Journal of Universal Computer Science, vol. 28, no. 10 (2022), 1108-1133
submitted: 16/2/2022, accepted: 13/9/2022, appeared: 28/10/2022 CC BY-ND 4.0

Climatological parameters estimation based on artificial intelligence techniques with particle swarm optimization and deep neural networks

Sercan Yalçın
(Adiyaman University, Adiyaman, Turkiye
https://orcid.org/0000-0003-1420-2490, svancin@adiyaman.edu.tr)

Musa Eşit
(Adiyaman University, Adiyaman, Turkiye
https://orcid.org/0000-0003-4509-7283, mesit@adiyaman.edu.tr)

Mehmet İshak Yüce
(Gaziantep University, Gaziantep, Turkiye
https://orcid.org/0000-0002-6267-9528, yuce@gantep.edu.tr)

Abstract: Climate forecasting plays an important role for human life in many areas such as water management, agriculture, natural hazards including drought and flood, tourism, business, and regional investment. Estimating these data is a difficult task as the time series climate parameter values vary monthly and seasonally. Therefore, predicting climate parameters based on learning and artificial intelligence is important to long-term efficient results in these fields. For this purpose, in this study, a time-series based Long Short-Term Memory (LSTM) deep neural network is proposed to predict future climate in Çankırı and Adıyaman cities in Turkey. With the help of this network, the average temperature, relative humidity, and precipitation values, which are known as the most effective climate parameters, have been estimated. An improved Particle Swarm Optimization (PSO) technique is also proposed to optimize input weight values of the LSTM deep network, and reduce the estimation errors. The proposed algorithm is compared with deep models of LSTM variants based on Root Mean Square Error (RMSE), Mean Absolute Deviation (MADE), and Mean Absolute Percentage Error (MAPE) metrics. The proposed adaptive LSTM-PSO and non-adaptive LSTM-PSO models achieved at RMSE 0.98 and 1.05 for temperature, 1.19 and 1.27 for relative humidity, and 4.21 and 4.67 for precipitation estimation, respectively. The RMSE is %7 lower with the proposed adaptive LSTM-PSO method than proposed non-adaptive LSTM-PSO method.

Keywords: Climate estimation, Artificial intelligence, Deep neural networks, Long short term memory, Data analysis
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
DOI: 10.3897/jucs.82370

1 Introduction

Climate variables such as precipitation, temperature, and relative humidity affect agricultural water management and processes in many climatic zones, whether arid, semi-arid or tropical [Alizamir et al. 2020]. Today, the best reliable measure of these parameters at a specific location is still derived from meteorological stations on the ground [Hasan et al. 2016, Wang et al. 2019]. In most parts of the world, these type of data with significant temporal span is not readily available for years [Buytaert et al. 2012]. But, in many areas, such as water resources management, flood and drought risk assessment, and meteorological forecasting, it is necessary to capture possible climatic patterns in variables on-site climatological information with such a temporal coverage [Yuce and Esit 2021, Sun et al. 2018, Park et al. 2019, Pumo et al. 2017]. Many of these areas are characterized by highly complex hydrological systems that frequently display outlier data, such as due to prolonged drought or intense monsoon seasons. Models and analysis approaches that are commonly utilized may not accurately describe these processes, and they are rarely assessed for appropriateness and robustness [Nkiaka et al. 2017].

In many parts of the world, especially in developing nations, a lack of meteorological data is a key obstacle impeding the growth of knowledge on water management and climate change. Climate data that is reliable, long-term, and well-distributed is critical for guiding policies aimed at mitigating the effects of climate variability and change [Van de Giesen et al. 2014]. Climatic change, as well as the inherent perturbations in any climate variable induced by natural climate variability, can have a substantial impact on precipitation patterns, affecting agriculture, water resources, and heightening uncommon events [Ali et al. 2018]. For example, a major change in climate variables can have a negative impact on economic growth, especially in developing countries [Odusola and Abidoye 2015]. Climate change affects crop productivity, which in turn cripples commodity-based economies, particularly those from developing countries. According to these factors, developing climate risk assessments and appropriate mitigation and adaptation measures necessitates a high level of ability to forecast climate extreme weather events, and such tasks can be dependent on how well such as precipitation, temperature, and relative humidity climate variables, can be predicted ahead of time [Ali et al. 2020].

Historical climate variables data can be gathered from various sources to complement instrumental records, such as radars [Austin and Seed 2005], satellites, and numerical model simulations [Huffman et al. 2010]. Short-term hydrometeorological variable time series are provided by satellites and radars, which begin in the 1990s in the best-case scenario [Chen et al. 2020, Li et al. 2020]. Reanalysis models [Pfeifroth et al. 2013], on the other hand, can supply continuous data for more than 50 years all over the world. However, the reanalysis databases have numerous drawbacks, such as a significant bias despite their long-time coverage [Wang et al. 2019]. As a result, it is critical to reproduce such long-term evolution in hydroclimate variables.

To address the aforementioned constraints of the various sources of information, several researchers have examined alternative strategies for predicting time series of historical hydrometeorological variables covering decades in the past [Li et al. 2020]. Data-driven models have been effectively utilized to estimate climate variable evolutions with historical records. These models can forecast future precipitation patterns with non-linear input features. The non-linear input features are important to analysis and optimize the climate data. Hung et al. [Hung et al. 2009] used an artificial neural network to predict rainfall in Thailand. Lin et al. [Lin et al. 2009] used the Support Vector Machines (SVMs) to predict precipitation in Taiwan. Several hybrid strategies have recently been used to forecast water demand [Altunkaynak and Nigussie 2018, Seo et al. 2018, Zubaidi et al. 2020], prediction of long and short-term drought prediction [Hassanzadeh et al. 2020, Nasir and Hamdan 2021], low-flow [Nejat et al. 2020], stream-flow prediction [Ghimire et al. 2021, Cheng et al. 2020] and flood forecasting [Hussain et al. 2021]. Considering the climate system of Turkey, Bayrak et al. [Bayrak et al. 2021] estimated the highest monthly average stream-flow in Ergene River using Artificial Neural Network (ANN), Multiple Linear Regression (MLR) and SVM, Apaydin et al. [Apaydin et al. 2020] utilized Deep Recurrent Neural Network (DRNN) designs such as bidirectional long short-term memory (Bi-LSTM), gated recurrent unit (GRU), LSTM, and simple recurrent neural networks to simulate daily streamflow to the Ermenek hydroelectric dam reservoir, Yakut and Suzulmus [Yakut and Süzülmüs 2020] modelled monthly mean air temperature using ANN, adaptive neuro-fuzzy inference system and support vector regression methods, and Akdi and Unlu [Akdi and Unlü 2021] modelled and forecasted monthly average temperature, and monthly average precipitation of Turkey by employing periodogram-based time series methodology.

In this study, we propose a novel Long Short-Term Memory-Particle Swarm Optimization (LSTM-PSO) deep learning method using time series and parameter values estimated by training the climate parameters of temperature, relative humidity, and precipitation. The study data are climatic parameters including temperature, relative humidity, and precipitation of Adıyaman and Çankırı provinces of Turkey. Previously, no study was conducted with the data obtained from the stations to which these two cities were connected. Adıyaman and Çankırı provinces are located in different climatic geographies in Turkey. Estimating the climate variables of these two provinces with different regional characteristics constitutes the basic basis of this study. There are many metaheuristic optimization methods in the literature. However, a detailed literature review concludes that a basic and uncomplicated particle swarm model is more useful than others. The PSO algorithm was used in this study and an algorithm framework was created together with the LSTM deep architecture. In the training and testing process of the method, the number of neurons and layers in the network are determined by adaptively adjusting the presented LSTM deep network architecture according to the data set and parameters used. In addition, the PSO algorithm is combined into the proposed scheme in order to optimize the LSTM input weights. The proposed deep architecture and optimization-based method is compared with other existing LSTM-derived methods in terms of estimation error metric, and its performance advantage is revealed.

The main contributions of this study can be presented briefly as follows.

- In this study, unlike other studies for Turkey, we present a hybrid approach by combining time series-adapted deep architecture and optimization technique to predict climate parameters in regions belonging to different meteorological basins.
- In the LSTM deep learning section of this hybrid approach, we further reinforce the learning by optimizing the input weight values with the PSO method, without memorizing by adding different blocks.
- We reveal the results of many analyzes by comparing the proposed method with different methods in terms of various performance criteria such as Root Mean Square Error (RMSE), Mean Absolute Deviation (MADE), and Mean Absolute Percentage Error (MAPE).

The rest of the paper is as follows. In Section 2, the proposed method is presented. In this section, firstly, we design an LSTM based recurrent deep neural network. After that, we construct the proposed adaptive LSTM-PSO based method. Section 3 presents the experimental analysis and results of the study. In this section, firstly, we mention related to the study area and data. After that, we set up the proposed LSTM-PSO deep neural network, and model the climate parameters for Çankırı and Adıyaman Case. Finally, we conclude the study by talking about future plans in Section 4.

2 Materials and Method

In this study, it is presented an approach that uses LSTM deep learning model and PSO algorithm together to predict climate parameters. In the proposed LSTM model, the number of input neurons and layers, the number of hidden layers and the output layer are designed specifically for the study. To optimize the input parameter weight values, the proposed approach has been made adaptive and more efficient by using the improved PSO method.

2.1 Designing the LSTM Based Recurrent Deep Neural Network

Neurons called neurones or nerve cells are the fundamental units of the brain and nervous system. An activation function is a function that is added into an artificial neural network in order to help the network learn complex patterns in the data. Deep Neural Networks (DNNs) are learning networks inspired by the interaction and communication of neurons in the human brain [Sarker 2021]. It realizes learning by labeling and processing neural data. They can process real data in the form of digital vectors, including images, audio, text or time series [Greff et al. 2017, Kim et al. 2021]. DNNs are complex networks consisting of many neurons interacting with each other. A DNN is a network architecture that can operate in parallel, which consist of one or multiple layers of input, hidden and output layers. In this aspect, it differs from standard ANNs. [Hanab et al. 2021, Wang et al. 2021]. The input layer in a DNN is known as raw data inputs, similar to optic neurons [Bouktif et al. 2020]. Hidden layers take the data from input layer and forward it to the output layer. Figure 1 depicts a conventional DNN model.

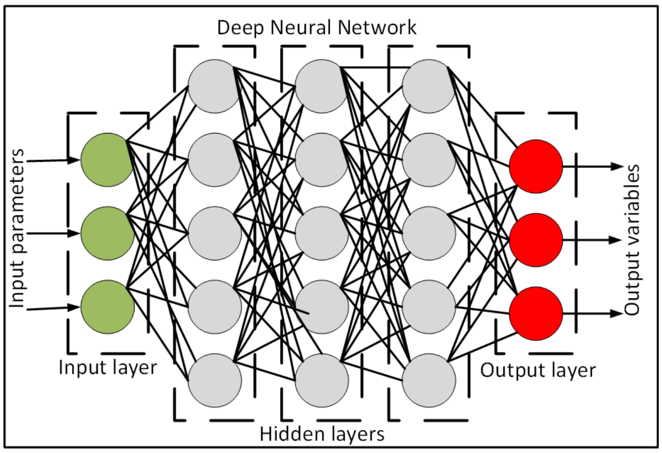


Figure 1: Description of the conventional DNN models

2.2 Overview of deep recurrent neural networks

Recurrent Neural Networks (RNNs) are improved version of DNNs, one output of which is passed to the input of a feed-forward network, taking into consideration past values. The past output is calculated and included in the next input, making data learning functional. The output data consists of an iterative series of iterations that facilitate learning without memorization [Bouktif et al. 2020].

RNNs, like feed-forward DNNs, do not simply process data inputs and transfer them to the next state. At the same time, some parts learned in memory using their internal memory are not forgotten. These features have gained importance in applications such as handwriting and speech recognition. On the other hand, in other DNNs, all data inputs can work independently with one another. In RNNs, interdependent data series can be modeled and each produced sample output cannot be considered independent from the previous one. The RNNs can also be used with convolutional layers in applications such as image processing and object recognition, where pixel neighborhood should be taken as a basis [Hoang et al. 2020]. Besides, LSTM networks are RNNs in which RNNs have been developed and their layers and the way their neurons work are different as the past learned data in the memory is prevented from being forgotten, and makes it easier to remember when necessary. In addition, the disappearing gradient problem is also eliminated. An LSTM is often used to process, classify and estimate time series data with unknown time delays [Pei et al. 2019, Ravuri et al. 2021]. The LSTM data output is transferred to the input of another one, that is, it works as back propagation principle. There are three gates in the internal structure of an LSTM network: input, forget and output gates [Sarker 2021, Wang et al. 2021]. The structure of an LSTM model architecture is shown in Figure 2.

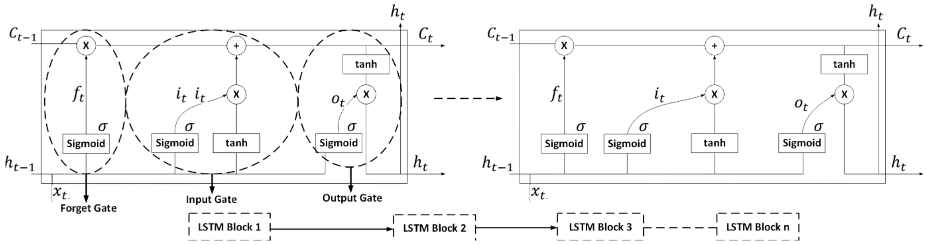


Figure 2: An architecture of the LSTM

Input gate: The input gate determines which values must be given from the input to replace the memory in the LSTM. The input gate i_t is defined as Eq. (1). Sigmoid function σ decides which of 0 and 1 will pass. Also, a $tanh$ function shown in Eq. (2) with cell state C_t assigns a weight value to the values passed from -1 to 1 to decide according to its priority.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (1)$$

$$C_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \quad (2)$$

where C_t and h_t are the cell and hidden states, respectively. Also, W_i is weight vector for different timesteps.

Forget gate: In the forget gate, it is determined which unlearned data will be forgotten from the block. This decision is made by the Sigmoid function. This function evaluates both the previous h_{t-1} state and the content input x_t. It then returns a number between 0 and 1 for each number in the cell state C_{t-1}. The expression of the forget gate f_t is given as Eq. (3).

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \tag{3}$$

Output gate: When determining the output h_t of the LSTM, input x_t and memory are used to decide the output h_t. In this gate, as shown in Fig. 2, the sigmoid function σ decides which of 0 and 1 will pass. Also, the $tanh$ function with C_t assigns a weight value to the values passed from -1 to 1 to decide according to its priority, it is multiplied by the sigmoid output out_t as shown in Eq. (5).

$$o_t = \sigma(W_{out} . [h_{t-1}, x_t] + b_{out}) \tag{4}$$

$$h_t = o_t * tanh(C_t) \tag{5}$$

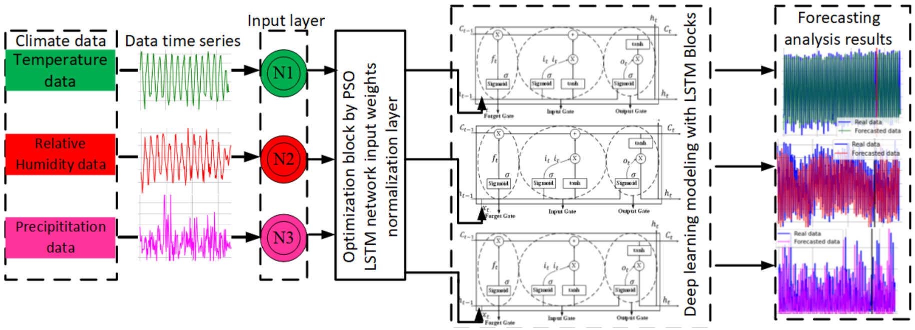


Figure 3: The block diagram of the proposed scheme

Figure 3 illustrates the block diagram of the proposed scheme. Neurons represented by N1, N2, and N3 are designed as input neurons for temperature, relative humidity, and precipitation, respectively. The important point here is that the LSTM input weight values optimized with PSO are given as input to the LSTM block. In this way, a more adaptive time series data transferred to the layers in the LTSM-RNN model are trained and tested. Finally, the forecasting analysis results are visualized after training.

2.3 Constructing the proposed adaptive LSTM-PSO based method

Before giving the details of the proposed method, it is useful to present the PSO method. PSO is a metaheuristic-optimization method in which the behavior of various birds in nature is artificially tackle to obtain a desired target or optimum value [Gundu and Simon 2021]. In a flock, a bird always flies towards a better position in its surroundings. Each individual in the PSO is like a bird, following various rules and interacting with other individuals around it in order to complete the final task. In a PSO algorithm, each individual in the swarm is represented as a particle. The particles follow several simple rules and benefit from the experience of neighboring particles or their own during swarm updates. They update their position to head to the destination and their speed to arrive at the destination just in a given time [Liu et al. 2021]. The velocity and position equations of the basic PSO algorithm are given in Eqs. (6) and (7).

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(o_b(t) - x_i(t)) + c_2 r_2(t)(g_b(t) - x_i(t)) \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{7}$$

where v_i and x_i are the velocity and position of the particle i at time t, respectively. Also, c_1 and c_2 are acceleration constants that are positive numbers used in the contribution of cognitive and social components to rate the updates. Random numbers r_1 and r_2 can take various values between 0 and 1. The inertia weight ω controls the inertia of a particle, and measures the effect of the velocity of the previous instant on the next displacement. $o_b(t)$ and $g_b(t)$ are the optimal positions calculated by the particle itself and all particles in the population, respectively. All particles know the population information. Each particle moves towards the global optimal solution and aims to find the optimal solution in the population. Before the update, the particle decides whether the reached location is better than the known location or the location known by someone else. Otherwise, the particles retain the previous optimal values and continue to move. The PSO algorithm only terminates when a successful solution is found or the iteration count reaches its maximum value.

2.4 Making adaptive the learning factors with improved PSO

The learning factors c_1 and c_2 are mainly used to determine the step size of a particle moving to the individual optimal and global optimal positions. In the standard PSO, the value of c_1 and c_2 is assigned as 2. This cannot meet the requirements of practical applications. However, in the first iterations, the value c_1 should change little from large to small [Shao et al. 2019]. Because, it is aimed to increase the search speed. As the number of iterations progress, it is necessary to increase the value of c_2 from small to large to improve local search. Therefore, the sine function as in Eqs. (8) and (9) is used to adjust the adaptive learning factors. Here, N_e is the number of epochs.

$$c_1 = 2\sqrt{1 - \sin\left(\frac{\pi}{2} * \frac{i}{N_e}\right)} \tag{8}$$

$$c_2 = 2\sqrt{\sin\left(\frac{\pi}{2} * \frac{i}{N_e}\right)} \tag{9}$$

2.5 Adaptive LSTM-PSO based method

For effective climate parameter estimation, it is required to select appropriate input data. In this study, we design the LSTM layer and the number of neurons adaptive to our study. In the proposed model, 3-L_i (x_j) -1 is the LSTM learning model with N-1, N-2, and N-3 three input neurons, three LSTM layers with 10 hidden units and one output neuron for RMSE. In this case, all three input signals are utilized for training. The proposed output h_t can be expressed as in Eq. (10).

$$h_t \leftarrow 3 - L_i\ (\ x_j\ )\ - 1 \tag{10}$$

In the proposed learning model, an LSTM model with 3 layers and 10 hidden units is proposed, x_j = 10, where j=1 and L_i= 3, where i =1. In this model, the output node is set to 1. In input layer, 3 different input nodes are defined as temperature, humidity, and precipitation values for climate analysis.

We analysis the model based on the RMSE, MADE, and MAPE metrics in this study. The RMSE is usually utilized measuring of the differences between samples or all data values forecasted by a model or an estimator and the observed values. The RMSE depicts the square root of the second sample moment of the differences between forecasted values and observed real values or the quadratic mean of these differences. The RMSE of forecasted values x'_t for t times a regression's dependent variable x_t with variables observed over T sample times, is computed for T different samples. The RMSE calculation is given as Eq. (11). The lower the RMSE value, the more successful the data estimation is expected. Therefore, when the high performance of the proposed models is aimed, the RMSE values should be considerably lowered. The number of layers and associated hidden nodes are determined and selected according to the LSTM and PSO model. The closer the RMSE, MADE, and MAPE values is to 0, the better predictive performance will be.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(x_t - x'_t)^2}{T}} \tag{11}$$

MADE avoids the problem that positive and negative errors in estimations dampen each other. The calculation of the MADE is presented as given in Eq.(12).

$$MADE = \frac{1}{T}\sum_{t=1}^{T}|x_t - x'_t| \tag{12}$$

MAPE calculates the deviation between the predicted value and the actual value. At the same time, it is based on the correlation between the error and the true value, which can better show the accuracy of the prediction result. The calculation of the MAPE is presented as given in Eq.(13).

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}\left|\frac{x_t - x'_t}{x_t}\right| \tag{13}$$

While modeling the LSTM network, the number of layers and the number of hidden units are chosen adaptively. After the layers and neurons are determined, the number of layers i is assigned from 1 to k. Then the i is set to the fixed value where the error metrics are minimum. The neuron number j is assigned from 1 to m for i =1. Also, j where RMSE is minimum is set to fixed value. These operations are also valid for $k-1$ layers. Hence, $L_{ib}(x_{jb})$ architecture is created according to the fitness functions F(RMSE), F(MADE) and F(MAPE) given in Eqs. (14)-(16).

$$F_{\text{RMSE}} = RMSE\{L_i(x_j)\} \tag{14}$$

where ∀ i = 1, 2 …, k and j = 1, 2, …, m

$$F_{\text{MADE}} = MADE\{L_i(x_j)\} \tag{15}$$

$$F_{\text{MAPE}} = MAPE\{L_i(x_j)\} \tag{16}$$

The total fitness function is the sum of these three functions as in Eq.(17).

$$F_{\text{total}} = \alpha F_{\text{RMSE}} + \beta F_{\text{MADE}} + \gamma F_{\text{MAPE}} \tag{17}$$

where α, β, and γ determine the severity of these error parameters, respectively. It is assumed that the importance of each error parameter is equal, their values are 0.33, 0.33 and 0.34 respectively, their sum being 1.

To use the PSO method in the proposed scheme, we used a particle population with the input parameters having the size equal to the number of parameters in the LSTM model. We set the population size to 30 based on size, time complexity, and adaptability of the problem. The task of each particle in the PSO is to minimize the difference between the actual values and the predicted values in the LSTM-PSO model. To reduce the probability of the particle moving away from the search area, the velocity value of the particle is set in the range of V_{min} and V_{max}. LSTM block cells eliminate the possibility of memorization in the data that can occur as a result of long-term learning. The number of LSTM cells is defined according to how the proposed model predicts the climate parameters. Batch size is an important parameter that affects the training efficiency of the LSTM neural network, which determines the number of samples to be detected over the deep network before the internal parameters of the LSTM model are updated. The data is divided into groups of several small blocks and at the same time the neural network is trained for multiple stages due to the fact that these networks work faster with mini-groups. On the other hand, with a very small batch size, the parameter value becomes harder to estimate and requires more memory space in the LSTM when a larger batch size is assigned in the network.

The pseudocode of the proposed PSO-based LSTM model is given in Algorithm 1. According to this algorithm, the output is the minimized error values followed by the estimation results of the climate parameters. The best position of a particle in the PSO swarm is assigned as the initial value of each weight in the LSTM model. That is, optimization of a particle is also intended. In Algorithm 1, at each iteration, the particle with the least error is reassigned as the new optimum particle. Climate parameter values are entered. We compute the error metrics and F_{total} with Eq.(17). Thus, the F_{total} of the output neuron is defined as the fitness function. If best values meet the condition in fitness function, the algorithm initializes the deep LSTM input weights obtained by

PSO. After that, it reads the training data and compare them with real values. Figure 4 shows the flowchart of the proposed method based on LSTM-PSO. Climate parameter data is given as input and the PSO population is started. After applying of the PSO, the climate parameter values are trained. The position and velocity of the PSO particles are updated. In the algorithm, observed data are identified. The RMSE, MADE, and MAPE values are calculated and checked according to the defined fitness function. Once sufficient best solution conditions are met, LSTM input weight values optimized by PSO are applied to the deep architecture. If a sufficient number of epochs has been reached, the real and forecasted values are displayed as a result of the trained values.

Algorithm 1: **Proposed Adaptive LSTM-PSO based Method**

```
1: Input: $P$ // $P$ is the number of particles and LSTM arguments
2: Output: Minimized error values and accordingly forecasted climate parameters
3: Initialize the population $P$
4: Enter the training climate parameter data
5: while $j \leq N_e$ and $j$++ do
6:      for each $p_i \in P$ and do
7:          Update $v_i$ and $x_i$ using Eqs. (6) and Eq. (7)
8:          Compute $c_1$ and $c_2$ using Eqs. (8) and (9)
9:          Compute fitness function $F_{\text{total}}$ with Eq.(17).
10:             Select the best $p_i$ position
11:             if best values meet the condition then
12:                 Initialize the deep LSTM input weights obtained by PSO
13:                 Read the training data and compare with real values
14:             else then
15:                 Go to step 4
16:             end if
17:     Obtain the errors with adaptive the LSTM input weights
18:   end for
19: end while
20: return error values and forecasted data
```

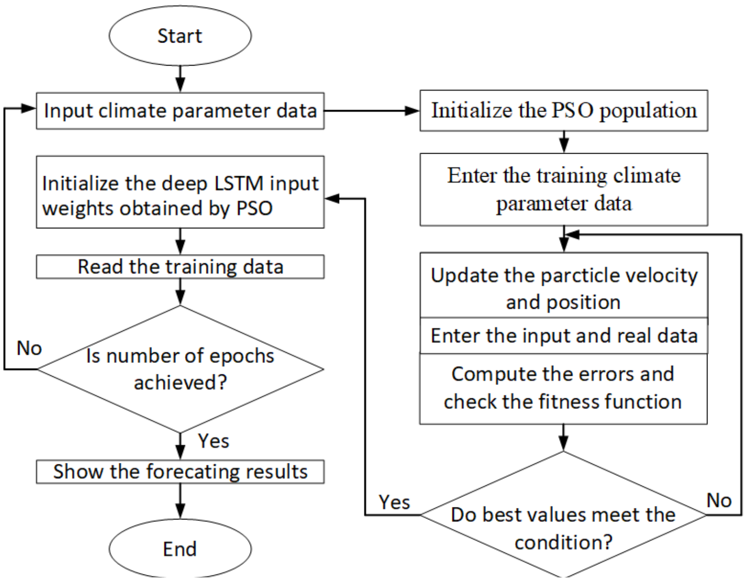

Start
Input climate parameter data
Initialize the PSO population
Initialize the deep LSTM input weights obtained by PSO
Enter the training climate parameter data
Read the training data
Update the parcticle velocity and position
Enter the input and real data
No
Is number of epochs achieved?
Compute the errors and check the fitness function
Yes
Show the forecating results
Yes
Do best values meet the condition?
No
End

Figure 4: The flowchart of the LSTM-PSO based method

3 Experimental Analysis and Results

To carry out the applications of this study, we performed the following installations. This study is closely related to open-source software conditions. We used the Python 3.8.5 programming language and Jupyter Notebook interface that came with Anaconda 3 distribution. We used the Numpy, Pandas, Tensorflow and scikit-learn libraries in Python coding. Moreover, some python codes were compiled via PyCharm.

3.1 Study area and data

We focus on the cities of Çankırı and Adıyaman in Turkey for study area. Therefore, we give the geographical and meteorological general characteristics of these two cities separately. The data used in this study is provided by General Directorate of Meteorology of the Ministry of Agriculture and Forestry in Turkey. The dataset contains monthly measured temperature, relative humidity, and precipitation data from 1948 to 2020. Table 1 shows some values of the climate data in the data set. In this way, the details are explained with the data set. Data analysis was made in these files with the Python language, and the models were trained and tested with the proposed algorithm.

Çankırı is located in the transition of the Central Anatolia Region to the Black Sea region in the East-West direction, and its length and width are 130 and 80 km, respectively. Its surface area is 7.490 km^2 and it constitutes 1% of Turkey's surface area. It is located between 40° 30' and 41° north latitudes and 32° 30' and 34° east longitudes. While the highest temperature in Çankırı is 41.8 degrees, the lowest temperature is -30.7 °C. In winter, snow stays on the ground for up to two months and there are even times when there is frost for more than four months. The hottest months are July and August, and the coldest months are January and February.

Adıyaman city is located in the Middle Euphrates Region. A part of Çelikhan and Gerger districts in the north are included in the Eastern Anatolia Region, and a part of Gölbaşı and Besni districts in the west are included in the Mediterranean Region. Adıyaman is located between 370 25' and 380 11' north latitude, 370 and 390 east longitude. The surface area of Adıyaman is 7.14 km^2, 7,871 km^2 with lakes, and its altitude is 669 m. After the formation of the Atatürk Dam, there has been a softening in the climate of the province and an increase in humidity. The prevailing winds in the province are in the north, northeast and northwest directions. The lowest temperatures of the year are between -10°C and -2°C, and it has been observed that the temperature does not fall below zero in some years. The minimum temperature averages in winter are between 0°C and 10°C. The annual precipitation average is 835 millimeters. The location of observed the corresponding stations are displayed in Figure 5. Statistical parameters for observed stations are presented in Table 1. In Table 1, L_t, L_g, M_v, S_d, C_v, C_s, and R_1 denote the latitude, longitude, mean value, standard deviation, coefficient of a variance, skewness, and kurtosis, respectively. Table 2 shows the detail information of the dataset used in this study.

Station	**Parameter**	**First record year**	**Last record year**	L_t	L_g	M_v	S_d	C_v	C_s	R_1
17080 Çankırı	Precipitation (mm)	1948	2020	40.6	33.6	34.57	27.58	0.80	1.13	0.18
	Temperature (°C)					11.31	8.43	0.74	-0.07	0.84
	Relative Humidity (%)					66.06	3.64	0.06	-0.43	0.57
17265 Adiyaman	Precipitation (mm)	1963	2020	37.7	38.2	59.72	68.18	1.14	1.50	0.48
	Temperature (°C)					17.32	9.44	0.55	0.07	0.85
	Relative Humidity (%)					48.68	16.58	0.34	-0.01	0.75

Table 1: Statistical parameters of observed stations

Station	**Parameter**	**Maximum**	**Year**	**Number of Dry Months**
7265 Adiyaman	Precipitation (mm)	369.1	2019-December	225
	Temperature (°C)	33.5	2000-July	
	Relative Humidity (%)	83.2	2018-December	

17080 Çankırı	Precipitation (mm)	149.5	2001-December	160
	Temperature (°C)	27.1	2006-July	
	Relative Humidity (%)	89.5	2017-December	

Table 2: The detail information of the dataset used in this study

3.2 Setting up proposed LSTM-PSO deep neural network

We use a function to define the LSTM TensorFlow Keras model with layers. The model uses a neuron for the output layer because we are predicting a real-valued number here. In the first LSTM block, 64 layers are used, and 32 layers are used in the second and third. As activation function, the deep network utilizes the rectified linear unit (ReLU) function which can take its own value in the range of 0 to maximum value. Also, the model uses the PSO optimizer model. The following observations were obtained as a result of various calculations in the proposed PSO-LSTM model.

- Too small and too long embedding size will not perform well. Note that we analyze the monthly data. The embedding size of 12 is good selection according to the dataset used.
- More epoch counts improve the algorithm. However, we set the epoch number to 500 in the analysis after a few tries.
- A batch size of 16 seems optimal and suitable.
- The predictions are performed for each climate variable but using the same model.

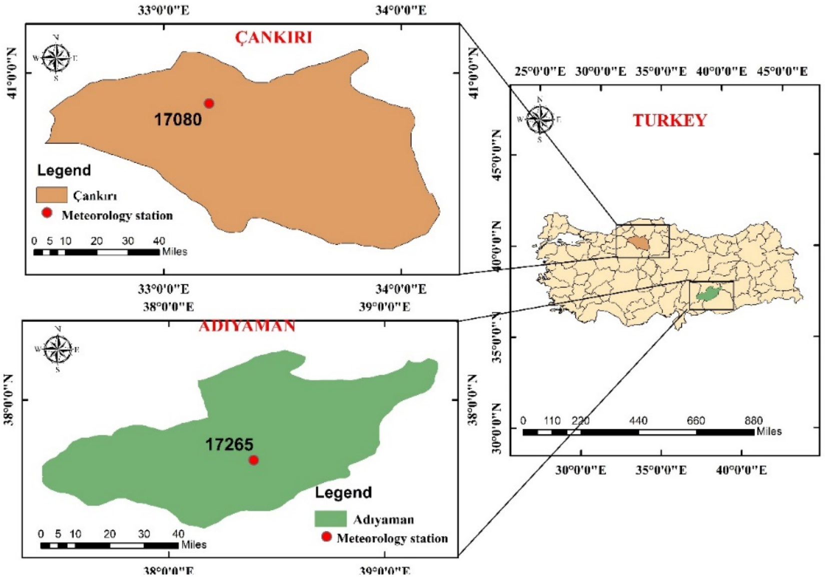


Figure 5: The location of the meteorological stations for Çankırı and Adıyaman regions

Table 3 shows some of the parameters used in training and testing the data.

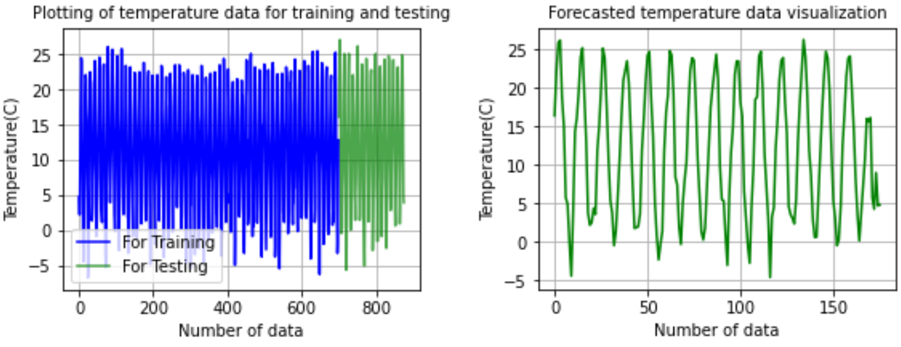
Parameter	Description	Value
num_epochs	Number of epochs (N_e)	500
Population size	PSO particle size	30
num_dense	Number of neurons in the dense layer followed by the LSTM layer	64,32,32
embedding	Time step length as embedding length	8
num_units	Number of units of the LSTM layer	128
lr	Learning rate	0.0005

Table 3: The training and testing parameters of the LSTM-PSO deep neural network

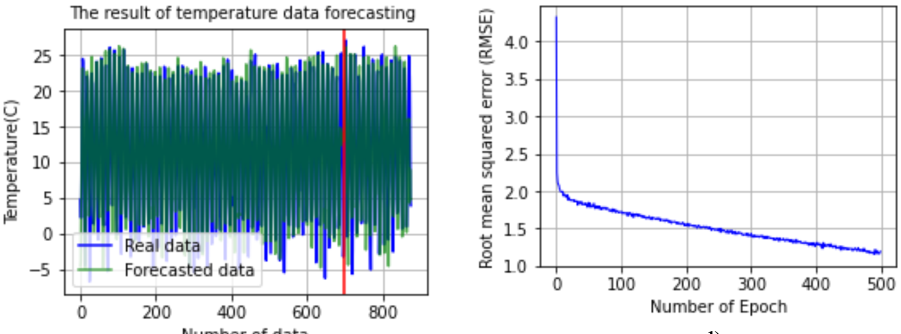
3.3 Modeling the climate parameters for Çankırı Case

In this part of the study, we model the average temperature, relative humidity, and precipitation data of Çankırı station in Turkey. We used 80% of the total 696 data for each climate parameter for training, 10% for testing, and the remaining 10% for validation. However, we combined testing and validation in experiments because the results were very similar and the analysis parameters were properly optimized.

Temperature data analysis: In this section, the temperature data received from the Çankırı station is analyzed. As seen in Figure 6(a), we used the remaining 176 pieces of data in the data set to predict the future for testing. After running the proposed method, the estimated data model and the predicted temperature data are given in Figures 6(b) and 6(c), respectively. After training, the forecasted data is separated by a red line for training and testing. From these results, it is understood that with the proposed PSO-LSTM model, the actual values and the estimated values produce very close results. As seen in Figure 6(d), the RMSE value is approximately 1.12 after sufficient iterations. We note that as the number of epochs increases, the RMSE value decreases and the prediction performance improves.



a) b)

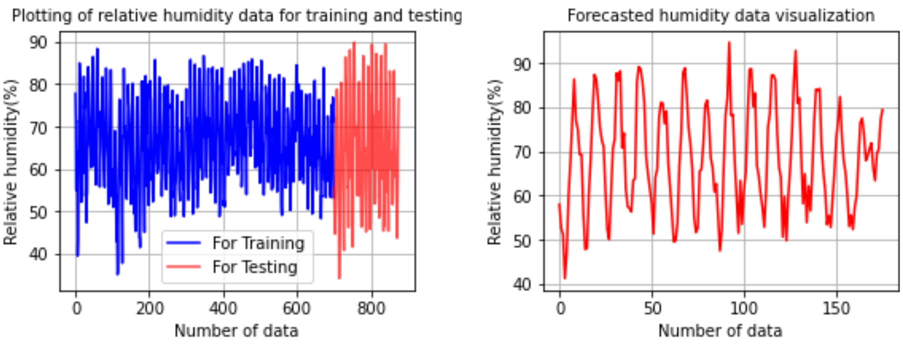


d)

c)

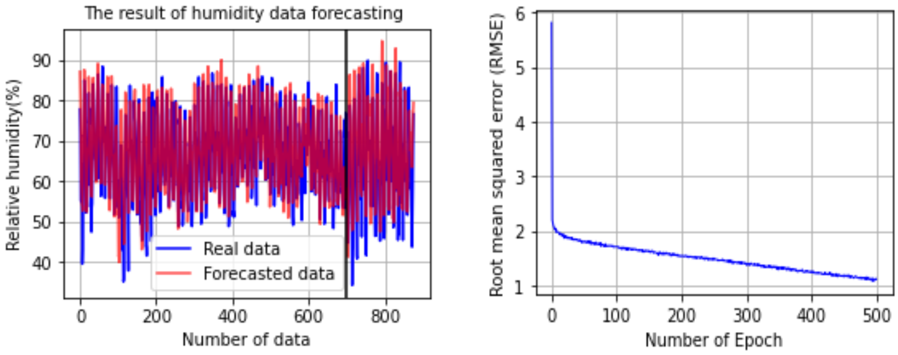
Figure 6: Experimental results for temperature data for Çankırı City a) plotting of the temperature data for training and testing b) forecasted temperature data visualization c) the results of temperature data forecasting d) RMSE result for temperature

Relative humidity data analysis: In this section, the relative humidity data received from the Çankırı station is analyzed. As seen in Figure 7(a), we used the remaining 176 pieces of data in the data set to estimate the future trends for testing. After running the proposed method, the estimated data model and the predicted relative humidity data are given in Figures 7(b) and 7(c), respectively. We understand that with the proposed PSO-LSTM model, the real values and the forecasted values produce very close series. As seen in Figure 7(d), the RMSE value is approximately 1.03 after sufficient iterations.



a)

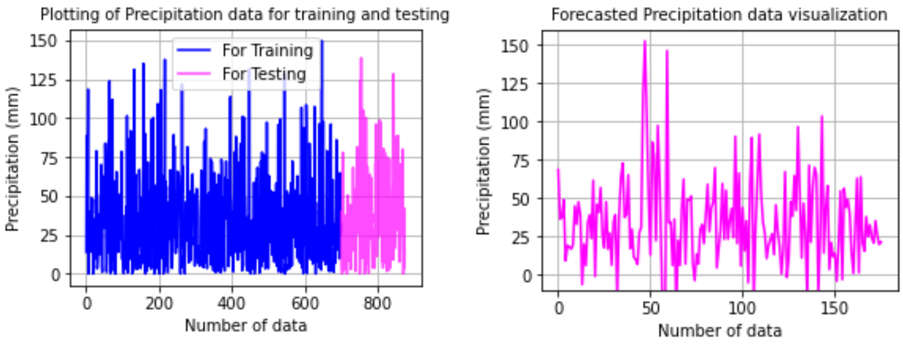
b)



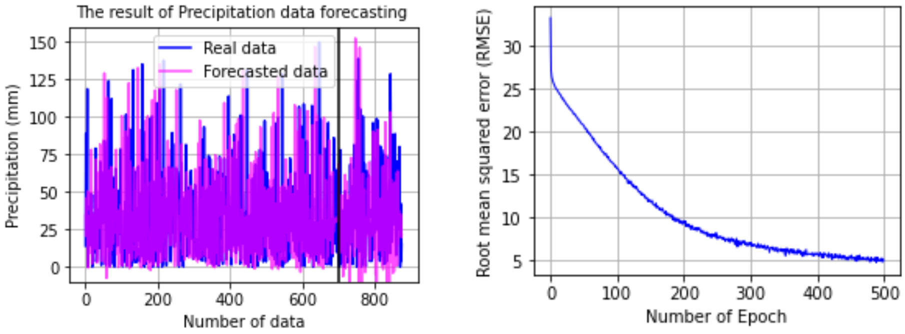
c) d)

Figure 7: Experimental results for relative humidity data for Çankırı City a) plotting of the relative humidity data for training and testing b) forecasted relative humidity data visualization c) the results of relative humidity data forecasting d) RMSE result for relative humidity

Precipitation data analysis: In this section, the precipitation data received from the Çankırı station is analyzed. As seen in Figure 8(a), we used the rest 176 pieces of data in the data set to estimate the future series for testing. After running the proposed method, the estimated data model and the predicted precipitation data are given in Figures 8(b) and 8(c), respectively. Actually, less performance was obtained in precipitation data estimation compared to other climate variables. As seen in Figure 8(d), the RMSE value is approximately 5.14 after some iterations.



a) b)



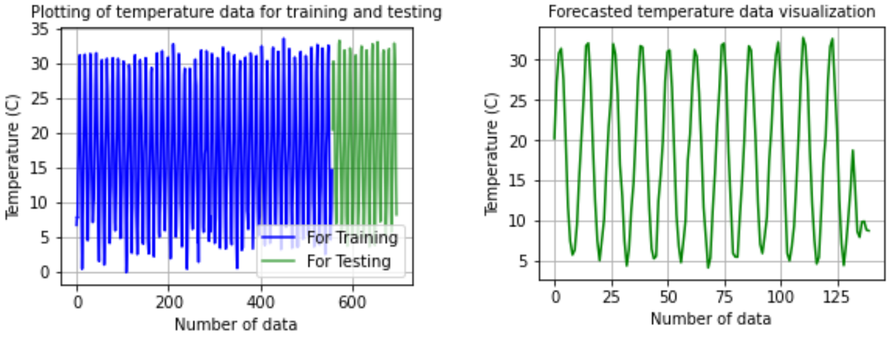
c) d)

Figure 8: Experimental results for precipitation data for Çankırı City a) plotting of the precipitation data for training and testing b) forecasted precipitation data visualization c) the results of precipitation data forecasting d) RMSE result for precipitation

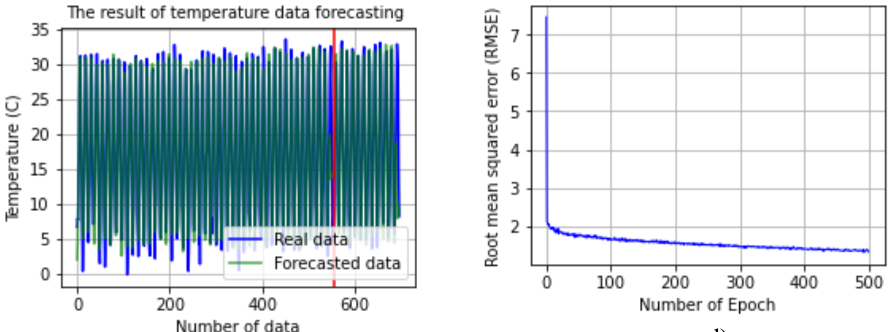
3.4 Modeling the climate parameters for Adıyaman Case

In this part of the study, we model the average temperature, relative humidity, and precipitation data of Adıyaman station in Turkey. We used 80% of the total 696 data for each climate parameter for training, 10% for testing, and the remaining 10% for validation. However, we combined testing and validation in experiments because the results were very similar and the analysis parameters were properly optimized. Data analyzes for each parameter are given in subsections.

Temperature data analysis: In this section, the temperature data received from the Adıyaman station is evaluated. As seen in Fig. 8(b), we use the remaining 140 data values in the data set to predict the future for testing. After executing the proposed method, the estimated data model and the predicted temperature data are given in Figs. 8(c) and 8(d), respectively. From these results, it is understood that with the proposed LSTM-PSO model, the actual values and the estimated values produce very close results. As can be seen from the graph in Fig. 8(c), the predicted temperature data after the red line can be representative of future data. As seen in Fig. 8(e), the RMSE value is 0.98 after sufficient iterations.



a) b)

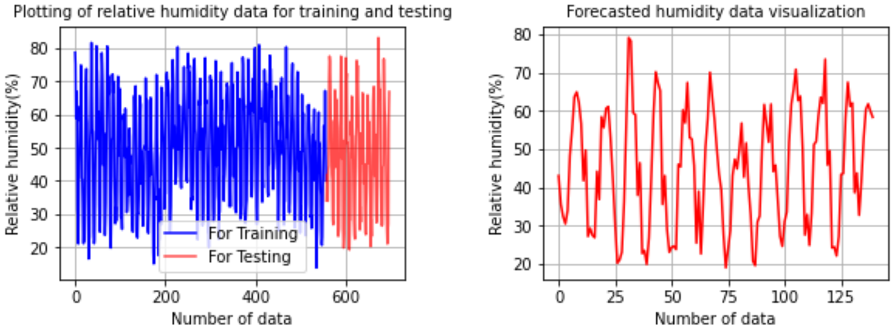


c)

d)

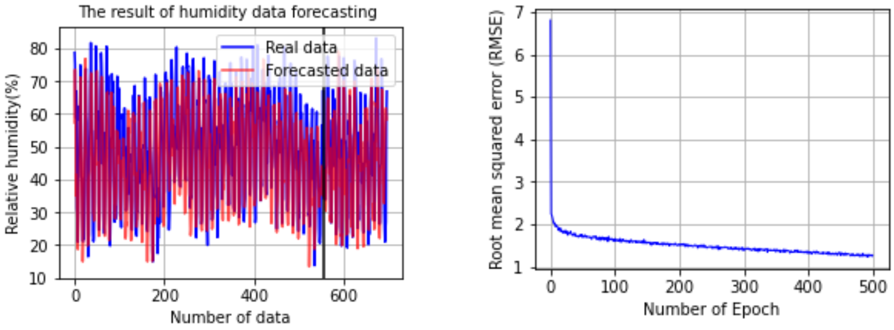
Figure 9: Experimental results for temperature data for Adıyaman City a) plotting of the temperature data for training and testing b) forecasted temperature data visualization c) the results of temperature data forecasting d) RMSE result for temperature

Relative humidity data analysis: In this section, the relative humidity data received from the Adıyaman station is analyzed. Fig. 9(a) gives the model of relative humidity data for first 556 values. As seen in Fig. 9(b), we used the remaining 140 data values for testing to make future predictions. After running the proposed method, the estimated data model and the predicted relative humidity data are given in Figs. 9(c) and 9(d), respectively. We understand that with the proposed adaptive LSTM-PSO model, successful performance is obtained, and the error rate is very low. We can say that very successful estimations have been made in the estimation of relative humidity. As seen in Fig. 9(e), the RMSE value is 1.19 after sufficient iterations.



a)

b)

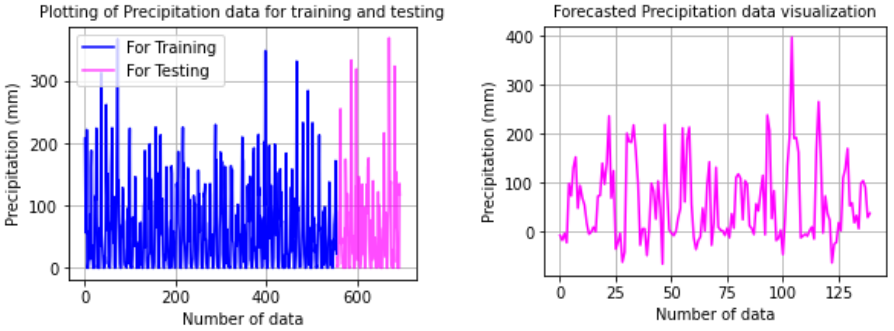


c)

d)

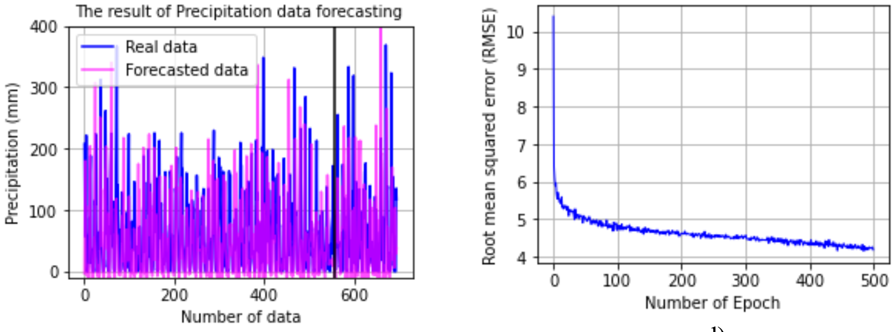
Figure 10: Experimental results for relative humidity data for Adıyaman City a) plotting of the relative humidity data for training and testing b) forecasted relative humidity data visualization c) the results of relative humidity data forecasting d) RMSE result for relative humidity

Precipitation data analysis: In this section, the precipitation data received from the Adıyaman station is modeled. As seen in Figure 11(a), we used the remaining 140 data values for testing to make future estimations. After carried out the proposed method, the estimated data model and the predicted precipitation data are given in Figures 11(b) and 11(c), respectively. As seen in Figure 11(d), the RMSE value is 4.21 after some iterations. Large forecast errors were observed at some data points in the precipitation data. However, its performance is higher than other methods compared.



a)

b)



c)

d)

Figure 11: Experimental results for precipitation data for Adıyaman City a) plotting of the precipitation data for training and testing b) forecasted precipitation data visualization c) the results of precipitation data forecasting d) RMSE result for precipitation

3.5 Performance comparison

The proposed LSTM-PSO models have been compared in performance with simple RNN, conventional LSTM, and LSTM-GRU models. The values of the c_1 and c_2 variables are set to 2. That is, the PSO algorithm is directly applied to LSTM. In the adaptive one, since the c_1 and $c_2$values were determined as a result of the training according to the Eqs.(8) and (9), dynamic training was carried out and very successful results were obtained. In the proposed non-adaptive LSTM-PSO Table 4 and 5 present the performance results according to error metrics for all climatic parameters of Çankırı and Adıyaman cities, respectively. From Table 4, we observed that with the proposed adaptive LSTM-PSO, the RMSE, MADE, and MAPE values for temperature are 1.12, 1.08, and 0.26, for relative humidity 1.03, 0.96, and 0.34, for precipitation 5.14, 1.25, and 0.54, respectively. That is, when compared to other methods, minimal errors were experienced with the proposed methodology.

		Models				
Climate parameters	**Error metrics**	**Simple RNN**	**LSTM**	**LSTM-GRU**	**Proposed non-adaptive LSTM-PSO** $c_1 = c_2 = 2$	**Proposed adaptive LSTM-PSO** c_1 and c_2 adaptive
Temperature	RMSE	12.67	6.24	6.58	**1.25**	**1.12**
	MADE	9.62	3.47	4.69	**1.35**	**1.08**
	MAPE	4.36	2.14	1.68	**0.44**	**0.26**
Relative humidity	RMSE	9.63	5.32	5.47	**1.39**	**1.03**
	MADE	8.71	3.86	3.56	**1.28**	**0.96**

	MAPE	3.94	1.76	1.34	**0.46**	**0.34**
Precipitation	RMSE	8.46	4.91	5.25	**5.68**	**5.14**
	MADE	6.35	2.56	1.81	**1.53**	**1.25**
	MAPE	3.54	1.28	0.93	**0.71**	**0.54**
Complexity of the models		O (n)	$O(n^2)$	O (n^2)	O (n^2)	O (n^2)

Table 4: Comparison of the error values in each climate parameter for Çankırı City

		Models				
Climate parameters	**Error metrics**	**Simple RNN**	**LSTM**	**LSTM-GRU**	**Proposed non-adaptive LSTM-PSO** c_1= c_2 = 2	**Proposed adaptive LSTM-PSO** c_1 and c_2 adaptive
Temperature	RMSE	6.35	5.24	4.86	**1.05**	**0.98**
	MADE	5.21	3.11	3.37	**0.72**	**0.68**
	MAPE	3.89	1.88	1.09	**0.29**	**0.13**
Relative humidity	RMSE	7.18	5.49	5.32	**1.27**	**1.19**
	MADE	5.42	3.53	3.18	**0.86**	**0.61**
	MAPE	2.47	1.45	1.06	**0.34**	**0.19**
Precipitation	RMSE	11.61	8.76	8.15	**4.67**	**4.21**
	MADE	5.67	4.68	1.52	**1.11**	**0.84**
	MAPE	2.09	1.13	0.54	**0.34**	**0.18**
Complexity of the models		O (n)	$O(n^2)$	O (n^2)	O (n^2)	O (n^2)

Table 5: Comparison of the error values in each climate parameter for Adıyaman City

We calculated the complexity of the proposed model according to the worst case. The Algorithm 1 is run as 1 nested while and 1 for loops. While j≤ N_e and j++ do, this cost is $O(N_e)$. N_e is the number of epochs. The for loop is run n times as for each p_i ∈ P do, P is the number of particles. So, the complexity of the proposed Algorithm 1 is O (N_e xP). The complexity of the algorithm is O (n^2) So, the complexity cost of Algorithm 2 is low. Also, the iteration time is 7 min and 36 secs, 6 min and 42 secs, 5 min and 27 secs, 4 min and 56 secs, and 4 min and 21 secs for Simple RNN, Conventional LSTM, LSTM-GRU, Proposed adaptive LSTM-PSO, and Proposed non-adaptive LSTM- PSO c_1= c_2 = 2, respectively.

From Table 5, we obtained with the proposed adaptive LSTM-PSO, the RMSE, MADE, and MAPE values of 0.98, 0.68, and 0.13 for temperature, 1.19, 0.61, and 0.19 for relative humidity, 4.21, 0.84, and 0.18 for precipitation, respectively. That is, when compared to other methods, minimal errors were achieved with the proposed methodology. From the results, the most successful results for both Çankırı and Adıyaman cities were obtained with the proposed method. Moreover, the estimation results in Adıyaman data are more encouraging than Çankırı. However, RMSE values of over 4% were obtained in the precipitation forecasts for both cities. It is clear that further refinement and optimization are required for this climate parameter data. Considering the average error rates, the proposed adaptive method achieved approximately 10% less error than the proposed non-adaptive, while the error rate was 40% less than LSTM-GRU, approximately 43% and 49% less than simple RNN, and conventional LSTM, respectively. From the results, we conclude that the worst method is simple RNN, and the best method is the proposed adaptive LSTM-PSO. In precipitation estimation, the proposed non-adaptive LSTM-PSO yielded 0.71 and 0.34 MAPE for Çankırı and Adıyaman, whereas the proposed adaptive LSTM-PSO obtained 0.54 and 0.18 MAPE for the two provinces, respectively.

Finally, the RMSE values obtained are lower than the other LSTM derivatives compared, and when deep learning and PSO optimization algorithms are used, it shows that there is a minimal difference between the observed values and the predicted values when estimating the climate parameters. From these results, it is understood that high success is shown in climate parameter estimation with time series. The reason why the results were successful and gave the least error is that we designed the LSTM architecture in an adaptive structure in accordance with current artificial intelligence technologies in the proposed approach, and we optimized the LSTM weight values by using the PSO metaheuristic method. We adjusted the hyperparameters and the number of neurons in Table 3 very well so that we could obtain better performance results than other studies in the literature. The LSTM-PSO hybrid model and the improvement techniques used show the superiority of this study over other studies.

4 Conclusion

In this paper, we propose an adaptive LSTM-PSO approach to forecast climate parameters. In the proposed approach, we used an LSTM deep architecture that is sensitive to input parameters and layers. We designed the number of layers and neurons in the deep network in an adaptive structure based on the data and parameters. We integrated the adaptive PSO algorithm into our work in order to optimize the LSTM input values and adapt the output data. In the proposed study, the time series of temperature, relative humidity, and precipitation, which are significant parameters for climate forecasting in Çankırı and Adıyaman, Turkey. For each parameter, the big data actual values and the estimated values are compared with each other in the proposed model. The proposed algorithms are modeled and coded using Python programming and libraries. In addition, the proposed approach is compared with the standard RNN, conventional LSTM, and LSTM-GRU models in terms of Root Mean Square Error (RMSE), Mean Absolute Deviation (MADE), and Mean Absolute Percentage Error (MAPE) metrics that reflect the prediction performance well. The proposed methods outperformed conventional LSTM models. The proposed adaptive method achieved approximately 6.3% less RMSE in temperature data prediction than the proposed non-adaptive LSTM-PSO. This study used both artificial intelligence technology and metaheuristic methods to predict climate data very efficiently. In future works, we plan to forecast natural disasters such as drought, flood, earthquake using improved LSTM and other time series-based hybrid DNNs.

Acknowledgments

We are grateful to the General Directorate of Meteorology of the Ministry of Agriculture and Forestry in Turkey for providing us with the data used in this study.

References

[Akdi and Unlü 2021] Akdi, Y., & Unlü, K. D. (2021). Periodicity in precipitation and temperature for monthly data of Turkey. Theoretical and Applied Climatology, 143(3), 957-968.

[Ali et al. 2018] Ali, M., Deo, R. C., Downs, N. J., & Maraseni, T. (2018). An ensemble-ANFIS based uncertainty assessment model for forecasting multi-scalar standardized precipitation index. Atmospheric Research, 207, 155-180.

[Ali et al. 2020] Ali, M., Deo, R. C., Xiang, Y., Li, Y., & Yaseen, Z. M. (2020). Forecasting long-term precipitation for water resource management: a new multi-step data-intelligent modelling approach. Hydrological Sciences Journal, 65(16), 2693-2708.

[Alizamir et al. 2020] Alizamir M, Kisi O, Ahmed AN, Mert C, Fai CM, Kim S, et al. (2020) Advanced machine learning model for better prediction accuracy of soil temperature at different depths. PLoS ONE 15(4): e0231055. https://doi.org/10.1371/journal.pone.0231055

[Altunkaynak and Nigussie 2018] Altunkaynak, A., & Nigussie, T. A. (2018). Monthly water demand prediction using wavelet transform, first-order differencing and linear detrending techniques based on multilayer perceptron models. Urban Water Journal, 15(2), 177-181.

[Apaydin et al. 2020] Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., & Chau, K. W. (2020). Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. Water, 12(5), 1500.

[Austin and Seed 2005] Austin, G., & Seed, A. (2005). Special issue on the hydrological applications of weather radar—guest editors' preface. Atmospheric Science Letters, 6(1), 1-1.

[Bayrak et al. 2021] Bayrak, G., Sevgen, S., & Samli, R. (2021). The Flow-Rate Prediction In Ergene Watershed. Carpathıan Journal Of Earth And Envıronmental Scıences, 16(2), 293-303.

[Bouktif et al. 2020] Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M.A. (2020). Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting, Energies, 13, 391.

[Buytaert et al. 2012] Buytaert, W., Friesen, J., Liebe, J., & Ludwig, R. (2012). Assessment and management of water resources in developing, semi-arid and arid regions. Water Resources Management, 26(4), 841-844.

[Chen et al. 2020] Chen, H., Yong, B., Shen, Y., Liu, J., Hong, Y., & Zhang, J. (2020). Comparison analysis of six purely satellite-derived global precipitation estimates. Journal of Hydrology, 581, 124376.

[Cheng et al. 2020] Cheng, M., Fang, F., Kinouchi, T., Navon, I. M., & Pain, C. C. (2020). Long lead-time daily and monthly streamflow forecasting using machine learning methods. Journal of Hydrology, 590, 125376.

[Ghimire et al. 2021] Ghimire, S., Yaseen, Z. M., Farooque, A. A., Deo, R. C., Zhang, J., & Tao, X. (2021). Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. Scientific Reports, 11(1), 1-26.

[Greff et al. 2017] Greff, K., Srivastava, R. K., Koutnık, J., Steunebrink, B.R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey, Transactions on Neural Networks and learning systems, 1-12.

[Gundu and Simon 2021] Gundu, V., & Simon, S. P. (2021). PSO–LSTM for short term forecast of heterogeneous time series electricity price signals, Journal of Ambient Intelligence and Humanized Computing 12:2375–2385.

[Hanab et al. 2021] Hanab, J. M., Ang, Y. Q. Malkawi, A., & Samuelson, H. W. (2021). Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements, Building and Environment, 192, 107601.

[Hasan et al. 2016] Hasan, M. M., Sharma, A., Johnson, F., Mariethoz, G., & Seed, A. (2016). Merging radar and in situ rainfall measurements: An assessment of different combination algorithms. Water Resources Research, 52(10), 8384-8398.

[Hassanzadeh et al. 2020] Hassanzadeh, Y., Ghazvinian, M., Abdi, A., Baharvand, S., & Jozaghi, A. (2020). Prediction of short and long-term droughts using artificial neural networks and hydro-meteorological variables. arXiv preprint arXiv:2006.02581.

[Hoang et al. 2020] Hoang, D. T., Yang, Pr. L., Cuong, L. D. P., Trung, P. D., Tu, N. H., Truong, L. V. , Hien, T. T., Nha, V. T. (2020). Weather prediction based on LSTM model implemented AWS Machine Learning Platform. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 8(5), pp. 283-290.

[Huffman et al. 2010] Huffman, G. J., Adler, R. F., Bolvin, D. T., & Nelkin, E. J. (2010). The TRMM multi-satellite precipitation analysis (TMPA). In Satellite rainfall applications for surface hydrology (pp. 3-22). Springer, Dordrecht.

[Hung et al. 2009] Hung, N. Q., Babel, M. S., Weesakul, S., & Tripathi, N. K. (2009). An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrology and Earth System Sciences, 13(8), 1413-1425.

[Hussain et al. 2021] Hussain, F., Wu, R. S., & Wang, J. X. (2021). Comparative study of very short-term flood forecasting using physics-based numerical model and data-driven prediction model. Natural Hazards, 107(1), 249-284.

[Kim et al. 2021] Kim, H., Ham, Y. G., Joo, Y. S. & Son, S. W. (2021). Deep learning for bias correction of MJO prediction. Nature Communications 12, 3087.

[Li et al. 2020] Li, Y., Wang, Q. J., He, H., Wu, Z., & Lu, G. (2020). A method to extend temporal coverage of high quality precipitation datasets by calibrating reanalysis estimates. Journal of Hydrology, 581, 124355.

[Li et al. 2020] Li, Y., Wang, Q. J., He, H., Wu, Z., & Lu, G. (2020). A method to extend temporal coverage of high quality precipitation datasets by calibrating reanalysis estimates. Journal of Hydrology, 581, 124355.

[Lin et al. 2009] Lin, G. F., Chen, G. R., Wu, M. C., & Chou, Y. C. (2009). Effective forecasting of hourly typhoon rainfall using support vector machines. Water Resources Research, 45(8).

[Liu et al. 2021] Liu, W., Wang, Z., Zeng, N., Alsaadi, F. E., & Liu, X. (2021). A PSO-based deep learning approach to classifying patients from emergency departments, International Journal of Machine Learning and Cybernetics 12, 1939–1948.

[Nasir and Hamdan 2021] Nasir, H. N., & Hamdan, A. N. A. (2021, March). Short-term and Long-term Drought Forecasts in Iraq Using Neural Networks and GIS. In IOP Conference Series: Materials Science and Engineering (Vol. 1090, No. 1, p. 012112). IOP Publishing.

[Nejat et al. 2020] Nejat, Z., Reza, N. H., Alireza, S., & Farshad, A. (2020). Prediction of the karstic spring flow rates under climate change by climatic variables based on the artificial neural network: a case study of Iran. Environmental monitoring and assessment, 192(6).

[Nkiaka et al. 2017] Nkiaka, E., Nawaz, N. R., & Lovett, J. C. (2017). Evaluating global reanalysis precipitation datasets with rain gauge measurements in the Sudano-Sahel region: case study of the Logone catchment, Lake Chad Basin. Meteorological Applications, 24(1), 9-18.

[Odusola and Abidoye 2015] Odusola, A., & Abidoye, B. (2015). Effects of temperature and rainfall shocks on economic growth in Africa. Available at SSRN 3101790.

[Park et al. 2019] Park, S., Berenguer, M., & Sempere-Torres, D. (2019). Long-term analysis of gauge-adjusted radar rainfall accumulations at European scale. Journal of hydrology, 573, 768-777.

[Pei et al. 2019] Pei, J., Deng, L., Song, S., et al. (2019). Towards artificial general intelligence with hybrid tianjic chip architecture", Nature, 572 (7767), 106-111.

[Pfeifroth et al. 2013] Pfeifroth, U., Mueller, R., & Ahrens, B. (2013). Evaluation of satellite-based and reanalysis precipitation data in the tropical Pacific. Journal of Applied Meteorology and Climatology, 52(3), 634-644.

[Pumo et al. 2017] Pumo, D., Arnone, E., Francipane, A., Caracciolo, D., & Noto, L. V. (2017). Potential implications of climate change and urbanization on watershed hydrology. Journal of Hydrology, 554, 80-99.

[Ravuri et al. 2021] Ravuri, S. et al. (2021). Skilful precipitation nowcasting using deep generative models of radar. Nature 597, 672–677.

[Sarker 2021] Sarker, I.H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN Computer Science 2, 420.

[Seo et al. 2018] Seo, Y., Kwon, S., & Choi, Y. (2018). Short-term water demand forecasting model combining variational mode decomposition and extreme learning machine. Hydrology, 5(4), 54.

[Shao et al. 2019] Shao, B., Li, M., Zhao, Y. & Bian, G. (2019). Nickel Price Forecast Based on the LSTM Neural Network

[Sun et al. 2018] Sun, Q., Miao, C., Duan, Q., Ashouri, H., Sorooshian, S., & Hsu, K. L. (2018). A review of global precipitation data sets: Data sources, estimation, and intercomparisons. Reviews of Geophysics, 56(1), 79-107.

[Van de Giesen et al. 2014] Van de Giesen, N., Hut, R., & Selker, J. (2014). The trans-African hydro-meteorological observatory (TAHMO). Wiley Interdisciplinary Reviews: Water, 1(4), 341-348.

[Wang et al. 2019] Wang, G., Zhang, X., & Zhang, S. (2019). Performance of three reanalysis precipitation datasets over the qinling-daba mountains, eastern fringe of tibetan plateau, China. Advances in Meteorology.

[Wang et al. 2019] Wang, G., Zhang, X., & Zhang, S. (2019). Performance of three reanalysis precipitation datasets over the qinling-daba mountains, eastern fringe of tibetan plateau, China. Advances in Meteorology, 2019.

[Wang et al. 2021] Wang, K., Ma, C., Qiaoa, Y., Lua, X., Hao, W., & Dong, S. (2021). A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction", Physica A: Statistical Mechanics and its Applications, 583, 126293.

[Yakut and Süzülmüs 2020] Yakut, E., & Süzülmüş, S. (2020). Modelling monthly mean air temperature using artificial neural network, adaptive neuro-fuzzy inference system and support vector regression methods: A case of study for Turkey. Network: Computation in Neural Systems, 31(1-4), 1-36.

[Yuce and Esit 2021] Yuce, M.I. & Esit, M. (2021) Drought monitoring in Ceyhan Basin, Turkey. Journal of Applied Water Engineering and Research. 0 (0), 1–22. doi:10.1080/23249676.2021.1932616.

[Zubaidi et al. 2020] Zubaidi, S. L., Abdulkareem, I. H., Hashim, K. S., Al-Bugharbee, H., Ridha, H. M., Gharghan, S. K., Al-Qaim, F. F., Muradov, M., Kot, P., & Al-Khaddar, R. (2020). Hybridised artificial neural network model with slime mould algorithm: a novel methodology for prediction of urban stochastic water demand. Water, 12(10), 2692.