Environmental Engineering and Management Journal

February 2024, Vol. 23, No. 2, 287-300
http://www.eemj.icpm.tuiasi.ro/; http://www.eemj.eu



"Gheorghe Asachi" Technical University of Iasi, Romania



COMPARISON OF LONG SHORT-TERM MEMORY NETWORKS FOR DAILY AHEAD-FLOW PREDICTIONS ON A RIVER

Ekin Ekinci[1], Önder Ekinci[2], Arzu Morkoyunlu[3]*

[1]Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, 54187, Sakarya, Turkey
[2]Kocaeli University, Hereke Asım Kocabıyık Vocational School, Department of Construction Technology, 41800 Kocaeli, Turkey
[3]Kocaeli University, Hereke Asım Kocabıyık Vocational School, Department of Environment and Cleaning, 41800 Kocaeli, Turkey

PROOF

Abstract

Prediction of peak river flows is an important problem for the design, management, and safety of hydraulic structures. The aim of the present study was to investigate the prediction capabilities of Long Short-Term Memory (LSTM) networks in predicting daily river flows of the Lüleburgaz stream-flow gauging station (SGS) for the observation period of 1987-2015. The Lüleburgaz SGS is on the Ergene River in the Meriç Ergene Basin in Turkey. Daily measured values of river flow at times seven-days (T-6 to T), two-days (T-1 to T), and one-day (T) were used as input to predict the univariate output of one day ahead flow for an acceptable estimation pattern. The dataset was divided into 75% for the training set and 25% for the test set, and then LSTM, Stacked LSTM, Bidirectional LSTM (BiLSTM), Convolutional LSTM (Conv-LSTM) and Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) were implemented. The river flow predictions were evaluated by using root mean square error (RMSE), mean absolute error (MAE), regression accuracy (R^2) and Nash–Sutcliffe efficiency (NSE). The calculated performance metrics during the test phase showed that, predictions using only one-day of historical data provided the highest performance compared with two-days and seven-days. It is concluded that the river flow predictions would be accurate, especially when using BiLSTM as this was superior to the other models with RMSE of 20.76 m^3/s and MAE of 5.33 m^3/s. Our comparisons show that LSTM networks can be a viable alternative for estimating future river flows.

Key words: Deep learning, Long Short-Term Memory networks, river flow, prediction

Received: September, 2023; Revised final: November, 2023; Accepted: December, 2023; Published in final edited form: February, 2024

1. Introduction

Forecasting peak river flows is decisive for engineering design to mitigate or prevent flood damage to property and lives. One of the primary challenges in hydrology is predicting future peak flows in rivers, which is essential for the design and operation of water structures such as dams. In recent years, computational deep learning models, known as black box or data-driven models, have gained popularity for forecasting stream flows in rivers, surpassing physically based models (Ebrahimi and Shourian, 2020; Karim et al., 2023; Mehedi et al., 2022; Sharma and Singh, 2021; Young et al., 2017). Deep learning offers significant improvements over traditional neural networks by automatically extracting features and handling complex data distributions (Shen, 2018). However, the reliability of deep learning algorithms can be limited due to insufficient historical data, such as accurate flow records of the river.

Many deep learning algorithms have shown promising results in tackling real-world problems (Bhandari et al., 2023; Sarker, 2021; Wang et al., 2023a). Recurrent Neural Network (RNN), in particular, has a notable capacity for learning from time series data (Alsavaf and Teksoy, 2022; Apaydın et al., 2022; Khaldi et al., 2023; Yılmaz et al., 2022),

* Author to whom all correspondence should be addressed: e-mail: arzu.yuce@kocaeli.edu.tr; Phone: +902625113715; Fax: +902625113705

as it incorporates loops that allow the transmission of information from one time step to the next. However, RNN quickly loses its ability to model long-distance information due to the vanishing or exploding gradient problem (Wang et al., 2022). To address this issue and capture long-term dependencies in the data, Long Short-Term Memory (LSTM) networks are employed.

LSTM networks have been extensively tested and explored in watershed hydrological modeling in recent years, showing promise in various applications including river flow and flood predictions. Pathirana et al. (2019) used LSTM to predict floods in Sri Lanka at 5-minute intervals for one year. Li et al. (2021) compared LSTM models with synced sequence input and output (SSIO) to those with sequence input single output (SISO) and found that SSIO-based LSTM outperformed SISO-based LSTM in estimating flood extent during Hurricane Harvey. Liu et al. (2020a) developed an LSTM-based K-Nearest Neighbours (KNN) model for real-time flood forecasting under different climate conditions, which outperformed other models including LSTM, RNN, and Xinanjiang model (XAJ). Liu et al. (2017) forecasted peak flows by integrating stacked autoencoders (SAE) and backpropagation neural networks (BPNN) into LSTM. Their method, utilizing 12 years of flow data at hourly intervals to predict the next six hours, showed superior performance compared to baseline methods.

Spatio-Temporal Attention LSTM (STA-LSTM) was proposed for flood prediction tasks in three basins in China (Ding et al., 2020). Hourly rainfall and flow data were used to predict floods between 1- and 6-hour-ahead, resulting in significantly reduced root mean square error (RMSE) values compared to baselines for all three basins. Lee et al. (2020) utilized LSTM for flood prediction using data spanning 66 months, collected at one-hour intervals for water elevation data from the Namgang Dam basin. The authors reported successful predictions using only water elevation. In a study on the Klang River in Malaysia, Faruq et al. (2020) recorded water level for two years at 15-minute intervals and applied LSTM and radial basis function neural network (RBFNN) to forecast floods, both of which achieved good results. Le et al. (2019) applied LSTM to obtain multi-step output for one-day, two-days, and three-days flows in Hoa Binh Station on the Da River using daily data. Liu et al. (2018) employed context-aware attention LSTM (CA-LSTM) for flood modeling using hourly data from a river gauging station on the Changhua river. They found that CA-LSTM outperformed LSTM and Fully Convolutional Networks (FCN) on average for two-days (T+1) to seven-days (T+6) predictions.

Some LSTM-based models like Convolutional Neural Network-LSTM (CNN-LSTM), Convolutional LSTM (Conv-LSTM), and bidirectional LSTM (BiLSTM) were found to be too complex for sequence prediction tasks (Liu et al., 2020b). Won et al. (2022) compared Artificial Neural Network (ANN), LSTM, Stacked-LSTM, and BiLSTM for flood forecasting in the Dorim River Basin, Seoul, and found that BiLSTM achieved the best accuracy. Li et al. (2022) applied CNN-LSTM to spatial and temporal features for flow prediction, while Yao et al. (2023) proposed an improved sparrow search algorithm-CNN-LSTM&GRU-based hybrid runoff prediction model for river runoff prediction. Wu et al. (2023) used CNN-BiLSTM based on variational mode decomposition, attention mechanism, and Bayesian optimization algorithm to forecast daily runoff in the Wei River Basin.

This study aims to apply various LSTM models for peak river flow forecasting with one-day (T), two-days (T-1 to T), and seven-days (T-6 to T) intervals. Data from the Lüleburgaz stream-flow gauging station (SGS) on the Ergene River in the Meriç-Ergene Basin in Turkey was used for modeling due to its complete data. Different LSTM architectures were investigated and compared, including LSTM, Stacked-LSTM (LSTM+LSTM and LSTM+BiLSTM), BiLSTM, Conv-LSTM, and CNN-LSTM. Additionally, the prediction accuracy as the time interval increased was assessed. One of the LSTM-based models, ConvLSTM, was applied for the first time in this field.

2. Methodology

2.1. LSTM networks

Considered to be a deep feed-forward neural network, RNN was customized to handle sequential data while coping with long-term dependencies. However, RNN, which was successful in processing sequential data, was less useful when assessing long-term dependencies. So, a new method was developed.

Introduced by Hochreiter and Schmidhuber (1996), the LSTM network a variant of RNN has the ability to learn long-term dependencies by using external memory and overcome exploding and vanishing gradient problems encountered in RNN (Fischer and Krauss, 2018). An LSTM network consists of an input layer, one or more hidden layers, and an output layer. In the input layer, there are neurons up to the number of attributes in the input. In the time series problems, the number of neurons in the output layer is equal to the number of predicted time steps. The feature that makes LSTM superior to RNN is the recurrent hidden layer(s). While in RNN, this layer has quite a simple structure, in LSTM this layer has one or more memory blocks which are composed of one or more memory cells. The hidden layer of LSTM is given in Fig. 1.

In the hidden layer of LSTM, the memory cell has three gates controlling cell states: a forget gate (f_t), an input gate (i_t), and an output gate (o_t). The following notations are used to express LSTM mathematically: t represents a time step; X_t is the input vector and h_t is the output vector of the LSTM layer, C_t and $\tilde{C}_t$ are the cell states and candidate values. W_C, W_f, W_i and W_o are used to represent weight matrices. b_C, b_f, b_i and b_o are the bias vectors. σ is the sigmoid function.

PROOF

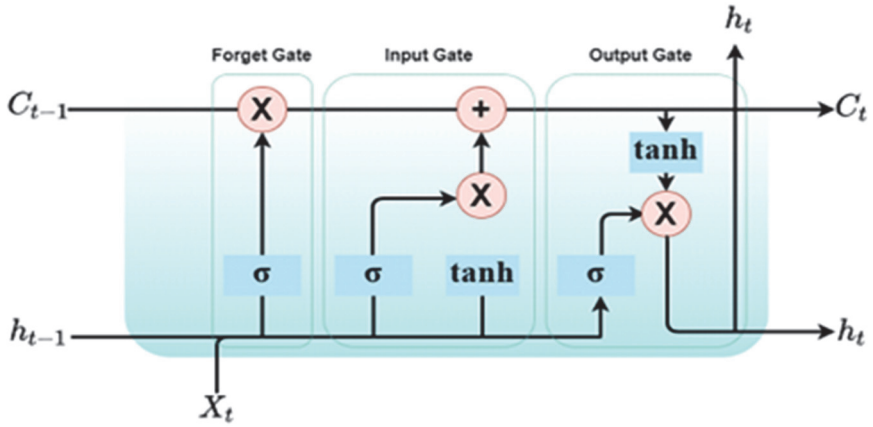


Fig. 1. Hidden layer of LSTM

The forget gate determines which information to retain or discard from the cell state. It takes as input the current input vector, X_t, and the output of the previous unit, ht-1. The output of the forget gate (f_t) is calculated using the sigmoid function, resulting in a value in the range [0, 1]. A value of f_t equal to 0 indicates that all data will be discarded, while a value of 1 indicates that all data will be retained (Eq. 1):

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, X_t\right] + b_f\right) \tag{1}$$

The task of the input gate is to decide on what information is to be added to the cell state. To do this, at first candidate values are calculated, then i_t is calculated by using Eqs. (2-3).

$$\tilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, X_t\right] + b_C\right) \tag{2}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, X_t\right] + b_i\right) \tag{3}$$

Using the results from the three equations above the new cell state is calculated by using the element-wise product (°) (Eq. 4):

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{4}$$

The output gate decides which of the information selected by the cell state will be output. For this, first, previous and current states are passed through the sigmoid function. Then the element-wise product of the result of the sigmoid function and new cell state which is processed by the hyperbolic tangent (*tanh*) function is calculated. The output is derived as follows (Eqs. 5-6):

$$o_t = \sigma\left(W_o X_t + W_o h_{t-1} + b_o\right) \tag{5}$$

$$h_t = o_t \circ \tanh\left(C_t\right) \tag{6}$$

2.2. Stacked LSTM

Making LSTM deeper is one of the simplest and most effective methods to increase the learning capacity of the network. A way to do this is to stack the hidden layers of LSTM, that is, the output of the previous LSTM recurrent layer is the input of the next LSTM recurrent layer. For each timestamp, Stacked LSTM gives a separate output.

Stacked LSTM maps an input sequence from one space to another with an automated feature learning process using a hierarchy of LSTM networks (Jahromi et al., 2020). The network is unrolled for 4 timesteps is given with Fig. 2.

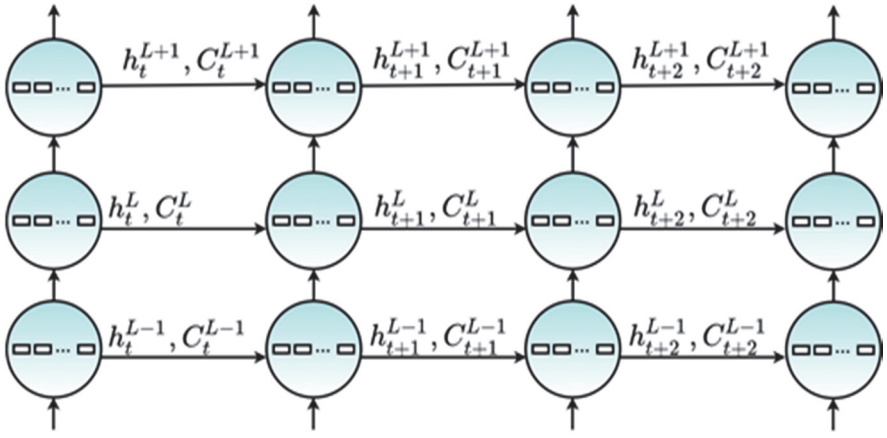


Fig. 2. Unrolled stacked LSTM

PROOF

Mathematical model of *L*th LSTM layer for an unrolled Stacked LSTM network is given by Eqs. (7-12):

$$f_t^L = \sigma\left(W_{fh}^L h_{t-1}^L + W_{fx}^L h_{t-1}^L + b_f^L\right) \quad (7)$$

$$i_t^L = \sigma\left(W_{ih}^L h_{t-1}^L + W_{ix}^L h_{t-1}^L + b_i^L\right) \quad (8)$$

$$\tilde{C}_t^L = \tanh\left(W_{\tilde{C}h}^L h_{t-1}^L + W_{\tilde{C}x}^L h_{t-1}^L + b_{\tilde{C}}^L\right) \quad (9)$$

$$C_t^L = f_t^L \circ C_{t-1}^L + i_t^L \circ \tilde{C}_t^L \quad (10)$$

$$o_t^L = \sigma\left(W_{oh}^L h_{t-1}^L + W_{ox}^L h_{t-1}^L + b_o^L\right) \quad (11)$$

$$h_t^L = o_t^L \circ \tanh\left(C_t^L\right) \quad (12)$$

In the equations above, h_t^{L-1}, which represents the output of the L_{t-1}th LSTM layer at time t, is the input of the neighborhood (*L*th) layer.

2.3. BiLSTM

Both the RNN model and the LSTM model allow the information to propagate only forward in time. That is, the hidden state of time t is dependent on previous context obtained from hidden state of time *t-1*. To capture context information from past and future time direction simultaneously, the bidirectional RNN (BRNN) was devised by Schuster and Paliwal (1997). Then, Graves and Schmidhuber (2005) proposed BiLSTM which combines BRNN and LSTM units to make effective use of past and future information.

The architecture of BiLSTM is given with Fig. 3. The forward layer of the BiLSTM and the forward layer of the Stacked LSTM have the same structure (Yu et al., 2019). The difference in this model is due to the backward layers. An LSTM cell in this layer at time *t* is defined as (Eqs. 13-19):

$$\overleftarrow{f}_t^L = \sigma\left(W_{\overleftarrow{f}h}^L h_{t-1}^L + W_{\overleftarrow{f}x}^L h_{t-1}^L + b_{\overleftarrow{f}}^L\right) \quad (13)$$

$$\overleftarrow{i}_t^L = \sigma\left(W_{\overleftarrow{i}h}^L h_{t-1}^L + W_{\overleftarrow{i}x}^L h_{t-1}^L + b_{\overleftarrow{i}}^L\right) \quad (14)$$

$$\overleftarrow{\tilde{C}}_t^L = \tanh\left(W_{\overleftarrow{\tilde{C}}h}^L h_{t-1}^L + W_{\overleftarrow{\tilde{C}}x}^L h_{t-1}^L + b_{\overleftarrow{\tilde{C}}}^L\right) \quad (15)$$

$$\overleftarrow{C}_t^L = \overleftarrow{f}_t^L \circ \overleftarrow{C}_{t-1}^L + \overleftarrow{i}_t^L \circ \overleftarrow{\tilde{C}}_t^L \quad (16)$$

$$\overleftarrow{o}_t^L = \sigma\left(W_{\overleftarrow{o}h}^L h_{t-1}^L + W_{\overleftarrow{o}x}^L h_{t-1}^L + b_{\overleftarrow{o}}^L\right) \quad (17)$$

$$\overleftarrow{h}_t^L = \overleftarrow{o}_t^L \circ \tanh\left(\overleftarrow{C}_t^L\right) \quad (18)$$

$$y_t = W_{\overleftarrow{y}t}\overleftarrow{h}_t + b_y \quad (19)$$

2.4. Conv-LSTM

Conv-LSTM is a variant of LSTM which uses the convolutional operation instead of matrix multiplication in each LSTM unit to calculate the next cell state after the current state by using inputs and past states of its neighbours (Fig. 4). Thus, both the number of fully-connected operations and the number of parameters are reduced.

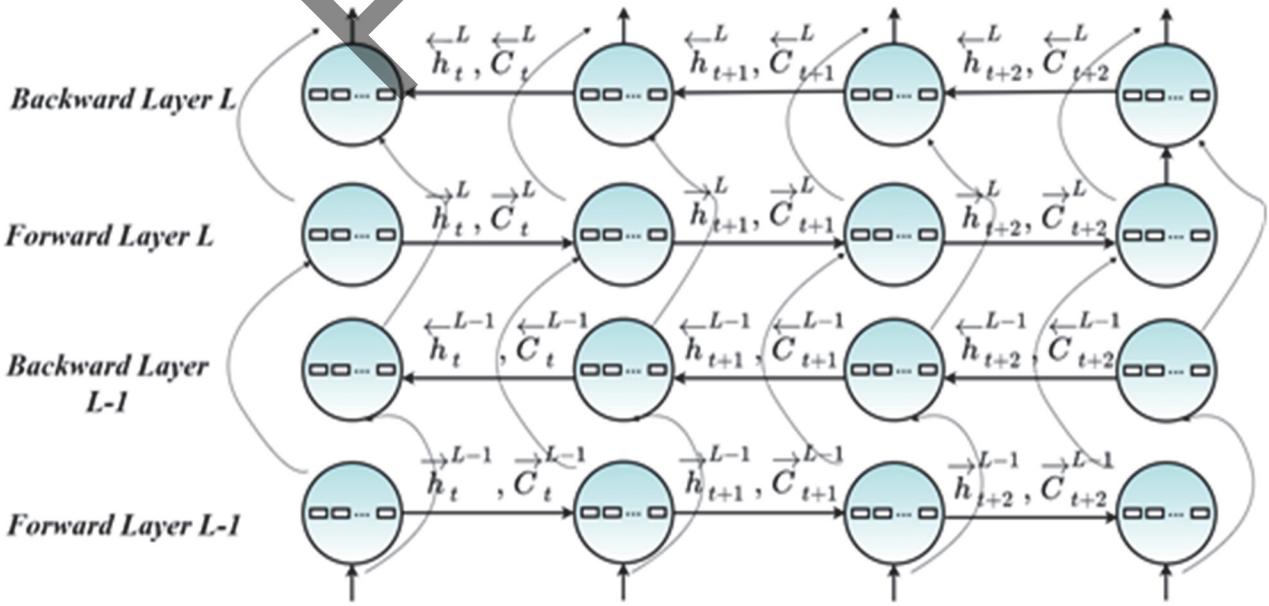


Fig. 3. Inner structure of BiLSTM

PROOF

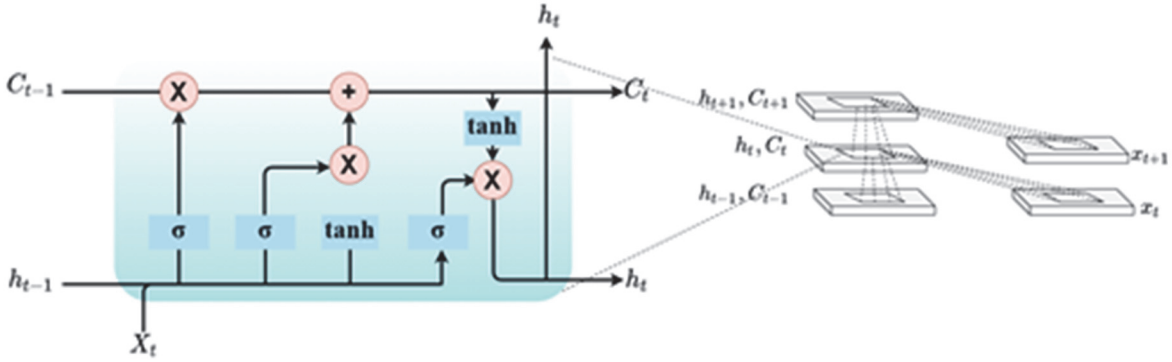


Fig. 4. Structure of Conv-LSTM

As in other networks, Conv-LSTM has three gates which are the input gate, forget gate and output gate. The states, cell memory, gates, parameters and input data are extended from one-dimensional to high-dimensional tensor (Liu et al., 2020c). The equations for the Conv-LSTM are given below (Eqs. 20-25):

$$f_t = \sigma\left(W_{fh} * h_{t-1} + W_{fx} * x_t + b_f\right) \quad (20)$$

$$i_t = \sigma\left(W_{ih} * h_{t-1} + W_{ix} * x_t + b_i\right) \quad (21)$$

$$\tilde{C}_t = \tanh\left(W_{\tilde{C}h} * h_{t-1} + W_{\tilde{C}x} * x_t + b_{\tilde{C}}\right) \quad (22)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \quad (23)$$

$$o_t = \sigma\left(W_{oh} * h_{t-1} + W_{ox} * x_t + b_o\right) \quad (24)$$

$$h_t = o_t \circ \tanh\left(C_t\right) \quad (25)$$

While * represents the convolution operation, ∘ represents the element-wise product in the equations. The architecture of the Conv-LSTM is given with Fig. 4.

2.5. CNN-LSTM

CNN is a kind of feed-forward neural network which was devised by LeCun et al. (1998) for document recognition tasks. This method, which is quite powerful, usually consists of two parts, the feature extraction part and prediction part, based on data sequences, which is the core component of the model. The feature extraction part consists of one or more convolution layers and pooling layers. The extracted features are given as input to the prediction part. The power of CNN comes from its ability to capture local dependencies of features in the input data (Heryadi and Warnars, 2017).

The CNN-LSTM network is the combination of two deep learning models, CNN and LSTM. This network can extract high-level features by CNN, and then this information is fed into the LSTM network to make predictions. The CNN-LSTM model consists of three layers, similar to other models: input layer, hidden layer, and output layer. The hidden layer also contains layers within itself: the convolutional layer and the pooling layer, the LSTM network and the fully connected (FC) layer (Xie et al., 2020). The high performance of the CNN-LSTM model is derived from the combination of CNN, which learns short-term feature relations and LSTM, which learns long-term temporal feature relations.

3. Case study

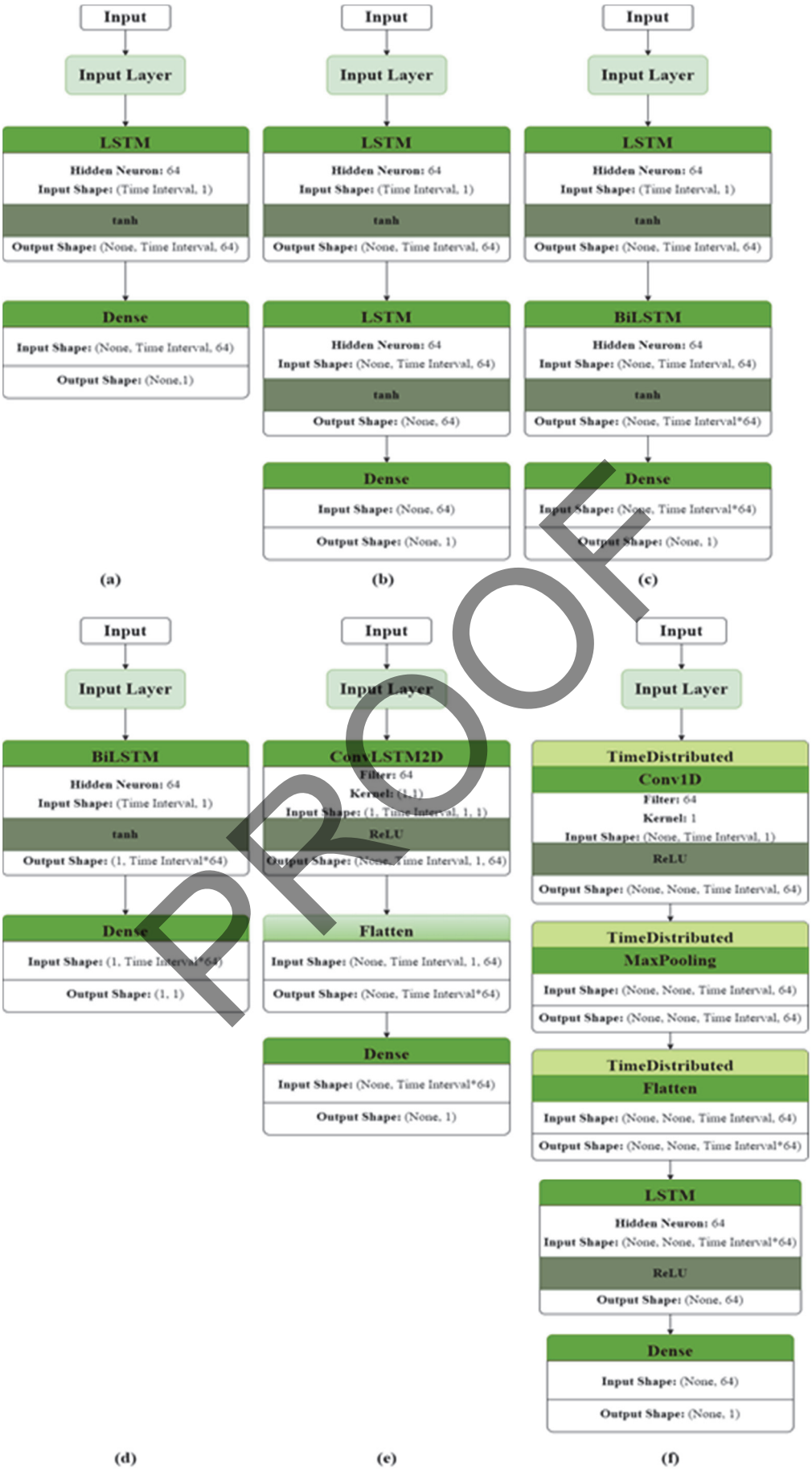
In this study we applied LSTM approaches to flow data to investigate which model was more effective and determine how many previous days would be appropriate to use to make predictions. To achieve this purpose Keras was used to generate the network topologies.

Keras is a high-level, deep learning API written in Python, which allows the implementation of deep learning models (Moolayil, 2019). Here, using Keras, we generated LSTM, Stacked LSTM (LSTM+LSTM and LSTM+BiLSTM), BiLSTM, Conv-LSTM and CNN-LSTM models. All the architectures used are given in Fig. 5.

In order to make the models usable, the model parameters should be decided in accordance with the dataset. In deep learning models it is necessary to first determine the number of input-layer neurons, hidden layer count and the neuron count per hidden layer, the number of neurons in the FC layer and the output layer. Afterwards, the number of epochs and batch size that provide optimization should be determined.

The number of input-layer neurons was determined by the number of features in the input dataset. In addition to comparing the models in the study, it was also important to determine how much historical data was used. Experiments were conducted by creating the data set in three different ways: "T-6 to T", "T-1 to T" and "T" to forecast "T+1". Therefore, the number of neurons in the input layer would be 7, 2 and 1, respectively. The number of hidden layers and dense layers were set to 1. While the number of hidden-layer neurons was 64, since only daily ahead flow would be forecast, the number of neurons in the output layer was set as 1. Epoch size was fixed to 1000 and batch size was fixed to 1.

PROOF



PROOF

Fig. 5. (a) LSTM, (b) LSTM+LSTM, (c) LSTM+BiLSTM, (d) BiLSTM, (e) Conv-LSTM (f) CNN-LSTM

These parameters are usually decided empirically by performing experiments with different values of parameters using any search methods (Clemente, 2017). However, the large dataset in this study did not allow this approach because of the computational burden. Consequently, the experiments were performed with a fixed set of parameters.

Besides parameter settings, the selection of the loss function is also important. The ADAM stochastic gradient descent optimizer was employed to train the LSTM models. ADAM is the most widely used optimization algorithm that utilizes first and second-order moments as parameter update rules. With this feature, it provides high accuracy while minimizing the mean square error. It also provides fast convergence. Therefore, it was used as a loss function in this study. In addition, the chosen learning rate was 0.001.

Another important problem in constructing the structure of deep learning models is the activation function which describes the output of a neuron in terms of the inputs of that neuron (Clemente, 2017). In the experiments, the most widely used non-linear activation functions namely tanh and Rectified Linear Unit (ReLU) were preferred because they have been successfully implemented for solving a large class of learning problems in the literature (Banerjee et al., 2019). The LSTM and BiLSTM models were constructed taking into account the parameter settings given above. The difference between them was that in BiLSTM, the hidden layer was bidirectional instead of the LSTM layer. We designed two different Stacked-LSTM models here. One was created by stacking two hidden LSTM layers where each layer was composed of multiple memory cells (LSTM+LSTM), while the other was created by stacking a hidden LSTM layer and a hidden bidirectional layer (LSTM+BiLSTM). The activation function of all these models was selected to be *tanh*.

The Conv-LSTM and CNN-LSTM differ from LSTM models above in their network structures. For both of these networks, the convolutional layer and number of filters it has and size of kernel to minimize loss needs to be defined. These parameters are also crucial for the performance of the models. The Conv-LSTM consists of one convolutional layer, one flatten layer and one dense layer. In the convolutional layer, there are 64 filters and a kernel of size 1x1. As an activation function ReLU was utilized. CNN-LSTM has a one time-distributed convolutional layer with 64 filters. As an activation function, ReLU was used. The kernel of size 1 was set. In the structure, there was a time-distributed flatten layer and a one time-distributed pooling layer with a size of 1. In this model, the convolutional layer was fed with input data and the output of the convolutional layer was then transmitted to the pooling layer. The output of the pooling layer was subsequently fed to the LSTM layer with 64 hidden neurons (Table 1). Finally, a dense layer was used.

4.1. Study area and data preperation

Lüleburgaz SGS (27°21'6"E - 41°21'3"N) is located on the Ergene River in the Meriç-Ergene Basin in Turkey. The rainfall catchment area of the station is 2794.00 km^2 and its approximate elevation is 38 m. The location of the station on the Ergene River is given in Fig. 6.

The accuracy of the deep learning algorithms depends on sufficient observation data samples. The observed dataset included 28-years of flow data which was taken with one-day intervals from 1 October 1987 to 23 September 2015 and was obtained from the General Directorate of Turkish State Hydraulic Works (DSI, 2023). The dataset contains a total of 10221 data points over the 28-year period. Time series plot of the runoff volume was given in Fig. 7.

PROOF

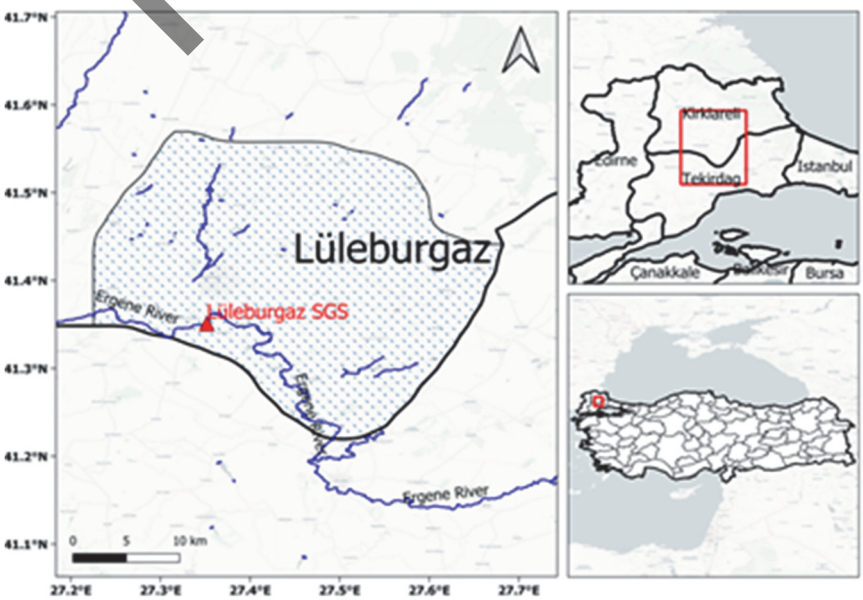


Fig. 6. Location of the Lüleburgaz SGS on Ergene River

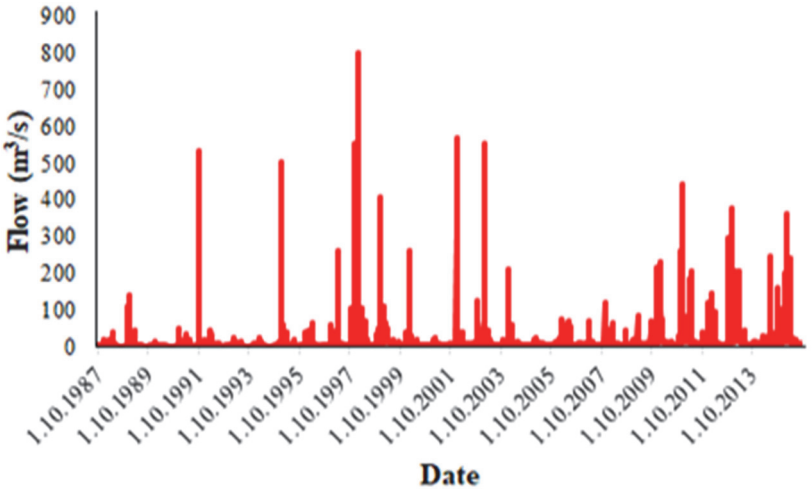


Fig. 7. Time series plot of the runoff volume

Table 1. Parameter settings of models

Model	*Optimizer*	*Activation function*	*# LSTM layer*	*#Hidden neurons/filters*	*#Kernel*
LSTM	ADAM	tanh	1	64	-
LSTM+LSTM	ADAM	tanh	2	64	-
STM+BiLSTM	ADAM	tanh	2	64	-
BiLSTM	ADAM	tanh	1	64	-
ConvLSTM	ADAM	ReLU	1	64	1x1
CNN-LSTM	ADAM	ReLU	1	64	1

PROOF

The observed flow data that was used in this study had the range 0 to 819.000. Imbalance in the range of flow negatively affects the accuracy of the models. In order to deal with this, normalization was used to map the range to [0, 1] by applying min-max normalization. Min-max normalization maintains the relationship among data instances thus enabling accurate predictions. The min-max normalization was calculated as follows (Wang et al., 2023b) (Eq. 26):

$$y_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{26}$$

where x_i represents the ith data point, x_{min} represents the minimum-valued data point and x_{max} represents the maximum-valued data point. y_i is the normalized form of x_i and in the range of [0,1]. Statistical characteristics of the Lüleburgaz SGS original data and normalized data was given in Table 2.

The distribution of values for original data could was right-skewed because the skewness value was positive (Table 2). The flow chart of proposed system is given in Fig. 8.

4.2. Performance metrics

In order to build a reliable model, the dataset should be divided into training and test sets. Based on this assumption, for the experimental dataset containing 10221 daily observation values in total we aimed for a 75% train (1 October 1987 to 24 September 2008) and 25% test (25 September 2008 to 23 September 2015) split which is one of the commonly used ratio in data splitting methods (Kanakaraddi et al., 2021). Thus, the training set contained 7666 observations and the test set contained 2555 observations.

In this study daily measured values of flow at time seven-days, two-days and one-day were used as input variables in order to predict the daily ahead flow (T+1); that is, it was aimed to obtain univariate output for one-day. In order to validate the LSTM networks model results, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2) and Nash-Sutcliffe efficiency (NSE) were assessed between observed data and predicted values.

Lower RMSE suggests better central tendency and smaller extreme errors, higher RMSE suggests worse central tendency and a larger dispersion. Thus, a lower value would be more favorable for RMSE. The formula is given below (Lu et al., 2023) (Eq. 27):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{27}$$

where y_i is the observed result and $\hat{y}_i$ is the predicted value; n is the number of observations in the test set.

The MAE is a measure of the difference between two continuous variables, that is, a measure of the average error. A model that achieves a lower

MAE is considered a better model and vice versa. When both metrics are compared, RMSE tends to be less than MAE, since the MAE assigns the same weight to all errors, while the RMSE penalizes large errors at a higher rate (Alabdrabalnabi et al., 2022). The MAE is mathematically expressed as (Lu et al., 2023) (Eq. 28):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{28}$$

Regression accuracy (R^2) is the measure of linear correlation between two variables (Lu et al., 2023). While a high R^2 value indicates a high correlation between predicted and observed value, a low R^2 value indicates a high variation between them. In the equation below, $\bar{y}$ is the mean value of observed values of variable y (Eq. 29).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{29}$$

The NSE score is frequently utilized performance measure to evaluate the forecasting abilities of hydrological models (Tunalı et al., 2023). If the Nash-Sutcliffe efficiency of the model is higher than 50%, it can be considered satisfactory. The range of the NSE is -∞ to 1. When the predicted data perfectly matches the observed data, NSE = 1. When the NSE is zero, it means that the predicted data is exactly as accurate as the observed data mean. If NSE<0 indicates that the predicted data is not as good of a predictor as the observed data (Reddy and Saravanan, 2023). In the equation below, $\bar{\hat{y}}$ is the mean value of predicted values of variable $\hat{y}$ (Yao et al., 2023) (Eq. 30).

$$NSE = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{\hat{y}})^2} \tag{30}$$

4. Results and discussions

The results of time interval impact show that there are differences in the performances of the LSTM, Stacked LSTM (LSTM+LSTM and LSTM+BiLSTM), BiLSTM, Conv-LSTM and CNN-LSTM models with input information spanning seven-days, two-days and one-day continuous days as shown in Fig. 9. In order to analyze this situation in detail, firstly the input sequences two-days and one-day are compared. When the input sequences for two-day interval and one-day interval were compared, the RMSE, MAE, R^2 and NSE values increased by 10%, 12%, 77% and 82% respectively for LSTM. Analysis based on Stacked LSTM (LSTM+LSTM) showed that RMSE and MAE values decreased by 8% and 15%, respectively, while R^2 and NSE values increased 65% and 80% respectively. In the LSTM+BiLSTM (stacked) model, while RMSE value increased 3%, R^2 value increased 64% and NSE value increased 73%, MAE value decreased 2%. For BiLSTM, while RMSE and MAE values decreased 17% and 14% respectively, R^2 and NSE values increased 75% and 72% respectively. In Conv-LSTM, RMSE and MAE values decreased 8% and R^2 and NSE values increased 70% and 88% respectively. In CNN-LSTM, RMSE value decreased 8% but MAE, R^2 and NSE values increased 5%, 47% and 48% respectively.

There was no significant difference between the RMSE and MAE values in these results. However, overfitting was observed in the two-day input sequence based on the R2 value from the test set. The most accurate results were obtained with the one-day input sequence, which was consistent when compared with the seven-day sequence. Since the NSE value ranges between -∞ and 1, it requires separate evaluation (Table 3). Upon examination of Table 3, it is evident that the NSE values for the one-day models are higher than 50%, indicating satisfactory performance. For the two-day models, all NSE values are lower than 50%, indicating unsatisfactory performance. Additionally, the NSE values for the seven-day models are smaller than zero, suggesting that the predicted data is not as reliable as the observed data.

PROOF

Table 2. Statistical characteristics of the Lüleburgaz SGS original data and normalized data

Statistic type	*Value (Original)*	*Value (Normalized)*	*Statistic type*	*Value (Original)*	*Value (Normalized)*
Data count (N)	10221	10221	Skewness	13.097	13.1
Minimum flow (m^3/s)	0	0	Kurtosis	256.624	256.796
Maximum flow (m^3/s)	819.000	1	Interquartile range (m^3/s)	5.26	0.007
Average (m^3/s)	10.32	0.13	Median (m^3/s)	5.8	0.007
Standard deviation (m^3/s)	24.955	0.031	Median absolute deviation (m^3/s)	3.707	0.009

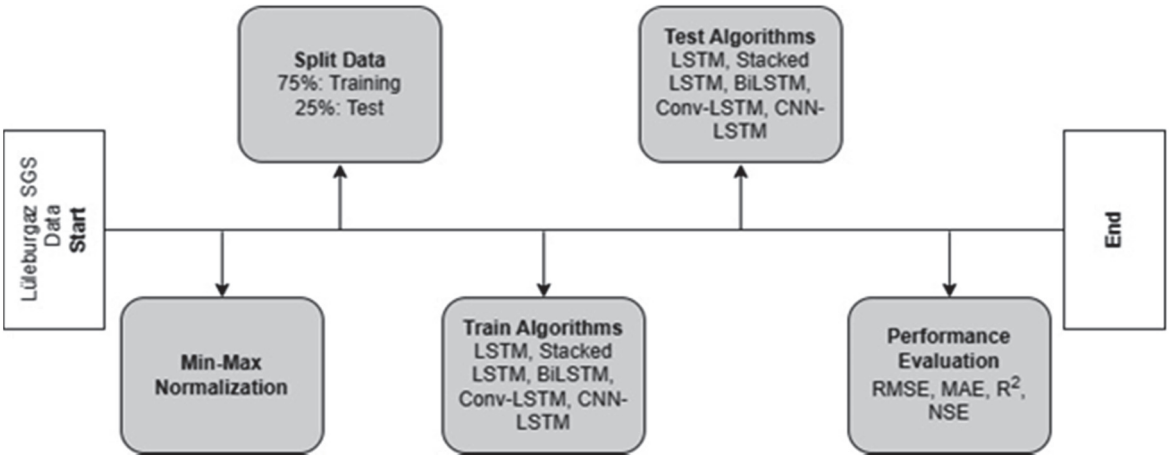


Fig. 8. Flow chart of proposed system

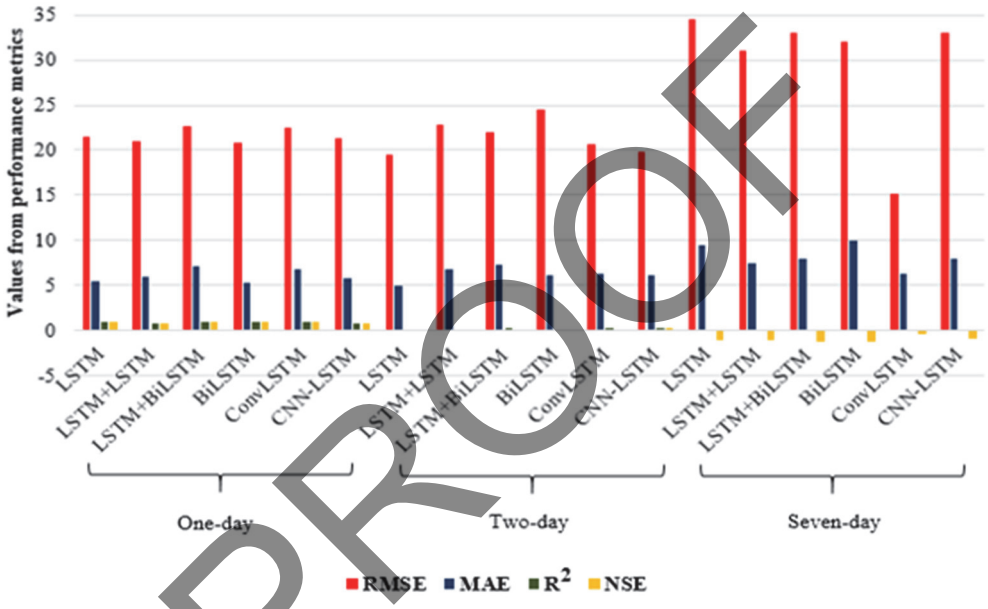


Fig. 9. Comparison of all models for each time-intervals for test data

PROOF

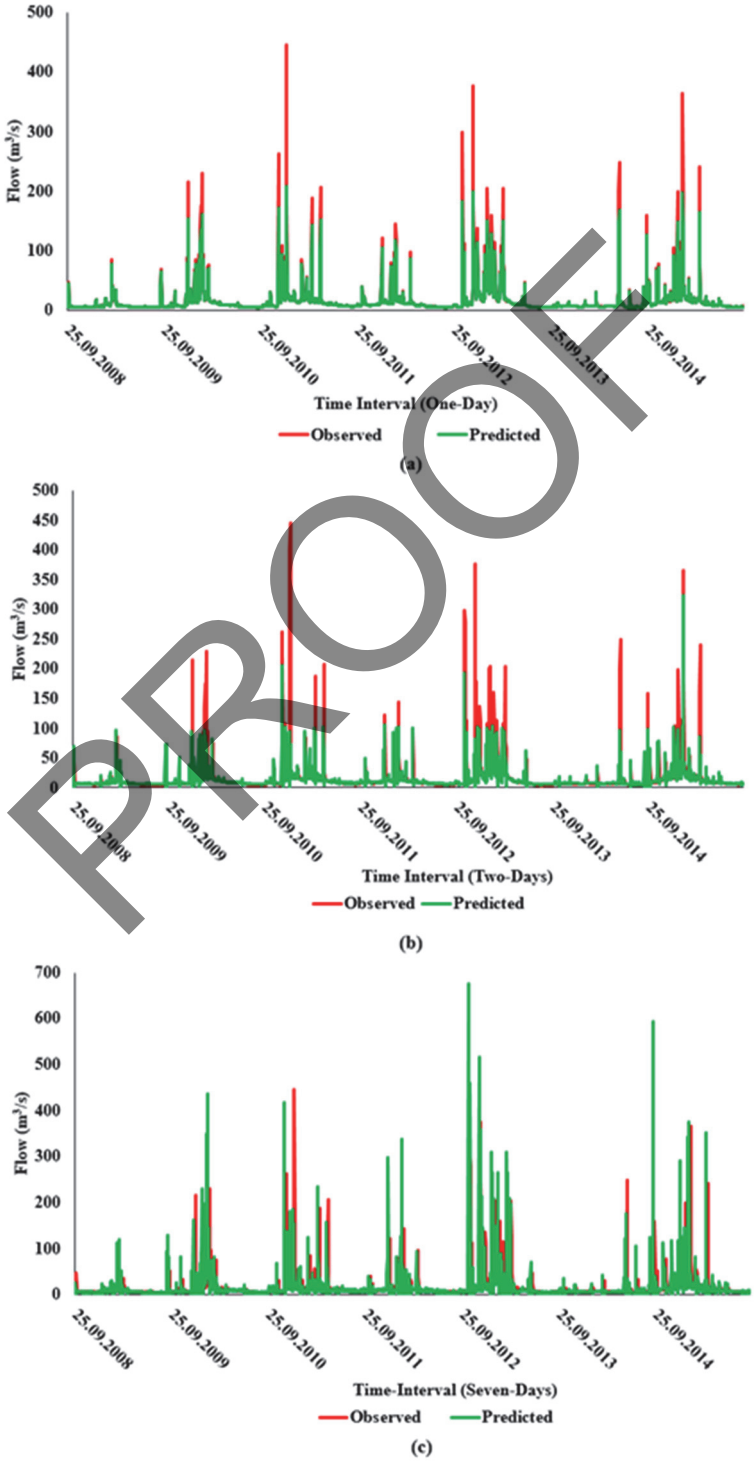
Table 3. Performance of the LSTM networks models for the time interval of one-day

Model	*RMSE*	*MAE*	*R^2*	*NSE*
LSTM	21.55	5.54	0.955	0.905
LSTM+LSTM	20.92	5.92	0.834	0.832
LSTM+BiLSTM	22.62	7.07	0.886	0.881
BiLSTM	**20.76**	**5.33**	0.951	0.887
ConvLSTM	22.43	6.76	**0.968**	**0.954**
CNN-LSTM	21.41	5.78	0.834	0.795

Although the best error metrics that distinguishes models from each other are R^2 and NSE, a comprehensive evaluation should be made by also including RMSE and MAE. According to the flow prediction results for the time interval of one-day, the BiLSTM model was superior with respect to RMSE, MAE, R^2 and NSE values, and achieves better performance in most cases while the trend in error variation was similar in all the applied models (Table 3). The reason why BiLSTM was most successful was that it traverses sequences in the forward and backward directions, so more time-series features can be captured and extracted. The structures of the LSTM models used in our study (Fig. 5) were different, so the training algorithms were also different. Therefore, we proposed to investigate the effectiveness and reliability of the different applied models in terms of flow prediction.

The implications of the differences in flow prediction have been compared by using observed and predicted values. Fig. 10 indicates differences between expected and predicted values for one-day, two-day and seven-day time intervals for the test data. While there was good agreement between the observed and the predicted flow series for a one-day interval, there are some underestimations of the predicted flow compared to the actual measured flow for two-day and seven-day time intervals. This has been reported previously (Ananthakrishnan et al., 2020; Cardone, 2019). When the results are examined, the prediction errors on the test set of the models are clearly seen.

The errors also tend to change significantly with the time interval used for prediction. This is because important information regarding the flow is lost over a large time interval. We found that the longer the forecast period, the worse the forecast. The LSTM models used in our study, especially BiLSTM, reliable and acceptable prediction results in short-term predictions (Fig. 10).



PROOF

Fig. 10. Comparisons of BiLSTM modelled flow data for (a) one-day, (b) two-day, (c) seven-day time intervals for test data

Numerous studies have been conducted into river flow prediction, and they have all concluded that the overall prediction error of models is reasonably low (Ghorbani et al., 2016; Ghorbani et al., 2020; Khazaee et al., 2019; Shahriar et al., 2016; Tabbussum and Dar, 2021). The common feature of these studies is that they use machine learning methods, because in difficult time series forecasting, machine learning algorithms have been shown to produce outstanding results.

However, when setup and trained appropriately, deep learning techniques provide superior representation and classification in time series problems compared to statistical analysis and machine learning methods (Dwivedi et al., 2021; Fawaz et al., 2019). In summary, predictions using only one-day of historical data provided the highest performance compared to intervals of two-days and seven-days from the target observation. At intervals of seven-days and two-days all models had unacceptable accuracy based on R^2 and NSE results. The ConvLSTM method and one-day intervals produced the highest R^2 value (0.968) and NSE value (0.954) for the test data. However, when were also considered as reference quantities than BiLSTM was the best performing model with RMSE of 20.76 m^3/s and MAE of 5.33 m^3/s, which are the lowest amongst the models tested for the one-day interval dataset, despite BiLSTM having a little less 95.1% regression accuracy compared ConvLSTM method.

4. Conclusions

We aimed to obtain univariate output for one-day inquiries using LSTM, LSTM+LSTM, LSTM+BiLSTM, BiLSTM, ConvLSTM, CNN-LSTM models. RMSE, MAE, R^2, NSE were taken into consideration to evaluate performance of each model using daily observed values of river flow at intervals of seven-days (T-6 to T), two-days (T-1 to T), and one-day (T) as the input variables. The prediction error of each model increased in direct proportion to the increasing time intervals.

It was seen that LSTM networks have an acceptable performance in predicting the daily river flows of the study area and the most successful predictions were realized in the one-day time interval. Among the models, BiLSTM is preferable to RMSE 20.76 m^3/s and MAE 5.33 m^3/s.

If it is not easy to decide which probability distribution has the best fit, deep learning methods such as BiLSTM may be used as an alternative approach for the future applications of river flow prediction.

One of the main limitations of this study is to not consider the arid or semi-arid climate with a pattern of scarce and variable river flow. In future researches this limitaion will be examined by taking into account the rivers in different climates. Also, future studies may focus on establishing a system with more sites to monitor the performance of LSTM models in daily-ahead flow prediction.

References

Alabdrabalnabi A., Gautam R., Sarathy S.M., (2022), Machine learning to predict biochar and bio-oil yields from co-pyrolysis of biomass and plastics, *Fuel*, **328**, 125303, https://doi.org/10.1016/j.fuel.2022.125303.

Alsavaf Y., Teksoy A., (2022), Applicability of recurrent neural networks to retrieve missing runoff records: challenges and opportunities in Turkey, *Environmental Monitoring and Assessment*, **194**, 28, https://doi.org/10.1007/s10661-021-09681-z.

Ananthakrishnan S., Geetha P., Soman K.P., (2020), *Temperature Forecasting using Dynamic Model Decomposition*, 3rd Int. Conf. on Intelligent Sustainable Systems, 3-5 December, Coimbatore, India, 1590-1596.

Apaydın H., Feizi H., Akçakoca F., Sattari M.T., (2022), *Daily Streamflow Modelling in the Nalli River Using Recurrent Neural Networks*, Int. Conf. New Technologies, Development and Applications, 22-24 June, Sarajevo, Bosnia and Herzegovina, 813-822.

Banerjee C., Mukherjee T., Jr E.P., (2019), *An Empirical Study on Generalizations of the ReLU Activation Function*, ACM Southeast Conf., 18-20 April, Kennesaw, USA, 164-167.

Bhandari S., Pathak S., Jain S.A., Cheng Z., (2023), A literature review of early-stage diabetic retinopathy detection using deep learning and evolutionary computing techniques, *Archives of Computational Methods in Engineering*, **30**, 799-810.

Cardone A., (2019), *Machine learning methods for adaptive test case generation for Android activities*, PhD Thesis, University of Illinois, Chicago, USA.

Clemente A.V., (2017), *Decoupling deep learning and reinforcement learning for stable and efficient deep policy gradient algorithms*, MSc Thesis, Norwegian University of Science and Technology, Norway.

Ding Y., Zhu Y., Feng J., Zhang P., Cheng Z., (2020), Interpretable spatio-temporal attention LSTM model for flood forecasting, *Neurocomputing*, **403**, 348-359.

DSI, (2023), Current Observation Annals, On line at: https://www.dsi.gov.tr/Sayfa/Detay/744.

Dwivedi S.A., Attry A., Parekh D., Singla K., (2021), *Analysis and Forecasting of Time-Series Data Using S-ARIMA, CNN and LSTM*, 2021 International Conference on Computing, Communication, and Intelligent Systems, 19-20 February, India, 131-136.

Ebrahimi E., Shourian M., (2020), River flow prediction using dynamic method for selecting and prioritizing K-nearest neighbors based on data features, *Journal of Hydrologic Engineering*, **25**, 04020010, https://doi.org/10.1061/(ASCE)HE.1943-5584.0001905.

Faruq A., Arsa H.P., Hussein S.F.M., Razali C.M.C., Marto A., Abdullah S.S., (2020), Deep learning-based forecast and warning of floods in Klang River Malaysia, *Ingénierie des Systèmes d'Information*, **25**, 365-370.

Fawaz H.I., Forestier G., Weber J., Idoumghar L., Muller P.A., (2019), Deep learning for time series classification: a review, *Data Mining and Knowledge Discovery*, **33**, 917-963.

Fischer T., Krauss C., (2018), Deep learning with long short-term memory networks for financial market

PROOF

predictions, *The European Journal of Operational Research*, **270**, 654-669.

Ghorbani M.A., Deo R.C., Kim S., Hasanpour Kashani M., Karimi V., Izadkhah M., (2020), Development and evaluation of the cascade correlation neural network and the random forest models for river stage and river flow prediction in Australia, *Soft Computing*, **24**, 12079-12090.

Ghorbani M.A., Zadeh H.A., Isazadeh M., Terzi O., (2016), A comparative study of artificial neural network (MLP, RBF) and support vector machine models for river flow prediction, *Environmental Earth Sciences*, **75**, 476, https://doi.org/10.1007/s12665-015-5096-x.

Graves A., Schmidhuber J., (2005), Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*, **18**, 602-610.

Heryadi Y., Warnars H.L.H.S., (2017), *Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM*, IEEE Int. Conf. on Cybernetics and Computational Intelligence, 20-22 November, Phuket, Thailand, 84-89.

Hochreiter S., Schmidhuber J., (1996), LSTM can solve hard long time lag problems, *Advances in Neural Information Processing Systems*, **33**, 473-479.

Jahromi A.N., Hashemi,S., Dehghantanha A., Parizi R.M., Choo K.K.R., (2020), An enhanced stacked LSTM method with no random initialization for malware threat hunting in safety and time-critical systems, *IEEE Transactions on Emerging Topics in Computational Intelligence*, **4**, 630-640.

Kanakaraddi S.G., Gull K.C., Bali J., Chikaraddi A.K., Giraddi S., (2021), *Disease Prediction Using Data Mining and Machine Learning Techniques*, In: *Advanced Prognostic Predictive Modelling in Healthcare Data Analytics*, Roy S., Goyal L.M., Mittal M. (Eds.), Springer, 71-92.

Karim F., Armin M.A., Ahmedt-Aristizabal D., Tychsen-Smith L., Petersson L., (2023), A Review of hydrodynamic and machine learning approaches for flood inundation modelling, *Water*, **15**, 566, https://doi.org/10.3390/w15030566.

Khaldi R., Afia A.E., Chiheb R., Tabik S., (2023), What is the best RNN-cell structure to forecast each time series behavior?, *Expert Systems with Applications*, **215**, 119140, https://doi.org/10.1016/j.eswa.2022.119140.

Khazaee Poul A., Shourian M., Ebrahimi H., (2019), A comparative study of MLR, KNN, ANN and ANFIS models with wavelet transform in monthly stream flow prediction, *Water Resources Management*, **33**, 2907-2923.

Le X.H., Ho H.V., Lee G., Jung S., (2019), Application of long short-term memory (LSTM) neural network for flood forecasting, *Water*, **11**, 1387, https://doi.org/10.3390/w11071387.

LeCun Y., Bottou L., Bengio Y., Haffner P., (1998), Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, **86**, 2278-2324.

Lee S., An H., Hur Y., Kim Y., Byun J., (2020), Flood prediction in the Namgang Dam basin using a long short-term memory (LSTM) algorithm, *Korean Journal of Agricultural Science*, **47**, 471-483.

Li W., Kiaghadi A., Dawson C., (2021), Exploring the best sequence LSTM modeling architecture for flood prediction, *Neural Computing & Applications*, **33**, 5571-5580.

Li X., Xu W., Ren M., Jiang Y., Fu G., (2022), Hybrid CNN-LSTM models for river flow prediction, *Water Supply*, **22**, 4902-4919.

Liu D., Jiang W., Mu L., Wang S., (2020b), Streamflow Prediction Using Deep LearningNeural Network: Case Study of Yangtze River, *IEEE Access*, **8**, 90069-90086.

Liu F., Xu F., Yang S., (2017), *A Flood Forecasting Model Based on Deep Learning Algorithm via Integrating Stacked Autoencoders with BP Neural Network*, IEEE 3rd International Conference on Multimedia Big Data, 19-21 April, Laguna Hills, California, USA, 58-61.

Liu M., Huang Y., Li Z., Tong B., Liu Z., Sun M., Jiang F., Zhang H., (2020a), The applicability of LSTM-KNN model for real-time flood forecasting in different climate zones in China, *Water*, **12**, 440, https://doi.org/10.3390/w12020440.

Liu Z., Chen T., Ding E., Liu Y., Yu W., (2020c), Attention-based convolutional LSTM for describing video, *IEEE Access*, **8**, 133713-133724.

Liu Z., Xu W., Feng J., Palaiahnakote S., Lu T., (2018), *Context-Aware Attention LSTM Network for 3,* 24th Int. Conf. on Pattern Recognition, 20-24 August, Beijing, China, 1301-1306.

Lu W., Long W., Yuan L., (2023), A machine learning regression approach for pre-renovation construction waste auditing, *Journal of Cleaner Production*, **397**, 136596, https://doi.org/10.1016/j.jclepro.2023.136596.

Mehedi M.A.A., Khosravi M., Yazdan M.M.S., Shabanian H., (2022), Exploring temporal dynamics of river discharge using univariate Long Short-Term Memory (LSTM) recurrent neural network at east branch of Delaware River, *Hydrology*, **9**, 202, https://doi.org/10.3390/hydrology9110202.

Moolayil J., (2019), *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*, Apress Berkeley, USA.

Pathirana D., Chandrasiri L., Jayasekara D., Dilmi V., Samarasinghe P., Pemadasa N., (2019), *Deep Learning-Based Flood Prediction and Relief Optimization*, 2019 Int. Conf. on Advancements in Computing (ICAC), 5-7 December, Malabe, Sri Lanka, 481-486.

Reddy N.M., Saravanan S., (2023), Evaluation of the accuracy of seven gridded satellite precipitation products over the Godavari River basin, India, *International Journal of Environmental Science and Technology*, **20**, 10179-10204.

Sarker I.H., (2021), Deep Learning: A comprehensive overview on techniques, taxonomy, applications and research directions, *SN Computer Science*, **2**, 420, https://doi.org/10.1007/s42979-021-00815-1.

Schuster M., Paliwal K.K., (1997), Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, **45**, 2673-2681.

Shahriar M., Kamruzzaman M., McCulloch J., Beecham S., (2016), Multiple step ahead river flow modelling for South East Tasmanian aquaculture, *Earth Science Informatics*, **9**, 271-279.

Sharma P., Singh S., (2021), Artificial neural network approach for hydrologic river flow time series forecasting, *Agricultural Research*, **11**, 446-476.

Shen C., (2018), A transdisciplinary review of deep learning research and its relevance for water resources scientists, *Water Resources Research*, **54**, 8558-8593.

Tabbussum R., Dar A.Q., (2021), Performance evaluation of artificial intelligence paradigms-artificial neural networks, fuzzy logic, and adaptive neuro-fuzzy inference system for flood prediction, *Environmental Science and Pollution Research*, **28**, 25265-25282.

Tunalı U., Tüzel I.H., Tüzel Y., Şenol Y., (2023), Estimation of actual crop evapotranspiration using artificial neural

PROOF

networks in tomato grown in closed soilless culture system, *Agricultural Water Management*, **284**, 108331, https://doi.org/10.1016/j.agwat.2023.108331.

Wang J., Li X., Li J., Sun Q., Wang H., (2022), Ngcu: A New Rnn Model for Time-Series Data Prediction, *Big Data Research*, **27**, 100296, https://doi.org/10.1016/j.bdr.2021.100296.

Wang Y., Li H., Xu J., Liu S., Tan Q., Wang X., (2023b), Machine learning assisted two-phase upscaling for large-scale oil-water system, *Applied Energy*, **337**, 120854, https://doi.org/10.1016/j.apenergy.2023.120854.

Wang Z., Wang Q., Wu T., (2023a), A novel hybrid model for water quality prediction based on VMD and IGOA optimized for LSTM, *Frontiers of Environmental Science & Engineering*, **17**, 88, https://doi.org/10.1007/s11783-023-1688-y.

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**, 187, https://doi.org/10.3390/w14020187.

Wu J., Wang Z., Hu Y., Dong J., (2023), Runoff forecasting using convolutional neural networks and optimized bi-directional long short-term memory, *Water Resources Management*, **37**, 937-953.

Xie G., Shangguan A., Fei R., Ji W., Ma W., Hei X., (2020), Motion trajectory prediction based on a CNN-LSTM sequential model, *Science China Information Sciences*, **63**, 212207, https://doi.org/10.1007/s11432-019-2761-y.

Yao Z., Wang Z., Wang D., Wu J., Chen L., (2023), An ensemble CNN-LSTM and GRU adaptive weighting model based improved sparrow search algorithm for predicting runoff using historical meteorological and runoff data as input, *Journal of Hydrology*, **625**, 129977, https://doi.org/10.1016/j.jhydrol.2023.129977.

Yılmaz S., Ekinci Ö., Şentürk E., Morkoyunlu Yüce A., (2022), Recurrent neural networks for peak flow estimation, *Environmental Engineering and Management Journal*, **21**, 927-937.

Young C.C., Liu W.C., Wu M.C., (2017), A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events, *Applied Soft Computing*, **53**, 205-216.

Yu Y., Si X., Hu C., Zhang J., (2019), A review of recurrent neural networks: LSTM cells and network architectures, *Neural Computation*, **31**, 1235-1270.

PROOF