



A hybrid deep learning model for rainfall in the wetlands of southern Iraq

Fehaid Alqahtani¹ · Mostafa Abotaleb² · Alhumaima Ali Subhi³ · El-Sayed M. El-Kenawy⁴ · Abdelaziz A. Abdelhamid^{5,6} · Khder Alakkari⁷ · Amr Badr⁸ · H. K. Al-Mahdawi⁹ · Abdelhameed Ibrahim¹⁰ · Ammar Kadi¹¹

Received: 6 December 2022 / Accepted: 18 January 2023 / Published online: 7 March 2023
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

Machine learning is being used by researchers and computer scientists to develop a new method for predicting rainfall. Due to the non-linear relationship between input data and output conditions, rainfall prediction is hard, so deep neural network (DNN) models substitute for costly, complex systems. Deep neural network-based weather forecasting models can be designed quickly and cheaply to predict rainfall. On the other hand, water levels depend on rainfall. Unpredictable rainfall due to climate change might cause floods or droughts. Many individuals, especially farmers, rely on rain forecasts. In our study, we focus on the area of marshes in southern Iraq, some of the most famous landmarks in the area (and the world), where the Shatt al-Arab flows into the Arabic Gulf and the Tigris and Euphrates rivers developed within the Mesopotamian plain to create a natural balance. Since the beginning of the 1980s, the wetlands, sometimes known as "the marshes," have experienced droughts. And by the late 1990s, a sizable portion of the marshes had dried up, leaving the arid and salty Sabkha lands void of life, particularly lands with vast bodies of water and high levels of human activity. Moreover, the corresponding regions have undergone visible hydrological and climatic changes. In this study focuses on the marshes of southern Iraq and aims to develop a rainfall forecasting model. We propose a novel approach based on optimized LSTM and hybrid deep learning algorithms to improve the forecasting of average monthly rainfall. To evaluate the efficiency of the predictions, a comparison of the predicted rainfall and the actual recorded rainfall is made, and the performance and accuracy of the models are examined. The hybrid convolutional stacked bidirectional long-short term memory (CNN-BDLSTMs) outperformed the other models.

Keywords Wetlands · Mesopotamia marshes · Hybrid deep-learning · Optimization · CNN · Machine learning

Introduction

Wetlands are an important natural resource because they have a wide variety of plants and animals and can be used to make money for local economies by doing things like grazing cattle or harvesting sugar cane (Maxwell 1957; Young 1977; Tiner et al. 2015; Albarakat et al. 2018). Some of the most densely populated parts of the Earth's water surface are swamps, which are wetland ecosystems inhabited by large quantities of aquatic plants. Among these is Phragmites in Australia, which can be found in almost every swamp (Al-Handal and Hu 2015). Hydrological changes can destroy wetlands, and climate change and human intervention have

led to the loss of wetlands in many parts of the world. Land use and hydrological changes have affected climatic conditions at the local level (Albarakat et al. 2018).

The swamps of Mesopotamia are one of the oldest ecosystems in the world. They are located between three Iraqi governorates: Basra, Dhi Qar, and Maysan. The Mesopotamian Marshes are the major wetlands in the Middle East and Western Asia and have an important role in the region's ecosystem. The lower Mesopotamian basin between the Euphrates and Tigris rivers has flat areas known as flood plains, formed by the buildup of sedimentary material moved upstream by surface waters. Currently, the area of the three marshes ranges from about 10,500 square kilometers to 20,000 square kilometers. These include the marshes of Hammar, the Central Marshes, and the Al-Hawizeh marshes. The entire upper Arabian Gulf ecosystem depends on the Mesopotamian

✉ Mostafa Abotaleb
abotalebmostafa@bk.ru

Extended author information available on the last page of the article

marshes' hydrology (<https://earthobservatory.nasa.gov/images/1716>) (Albarakat et al. 2018). Due to their size, abundance of aquatic vegetation, and isolation from other similar systems, the marshes are crucial for maintaining biodiversity in the Middle East (Al-Handal and Hu 2015; Douabul et al. 2013). The Tigris and Euphrates rivers use them as natural wastewater treatment systems, filtering fertilizers out of the water before releasing it into the Arabic Gulf. The drying of over 10,000 square kilometers of wetlands and lakes will have a significant impact on the local microclimate. Removing vegetation from wetlands will result in significantly lower rates of evaporation and moisture, leading to changes in precipitation patterns (Partow 2001). As well as continuous temperature increases, particularly during the long and hot summer. The reed layer will no longer protect the marsh from strong, dry winds above 40 °C (Maltby 1994).

The results of water scarcity and pollution, extreme thermal conditions, and increased vulnerability to toxic dust storms that can devastate drinking salt ponds and dry swamp basins are just a few of the ways that ecosystem degradation on this scale can seriously harm human health (Pörtner et al. 2022). The exposed salt crusts and dry marsh soil will generate higher volumes of dust, and wind erosion will distribute various impurities, affecting thousands of square kilometers outside of Iraqi borders (Partow 2001). Additionally, due to wind erosion and sand erosion from dry swamps and surrounding deserts, the fragile arable land near the former swamp is likely to contribute to land degradation and desertification (Meng et al. 2020). The flow of the Tigris and Euphrates rivers changed in the late 1980s and early 1990s after the construction of dams and canals; swamps dried up due to human-made dams and politically motivated drainage practices (Parsaie 2016).

Degradation is the shrinking of the area covered in vegetation into arid land. All three of the noted marshes have shrunk, which has caused a massive increase in arid areas. The swamp has degenerated into a wasteland as a result. The Hammar and Fasat Marshes are the most severely degraded, with a 95% degradation rate. The Karkheh River continues to supply water from Iran to the northeastern portion of the Hawizeh swamp, preserving about 30% of the land area (Partow 2001). One of the biggest ecological disasters affecting wetlands worldwide is large-scale drainage modification (Mohamed and Hussain 2016).

The majority of the embankment dams and dams on the Tigris and Euphrates rivers were uprooted by the swamp's residents after the regime responsible for these drainage changes was overthrown in late 2003, and water started to flow back into the swamp (Fitzpatrick 2004). After three years of natural flow, Mesopotamia's swamps started to recover. Between 50 and 60% of the original population of plant and animal species have returned, demonstrating the

wetlands' resilience (Richardson 2005; Richardson et al. 2005).

Although drought is a natural, somewhat unpredictable phenomenon, it can be observed, studied, and predicted using contemporary techniques. A catastrophic drought occurs when the precipitation system fails, affecting the water supply for both natural and agricultural systems and human activities. Because rain is one of the most important sources of water, its presence or absence can have a significant impact on wetlands, particularly due to dams built by neighboring countries with a lack of rainfall, which caused drought and reduced the wetlands area (Raj et al. 2018; Adham 2018; Awchi and Jasim 2017).

The main goal of our research is to study and predict rainfall. The University of East Anglia Climatic Research Unit (CRU) studied rainfall from 1901 to 2020 and created long-term rainfall predictions. We analyzed these data via the Google Earth based CRU TS add-on interface. Previous studies have used satellite imagery to examine the impact of rainfall and climate changes on the landscape over 16 years (Rabbani et al. 2022; Alhumaima and Abdullaev 2020). We use hybrid deep learning models for modeling and predicting rainfall using univariate time series data. This research aims to improve monthly rainfall forecasts for the marshes of Hawizeh, Central, and Al Hammar. For this purpose, we employ data visualization techniques, such as data exploration (patterns, unusual observations, changes over time, or structural breaks). Different underlying assumptions concerning the estimate of data were employed in the hybrid machine learning models used in this research.

Our approach that combines different types of deep neural networks with probabilistic approaches to model uncertainty. Different kinds of deep learning networks, however, deep learning algorithms do not model uncertainty, the way Bayesian, or probabilistic approaches do. Hybrid learning models combine the two kinds to leverage the strengths of each. Our approach (CNN-BDLSTMs) combines CNN and BDLSTMs, and we find that it outperforms the other models.

Materials and proposed algorithm framework

Study area

The Mesopotamian Marshes of Southern Iraq are situated between 46.4° E and 48° E longitude and 30.5° N and 32.2° N latitude. The wetlands consist of shallow freshwater lakes with varying levels of permanence. The mean annual precipitation and mean annual temperature are less than 25 mm/year and 26.5 °C, respectively, based on the GLDAS study, which allows this land area to be classified as arid (Albarakat et al. 2018; Peltier 1950; Fookes et al. 1971).

Figure 1 shows the normalized difference moisture index (NDMI) for the Mesopotamian marshes in southern Iraq in 2000, 2010, and 2020 based on MODS satellite data. Images were taken in October, when the climatic conditions improved. Despite this, drought rates were high. The lowest soil moisture index was noted in 2000 (12%) due to drought and lack of rain; in 2010, the percentage improved due to re-flooding (30%); and the highest index was recorded in 2020 (56%).

Dataset

To analyze CRU data we installed the CRU TS interface to Google Earth Pro. Then we selected the area of study and loaded the relevant data. The dataset is updated annually and includes data from 1901 to 2020. The interface is available on CRU website https://crudata.uea.ac.uk/cru/data/hrg/cru_ts_4.02/ge/.

The highest monthly average in the Hawizeh Marsh was recorded in January 1974 (see Fig. 2). We noted high

Fig. 1 Location of the Mesopotamian marshes in Southern Iraq and NDMI for 2000, 2010, and 2020

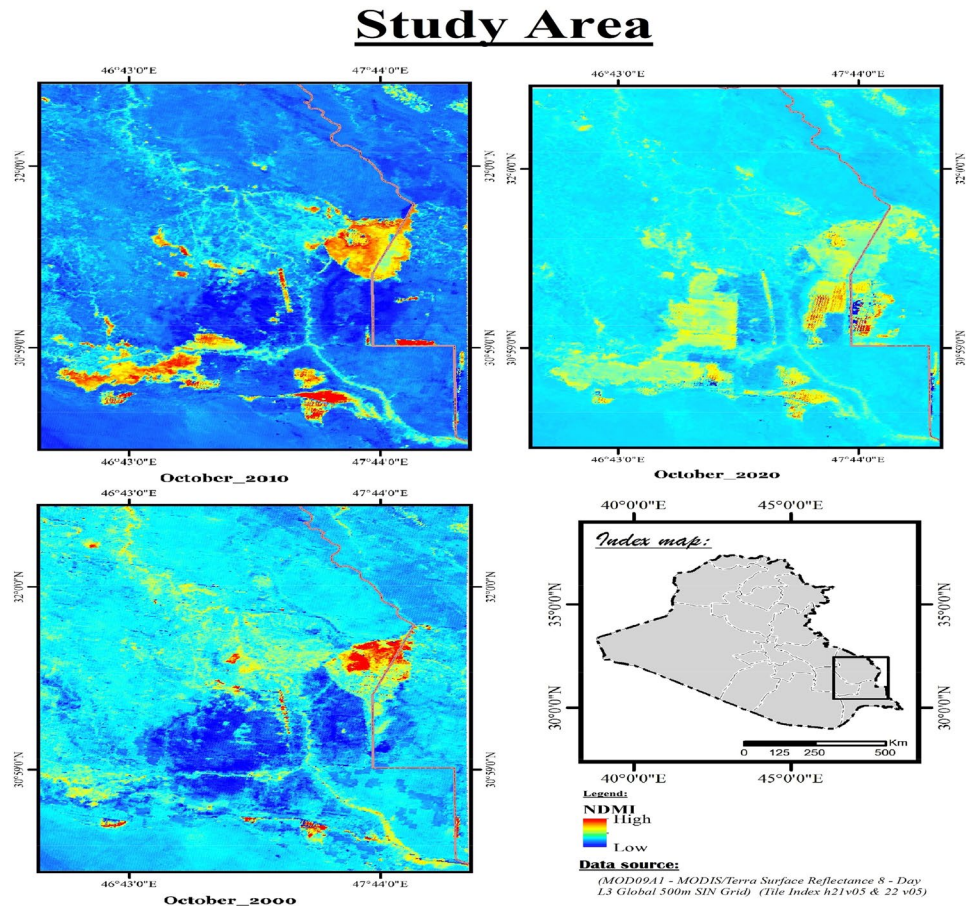


Fig. 2 Monthly average rainfall in the Hawizeh Marsh

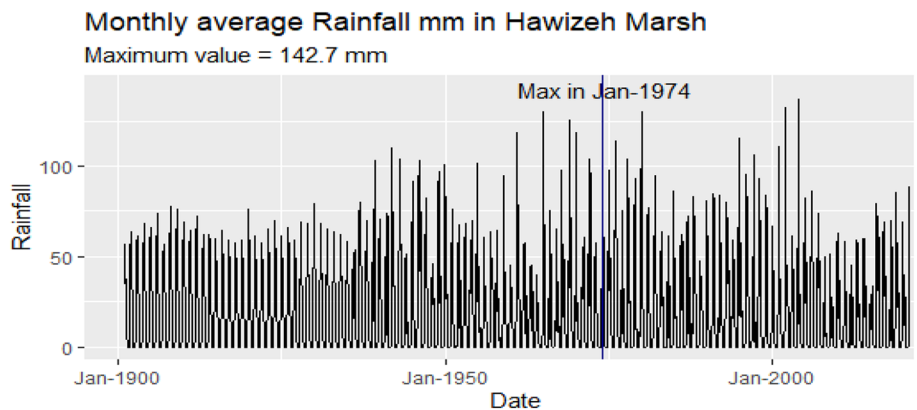


Fig. 3 Monthly average rainfall in the Central Marsh

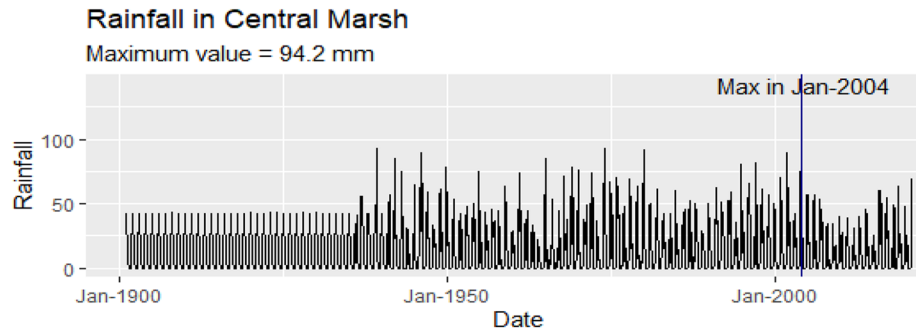
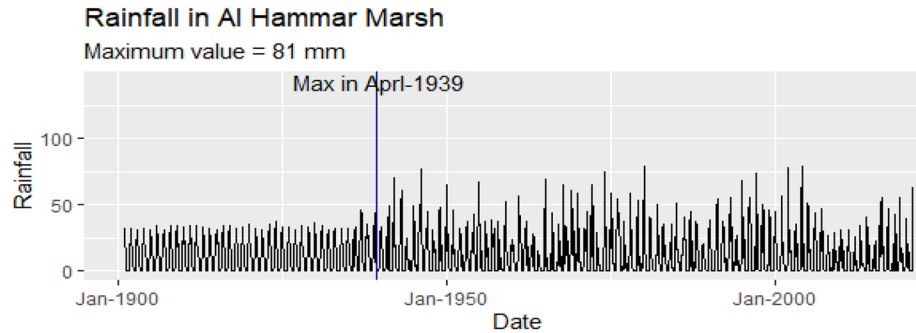


Fig. 4 Monthly average rainfall in the Al Hammar Marsh



volatility in rainfall after 1940, indicating a structural break at the variance level (shift). The highest rainfall levels in the Central Marsh were recorded in January 2004 (see Fig. 3). We noted a change in average rainfall after 1940, indicating a structural break at the monthly average level (location). The highest monthly average rainfall in the Al Hammar Marsh was recorded in April 1939. We noted a change in mean rainfall after 1940, indicating a structural break at the mean level (location) (Fig. 4).

Figure 5 shows the strength of the relationship between variables: there is a nearly perfect correlation between Al Hammar Marsh and Hawizeh Marsh and between central Marsh and Hawizeh Marsh, and a perfect correlation between central Marsh and Al Hammar Marsh. This indicates that the rainfall in one region coincides with the rainfall in the other regions.

Proposed algorithm framework

The mechanism underlying our proposed approach for modeling and forecasting average rainfall is depicted in Fig. 6. The algorithm consists of the following steps: CSV data and the developed hybrid deep learning algorithm are available on GitHub (Abotaleb 2022).

First step: Datasets on average rainfall in the Hawizeh Marsh, Central Marsh, and Al Hammar Marsh are generated in Google Earth Pro to train the algorithm. Then, the "cruts_4.06_gridboxes.kml" add-on interface is launched to display climatic data from January 1901 to December

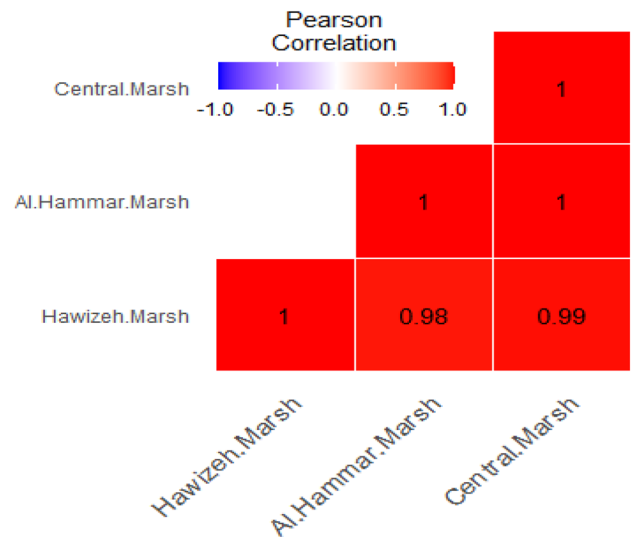
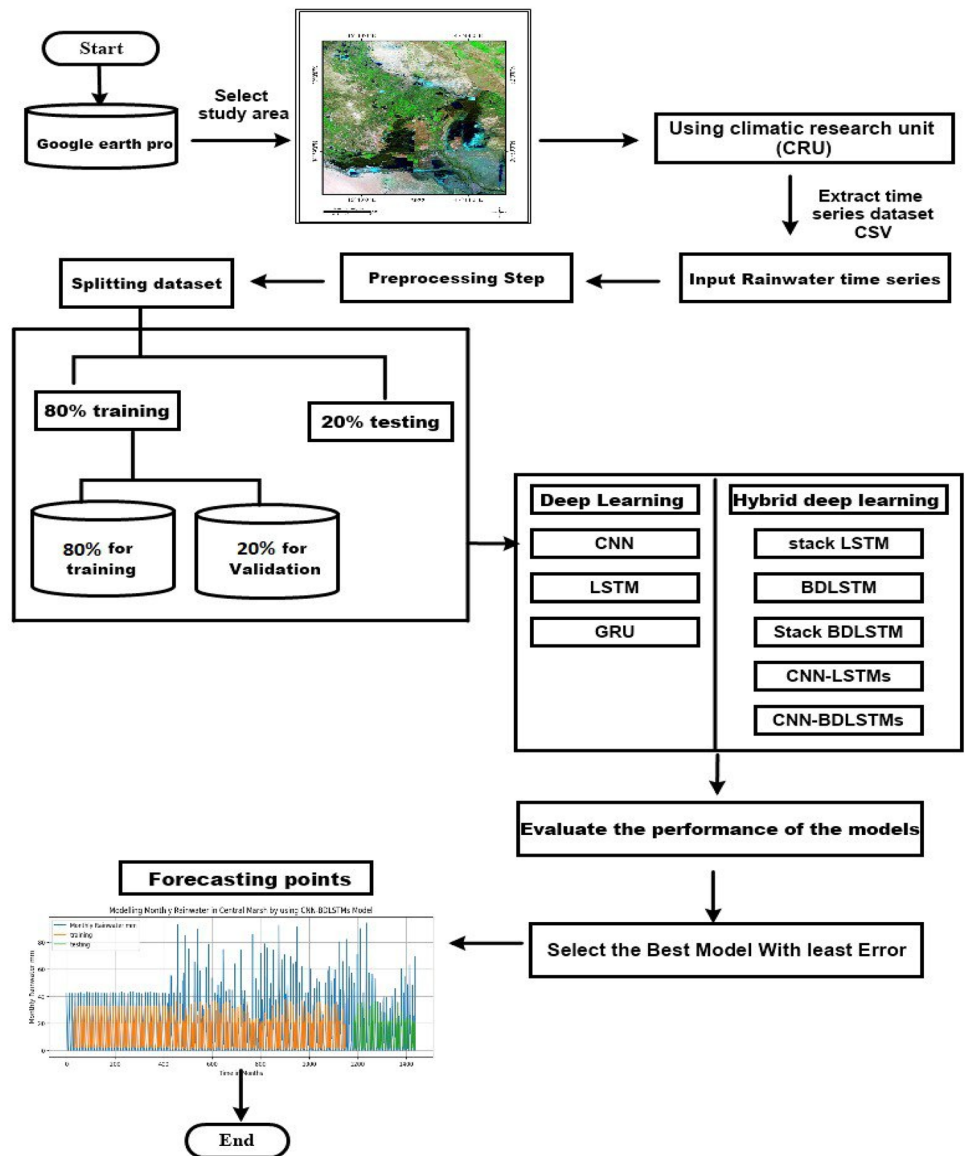


Fig. 5 Correlation heatmap

2020. We then load the average rainfall data for Hawizeh Marsh, Central Marsh, and Al Hammar Marsh. Detailed information about each rainfall dataset is stored in a separate CSV file. Each CSV file contains two columns: the first is the date, and the second is the average rainfall value. There are 1440 rows of data, resulting in a table of (1440). Al Hammar Marsh has a file size of 13.4 KB, Central Marsh—13.5 KB, and Hawizeh Marsh—13.7 KB.

Fig. 6 Schematic of the proposed algorithm framework



Second step: Input time series data for monthly average rainfall in Hawizeh Marsh, Central Marsh, and Al Hammar Marsh into our algorithm. Then the input parameters for the deep learning model (optimizer, loss function, number of epochs, and number of neural networks) are entered, and the algorithm is started.

Third step: Preprocessing and training require memory and time. Back propagating extended sequences create a trained model with poor performance. The data are prepared via normalization and standardization before being input into neural networks. Using data normalization, the standard deviation is set to 1 and the mean is set to 0.

Fourth step: The dataset is split into three sets namely, testing, validation, and training sets. The size of the test set is

20% of the dataset. The remaining 80% is split validation set of (20%) and training set of (80%). The model is trained using the training set to improve the model performance. During the training, the test for the overfitting is performed using the validation set. However, the model performance evaluation is performed using the test set.

Fifth step: The algorithms are executed for the CNN (Convolutional Neural Network), LSTM, LSTMs (Stacked LSTM), BDLSTM (Bidirectional LSTM), BDLSTMs (Stacked Bidirectional LSTM), GRU (Gated Recurrent Unit), Conv-LSTMs, and Conv-BDLSTMs models.

Sixth step: Evaluation of model performance.

Seventh step: Use best models in forecasting.

Methodology and optimization

In contrast to Bayesian and probabilistic approaches, deep learning algorithms do not account for uncertainty in their calculations. Several varieties of deep neural networks are integrated with probabilistic techniques to describe uncertainty in our suggested model. These hybrid models combine the best features of both types of deep learning networks. We discover that our suggested hybrid model outperforms the other models by combining a convolutional neural network (CNN) with bi-directional long short-term memory (BDLSTMs), and we call this method CNN-BDLSTMs. Both data and code come from (Abotaleb 2022).

Methodology

We use eight deep-learning models to forecast rainfall.

A. Convolutional neural network (CNN)

The convolutional neural network has a double-convolutional layer architecture to facilitate spatial advantage extraction. At each time step t , the flow data x_t^s is convolved with itself in a one-dimensional space. Specifically, the local perceptual domain is acquired using a sliding and one-dimensional convolution kernel filters (Graves et al. 2013). The method of the twisted kernel filter is demonstrated as follows:

$$Y_t^s = \sigma(W_s * x_t^s + b_s), \quad (1)$$

where Y_t^s is the convolutional layer output; W_s is the filter weights; x_t^s is the input traffic flow at time t ; and σ is the activation functions.

Because CNNs can be trained to recognize patterns in time series data and utilize that information to make predictions about the future, they are a valuable tool for anybody working with this type of information (Koprinska et al. 2018). CNN can also automatically recognize and capture features from class data without presuppositions and feature ordering. They may also work well with time series containing high noise by filtering out the noise in each subsequent layer, generating a set of useful information and features and extracting only meaningful features (Koprinska et al. 2018).

After moving the input information to the Conv layer, a ReLU is used to extract patterns. Then the max pooling layer is used to reduce the number of parameters and move the information to a lower dimension. Flatten is used in Keras to normalize data to the number of elements in the tensor. These mechanisms are displayed in Fig. 7 below.

According to the model's processing mechanism for the data from Fig. 7, the model is trained to extract data patterns with the training stopping at the point that achieves maximum accuracy and least information loss. The model weights are randomly optimized by improving the accuracy of the training data in the model from one layer to another using the keras time series data generator (Muftah et al. 2022).

B. Long short-term memory (LSTM)

When solving the problem of vanishing gradients, long short-term memory (LSTM) was one of the earliest and most effective methods to be developed (Hochreiter and Hochreiter 1997; Gers et al. 2002). In this context, "long-term" refers to simple recurrent neural networks storing information about their previous decisions as weights. A gradual shift in weights occurs throughout training when new information about the data is retrieved and used to calibrate the model. Short-lived activations hop from one node to another and are therefore referred to here. In the LSTM paradigm, a memory cell serves as intermediate storage. For the first time, multiplex nodes are incorporated into the construction of memory cells, making them a more complicated unit. Three gates (input, output, and forget) make up a generic LSTM unit (Huynh et al. 2017). With the input gate, LSTM may be programmed to either retain existing data or learn new information. The sigmoid layer and the tanh layer make up this gate's structure. The tanh layer generates a vector of potential new values to be added to LSTM (Zhang, et al. 2019a, b), whereas the sigmoid layer specifies which values will be modified. To derive the final result from these layers, we use:

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i), \quad (2)$$

$$u_t = \tanh(W^u x_t + U^u h_{t-1} + b^u), \quad (3)$$

where i_t is the updated value; u_t is new candidate values; σ is the sigmoid layer (or nonlinear function); x_t is a sequence of

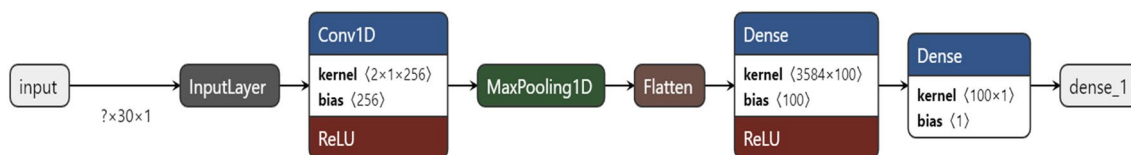


Fig. 7 Architecture and hyper parameters of the proposed convolutional neural network (CNN)

length t ; b is constant bias; h is RNN memory at time step t ; and W and U are weight matrices.

Forget gates, whose sigmoid functions are used to choose data for deletion from LSTM, are discussed in detail in (Song et al. 2020). The values of h and x_t are used heavily in making this determination. This gate has an output f that may take on the values 0 and 1, where 0 signifies full erasure of the acquired value and 1 represents complete preservation of the value. This result is derived by:

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f), \tag{4}$$

where f_t is updated value; σ is the sigmoid layer (or non-linear function); x_t represents a sequence of length t ; b is constant bias; h represents RNN memory at time step t ; and W and U are weight matrices.

The input gate uses a sigmoid layer to determine which sub-tree of the LSTM is responsible for the output. A non-linear tanh function is then used to assign values between -1 and 1 after that. The output from the sigmoid layer is then multiplied by the final product. Following are some formulae that are used to determine output:

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o), \tag{5}$$

$$h_t = o_t \tanh_t c_{t-1}, \tag{6}$$

where o_t is an output gate and h_t is a value between $[1, -1]$.

The LSTM is kept current by combining these two layers. The forget gate layer works by first doubling the previous value, c_{t-1} , and then adding the candidate value, $i_t u_t$, to

forget the current value. Specifically, this process requires the following equation:

$$c_t = i_t u_t + f_t c_{t-1}, \tag{7}$$

where c_t represents a memory cell and f_t represents a value between 0 and 1 produced by the forget gate. Specifically, a value of 0 denotes that the value is nullified, whereas a value of 1 indicates that it is retained (Van Houdt et al 2020). Figure 8 depicts a potential configuration including these components.

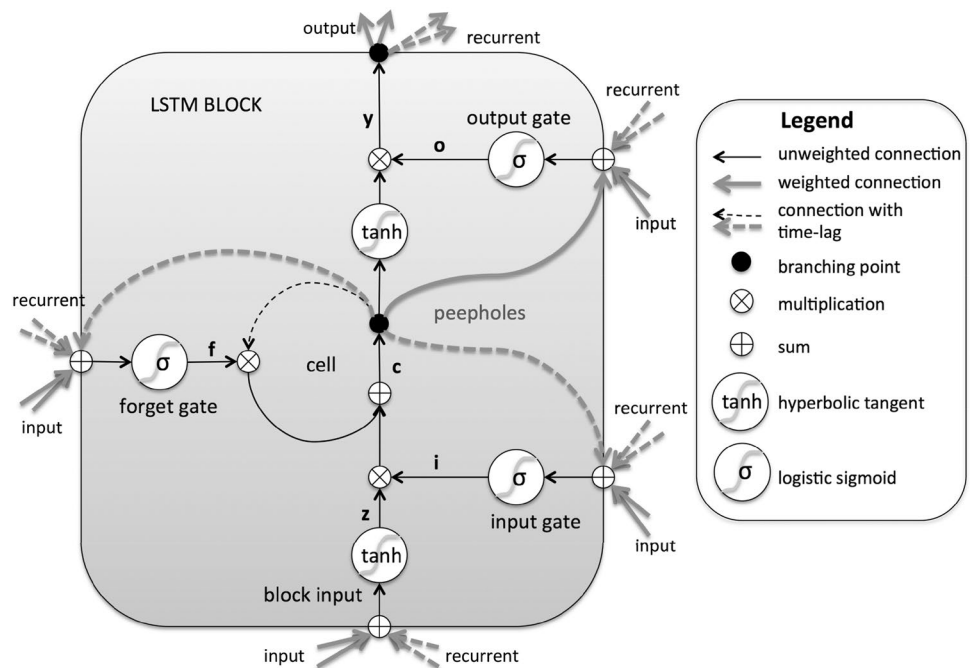
In the LSTM model, the input information is passed to the forget layer, at which point the model decides to: (a) keep the information in the past and use it for prediction, or (b) forget the information and rely on the instantaneous state, then send this information to a tanh function to normalize the information and extract features and patterns and remove noise from them (Reddy and Prasad 2018).

Figure 9 shows the characteristics of the kernel used to run the LSTM model which is used to fit the model to the training data, the memory used to store information, and key features in the data and used for forecasting.

C. Stacked long short-term memory (LSTMs):

Graves et al. (2013) first proposed this model after concluding that the number of memory cells in a given layer is less essential than the network depth for data modeling and pattern extraction. There are several nested layers in the stacked LSTM model, and each houses numerous memory nodes. Instead of sending a single value to the LSTM layer below, a

Fig. 8 The long short-term memory (LSTM) model (Mohamed and Hussain 2016)



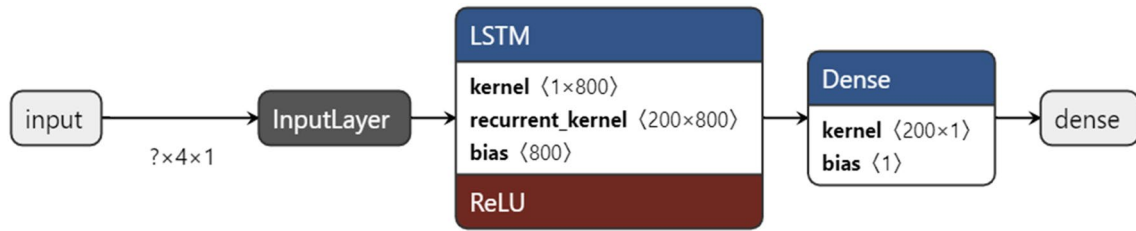


Fig. 9 Schematic and hyper parameters of the proposed LSTM model

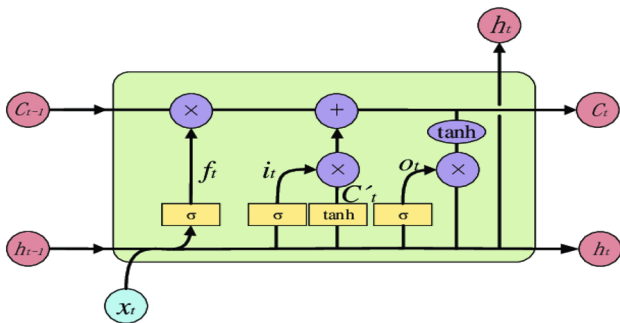


Fig. 10 A stacked LSTM architecture (Muftah et al. 2022)

stacked LSTM sends a sequence. In other words, rather than having a single output time step for all input time steps, there is one output per input time step (Cui et al. 2020).

Figure 10 shows the structure of a stacked LSTM; the mechanism is similar to the LSTM model, but with several layers $f, i, o(\sigma)$, which allows for additional features to be extracted from the data.

Figure 11 shows the properties of the kernel used to run the Stacked LSTM model and size of the storage memory for information storage in preparation for the production of predictions. As shown, the model is adapted to the training data in more than one layer, which allows the extraction of highly complex information (Dikshit et al. 2021).

D. Bidirectional long-short term memory model (BDLSTM):

The Bi-LSTM model integrates the strengths of two separate RNNs. With this setup, the network may exchange

sequence-related data in both directions at each time interval (Fernández et al. 2007). The input data is processed in both directions by the Bi-LSTM, from the future to the past and back again. If you utilize LSTM for your backwards estimations, you may save your future-oriented data and use the two hidden states in combination at any time. That way, we would not lose any of the knowledge from the past or the future (Shahid et al. 2020). The expression for the output y at time t is:

$$y_t = \sigma(W_y [h_t^+, h_t^-] + b_y), \tag{8}$$

where σ is nonlinear function; W_y are weight matrices that are used in deep learning model; b_y is a constant bias; and h_t is hidden states.

In the BDLSTM model, the hidden state h_t works to receive information from the past and future x_t and take advantage of these patterns in prediction y_t (see Fig. 12).

To forecast the future value of a variable, y_t , a kernel is used to extract features from a non-linear function (kernel) that is fed information from a time series of inputs, x_t , both past and future (see Fig. 13). These models allow full sequence information to be retrieved for all points before or after a given point in the sequence using a bidirectional recurrent neural network, which helps to improve prediction accuracy in some areas where past and future data are important (Zhang et al. 2022).

Bidirectional stacked long short-term memory model (BD-LSTM):

This model combines the features of the BD and LSTM models, allowing the user to obtain information about the sequence forward and backward at each time step (Fernández et al. 2007). The model provides multiple sequential

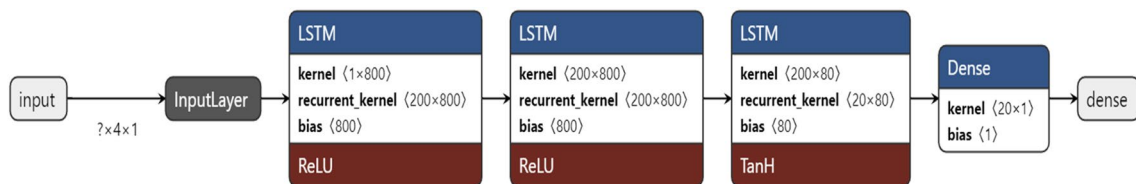


Fig. 11 Architecture and hyper parameters of the proposed stacked long short-term memory (LSTMs) model

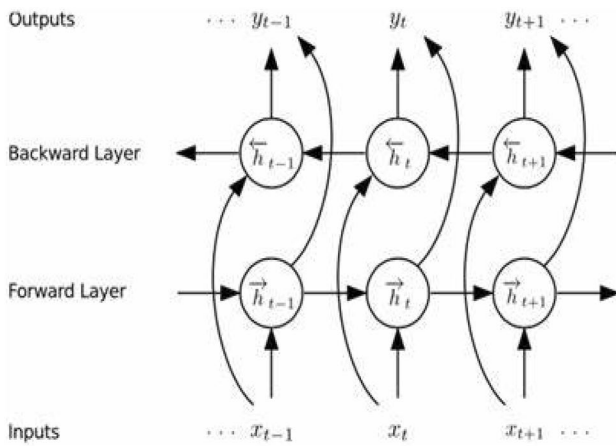


Fig. 12 Bidirectional long-short term memory model (BDLSTM)

values instead of a single value output to the LSTM layer (Shahid et al. 2020).

E. Stacked bidirectional long short-term memory model (BDLSTMs)

The BDLSTMs model uses information from the past and future with multiple LSTM layers for processing (see Fig. 14).

The BDLSTMs model processes information using the same method as the BDLSTM model with several layers from LSTM (see Fig. 15) (Biswas and Sinha 2021).

F. Gated recurrent unit model (GRU):

Compared to LSTM, the Gated Recurrent Unit (GRU) is far superior. In the same vein, this is a recurrent neural network. In comparison to LSTM, which employs three hyper parameters, RNG only needs two (a reset gate and an update gate) (Dey and Salem 2017). When deciding what data should be transmitted to the output, the update gate and reset gate act as vectors (Gulli and Pal 2017). The reset gate sets the

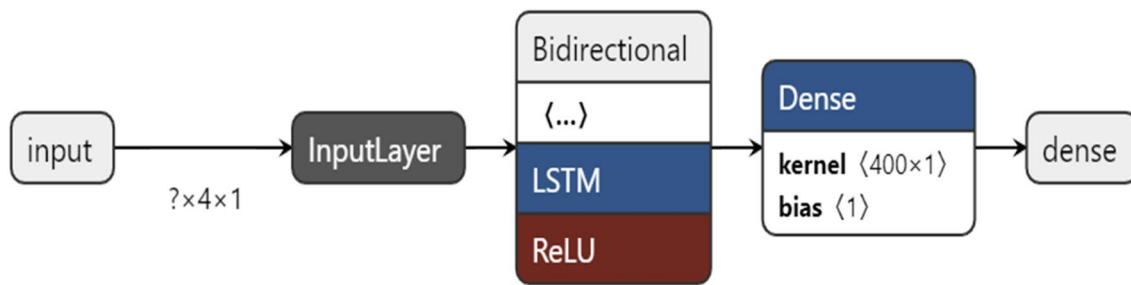
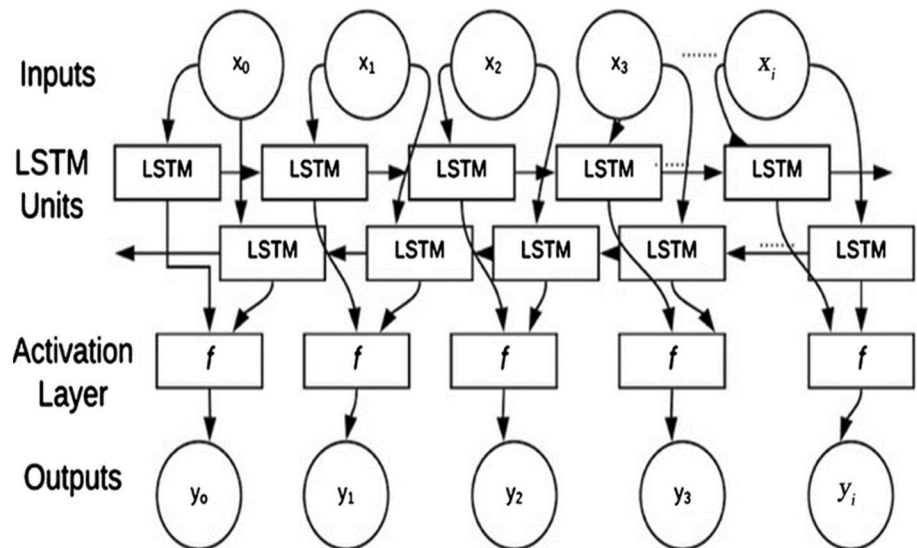


Fig. 13 Architecture and hyper parameters of the proposed bidirectional long-short term memory model BDLSTM

Fig. 14 Stacked bidirectional long short-term memory model (BDLSTMs)



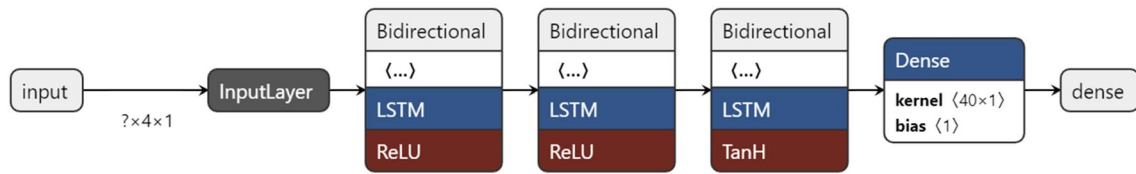


Fig. 15 Architecture and hyper parameters of the proposed stacked bidirectional long short-term memory model (BDLSTM)

amount of state that should be retained. The update gate then decides if the new state is an exact replica of the previous one. Two sigmoid-activation-function-equipped, fully-connected layers provide two gate outputs. All of the GRU (Wang et al. 2018) inputs are depicted in Fig. 6, including those for the reset and update gates. For a mathematical analysis of output, we have:

$$r_t = \sigma(W^r x_t + U^r h_{t-1} + b^r), \tag{9}$$

$$z_t = \sigma(W^z x_t + U^z h_{t-1} + b^z), \tag{10}$$

where r_t represents the reset gate, z_t represents the update gate, h_{t-1} represents the hidden state from the previous time step, σ represents the sigmoid activation function, W and U represent weight parameters, and b represents a constant bias. Then, we join the reset gate with the standard refresh system:

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i), \tag{11}$$

which leads the hidden state; the next candidate:

$$a_t = \tanh(w x_t + r_t U^i h_{t-1} + b^h), \tag{12}$$

where r_t is the reset gate, h_{t-1} is the hidden state from the preceding time step, w and U are weight parameters, \tanh is the activation function, and b is a constant bias. Last but not least, the update gate's impact must be factored in. This evaluates the degree of similarity between the current

hidden state and the previous state, as well as the similarities between the current hidden state and the candidate states. By selecting convex combinations of elements h_t and h_{t-1} element-wise, the update gate may be employed for this purpose (Seidu et al. 2022). The following equation is the result of this process and represents the final GRU update:

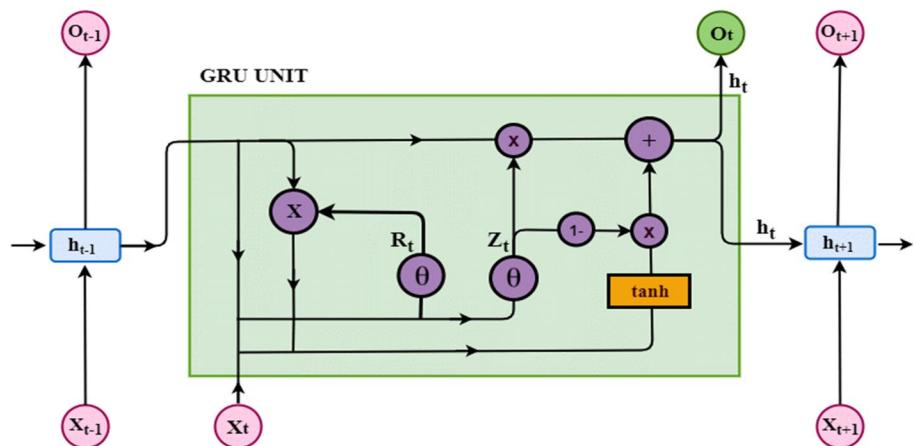
$$h_t = z_t h_{t-1} + (1 - z_t) a_t \tag{13}$$

where z_t the update is gate; r_t is the reset gate; a_t is the activation function; and h_t is the hidden state output gate.

The input x_t is sent to update gate z_t , then to reset gate r_t , and then to activation function \tanh , where the information properties are extracted less excessively (see Fig. 16).

The GRU model uses a recount to process information, which allows access to a shorter form of the previous models (see Fig. 17). The most prominent feature shared between LSTM and GRU model is the additive component of their update from t to $t + 1$, which is lacking in the traditional recurrent unit. The traditional recurrent unit always replaces the activation, or the content of a unit with a new value computed from the current input and the previous hidden state. On the other hand, both LSTM unit and GRU keep the existing content and add the new content on top of it (Chung et al. 2014). These two units however have a number of differences as well. One feature of the LSTM unit that is missing from the GRU is the controlled exposure of the memory content. In the LSTM unit, the amount of the memory content that is seen, or used by other units in the network is controlled by the output gate.

Fig. 16 Gated recurrent unit (GRU) layer



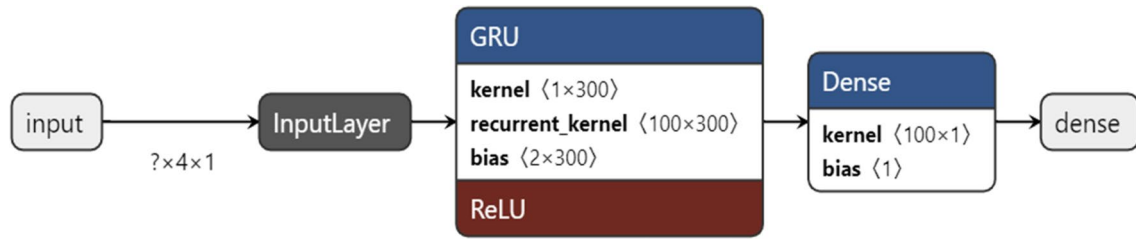


Fig. 17 Architecture and hyper parameters of the proposed gated recurrent unit (GRU) layer

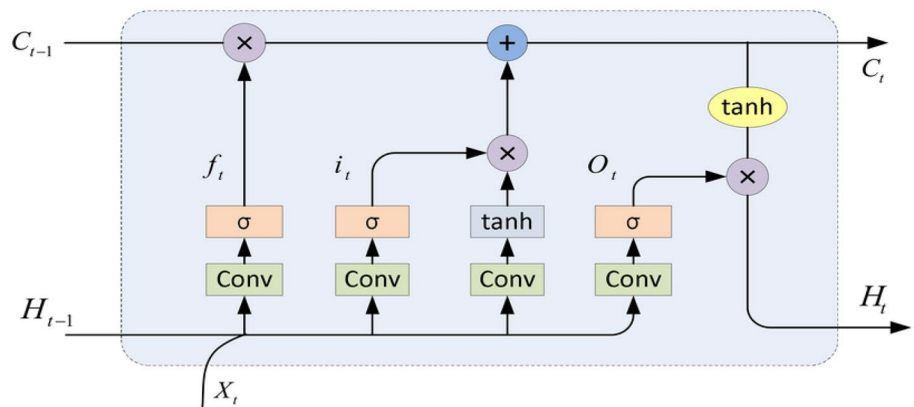
On the other hand the GRU exposes its full content without any control. Another difference is in the location of the input gate, or the corresponding reset gate. The LSTM unit computes the new memory content without any separate control of the amount of information flowing from the previous time step. Rather, the LSTM unit controls the amount of the new memory content being added to the memory cell independently from the forget gate. On the other hand, the GRU controls the information flow from the previous activation when computing the new, candidate activation, but does not independently control the amount of the candidate activation being added (the control is tied via the update gate) (Gaudio et al. 2021).

G. Convolutional neural network long-short term memory model (CNN-LSTM):

When combining Conv and LSTM, we get the CNN-LSTM Model, which takes as input a spatial–temporal traffic flow matrix of the form x_t^s (Mallah and Bagheri-Bodaghabadi 2022):

$$x_t^s = \begin{bmatrix} x_{t-n}^s \\ x_{t-(n-1)}^s \\ \vdots \\ x_t^s \end{bmatrix} \begin{bmatrix} f_{t-n}^1 & f_{t-(n-1)}^1 & \dots & f_t^1 \\ f_{t-n}^2 & f_{t-(n-1)}^2 & \dots & f_t^2 \\ \vdots & \vdots & \ddots & \vdots \\ f_{t-n}^m & f_{t-(n-1)}^m & \dots & f_t^m \end{bmatrix} \quad (14)$$

Fig. 18 The CNN-LSTM model is a combination of Conv and LSTM (Livieris et al. 2020)



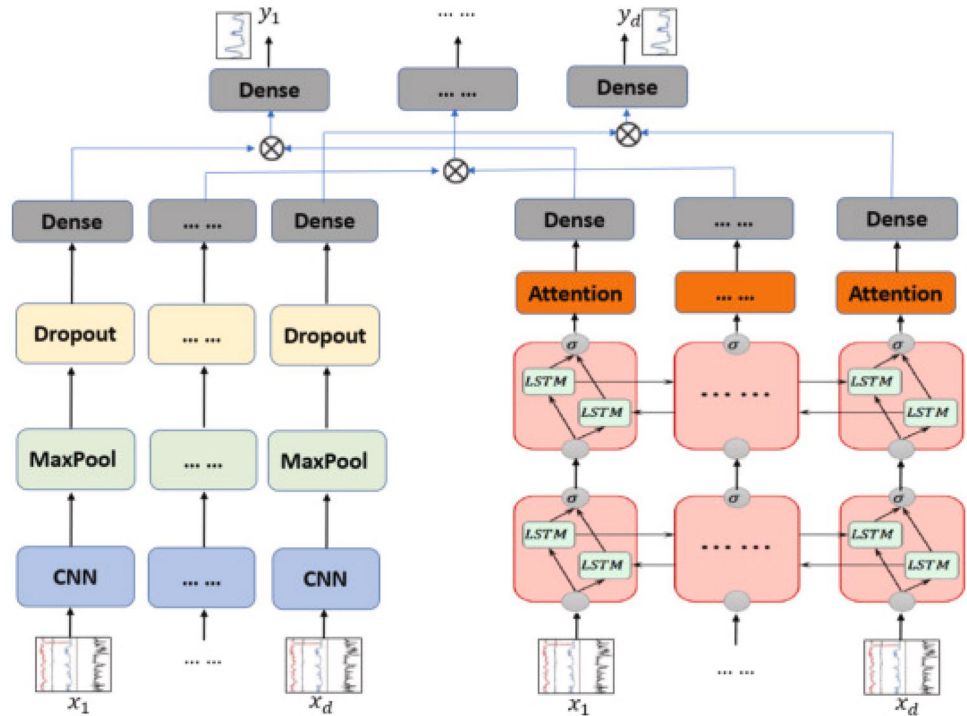
where $x_t^s = f_t^1 \dots f_t^m$ denotes the prediction region’s traffic flow at time t , representing the POI’s historical traffic flow to be forecasted and its neighbors (Livieris et al. 2020).

In the LSTM model, the initial estimation stage is a CNN layer (see Fig. 18), and the last level of estimation is an LSTM layer with a dense layer. Each time unit is processed by the LSTM model, which is responsible for interpreting steps, while the CNN model is responsible for extracting relevant data (see Fig. 18). The CNN-LSTM neural network architecture allows the hidden relationships to be automatically captured and used for prediction, which may lead to the method being more applicable and easy to implement (Zha et al. 2022).

H. Convolutional neural network bidirectional long-short term memory model (CNN-BDLSTM)

By utilizing CNN to capture characteristics and then feeding those features into a BDLSTM model, this model can fully use the capabilities of both models. In the next step, the outputs from each max pooling layer are combined to generate the BDLSTM input, before the layer’s three gates are used to perform a recursive backpropagation-style filtering operation. Input to the fully connected layer (Lu et al. 2021), which connects each input to a subset of the output (Nie et al. 2021; Casallas, et al. 2022), is the result of this stage.

Fig. 19 Mechanism of the CNN-BDLSTM model (Nie et al. 2021)



It can be seen how the CNN-BDLSTM model works in Fig. 19. For instance, the BDLSTM model lets you get forward and backward information about the sequence at each time step, while the CNN model is utilized to extract the relevant information. In the left panel, we can see the first CNN layer, followed by the subsequent LSTM layers and finally, the dense layer at the very end (right panel).

Optimization: Adam optimization algorithm

Adam optimization is an extension of stochastic gradient descent that allows for more effective updates to network weights. Adam optimization arises from a two-factor interaction (RMSprop and Momentum, see Fig. 19 and Pseudocode 1). In the field of stochastic optimization, adaptive moment estimation is employed (Jais et al. 2019; Kim and Choi 2021). Since Adam optimization also displays this behavior, understanding how the pace of learning might change over time is crucial.

Adam optimization is the stochastic optimization algorithm proposed in this work. The elementwise square g_t^2 is calculated for $g_t \odot g_t$. The default values are set as: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The elementwise operation is applied for all vectors. With β_1^t and β_2^t we denote β_1 and β_2 to the power t (Kingma and Ba 2015).

```

Require:  $\alpha$ : Stepsize
Require:  $f(\theta)$ : Stochastic objective function with parameters  $\theta$ 
Require:  $\beta_1, \beta_2 \in [0, 1)$ : Exponential decay rates for the moment estimates
Require:  $\theta_0$ : Initial parameter vector:
Initialize timestep:  $t \leftarrow 0$ 
Initialize 2nd moment vector:  $v_0 \leftarrow 0$ 
Initialize 1st moment vector:  $m_0 \leftarrow 0$ 
while  $\theta$  not converged do
     $t + t_1$ 
     $g_t \leftarrow \nabla_{\theta} f(\theta_{t-1})$  (Get gradients w.r.t. stochastic objective at timestep  $t$ )
     $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update biased first moment estimate)
     $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update biased second raw moment estimate)
     $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)
     $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)
     $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)
end while
return  $\theta_t$  (Resulting parameters)
    
```

Adaptive moment estimation (Adam)
Pseudocode: Adam algorithm for stochastic optimization
Note:
There are two separate beta coefficients \rightarrow one for each optimization component
We implement bias correction for each gradient

```

On iteration t:
Compuе dW, db for current mini-batch
# #Momentum
v_db=beta1 * v_db+(1 - beta1) db, v_db_corrected=v_db/(1 -
beta1 ** t)
v_dW=beta1 * v_dW+(1 - beta1) dW, v_dW_corrected=v_dW/
(1 - beta1 ** t)
# #RMSprop
s_dW=beta * v_dW+(1 - beta2) (dW ** 2), s_dW_corrected=s_
dW/(1 - beta2 ** t)
s_db=beta * v_db+(1 - beta2) (db ** 2), s_db_corrected=s_db/
(1 - beta2 ** t)
# #Combine
W=W - alpha * (v_dW_corrected/(sqrt(s_dW_corrected)+epsi-
lon))
b=b - alpha * (v_db_corrected/(sqrt(s_db_corrected)+epsilon))
Coefficients
alpha: the learning rate. 0.001
beta1: momentum weight. Default to 0.9
beta2: RMSprop weight. Default to 0.999
epsilon: Divide by Zero failsave. Default to 10 ** -8
    
```

Overfitting and under fitting

Overfitting and under fitting are a major contributing factor to poor performance in deep learning models. In overfitting the model (which performs consummately on the training set while fitting ineffectively on the testing set) the model begins by matching the noise to the estimation data and parameters, thus producing predictions with large out-of-sample errors that negatively impact the model’s ability to generalize. An over fit model shows low bias and high variance (He et al. 2016). Under fitting refers to the model’s inability to capture all the data’s characteristics and features, resulting in poor performance on the training data and an inability to generalize the model’s results (Zhang et al. 2019).

To avoid and detect overfitting and under fitting, we tested the validity of the data by training the model on 80% of the data subset and testing the other 20% using the set of performance indicators (Alqahtani et al. 2022; Abotaleb and Makarovskikh 2021) detailed in the next section.

Performance indicators:

To compare the prediction performance of the three models we:

Calculated mean square error (MSE):

$$\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n} \tag{15}$$

where \hat{y}_t the forecast is value; y_t is the actual value; and n is the number of fitted observed.

Calculated root mean square error (RMSE) between the estimated data and actual data:

$$\sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}} \tag{16}$$

where \hat{y}_t is the predicted value; y_t is the actual value; and n is number of fitted observed.

Calculated relative root mean square error (RRMSE):

$$\sqrt{\frac{\frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2}{\sum_{t=1}^n (\hat{y}_t)^2}} \tag{17}$$

Calculated mean absolute error (MAE):

$$\frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \tag{18}$$

Calculated mean bias error (MBE):

$$\frac{\sum_{t=1}^n (y_t - \hat{y}_t)}{n} \tag{19}$$

Calculated optimum loss error:

$$loss(y_t, \hat{y}_t) = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|^2 \tag{20}$$

The model with the lowest values of (RMSE – RRMSE – MAE – MBE – loss) is the best.

Results

Table 1 shows that mean > median > mode, which shows that the distribution is skewed to the right for all variables. Observations with a value larger than mean are more frequent. Kurtosis < 3 for all variables, indicating that there are no extreme outliers. The greatest difference between the maximum and minimum value for rainfall was noted in Hawizeh Marsh (0 mm to 142.7 mm). This lead to a larger

Table 1 Descriptive statistics of rainfall

	Mean	S.E.	Median	Mode	S.D.	Kurtosis	Skewness	Mini	Max
Hawizeh Marsh	21.37	0.70	7.9	0	26.42	1.57	1.36	0	142.7
Central Marsh	14.91	0.48	6.3	0	18.23	1.69	1.37	0	94.2
Al Hammar Marsh	11.99	0.40	5	0	15.08	2.17	1.48	0	81

S.D. and S.E (difficulty in prediction) than the rest of the variables.

Table 2 shows that CNN-BDLSTMs was the best model for predicting rainfall because it has the least values of MSE – RMSE – RRMSE – MAE – MBE – Optimum Loss Error and, therefore, the least difference between the actual and predicted values. This model achieves convergence between the training and test data's actual and predicted values, demonstrating their ability to capture data features.

Figures 20, 21 and 22 show the convergence of actual monthly rainfall in the Hawizeh Marsh, Central Marsh, and Al Hammar Marsh with the values predicted by the CNN-BDLSTMs model. There is good convergence between the actual and predicted data. This model is able to clarify volatility in rainfall and capture structural breaks, and can thus be used to predict monthly rainfall in this region.

Conclusion

Climate change has impacted Wetlands due to increased annual average maximum temperature and decreased rainfall. Since Google Earth Pro data has great potential for detecting changes that have already occurred, it can be used to monitor the climatic elements of marshes and water bodies. The Mesopotamian marshes are vital to Iraq's ecology and economy, so it is crucial to take measures to develop them and return them to their original state. We aim to continue our research in this field by developing a model for predicting monthly average rainfall which incorporates data on sea-surface temperature, global wind circulation, and a variety of other climatic variables. We described deep learning approaches for monthly average rainfall forecasting and proposed a hybrid deep learning CNN-BDLSTMs-based model for Hawizeh Marsh, Central Marsh, and Al Hammar Marsh. The dataset includes average monthly records for meteorological parameters such as maximum and minimum temperatures, precipitation, evaporation, and monthly average rainfall from Google Earth Pro for 1901 to 2020. Our

Table 2 Comparison of dataset evaluation methods(20%)

Model	MSE	RMSE	RRMSE	MAE	MBE	Optimum loss error
Monthly rainfall in the Hawizeh Marsh (80–20%)						
CNN	426.02	20.64	1.10	13.00	0.07	16.90
LSTM	530.83	23.04	1.21	14.24	3.31	937.52
LSTMs	549.41	23.44	1.23	13.98	3.42	966.59
BDLSTM	649.36	25.48	1.34	15.56	2.81	794.68
BDLSTMs	487.52	22.08	1.16	14.22	6.38	1805.91
GRU	414.67	20.36	1.07	13.50	2.76	781.66
CNN-LSTMs	407.92	20.20	1.07	14.14	6.44	1654.91
CNN-BDLSTMs	315.29	17.76	0.94	11.15	3.48	895.18
Monthly rainfall in the Central Marsh (80–20%)						
CNN	239.61	15.48	1.18	9.94	−0.44	−114.18
LSTM	297.59	17.25	1.30	10.54	2.45	693.57
LSTMs	231.50	15.22	1.15	9.64	0.92	261.14
BDLSTM	207.65	14.41	1.09	9.27	0.91	257.05
BDLSTMs	246.82	15.71	1.18	9.45	0.98	278.71
GRU	201.71	14.20	1.07	9.49	1.46	413.86
CNN-LSTMs	193.84	13.92	1.06	9.22	4.41	1132.20
CNN-BDLSTMs	166.18	12.89	0.99	8.03	−0.49	−125.15
Monthly rainfall in the Al Hammar Marsh (80–20%)						
CNN	173.45	13.17	1.25	8.75	0.73	187.60
LSTM	158.57	12.59	1.17	8.59	0.36	101.95
LSTMs	159.78	12.64	1.18	7.95	−0.28	−77.89
BDLSTM	180.02	13.42	1.25	9.09	2.61	737.29
BDLSTMs	171.39	13.09	1.22	8.02	−0.51	−143.87
GRU	156.46	12.51	1.16	8.18	0.63	178.15
CNN-LSTMs	151.99	12.33	1.17	7.96	1.42	364.60
CNN-BDLSTMs	133.59	11.56	1.10	7.16	−0.14	−34.86

The bold colour is mean the best model that has the least errors, and the is the output performance model

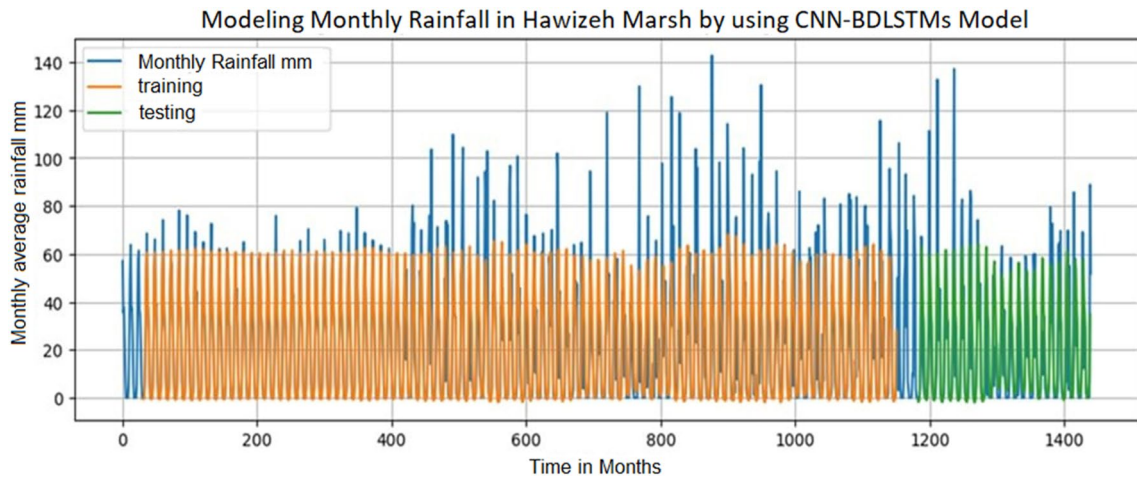


Fig. 20 The CNN-BDLSTMs model for forecasting monthly average rainfall in the Hawizeh Marsh

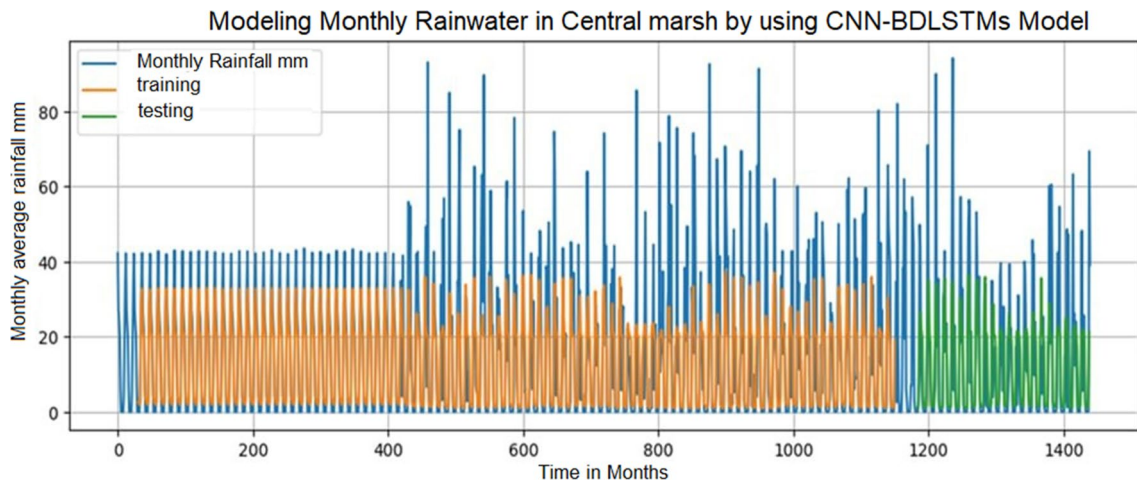


Fig. 21 The CNN-BDLSTMs model for forecasting monthly average rainfall in Central Marsh

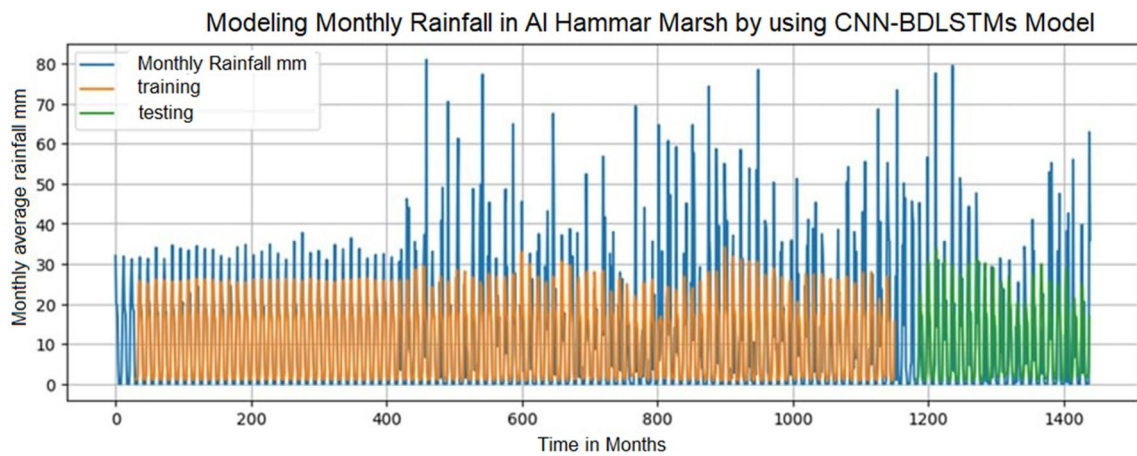


Fig. 22 The CNN-BDLSTMs model for forecasting monthly average rainfall in the Al-Hammar Marsh

tests showed that the proposed prediction model is accurate. Smart farming and other applications that require accurate rainfall forecasts might benefit from this model.

Data availability The datasets generated during and analysed during the current study are available in the Hybrid deep learning models algorithm for modelling and forecasting rainwater in Wetlands in south repository, <https://github.com/abotalebmostafal1/Hybrid-deep-learning-models-algorithm-for-modelling-and-forecasting-rainwater-in-Wetlands-in-south-I>.

References

- Abotaleb M, Makarovskikh T (2021) Analysis of neural network and statistical models used for forecasting of a disease infection cases. In: International conference on information technology and nano-technology (ITNT), pp 1–7. IEEE, Samara. <https://doi.org/10.1109/ITNT52450.2021.9649126>
- Abotaleb M (2022) Hybrid deep learning models algorithm for modelling and forecasting rainwater in Wetlands in south Iraq. <https://github.com/abotalebmostafal1/Hybrid-deep-learning-models-algorithm-for-modelling-and-forecastingrainwater-in-Wetlands-in-south-I>
- Adham A (2018) A GIS-based approach for identifying potential sites for harvesting rainwater in the Western Desert of Iraq. *Int Soil Water Conserv Res* 6(4):297–304. <https://doi.org/10.1016/j.iswcr.2018.07.003>
- Albarakat R, Lakshmi V, Tucker C (2018) Using satellite remote sensing to study the impact of climate and anthropogenic 561. *Remote Sensing, Iraq*
- Al-Handal A, Hu C (2015) Modis observations of human-induced changes in the mesopotamian marshes in Iraq. *Wetlands* 35:31–40
- Alhumaima A, Abdullaev M (2020) Tigris basin landscapes: sensitivity of vegetation index NDVI to climate variability derived from observational and reanalysis data. *Earth Interact* 24(7):1–18. <https://doi.org/10.1175/EI-D-20-0002.1>
- Alqahtani F, Abotaleb M, Kadi A, Makarovskikh T, Potoroko I, Alakkari K, Badr A (2022) Hybrid deep learning algorithm for forecasting SARS-CoV-2 daily infections and death cases. *Axioms* 11:620. <https://doi.org/10.3390/axioms11110620>
- Awchi T, Jasim I (2017) Rainfall data analysis and study of meteorological draught in Iraq for the period 1970–2010. *Tikrit J Eng Sci* 24(1):110–121. <https://doi.org/10.25130/tjes.24.2017.12>
- Biswas S, Sinha M (2021) Performances of deep learning models for Indian Ocean wind speed prediction. *Model Earth Syst Environ* 7(2):809–831
- Casallas A, Ferro C, Celis N, Guevara-Luna M, Mogollón-Sotelo C, Guevara-Luna F, Merchán M (2022) Long short-term memory artificial neural network approach to forecast meteorology and PM2.5 local variables in Bogotá, Colombia. *Model Earth Syst Environ* 8(3):2951–2964
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv*, 1412.3555.
- Cui Z, Ke R, Pu Z, Wang Y et al (2020) Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network wide traffic state with missing values. *Transport Res Part C Emerg Technol* 1:118
- Dey R, Salem F (2017) Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), pp 1597–1600
- Dikshit A, Pradhan B, Alamri A (2021) Long lead time drought forecasting using lagged climate variables and a stacked long short-term memory model. *Sci Total Environ* 755:142638
- Douabul A, Al-Saad H, Abdullah D, Salman N (2013) Designated protected marsh within mesopotamia: water quality. *Water Resour* 1:39–44
- Fernández S, Graves A, Schmidhuber J (2007) An application of recurrent neural networks to discriminative keyword spotting. *International conference on artificial neural networks*. Springer, Berlin, pp 220–229
- Fitzpatrick R (2004) Changes in soil and water characteristics of natural, drained and re-flooded soils in the mesopotamian marshlands: implications for land management planning. In: Client report. CSIRO land and water, Canberra
- Fookes P, Dearman W, Franklin J (1971) Some engineering aspects of rock weathering with field examples from Dartmoor and elsewhere. *Q J Eng Geol Hydrogeol* 4:139–185
- Gaudio M, Coppola G, Zangari L, Curcio S, Greco S, Chakraborty S (2021) Artificial intelligence-based optimization of industrial membrane processes. *Earth Syst Environ* 5(2):385–398
- Gers F, Eck D, Schmidhuber J (2002) Applying LSTM to time series predictable through time-window approaches. *Neural Nets WIRN Vietri* 01:193–200
- Graves A, Mohamed M, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: International conference on acoustics, speech and signal processing, pp 6645–6649. IEEE
- Gulli A, Pal S (2017) Deep learning with Keras. Packt Publishing Ltd
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference computer visual pattern recognition, pp 770–778. IEEE. <https://doi.org/10.1109/CVPR.2016.90>
- Hochreiter S, Hochreiter J (1977) Long short-term memory. *Neural Comput* 8(9):1735–1780
- Huynh H, Dang L, Duong D (2017) A new model for stock price movements prediction using deep neural network. In: Proceedings of the Eighth international symposium on information and communication technology, pp 57–62
- Jais I, Ismail A, Nisa S (2019) Adam optimization algorithm for wide and deep neural network. *Knowl Eng Data Sci* 2(1):41–46
- Kim K, Choi Y (2021) HyAdamC: a new Adam-based hybrid optimization algorithm for convolution neural networks. *Sensors* 21(12):4054
- Kingma D, Ba J (2015) Adam: a method for stochastic optimization. In: Published as a conference paper at ICLR 2015. *arXiv preprint arXiv:1412.6980*
- Koprinska I, Wu D, Wang Z (2018) Convolutional neural networks for energy time series forecasting. In: International joint conference on neural networks (IJCNN), pp 1–8. IEEE, New York
- Livieris I, Pintelas E, Pintelas P (2020) A CNN–LSTM model for gold price time-series forecasting. *Neural Comput Appl* 32(23):17351–17360
- Lu W, Li J, Wang J, Qin L (2021) A CNN–BiLSTM–AM method for stock price prediction. *Neural Comput Appl* 33(10):4741–4753
- Mallah S, Bagheri-Bodaghabadi M (2022) Towards a global soil taxonomy and classification tool for predicting multi-level soil hierarchy. *Model Earth Syst Environ* 8(2):1505–1517
- Maltby E (1994) An environmental and ecological study of the marshlands of Mesopotamia wetland ecosystem. University of Exeter, London
- Maxwell G (1957) People of the Reeds. ASIN: B0007DMCTC, 223
- Meng Z, Dang X, Gao Y (2020) Land degradation action plan in Inner Mongolia. In: Public private partnership for desertification control in Inner Mongolia, pp 171–194
- Mohamed A-R, Hussain N (2016) Evaluation of fish assemblage environment in Huwazah Marsh, Iraq using integrated biological index. *Int J Curr Res* 6:6124–6129

- Muftah H, Rowan T, Butler A (2022) Towards open-source LOD2 modelling using convolutional neural networks. *Model Earth Syst Environ* 8(2):1693–1709
- Nie Q, Wan D, Wang R (2021) CNN-BiLSTM water level prediction method with attention mechanism. *J Phys* 2078(1):012032
- Parsaie A (2016) Predictive modeling the side weir discharge coefficient using neural network. *Model Earth Syst Environ* 2(2):1–11
- Partow H (2001) The Mesopotamian Marshlands: demise of an ecosystem. *Division of Early Warning and Assessment*
- Peltier L (1950) The geographic cycle in periglacial regions as it is related to climatic geomorphology. *Ann Assoc Am* 40:214–236
- Pörtner H, Roberts D, Adams H, Adler C, Aldunce P, Ali A, Birkmann J (2022) Climate change 2022: impacts, adaptation and vulnerability. In: *IPCC sixth assessment report*
- Rabbani A, Samui P, Kumari S (2022) A novel hybrid model of augmented grey wolf optimizer and artificial neural network for predicting shear strength of soil. *Model Earth Syst Environ* 10(3144):1–21
- Raj A, Viswanath J, Oliver D, Srinivas Y (2018) Tollgate neural networks (TNN) model with time bound learning methodology for futuristic approach in climatic data analysis. *Model Earth Syst Environ* 4(4):1331–1339
- Reddy D, Prasad P (2018) Prediction of vegetation dynamics using NDVI time series data and LSTM. *Model Earth Syst Environ* 4(1):409–419
- Richardson C (2005) The status of Mesopotamian Marsh restoration in Iraq: a case study of transboundary water issues and internal water allocation problems. *Towards new solutions in managing environmental crisis*. University of Helsinki, Helsinki
- Richardson C, Reiss P, Hussain N, Alwash A, Pool D et al (2005) The restoration potential of the Mesopotamian marshes of Iraq. *Science* 307:1307–1311
- Seidu J, Ewusi A, Kuma J, Ziggah Y, Voigt H (2022) A hybrid ground-water level prediction model using signal decomposition and optimised extreme learning machine. *Model Earth Syst Environ* 8(3):3607–3624
- Shahid F, Zameer A, Muneeb M (2020) Predictions for COVID-19 with deep learning models of LSTM, GRU and BiLSTM. *Chaos Solit Fractals* 110212
- Song X, Liu Y, Xue L, Wang J, Zhang J, Wang J et al (2020) Time-series well performance prediction based on long short-term memory (LSTM) neural network model. *J Petrol Sci Eng* 186
- Tiner R, Lang M, Klemas V (2015) *Remote sensing of wetlands: applications and advances*. CRC Press and Taylor and Francis Group, Boca Raton
- Van Houdt G, Mosquera C, Nápoles G et al (2020) A review on the long short-term memory model. *Artif Intell Rev* 53:5929–5955
- Wang Y, Liao W, Chang Y (2018) Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies* 11(8):2163
- Young G (1977) *Return to the marshes: life with the marsh Arabs of Iraq*. Collins, London, p 224
- Zha W, Liu Y, Wan Y, Luo R, Li D, Yang S, Xu Y (2022) Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* 124889
- Zhang F, Fleyeh H, Bales C (2022) A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting. *J Oper Res Soc* 73(2):301–325
- Zhang H, Zhang L, Jiang Y (2019a) Overfitting and underfitting analysis for deep learning based end-to-end communication systems. In: *2019a 11th international conference on wireless communications and signal processing (WCSP)*, pp 1–6 IEEE, New York
- Zhang X, Liang X, Zhiyuli A, Zhang S, Xu R, Cheng Z et al (2019b) AT-LSTM: an attention-based LSTM model for financial time series prediction. In: *IOP conference series: materials science and engineering*, vol 569(5), p 052037

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Fehaid Alqahtani¹ · Mostafa Abotaleb²  · Alhumaima Ali Subhi³ · El-Sayed M. El-Kenawy⁴ · Abdelaziz A. Abdelhamid^{5,6} · Khder Alakkari⁷ · Amr Badr⁸ · H. K. Al-Mahdawi⁹ · Abdelhameed Ibrahim¹⁰ · Ammar Kadi¹¹

Fehaid Alqahtani
f-alqahtani@rsnf.gov.sa

Alhumaima Ali Subhi
alhumaimaali@uodiyala.edu.iq

El-Sayed M. El-Kenawy
skenawy@ieee.org

Abdelaziz A. Abdelhamid
abdelaziz@cis.asu.edu.eg; abdelaziz@su.edu.sa

Khder Alakkari
khderalakkari1990@gmail.com

Amr Badr
amr.mostafa@live.com

H. K. Al-Mahdawi
hssnkd@gmail.com

Abdelhameed Ibrahim
afai79@mans.edu.eg

Ammar Kadi
ammarka89@gmail.com

- 1 Department of Computer Science, King Fahad Naval Academy, Al Jubail 35512, Saudi Arabia
- 2 Department of System Programming, South Ural State University, Chelyabinsk, Russia
- 3 Electronic and Computer Center, University of Diyala, Baqubah MJJ2+R9G, Iraq
- 4 Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt

- ⁵ Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
- ⁶ Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra 11961, Saudi Arabia
- ⁷ Department of Statistics and Programming, Faculty of Economics, University of Tishreen, P.O. Box 2230, Latakia, Syria
- ⁸ Faculty of Science, School of Science and Technology, University of New England, Armidale, NSW 2350, Australia
- ⁹ Electronic Computer Centre, University of Diyala, Diyala 32001, Iraq
- ¹⁰ Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
- ¹¹ Department of Food and Biotechnology, South Ural State University, 454080 Chelyabinsk, Russia