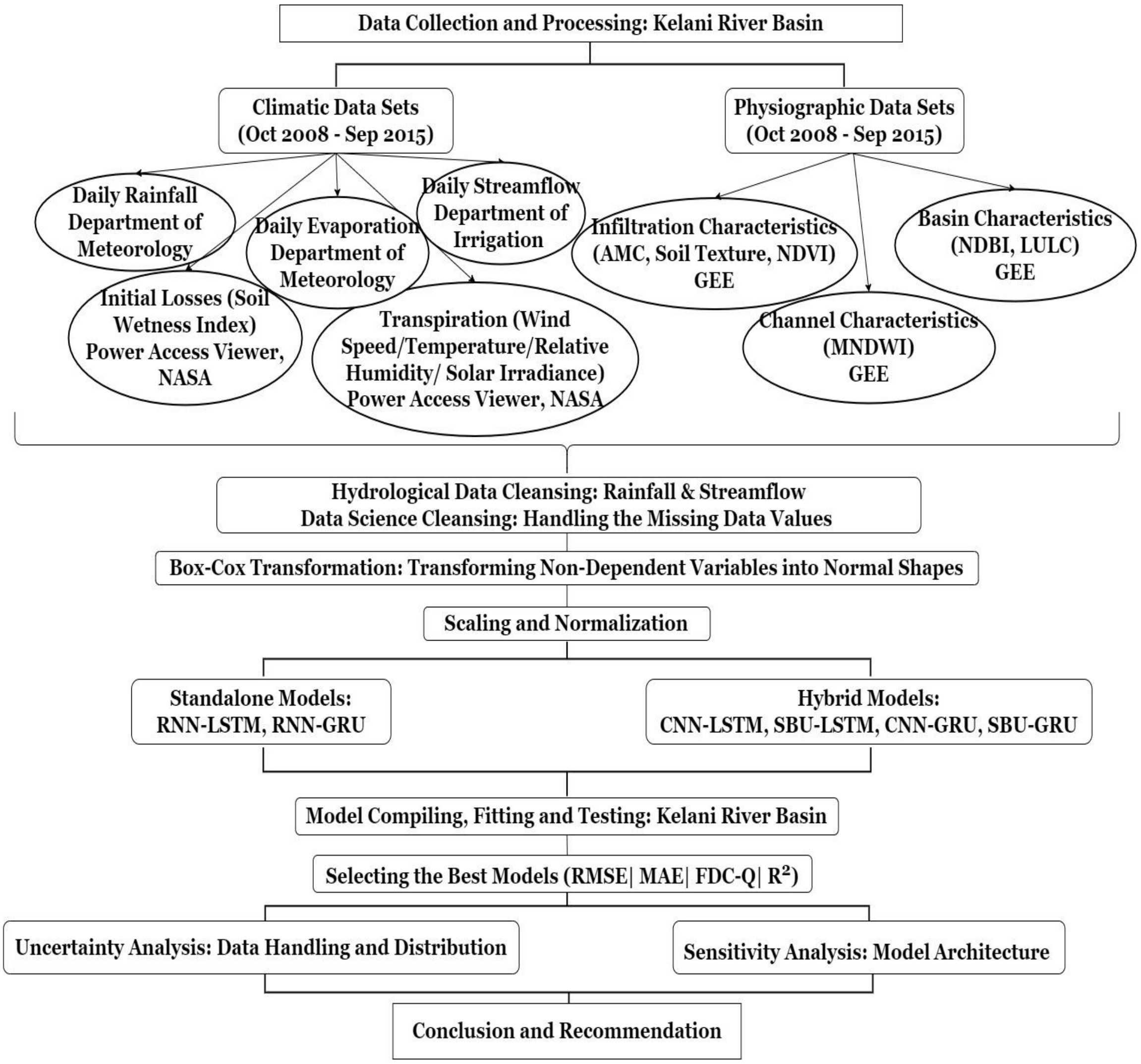
Data Collection and Processing: Kelani River Basin
Climatic Data Sets (Oct 2008 - Sep 2015)
Physiographic Data Sets (Oct 2008 - Sep 2015)
Daily Rainfall Department of Meteorology
Daily Evaporation Department of Meteorology
Daily Streamflow Department of Irrigation
Initial Losses (Soil Wetness Index) Power Access Viewer, NASA
Transpiration (Wind Speed/Temperature/Relative Humidity/ Solar Irradiance) Power Access Viewer, NASA
Infiltration Characteristics (AMC, Soil Texture, NDVI) GEE
Channel Characteristics (MNDWI) GEE
Basin Characteristics (NDBI, LULC) GEE
Hydrological Data Cleansing: Rainfall & Streamflow
Data Science Cleansing: Handling the Missing Data Values
Box-Cox Transformation: Transforming Non-Dependent Variables into Normal Shapes
Scaling and Normalization
Standalone Models: RNN-LSTM, RNN-GRU
Hybrid Models: CNN-LSTM, SBU-LSTM, CNN-GRU, SBU-GRU
Model Compiling, Fitting and Testing: Kelani River Basin
Selecting the Best Models (RMSE| MAE| FDC-Q| $R^2$)
Uncertainty Analysis: Data Handling and Distribution
Sensitivity Analysis: Model Architecture
Conclusion and Recommendation

*Figure 1 Methodology Flowchart*

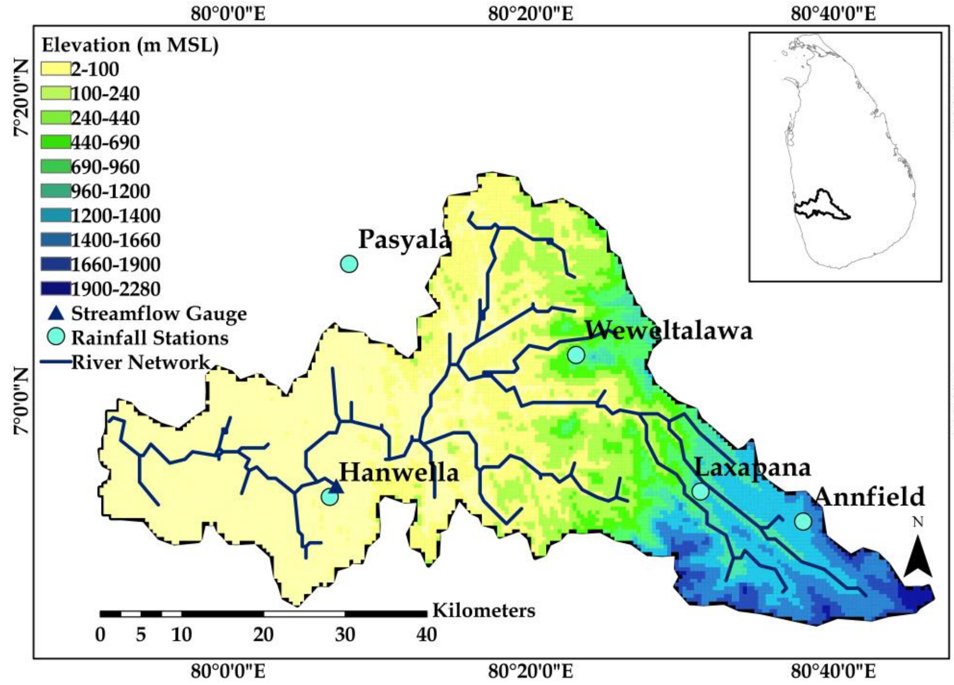

*Figure 2 Location Map of Kelani River Basin (DEM)*

Catchment analysis is the predominant process to extract significant information for hydrological function, where the shape of the catchment, land slope attributes, infiltration characteristics, soil moisture, and vegetation growth are extracted using the method (Vivekanandan, 2019). Besides, Google Earth Engine (GEE) provides a platform for geospatial analysis, where the cloud computing features support to derive the essential information (Nashwan et al., 2019). Soil Conservation Service Curve Number (SCS-CN) is employed to receive runoff coefficient series using rainfall, landcover and soil data (Jain et al., 2021). Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI), and Modified Normalized Difference Water Index (MNDWI) are directly obtained from the Landsat data as time series (Ashok et al., 2021). Table 1 represents the data sources and resolution of data sets.

*Table 1: Data Type and Data Resolution*

| Data Type | 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<br>(Wind Speed, Temperature, Relative Humidity, Solar Irradiance) | Daily | Oct 2008 – Sep 2015 | Power Data Access Viewer Website, NASA |
| Initial Losses<br>(Soil Wetness Index) | Daily | Oct 2008 – Sep 2015 | Power Data Access Viewer Website, NASA |
| Physiographic Data (NDVI, NDWI, MNDWI, and Runoff Coefficient) | Daily | Oct 2008 – Sep 2015 | Google Earth Engine |

### *Data and Data Checking*

Data preprocessing was conducted to improve input quality, which includes missing data handling and correcting the heterogeneity. Hydrological and data science cleanings were employed to elevate the quality of input parameters.

*Data Preprocessing of Rainfall, Streamflow, and Initial Losses:* Visual inspection was initially employed to verify the patterns of rainfall and streamflow at each station annually. In addition, annual water balance estimation was proposed to understand the deviation in data distribution. Besides, according to (Tang et al., 1996), the closest station patching method was involved to fill the missing values by capturing the trend of a single mass curve. Finally, based on (Subramanya, 2017), the double mass curves were developed to ensure the linearity of each station after filling in the missing values.

*Data Preprocessing of Transpiration and Initial Losses:* NASA POWER ACCESS platform is a beneficial source for the regions with unavailable ground weather stations. Most data in the portal are accurately identified, irrespective of the areas (Rodrigues & Braga, 2021). For further processing, the required data were gathered as time series sets for the Colombo region.

*Data Preprocessing of Physiographic Data:* Heterogeneity problems corrupt the radar sets due to instrument configurations, resolution, and band combination. Thus, morphological filters and speckle noise elimination were employed to calibrate the data sets.

*Data Preprocessing with Data Science:* Data division supports eliminating overfitting and ensuring forecasting ability, while data cleaning controls missing values and outliers (Jiang et al., 2021). Data imputation proceeded with data science techniques, especially for evaporation data. In addition, inappropriate data types were reorganized to support NN modelling.

*Data Preprocessing for NN Modelling:* Well-organized TensorFlow library is utilized to run the Python coding in Jupyter Notebook, which facilitates text editing with annotation to break a long script into smaller portions. Forecasting problems are categorized as regression machine learning programs, where pre-processing is essential to control accuracy deficiencies (Ketkar & Moolayil, 2021). Therefore, Box-Cox transformation, a one-dimensional conversion technique, is widely proposed to enhance the Gaussian data distribution; however, the application is only viable for some cases (Blum et al., 2022). Besides, feature-based normalization is appreciable to control the computational losses in the numerical data sets, where the values are re-distributed between 0 and 1 (Géron, 2019). The box-cox transformation and data-scaling were applied to ensure the learning capacity of machine learning techniques. In addition, recent hydrological data are assembled to generate batches and targets using a python generator, enhancing simulation reliability. Equation 1 illustrates the Box-Cox transformation, while Equation 2 explains the feature-scaling technique.

*Equation 1*

For $\lambda \neq 0$, $y_i = \frac{y_i^{\lambda}-1}{\lambda}$; For $\lambda = 0$, $y_i = \ln y_i$

*Equation 2*

Normalized Value = $\frac{value-mean}{standard\,deviation}$

### *Developing NN Models*

Model architecture is developed with optimizer Adam, loss function Root Mean Squared Error (RMSE), and monitoring metrics Mean Absolute Error (MAE) (Géron, 2019). Figure 3 represents the skeleton of the NN model, and Equation 3 explains the general formula behind NN modelling. In addition, consecutive windows without shuffles are allocated with input, target, and shift (Hassan & Hassan, 2021). The split window function was arranged with two sets: a batch of 30-time step 16-features ad a batch of 30-time step 1-feature. Further, the entire set was divided into three groups: training, validation, and testing by 70%, 20%, and 10%, respectively.

$y' = f\ (W^TX + b)$

y': Predicted Values; X: Input Values; W: Weight Values; b: Bias Values;
f: Activation Function

*Equation 3*

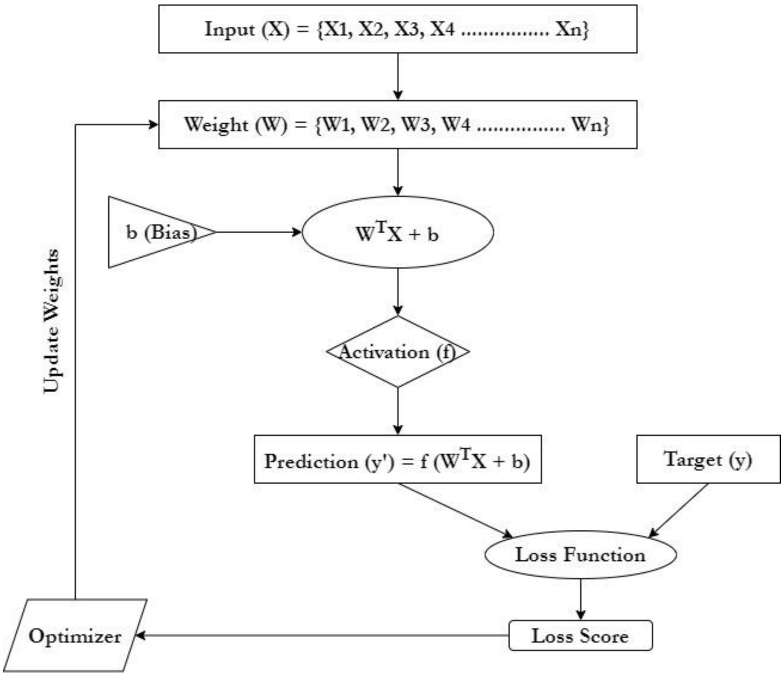
Input (X) = {X1, X2, X3, X4 ................ Xn}
Weight (W) = {W1, W2, W3, W4 ................ Wn}
b (Bias)
W^TX + b
Activation (f)
Prediction (y') = f (W^TX + b)
Target (y)
Loss Function
Loss Score
Optimizer
Update Weights

*Figure 3 Skeleton of NN Model*

*LSTM and GRU*: RNN-variants such as LSTM and GRU were developed as standalone models for flood forecasting. The LSTM model was initially proposed to control the vanishing gradient issues by Hochreiter and Schmidhuber in 1997. In 2014, Chung and team introduced GRU to run the model cheaply with low symbolic power. In addition, the dropout function is applied to overcome the overfitting issues, initiated in 2015 by Yarin Gal (Ketkar & Moolayil, 2021). A Keras framework was generated with in-built modules of LSTM and GRU to develop the model architecture. Besides, the return sequence argument was allocated as True to generate output for each input state.

*CNN-LSTM and CNN-GRU:* CNN-LSTM or CNN-GRU learns the temporal and spatial features of the sequential input sets, which integrates two standalone models, such as CNN and LSTM or CNN and GRU. The process begins with CNN, which understands the spatial features of the input. Consequently, the gathered patterns are converted into one-dimensional elements to feed the LSTM unit (Shen & Lin, 2020). According to the model architecture, data sets were passed through the units after extracting the sequential features by inserting them into a Fully Connected Dense Layer (FCN). A Conv1D layer with a kernel size of five was proposed as the first layer. Then, LSTM or GRU layer with a return sequence was allocated, a dense layer with ReLu activation followed the previous layer, and the dense layer was proposed at the bottom.

*SBU-LSTM and SBU-GRU:* The architecture obeys the feed-forward theories, where the previous unit of stacked LSTM/GRU feeds the following LSTM/GRU unit. BiLSTM/BiGRU captures the backward and forward dependencies of the data; thus, the model understands the spatial and temporal characteristics (Cui et al., 2020). A bidirectional layer with LSTM or GRU with a return state was proposed; consequently, deep LSTM or GRU was attached to form a fully connected layer to boost the model.

### *Ranking the Model Performance*

Four statistical tools, such as Residual Mean Maximum (RM), Flow Duration Curve Behavioral Error (FDC-Q), Determination of Coefficient ($R^2$), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE), are considered for ranking the model performance (Jiang et al., 2021). Analytic Hierarchy Process (AHP) was proposed to identify the weights of decision-making factors. Consequently, grid analysis was applied to rank the performance, where higher rank values were assigned for the worst performance. Table 2 illustrates the importance of statistical tools and formulas.

*Table 2: Weightages of the Statistical Tools for Grid Analysis*

| Equations | Range | Reason for Influence on Decision - Making | Weights |
|---|---|---|---|
| $R^2$ = NSE<br>$= 1 - \frac{\sum_{i=1}^{n}(O_i - E_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$ | [0, 1] | 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. | 0.44 |
| RM = max $\lvert(O_i - E_i)\rvert$ | [0, ∞) | It measures the scattered deviation of predicted values from the observed values. Therefore, it was assigned the second highest value. | 0.22 |
| $R_{FDC} = 1 - \frac{\sum_{1}^{N-1} S_i}{N-1}$ | [0, 1] | This tool expresses the behavioral error of observed and forecasted FDC. FDC greatly explicates the catchment characteristics. Therefore, it was assigned a fair value. | 0.15 |
| MAE = $\frac{\sum_{i=1}^{n}\lvert(O_i - E_i)\rvert}{n}$ | [0, ∞) | It was the monitoring metrics of the NN models. Therefore, it was assigned a low value. | 0.11 |
| RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(O_i - E_i)^2}{n}}$ | [0, ∞) | It was the loss function considered for the model compiling and fitting. Therefore, it was considered the lowest value. | 0.09 |

### *Uncertainty and Sensitivity Analysis*

Uncertainty analysis is a perfect tool to investigate the reliability of NN forecasting models, where $R^2$ values are implemented to express the performance based on the core model performance. In addition, data-driven forecasting models deal with various uncertainty sources, such as input variables and sampling sizes (Shamshirband et al., 2019). Sensitivity analysis assesses the importance of model architecture, which contributes to the functionality of NN models. Similar to uncertainty analysis, the sensitivity analysis of the model is expressed as a fraction of the core model performance using $R^2$ values (Song et al., 2020). Thus, learning rates, optimizer types, input parameters, and lead time sampling were considered to express the uncertainty and sensitivity analysis. Equation 4 illustrates the way to compute the scaled values. Table 3 represents the interpretation of scaled values regarding model performance. And Table 4 describes the uncertainty and sensitivity parameters.

$$Performance\,Measure = \frac{\text{Model's R}-\text{Sq}}{\text{Core Model's R}-\text{Sq}} \qquad \textit{Equation 4}$$

*Table 1 Scaled Values and the Interpretations*

| Scaled Values (x) | Decision Based on Values |
|---|---|
| X < 1 | The newly developed model performs worse than the core model |
| X = 1 | The newly developed model performs equal to the core model |
| X > 1 | The newly developed model performs better than the core model |

*Table 4: Uncertainty and Sensitivity Parameters*

| Sensitivity Parameters | Model 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 | |

## Results and Discussion

Both hydrographs and FDCs were plotted for 2015 streamflow values, where R2 and FDC-Q explicated the forecasting quality. Figure 4 to Figure 15 illustrate the performance of flood forecasting models. Initially, the performance of RNN variants such as LSTM and GRU was inspected as standalone models. Consequently, the hybrid models incorporated by CNN and RNN models were examined for their prediction quality.

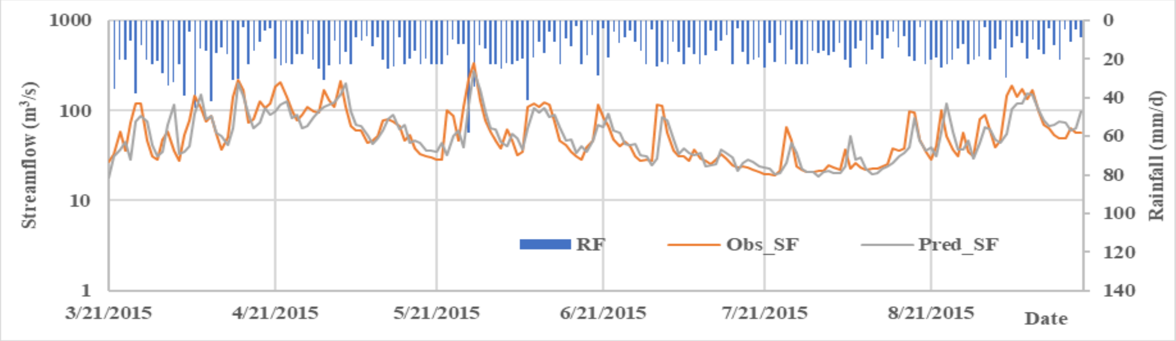

*Figure 4: Hydrograph for Kelani (LSTM)*

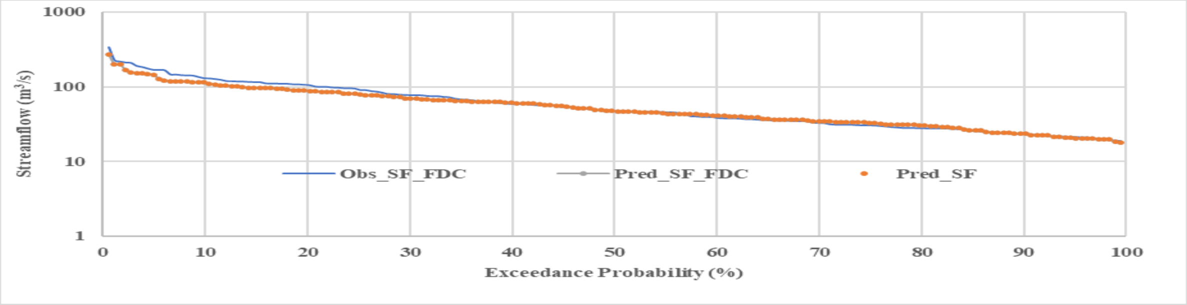

*Figure 5: FDC for Kelani (LSTM)*

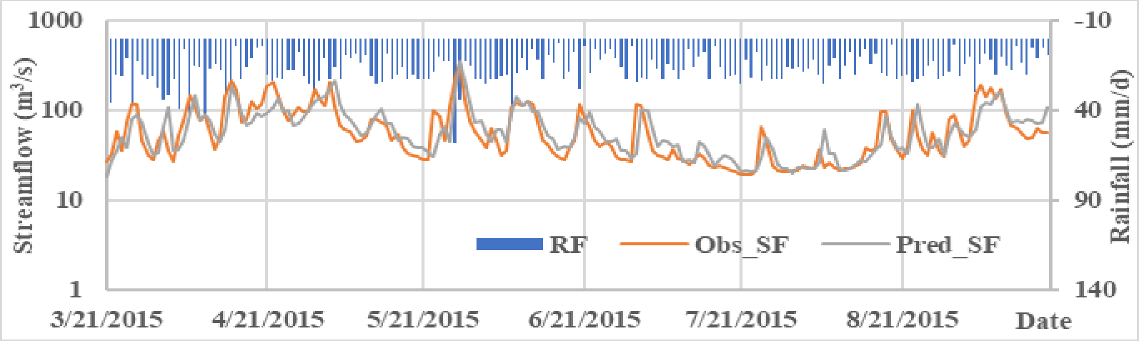

*Figure 6: Hydrograph for Kelani (GRU)*

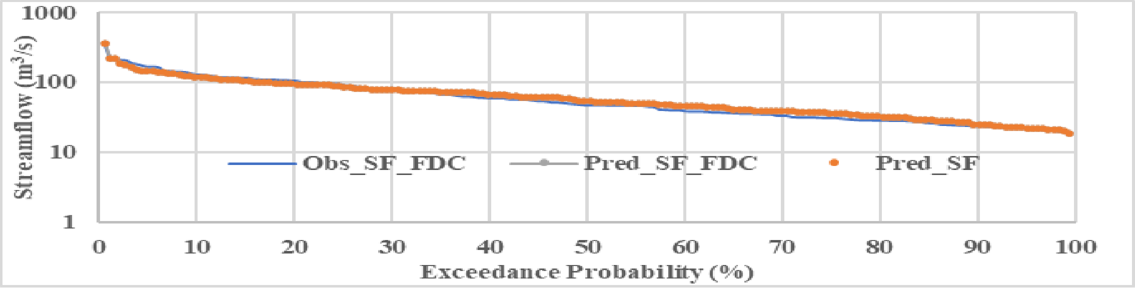

*Figure 7: Hydrograph for Kelani (GRU)*

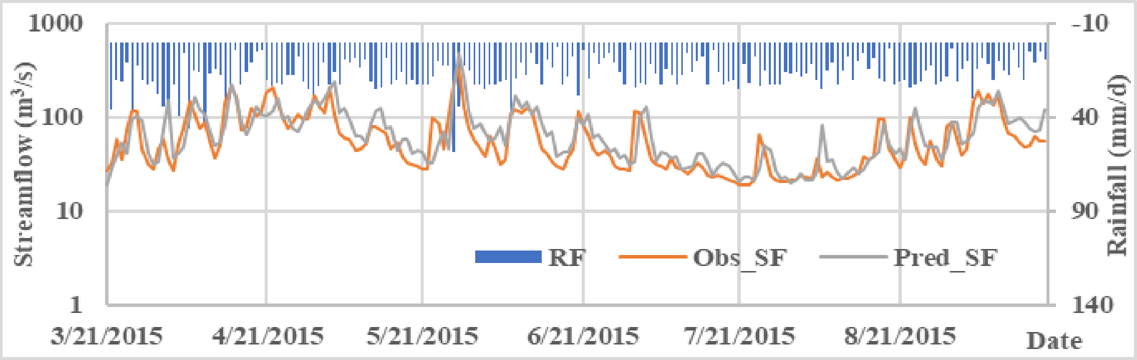

*Figure 8: Hydrograph for Kelani (CNN-LSTM)*

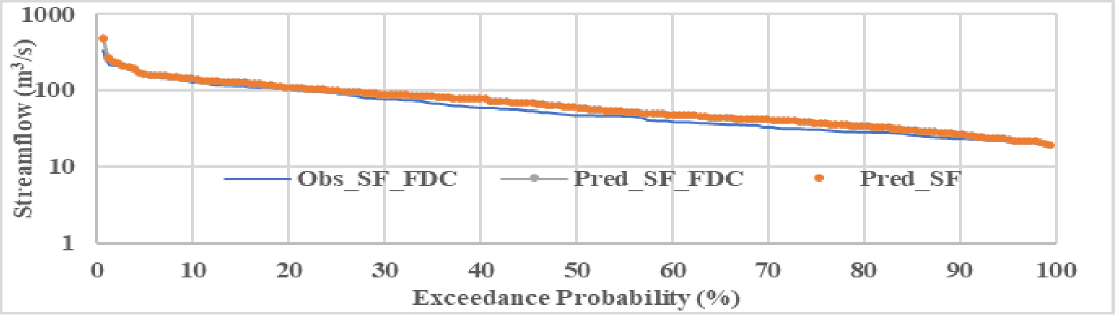

*Figure 9: Hydrograph for Kelani (CNN-LSTM)*

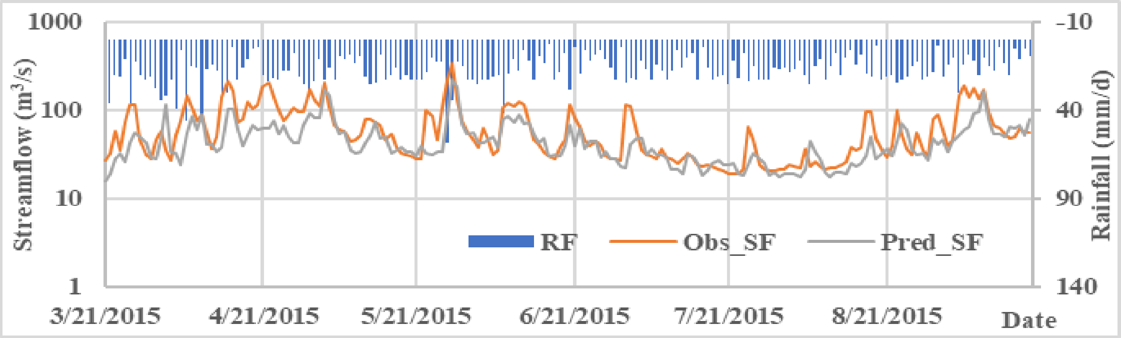

*Figure 10: Hydrograph for Kelani (CNN-GRU)*

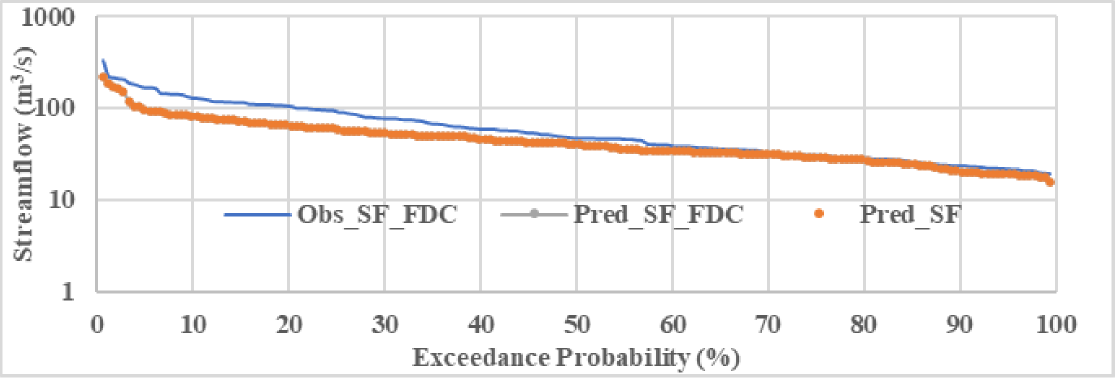

*Figure 11: Hydrograph for Kelani (CNN-GRU)*

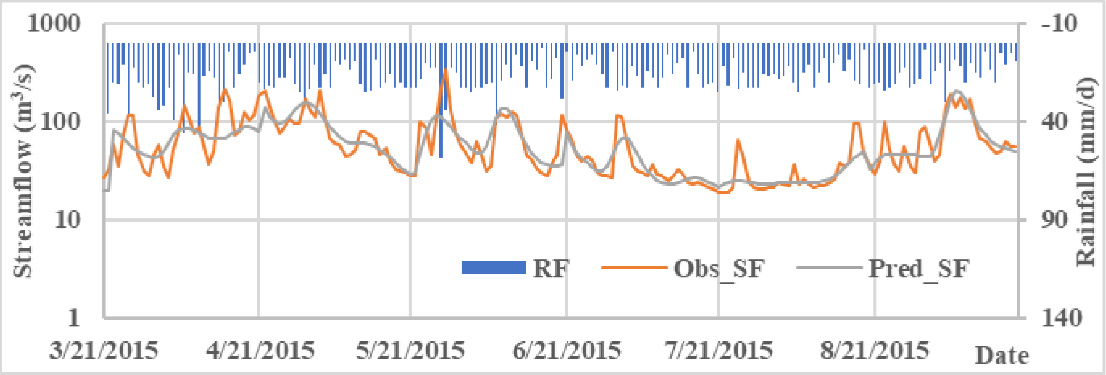

*Figure 12: Hydrograph for Kelani (SBU-LSTM)*

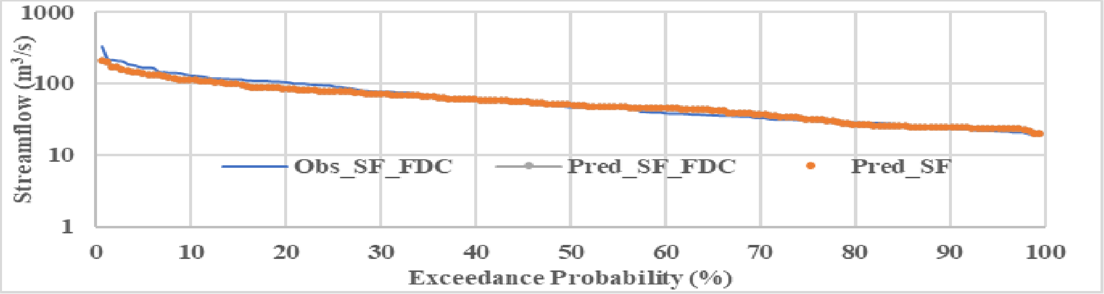

*Figure 13: Hydrograph for Kelani (SBU-LSTM)*

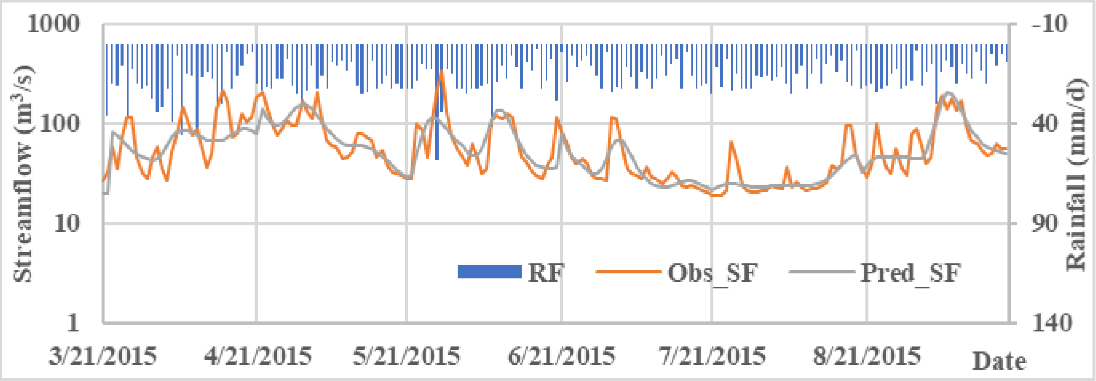

*Figure 14 Hydrograph for Kelani (SBU-GRU)*

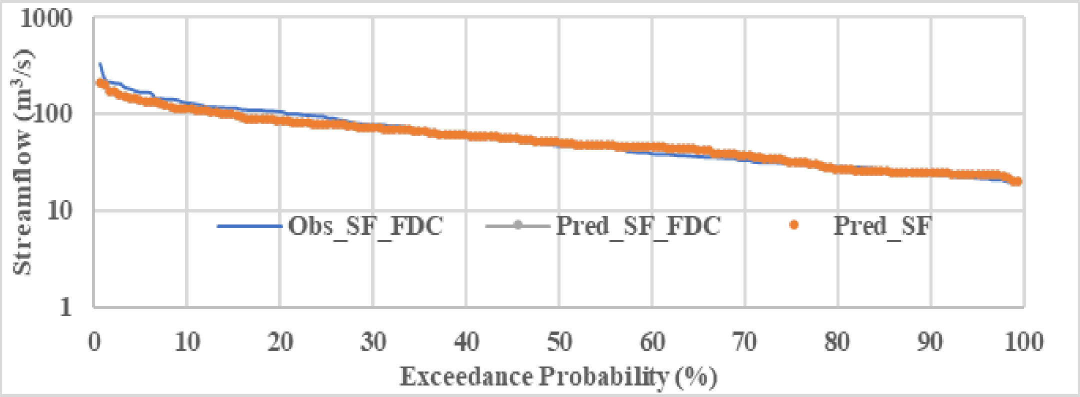

*Figure 15 Hydrograph for Kelani (SBU-GRU)*

According to the standalone models, the LSTM model perfectly forecasted intermediate and low flow rates. Besides, GRU predicted high, intermediate, and low streamflow values with adequate quality. The FDC-Q was identified as 0.98 for both models. The overall RM values were within the acceptable range of 0.24 and 0.40 for LSTM and GRU, respectively. Simultaneously, the linearity coefficient was 81% for both standalone models.

Based on the CNN-based hybrid models, CNN-LSTM poorly predicted low flow rates; however, the model perfectly forecasted high flow rates. On the other hand, CNN-GRU explicated the opposite behavior in predicting streamflow values. Therefore, FDC-Q values were 0.99 for both models. The overall RM was 0.38 and 0.37 for CNN-LSTM and CNN-GRU, respectively. Besides, the linearity coefficient was 76% and 62% for CNN-LSTM and CNN-GRU, respectively.

***Ranking the Model Performance***

Both training and testing set results were considered to develop the grid analysis to rank the model performance, where the AHP analysis was developed to derive the weightage values. Table 5 illustrates the performance of each model in every category.

*Table 5: Performance of NN Models for Statistical Tools*

| NN Models | | LSTM | GRU | CNN-LSTM | CNN-GRU | SBU- LSTM | SBU-GRU |
|---|---|---|---|---|---|---|---|
| $R^2$ | Training | 0.8987 | 0.8982 | 0.8556 | 0.8299 | 0.8827 | 0.8811 |
| | Rank | 1 | 2 | 5 | 6 | 3 | 4 |
| | Testing | 0.6915 | 0.7157 | 0.6791 | 0.3351 | 0.7482 | 0.7148 |
| | Rank | 4 | 2 | 5 | 6 | 1 | 3 |
| RM | Training | 0.2157 | 0.5212 | 0.5126 | 0.4058 | 0.4420 | 0.5093 |
| | Rank | 1 | 6 | 5 | 2 | 3 | 4 |
| | Testing | 0.2645 | 0.2780 | 0.2560 | 0.3374 | 0.2487 | 0.2565 |
| | Rank | 4 | 5 | 2 | 6 | 1 | 3 |
| MAE | Training | 0.0328 | 0.0356 | 0.0453 | 0.0481 | 0.0354 | 0.0388 |
| | Rank | 1 | 3 | 5 | 6 | 2 | 4 |
| | Testing | 0.0399 | 0.0409 | 0.0435 | 0.0609 | 0.0380 | 0.0398 |
| | Rank | 3 | 4 | 5 | 6 | 1 | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| RMSE | Training | 0.0459 | 0.0537 | 0.0587 | 0.0637 | 0.0500 | 0.0548 |
| | Rank | 1 | 3 | 5 | 6 | 2 | 4 |
| | Testing | 0.0549 | 0.0555 | 0.0572 | 0.0806 | 0.0503 | 0.0535 |
| | Rank | 3 | 4 | 5 | 6 | 1 | 2 |

Except for CNN-GRU, other models expressed a perfect linearity correlation between actual and predicted values, where the R$^2$ was above 0.75. GRU explicated higher residual error among the models; however, the value was acceptable because it was lesser than 0.5. Besides, the LSTM model expressed a lower value; other models were identified with similar performance. All the models behaved outstandingly for MAE and RMSE, where the values were less than and equal to 0.05. Table 6 expresses the grid analysis of the models.

According to the grid analysis, SBU-LSTM and LSTM performed extraordinarily among the models. However, CNN-based models such as CNN-LSTM and CNN-GRU expressed poor performance among the other models. Besides, GRU and SBU-GRU explicated intermediate performance.

*Table 6: Grid Analysis of NN Models*

| Weight and Tools | | LSTM | GRU | CNN-LSTM | CNN-GRU | SBU-LSTM | SBU-GRU |
|---|---|---|---|---|---|---|---|
| 0.4 | R$^2$ | 2.5 | 2.0 | 5.0 | 6.0 | 2.0 | 3.5 |
| 0.2 | RM | 2.5 | 5.5 | 3.5 | 4.0 | 2.0 | 3.5 |
| 0.1 | FDC-Q | 6.0 | 5.0 | 1.0 | 2.0 | 3.0 | 3.0 |
| 0.1 | MAE | 2.0 | 3.5 | 5.0 | 6.0 | 1.5 | 3.0 |
| 0.1 | RMSE | 2.0 | 3.5 | 5.0 | 6.0 | 1.5 | 3.0 |
| Weighted Ranks | | 2.9 | 3.5 | 4.1 | 5.0 | 2.0 | 3.3 |
| Ranks | | **2** | **4** | **5** | **6** | **1** | **3** |

### *Uncertainty and Sensitivity Analysis*

By adjusting the model architecture (learning rates, optimizers), lead time and input parameters, the newly prepared models were examined based on the core model performance; specifically, R$^2$ was applied to perform the sensitivity analysis. Figure 16 illustrates the scaled values of models in each parameter.

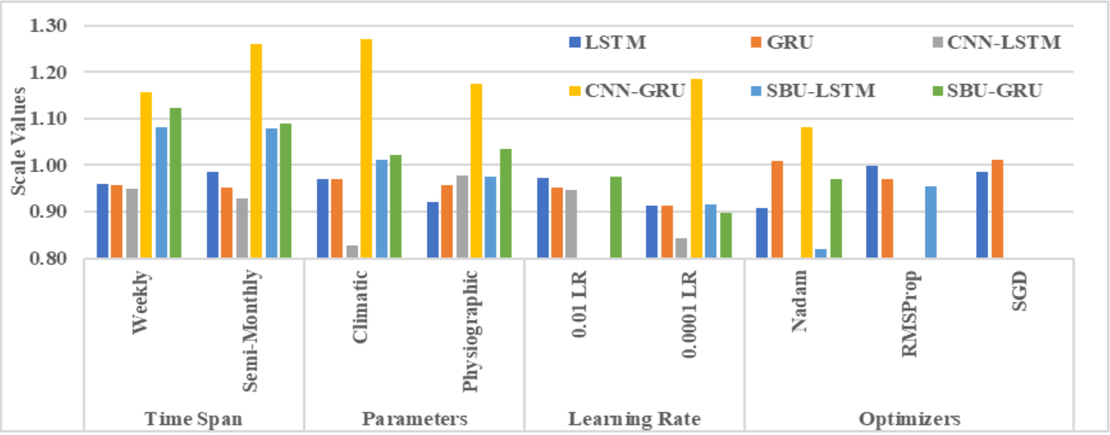

*Figure 16: Uncertainty and Sensitivity Analysis of NN Models*

Standalone models such as LSTM and GRU delivered reasonably good performance for all the sensitivity parameters, where the scale values were within 0.9 to 1.0. Besides, CNN-GRU expressed more extraordinary performance for semi-monthly and weekly spans than monthly ones. Thus, the performance based on span length was inconclusive; further, other models expounded inconclusive conclusions regarding the time span.

Except for CNN-LSTM, other models were perfectly performed forecasting for the adjusted input parameters, where the value falls between 0.9 to 1.3. CNN-GRU and SBU-LSTM expressed worse performance for a learning rate of 0.01 than the core model. Moreover, except CNN-LSTM, other models delivered a better performance for the learning rate lesser than the core model one.

Only SBU-LSTM and CNN-LSTM exposed poor performance for NAdam; except LSTM and GRU, none of the models expressed adequate performance for the models with SGD. Further, RMSProp supported the model performance of LSTM, GRU, and SBU-LSTM.

### *Comparing the Present Models' Performance with the Available Models*

The present study models were compared with the RNN-based models available in the literature. However, the type of input parameters and the forecasting horizon varied from the present study. Most of the available models performed more exceptionally than the present models; nevertheless, the present models adequately maintained the linearity correlation between observed and forecasted values.

The LSTM model was available in the literature to forecast a 2-day period with daily streamflow from 1995 to 2013 (Le et al., 2021). In addition, the GRU model in the literature offered an excellent performance than the GRU model of the present study. However, it predicted a one-day span using daily rainfall and runoff data for eight years from 2007 to 2014 (Wang et al., 2020). The models performed more outstandingly than the LSTM and GRU models available in the present study. However, the forecasting horizon of the present study was 30-day; further, the study was arranged to forecast streamflow by considering climatic and physiographic parameters.

The stacked LSTM (STA-LSTM) was modelled to predict 6-hour streamflow using hourly flow rate and rainfall from 1981 to 2007 (Ding et al., 2020). On the other hand, the feed-forward LSTM (FF-LSTM) forecasted 1 hour with daily streamflow and rainfall from 1980 to 2016 (Lin et al., 2021). SBU-LSTM and SBU-GRU in the present model expressed similar behavior to STA-LSTM and FF-LSTM based on their model architecture. However, the prediction capacity was quietly better than the present models. By reducing the forecasting span to either 15-day or 7-day, the present study models expressed similar abilities like STA-LSTM and FF-LSTM.

The CNN-LSTM model was available in the literature to forecast one-month streamflow with hourly streamflow data from 1996 to 2016 (Ghimire et al., 2021). The forecasting ability of CNN-based models in the present study could have been better than the model available in the literature. Even adjusting the input parameters and period was never supported to enhance the performance of the present study models. Figure 17 illustrates the graphical representation of the model performances.

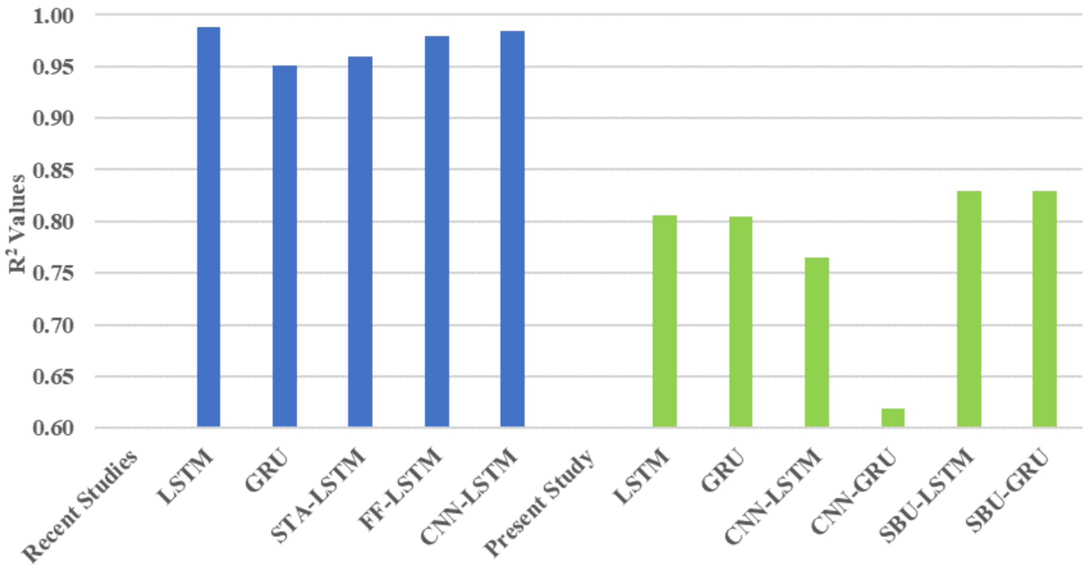

*Figure 17:Comparison of Present and Available NN Forecasting Models*

## Conclusions and Recommendations

The Box-Cox transformation supported the development of Gaussian distribution on the input set, where the extreme events were forced not to corrupt the computation. Further, the normalization encouraged computational effectiveness. Besides, sliding window sampling motivated the periodical chopping of hydrological data.

The CNN-based models performed poorly than other models, especially since CNN-GRU was identified with a lower value than others. Besides, among the other two sets, GRU-based models expressed underperformance than LSTM-based models. Longer sequences are poorly processed with GRU due to the less memory capacity. Therefore, GRU poorly performed forecasting than LSTM. Most significantly, both standalone models explicated better performance than the CNN-based models.

Hybrid models SBU-LSTM and SBU-GRU were designed with unidirectional and bidirectional units. Further, the model forecasting ability was similar to standalone models. Moreover, the stacked behavior of the NN represented the feed-forward nature. However, SBU-GRU explicated underperformance than SBU-LSTM.

### *Limitations in the Present Study*

The GEE contributes to performing geospatial analysis using satellite products; however, the missing data handling is required for the platform (Shelestov et al., 2017). Therefore, in the present study, the excellency of the NN model supported data imputation. In addition, the cloud system provides limited training and validation samples in large-scale data collection. Therefore, GEE must be incorporated with a management decision to ensure derived data quality (Zhao et al., 2021). There are no defined ways for input handling while modelling the NN; simultaneously, there is neither an empirical nor theoretical tool to select the best NN models (Andrea Sánchez-Sánchez et al., 2020). Therefore, the present study was allocated with grid and AHP analysis to rank the model performance.

### *Recommendations for Future Studies*

The loss function was RMSE for the present study, which optimized the error propagation by comparing actual and predicted values. In future studies, the loss function must be replaced with $R^2$ to check forecasting quality. Autocorrelation correlates with the interdisciplinary of errors, which is more applicable to the temporal nature of error propagation (Sun et al., 2021). Therefore, a future study must be developed with an autocorrelation technique to control the scattering of forecasted values. Further, the present study elaborated on the integrated models of unidirectional and bidirectional RNN variants. However, no models were developed on bidirectional RNNs alone. Therefore, the bidirectional models must be comprehensively studied similarly in the upcoming studies.

## Acknowledgments

I am genuinely pleased to express my sincere thanks to the Department of Civil Engineering at the University of Moratuwa for providing the research opportunity. A debt of gratitude is owed to the Department of Meteorology and Department of Irrigation, Sri Lanka, for offering adequate data sets. I especially thank my friends, Mr N. Kiruthihan and Mr P. Loghi, for their valuable guidance on online courses and materials.

## References

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

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

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

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

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

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

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

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

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

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

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

Shen, H., & Lin, J. (2020). Investigation of crowd shipping 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

Tang, W. Y., Kassim, A. H. M., & Abubakar, S. H. (1996). Comparative studies of various missing data treatment methods - Malaysian experience. Atmospheric Research, 42(1–4), 247–262. https://doi.org/10.1016/0169-8095(95)00067-4

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 correlated time series prediction. Knowledge-Based Systems, 191(xxxx), 105239. https://doi.org/10.1016/j.knosys.2019.105239

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

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