

Energy Sources, Part A: Recovery, Utilization, and Environmental Effects

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ueso20

Deep learning approach for one-hour ahead forecasting of weather data

Arif Ozbek

To cite this article: Arif Ozbek (2023) Deep learning approach for one-hour ahead forecasting of weather data, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 45:3, 7606-7628, DOI: 10.1080/15567036.2023.2222690

To link to this article: https://doi.org/10.1080/15567036.2023.2222690

Published online: 12 Jun 2023.

Submit your article to this journal

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ueso20

ENERGY SOURCES, PART A: RECOVERY, UTILIZATION, AND ENVIRONMENTAL EFFECTS
2023, VOL. 45, NO. 3, 7606–7628
https://doi.org/10.1080/15567036.2023.2222690

Taylor & Francis
Taylor & Francis Group

Check for updates

Deep learning approach for one-hour ahead forecasting of weather data

Arif Ozbek

Department of Mechanical Engineering, Ceyhan Engineering Faculty, Cukurova University, Adana, Türkiye

ABSTRACT

Weather is made up of multiple parameters, including solar radiation (SR), atmospheric pressure (AP), soil temperature (ST), atmospheric temperature (AT), wind speed (WS), relative humidity (RH), and sunshine duration (SD). These factors are also crucial for the renewable energy sector, solar simulation, agriculture, air pollution, water supply and distribution, avalanche warning, forestry, and town and regional planning. A deep learning method based on a neural network with Long Short-Term Memory (LSTM) was employed in this investigation for one-hour-ahead weather data forecasting. The ability of the LSTM model was compared with the Adaptive Neuro-Fuzzy Inference System (ANFIS) with that of the fuzzy c-means (FCM), Autoregressive Integrated Moving Average (ARIMA) model, and the Autoregressive Moving Average (ARMA) model. Mean absolute error (MAE), correlation coefficient (R), root means square error (RMSE), average bias, Nash – Sutcliffe efficiency coefficient (NSE), and mean absolute percentage error (MAPE) were selected as evaluation criteria. Results indicated that the proposed LSTM model presented good enough results compared to other used methods. 7 different types of meteorological data from a total of 4 years (35040 hours) were divided into 25% test data and 75% training data for the models. The best result was obtained for the hourly ST estimation of Adana province using the LSTM method, the MAE, RMSE, R, bias, NSE, and MAPE values were computed as 0.016°C, 0.078°C, 0.9999, −0.00018°C, 0.0805%, and 0.9999, respectively. On the other hand, the worst result was obtained for the hourly SD for Mardin province when ARIMA was used, and the statistical measures were derived as 0.128 hours for MAE, 0.215 hours for RMSE, 0.8851 for R, 0.00091 hours for bias, and 0.7657 for NSE. In this regard, it is demonstrated that the LSTM technique outperformed the other models in terms of all-weather data estimates and delivered highly sensitive outcomes.

ARTICLE HISTORY
Received 13 January 2023
Revised 8 May 2023
Accepted 3 June 2023

KEYWORDS
Atmospheric parameters; estimation; ARMA; ARIMA; LSTM; ANFIS FCM

Introduction

The ecosystem's quality is impacted by environmental factors brought on by numerous weather elements. The effects of meteorological factors have a domino effect since they are not limited to the atmosphere. Interactions between temperature, pressure, and humidity (moisture) can result in clouds. These clouds can then reduce the amount of solar energy that reaches plants or increase precipitation, which can fall into a body of water. In addition to heating the air, persistently high temperatures can promote heat transfer to neighboring bodies of water. Similarly to this, a lack of precipitation affects soil moisture and water levels due to evaporation, as well as the weather. Wind speed and direction might indicate the approach of a front. As a result, an accurate estimation of atmospheric characteristics is very important to more readily analyze the consequences of weather

CONTACT Arif Ozbek arozbek@cu.edu.tr Department of Mechanical Engineering, Ceyhan Engineering Faculty, Cukurova University, Adana 01950, Türkiye

© 2023 Taylor & Francis Group, LLC

parameters on the environmental ecosystem. Therefore, many papers have been written to forecast meteorological parameters.

The total amount of electromagnetic radiation that the sun emits across a given frequency range is known as solar irradiation (SR). Solar energy, which can be used directly or indirectly, is among the most available and adaptable renewable energy sources. Convolutional neural network – long short term memory – artificial neural network (CNN – LSTMANN) and convolutional neural network – artificial neural network (CNN – ANN) were developed by Mukhtar et al. (2022) to forecast hourly global solar radiation. LSTM was employed by Qing and Niu (2018) to forecast SR. Their study includes several variables over a period of 2.5 years, including wind speed, humidity, dew point, temperature, weather type, and visibility. It was discovered that LSTM outperforms linear regression and multilayered feedforward neural networks. Solano, Dehghanian, and Affonso (2022) compared the performance of multiple machine learning methods (Voting-Average (VOA), which combines SVR, CatBoost, and XGBT), Extreme Gradient Boosting (XGBT), Support Vector Regression (SVR), and Categorical Boosting (CatBoost)) for SR forecasting utilizing endogenous and exogenous inputs and offered an ensemble feature selection method that involves selecting the most relevant input variables and their historical observation values. One of the key elements controlling the temperature and air on Earth is atmospheric pressure (AP) (Erdil and Arcaklioglu 2013). Climate, agriculture, atmospheric and environmental research, meteorology, animal and human life, and the living ecosystem of Earth all benefit from atmospheric pressure (AP). Estimating air pressure in the short term is essential to tracking any changes in local weather. The storm's strength and potential danger in that area can be forecast if the AP is calculated accurately. The abrupt reduction in air pressure suggests the potential for inclement weather. Long short-term memory (LSTM) neural network, adaptive neuro-fuzzy inference system (ANFIS) with fuzzy c-means (FCM), the ANFIS with grid partition (GP), and the ANFIS with subtractive clustering (SC) were all employed by Bilgili, Ilhan, and Ünal (2022) to forecast AP one hour ahead. To estimate the daily minimum and maximum pressures, a model of an artificial neural network (ANN) was utilized by Erdil and Arcaklioglu (2013). They demonstrated that the suggested ANN model could have several benefits, simplicity, including learning from examples, and calculation speed. An ERA5-based hourly global pressure and temperature model (HGPT) based on Fourier analysis and temporal segmentation was demonstrated by Mateus et al. (2020). Existing soil temperature (ST) estimation approaches have recently been split into two broad categories: statistical methods and other data-driven methodologies (Sanikhani et al. 2018). Statistical methods heavily rely on actual ST data and physical soil model output to calculate the results. Statistical methods, in contrast to data-driven procedures, are often simple models with unambiguous explanations. Zeynoddin et al. (2019) claim that ST estimation can be viewed as a time series problem that can be solved using linear techniques (ARMA, ARIMA, SARIMA). They outperformed the most extensively used nonlinear ST modeling technique, the multilayer perceptron neural network (MLPNN). They measured ST based on previous observations of ST. Sanikhani et al. (2018) demonstrated that the ELM approximation produced the Mersin station's most precise performance when compared to the M5 Tree and ANN models. The fundamental support vector machine (SVM), which chooses a function that reliably predicts future values, is the basis for SVRs. Delbari, Sharifazari, and Mohammadi (2018) made the ST estimation as a result of the SVR model's capacity to reliably predict future values. The ANN architecture in the study has a lot in common with brain neurons. This technique is appropriate for modeling across a very broad variety of applications because of the flexibility of its architecture (Feng et al. 2019; Sanikhani et al. 2018). Other data-driven methods, such as eXtreme gradient boosting (XGBoost) and long short-term memory (LSTM) were rarely used in the ST estimation scenario; nonetheless, they had previously been successful in other estimating scenarios (Yao et al. 2020). To accurately predict soil temperature and moisture, Li et al. (2022) suggested an attention-aware LSTM method. They pointed out that in most situations, their suggested model performs better than the Random Forest (RF), Support Vector Regression (SVR), original Long Short-Term Memory (LSTM), Elastic-Net (ENET), and attention LSTM (A-LSTM) models. In order to calculate the water table's depth, Zhang et al. (2018) provided an LSTM layer with another fully connected layer on top

while using the dropout method to prevent over fitting. The LSTM network was employed by Guo et al. (2019) to reflect aberrant fluctuation times, and real-time data was used to predict short-term anomalous passenger flow. Meteorological data is an example of sequential data. As a result, time series prediction approaches have been employed to solve numerous weather forecast problems, including WS (Ozbek et al. 2022), precipitation (Wang et al. 2020), and atmospheric temperature (Sekertekin et al. 2021). One of the most significant responsibilities in meteorology is atmospheric temperature (AT) forecasting; accurate temperature forecasting is critical to the economy and human daily lives. When forecasting AT for a short-term time frame, using single statistical analysis methods such as hourly and daily statistical analysis, accurate outcomes can be obtained. To make climate predictions, Li et al. (2020) revealed a lot of statistical and machine-learning algorithms and examined how well they performed at downscaling long-term daily atmospheric air temperature. Prediction-based research has also used support vector regression (SVR), one of the supervised learning techniques. Utilizing weather information from the Republic of Georgia, Chevalier et al. (2010) improved the SVR approach to produce short-term AT predictions. An LSTM model was created for the estimation of AT (Ozbek et al. 2021). The forecast of the short-term temperature was mostly accurate compared to the LSTM model applied using long-term data. There have been three notable methods for predicting wind speed (WS) so far. The first way involves physical techniques, such as those for numerical weather prediction (NWP) (Piotrowski et al. 2019), which mostly estimate WS using mathematical models of the seas and atmosphere. NWP is frequently a physical approximation with great precision and a robust foundation. The second method uses statistical approaches to create input and output function models using observed and historical data. Regression with support vectors (SVR), the Kalman filter, the gray prediction method and Gaussian regression, and the autoregressive integrated moving average model (ARIMA) are among these methods (Xia et al. 2020). The third method involves using machine learning techniques to create a model of the relationship between inputs and outputs that is comparable to the neural network-based synaptic connection structure. Numerous fields, including wind power or wind speed forecasts, context categorization, and cloud services have successfully used machine learning (Rjoub et al. 2019; Sarker, Kayes, and Watters 2019). Deep learning has recently been widely utilized to predict WS (Ozbek et al. 2022). Khosravi, Machado, and Nunes (2018) used a multilayer feed-forward neural network (MLFFNN), the fuzzy inference system (FIS), and SVR methods to estimate WS. After applying wavelet transform to analyze the original data of WS, Pei et al. (2019) introduced a new cell updating network for long short-term memory (LSTM) to estimate the future WS. To estimate WS, Ma et al. (2019) employed the PSO algorithm to optimize the negative correlation neural network model. A shared weights LSTM model was presented by Zhang et al. (2019) to estimate WS. Yalçın (2022) suggested a method for forecasting meteorological conditions utilizing time series and deep hybrid neural networks. The method used long short-term memory (LSTM) – convolutional neural networks (CNNs) models to train and estimate the important variables for weather estimation, including relative humidity (RH), atmospheric pressure (AP), atmospheric temperature (AT), and wind speed (WS). The data were used to compare his suggested method to machine learning and deep learning techniques. The term relative humidity (RH) refers to the quantity of water vapor (or moisture content) in the air that may be retained by it at a specific temperature and pressure without forming condensation. An accurate RH estimate is essential for environmental and economic policies and industrial and agricultural operations. Models created utilizing the artificial neural network technique can very precisely estimate the outdoor atmospheric temperature and RH (Al-Shawwa et al. 2018). A study on the LSTM network technique for estimating RH was done in 2020 by Hutapea et al. (2020). RH estimation has only received a small amount of research due to its complexity and non-linear character. In the work by Qadeer et al. (2021) and Ozbek, Ş, and Bilgili (2022), RH was calculated using a machine-learning technique. According to studies, the machine learning approach can forecast RH with a high degree of accuracy, and many applications, including the operation and design of thermal power plants, cooling towers, and HVAC systems, can make use of this information. One of the most significant and useful meteorological indicators is the sunshine duration (SD), which is commonly used in solar radiation

forecasts (Badescu 1999), agriculture applications (Brown 2012), and studies of climate change (Matuszko and Węglarczyk 2014). Based on the percentage of cloud cover, the maximum and minimum air temperatures, and the ratio of real to astronomical sunshine durations, El-Metwally (2005) developed a nonlinear regression model to predict the relative SD in Egypt. El-Kenawy et al. (2021) constructed ensemble hybrid learning models for estimating daily SD using meteorological variables such as daily minimum, mean, and maximum air temperatures, daily temperature range, and daily mean relative humidity as input and compared the results obtained from the models with the results in the literature. In four dry locations in Iran, Rahimikhoob (2014) evaluated the capability of ANNs in predicting the length of the day. Temperature and air humidity were regarded as independent factors in this study. The findings demonstrated that ANNs could forecast sunshine duration with a level of acceptable accuracy. The goal of Uçkan and Khudhur (2022) was to compare sunshine duration-based models with other models based on meteorological parameters and to create new forecasting models. Their findings indicated that models based on additional meteorological factors produced more accurate forecasts. Kandirmaz, Kaba, and Avci (2014) used radial basis function, generalized regression neural network (GRNN), and multilayer perceptron (MLP) to estimate the monthly mean daily sunshine duration values in Türkiye based on the cloud cover index, month number, and day length. Using the SVR technique, Kaba, Kandirmaz, and Avci (2016) created a forecasting model for calculating the daily SD. RH, maximum and minimum AT, WS, day length, and cloud cover were taken into account as significant variables in that study. Machine learning models have been employed in the estimation of different parameters. Modeling the solubility of various acids in carbon dioxide was done by Bemani et al. (2020) using a least squares support vector machine (LSSVM), artificial neural networks (ANN), multilayer perceptron (MLP), and adaptive neuro-fuzzy inference system (ANFIS). LSSVM technique's efficiency in forecasting biodiesel production as a function of pressure, temperature, ethanol/oil ratio, and reaction time was assessed by Liu and Baghban (2017).

In this work, a deep learning method based on the LSTM neural network is suggested for the time series prediction of several meteorological parameters. The ability of the LSTM model was compared with the traditional statistical methods (ARMA and ARIMA), and a machine learning method (ANFIS FCM). Turkish measurement stations' time series data for several meteorological characteristics were utilized as sample data. This work aims to compare the capabilities of the LSTM model with the Adaptive Neuro-Fuzzy Inference System (ANFIS) with fuzzy c-means (FCM), the Autoregressive Integrated Moving Average (ARIMA) model, and the Autoregressive Moving Average (ARMA) model. The time-series technique based on the LSTM neural network enables the use of historical data as an independent variable by making predictions by accounting for the data's underlying periodicity. Therefore, using the LSTM neural network model to estimate different meteorological parameters over time series, the estimation capability of the LSTM approach was compared in this study between standard statistical models such as ARMA and ARIMA and machine learning methods such as ANFIS. In addition, it has been shown that one-hour forward forecasting of weather parameters can be made using the model created, without the need for complex calculations or large amounts of environmental, topographic or meteorological data.

Material and methods

Autoregressive moving average (ARMA) model

The ARMA model, also known as the Box-Jenkins model, is a standard statistical model for time series forecasts and analyses. An ARMA technique employs a moving average (MA) model with an autoregressive (AR) model. As ARMA models can simulate the unknown process with the fewest parameters when compared to pure AR and MA models, they offer the most effective stationary time series linear model (Jenkins and BGE, 1976).

To accurately model the underlying process in some experimental time series, an extremely high-order MA or AR model is necessary. In this aspect, a mixed ARMA model may be a better solution on occasion. The ARMA model defines the conditional average of y_t as a result of both previous observations, $y_{t-1}, \ldots, y_{t-p}$, and previous innovations, $\varepsilon_{t-1}, \ldots, \varepsilon_{t-q}$. The AR degree is the number of previous observations on which y_t depends, p. The MA degree is the number of previous inventions on which y_t is dependent. In general, ARMA is used to represent these models *(p,q)*. The ARMA*(p,q)* model is denoted as (BGE, Jenkins, and Reinsel 1994):

$$y_t = c + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{1}$$

where ε_t is the mean of an unrelated innovation procedure. Considering Eq. (1) differently, the polynomial notation for the lag operator is presented as

$$L^i y_t = y_{t-1} \tag{2}$$

The polynomial of degree p AR lag operator can be defined as

$$\phi(L) = \left(1 - \phi_1 L - \ldots - \phi_p L^p\right) \tag{3}$$

On the other hand, the MA lag operator polynomial degree q can be written as

$$\theta(L) = \left(1 - \theta_1 L - \ldots - \theta_q L^q\right) \tag{4}$$

According to this, the model of ARMA(p,q) can express as

$$\phi(L) y_t = c + \theta(L) \varepsilon_t \tag{5}$$

Autoregressive integrated moving average (ARIMA) model

The ARIMA method generates nonstationary integration series of order D. This procedure is indicated *I(D)*. A nonstationary *I(D)* process can be executed stationary obtaining D differences. These processes are frequently referred to as unit root or difference-stationary processes. After being differenced D times, ARIMA*(p,D,q)* is a series that can be represented by a stationary ARMA*(p,q)* process. The ARIMA*(p,D,q)* model's form is expressed as (BGE, Jenkins, and Reinsel 1994).

$$\Delta^D y_t = c + \Delta^D \phi_1 y_{t-1} + \ldots + \phi_p \Delta^D y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{6}$$

where $\Delta^D y_t$ is a time series with D^{th} differences. The ARIMA*(p,D,q)* model can also be expressed in lag operator notation as

$$\phi^*(L) y_t = \phi(L)(1 - L)^D y_t = c + \theta(L) \varepsilon_t \tag{7}$$

where ϕ^* is a polynomial with an unstable AR operator that has exactly D unit roots. $(1 - L)^D$ is the non-seasonal differencing operator, which compensates for nonstationarity in data taken throughout time.

Long short term memory (LSTM) neural network

LSTM neural network is one of the earliest techniques to address short-term input-skipping difficulties and long-term information preservation in latent variable methods (Hochreiter and Schmidhuber 1997). The LSTM memory cell has the same shape as the hidden state and is designed to store additional information. LSTM was designed with inspiration from a computer's logic gates. A set of gates is needed to control the memory cell. To read the inputs from the cell, a gate called the output gate is needed. A second gate called the input gate, is required to determine when to read data into the

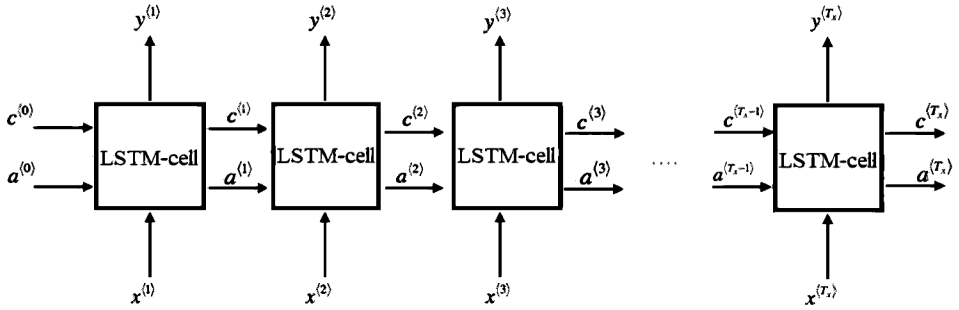

Figure 1. The structure of the basic LSTM cell for the time steps.

cell. Finally, a mechanism is required to reset the contents of the cell, which is controlled by a forget gate.

The basic LSTM cell's structure for the time steps is shown in Figure 1. An LSTM neural network, as illustrated in the figure, can be described as the repeat of a single cell. The order of the input $x = \left(x^{\langle 1\rangle}, x^{\langle 2\rangle}, \ldots, x^{\langle T_x\rangle}\right)$ is held over T_x time steps. The network outputs $y = \left(y^{\langle 1\rangle}, y^{\langle 2\rangle}, \ldots, y^{\langle T_x\rangle}\right)$.

Figure 2 depicts the processes of a single LSTM-cell time step. The LSTM-cell receives as input the hidden state from the current time step's input data, $x^{\langle t\rangle}$and the previous cell $a^{\langle t-1\rangle}$. Furthermore, LSTM-cell is tracked and updated by a memory variable $c^{\langle t\rangle}$ or a "cell state" at each time step, which can be various from $a^{\langle t\rangle}$. It is a hidden state $a^{\langle t\rangle}$and outputs an estimation $y^{\langle t\rangle}$for time-step.

$$\Gamma_f^{\langle t\rangle} = \sigma\left[W_f\left(a^{\langle t-1\rangle}, x^{\langle t\rangle}\right) + b_f\right] \tag{8}$$

Here, W_f and b_f are the weights that manage the functions of the forget gate. Eq. (8) results in a vector $\Gamma_f^{\langle t\rangle}$with values ranging from 0 to 1. The earlier cell state $c^{\langle t-1\rangle}$ will be multiplied element-wise by this forget gate vector. If one of the $\Gamma_f^{\langle t\rangle}$ values is 0, the information should be eliminated by the LSTM from the corresponding component of$c^{\langle t-1\rangle}$. Information is stored if one of the variables is 1.

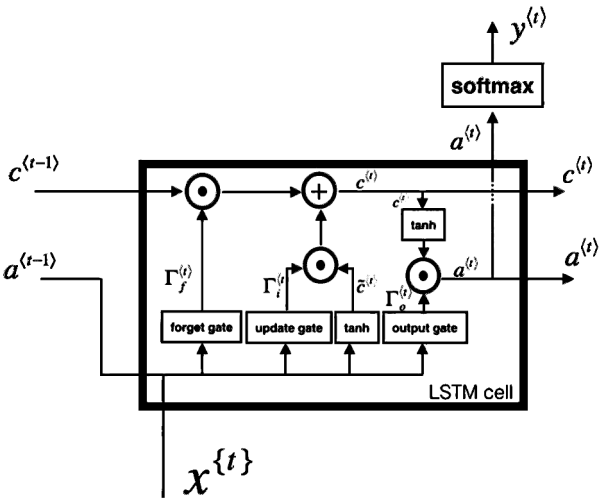

Figure 2. An LSTM-operations cell's.

Forget gate: In an LSTM-cell, the following mathematical expression is obtained in the forget gate $\Gamma_f^{\langle t \rangle}$:

Update gate: The update gate $\Gamma_u^{\langle t \rangle}$ determines which aspects of the candidate $c^{\langle t \rangle}$ to add to the cell state $c^{\langle t \rangle}$. It determines which components of a candidate tensor $c^{\langle t \rangle}$ are passed to the cell state$c^{\langle t \rangle}$.

The update gate $\Gamma_u^{\langle t \rangle}$ is used to decide what aspects of the candidate $c^{\langle t \rangle}$ to add to the cell state $c^{\langle t \rangle}$. It decides what parts of a candidate tensor are passed onto the cell state. The update gate formula is as follows:

$$\Gamma_u^{\langle t \rangle} = \sigma\left[W_u\left(a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right) + b_u\right] \tag{9}$$

where W_u and b_u are the weights that manage the update gate's behavior. $\Gamma_u^{\langle t \rangle}$is a vector of values between 0 and 1, similar to the forget gate. A unit in the update gate allows the value of the candidate to be passed onto the hidden state when it is close to 1. A unit in the update gate prevents the corresponding value in the candidate from being passed onto the hidden state when it is nearly 0.

Updating the cell: A new number vector that can be combined with the previous cell state must be created to update the new subject. In this sense, the candidate value $c^{\langle t \rangle}$ equation is written as

$$c^{\langle t \rangle} = \tanh\left[W_c\left(a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right) + b_c\right] \tag{10}$$

Here, W_c and b_c are the weights. The candidate value $c^{\langle t \rangle}$ is a tensor including information from the current time step that may be kept in the current cell state. The update gate determines which components of the candidate value are transferred. Last, the *tanh* function produces values between −1 and + 1 in the candidate function.

The memory that is passed on to future time steps is the cell state $c^{\langle t \rangle}$. The candidate value and the previous cell state are combined to create the new cell state, which is defined as

$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} * c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} * c^{\langle t \rangle} \tag{11}$$

Output gate: The output gate $\Gamma_o^{\langle t \rangle}$ is identified by the previous current input $x^{\langle t \rangle}$and the hidden state $a^{\langle t-1 \rangle}$ as follows:

$$\Gamma_o^{\langle t \rangle} = \sigma\left[W_o\left(a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right) + b_o\right] \tag{12}$$

Here, W_o and b_o are the output gate weight and output gate bias, respectively. The output gate $\Gamma_o^{\langle t \rangle}$ is similar to the other gates. It contains values between 0 and 1. It chooses what information is sent as the time step estimate.

Hidden state: The three gates of the following time step are chosen using the hidden state, which is passed to the LSTM cell's subsequent time step. Also, it is used for the prediction $y^{\langle t \rangle}$. This state is defined as

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} * \tanh\left(c^{\langle t \rangle}\right) \tag{13}$$

Adaptive neuro-fuzzy inference system (ANFIS)

Construction and training are the two processes that typically make up an ANFIS model. In the construction phase, the input-output data is separated into rule patches, and the types and numbers of membership functions are defined. Clustering algorithms are utilized in the ANFIS model to grasp and categorize inputs into groups that aid in the training phase. This explains why three clustering approaches were used: FCM, subtractive clustering (SC), and grid partitioning (GP). ANFIS is discussed extensively in FCM. MATLAB & Simulink (2022), and Karakuş, Kuruoğlu, and Altınkaya (2017).

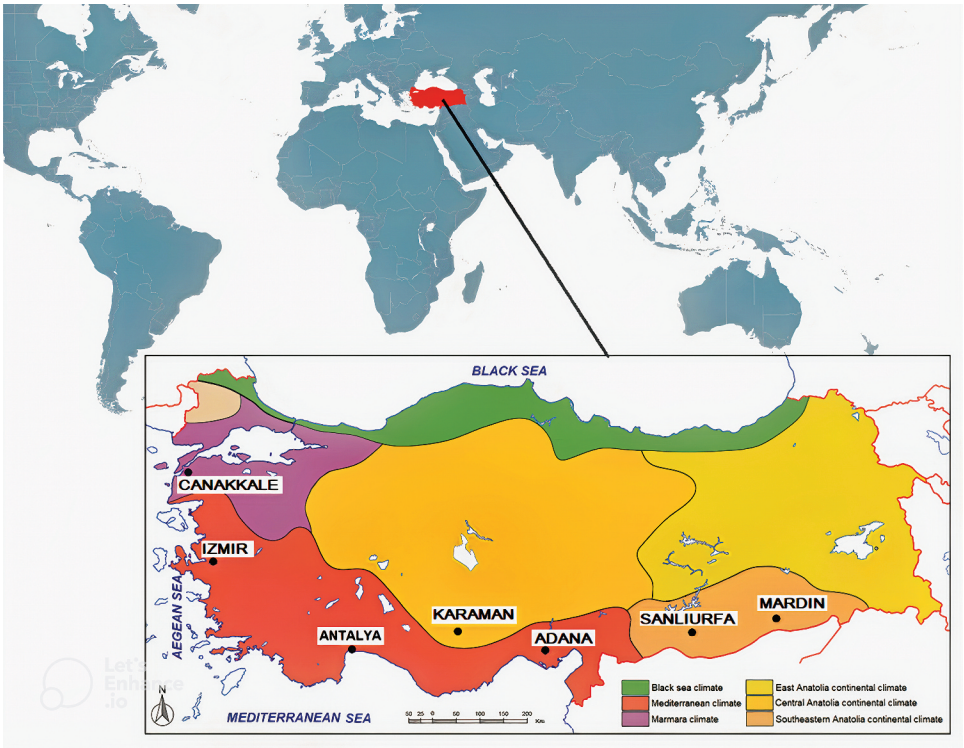

Figure 3. Locations of measuring stations.

Study area and data preparation

The study's locations of interest are placed on a map of Türkiye, as shown in Figure 3. The analyses were conducted using 4-year hourly weather data measured by The State Meteorological Department of Türkiye between 2016 and 2019 in the provinces of Karaman, Antalya, Adana, Izmir, Canakkale, Sanlıurfa, and Mardin (Figure 3). The locations of the stations are shown in Table 1, along with a statistical breakdown of the dataset. This table displays the maximum, minimum, average, and standard deviation values for all measured parameters.

Figure 4 illustrates a block diagram of the suggested LSTM estimation technique. This figure gives the basic processes for the model. After the weather data was collected from the stations, by interpolating missing values, it was pre-processed and modified for the subsequent stage. After normalizing the original data, the suggested model was trained, validated, and tested. The models designed as statistical, deep learning and machine-learning approach were then applied to these data separately as a time series prediction. To assess the precision of the proposed model, different statistical measures

Table 1. Stations' geographic locations and a statistical description of the dataset.

Station	Location		Parameter Measured	Max	Min	Ave	STD
Karaman	37°11′35.5″N 33°13′12.7″E	**SR**	Solar Radiation, (Wh/m^2)	1085.1	0.0	213.87	303.672
Antalya	36°54′22.7″N 30°47′56.4″E	**AP**	Atmospheric Pressure, (hPa)	1024.8	985.8	1006.59	5.862
Adana	37°00′14.8″N 35°20′39.5″E	**ST**	Soil Temperature, (°C)	33.4	9.3	22.26	7.337
Izmir	38°28′42.1″N 26°50′06.8″E	**AT**	Atmospheric Temperature, (°C)	40.5	−2.6	19.12	8.021
Canakkale	40°08′27.6″N 26°23′57.5″E	**WS**	Wind Speed, (m/s)	24.6	0.0	3.53	2.251
Sanlıurfa	37°26′43.4″N 38°54′12.6″E	**RH**	Relative Humidity, (%)	100.0	6.0	47.26	24.152
Mardin	37°18′37.1″N 40°43′42.2″E	**SD**	Sunshine Duration, (hour)	1.0	0.0	0.36	0.453

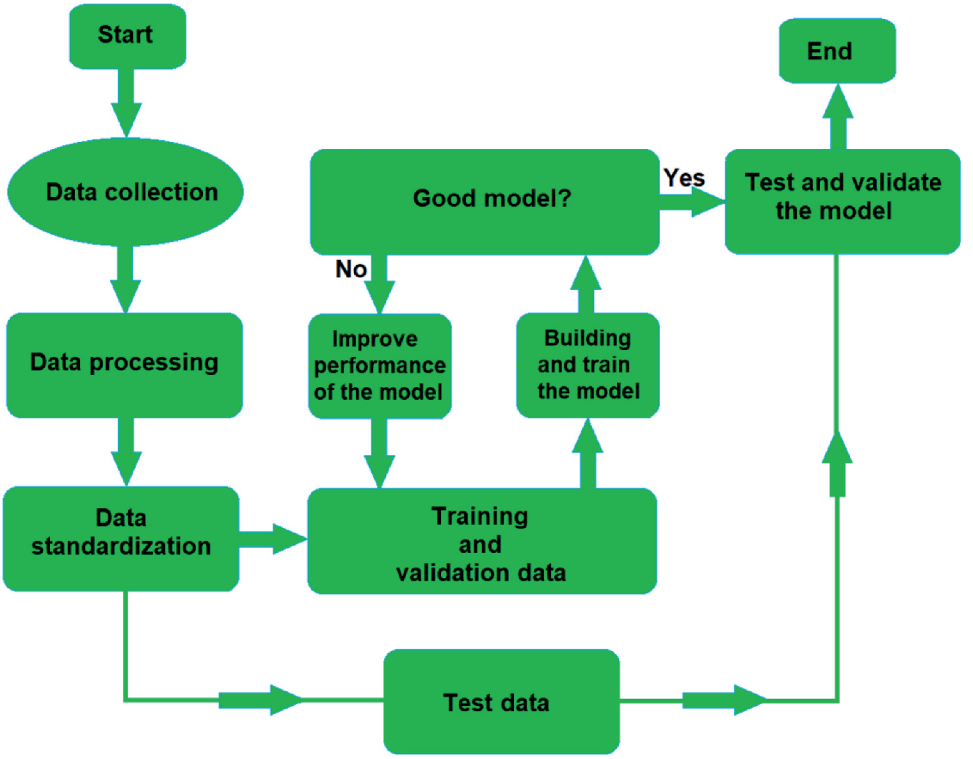

Figure 4. Flowchart for the model proposed in the estimation.

like MAE, RMSE, R, average bias, and MAPE were applied. To run the models, MATLAB R2018a (Trail Version) was utilized. In addition, a part of the LSTM model software used during the analysis is given in the Appendix section.

Error analysis for suggested approaches

In that study, model validity is evaluated using six statistical error criteria: mean absolute error (MAE), root mean square error (RMSE), correlation coefficient R, average bias, mean absolute percentage error (MAPE), and Nash-Sutcliffe efficiency coefficient (NSE) (Nash and Sutcliffe 1970). These error standards are used to control the accuracy of the estimations based on observable data. These error criteria's mathematical expressions are given in Eqs. (14), (15), (16), (17), (18) and (19) respectively.

The average of all absolute errors is known as the mean absolute error (MAE) and the smaller value, the better. The formula is shown in Eq. 14:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|p(i) - o(i)| \tag{14}$$

The RMSE measures the average mismatch between calculated and measured values. The lower the value, the better. It is given by Eq. 15:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[p(i) - o(i)]^2} \tag{15}$$

The correlation between two data sets is described by the correlation coefficient (R). The R-value is between 0 and 1. 1 point out that the model properly fits the relevant data set, whereas a value of 0 expresses that the model is not significant. It is given as Eq.16:

$$R = \left(\sum_{i=1}^{N}[p(i)-\bar{p}][o(i)-\bar{o}]\right) / \left(\sqrt{\sum_{i=1}^{N}[p(i)-\bar{p}]^2}\sqrt{\sum_{i=1}^{N}[o(i)-\bar{o}]^2}\right) \tag{16}$$

The bias is the average prediction error, which represents a forecast model's systematic error to under or forecast and its value is expected to be close to zero. (Eq.17).

$$BIAS = \frac{1}{N}\sum_{i=1}^{N}(p(i)-o(i)) \tag{17}$$

The mean absolute percentage variation between calculated and measured values are represented by the MAPE. The smaller the value, the better. It is given by Eq. 18.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|p(i)-o(i)|/o(i) * 100 \tag{18}$$

The Nash-Sutcliffe efficiency (NSE) is evaluated as one minus the ratio of the modeled time series' error variance divided by the variance of the observed time series. The NSE value ranges from 0 to 1. A value of 1 shows the model that fits perfectly, while a value of 0 displays that the model is not significant for the related data set.

$$NSE = 1 - \frac{1}{N}\sum_{i=1}^{N}|p(i)-o(i)|^2 / \frac{1}{N}\sum_{i=1}^{N}|o(i)-\bar{o}|^2 \tag{19}$$

where, respectively, p*(i)* and o*(i)* stand for the forecasted and real values at the time *i*. Additionally, the mean values of the forecasted and real values are shown, respectively, in, $\bar{p}$ and $\bar{o}$. N represents the total number of data.

Results and discussions

Results of the LSTM model

For the provinces to be employed in all models, Figure 5 depicts the meteorological data values obtained from training (3 years) and testing (1 year) data. In the training and testing procedures, 35040 hourly observed data points were employed. Within these measurement values, the models were trained using 26,280 pieces of data (75% of the whole dataset) from the 1st of January 2016 to the 31st of December 2018 and tested using the final 8760 hourly observed data (25% of the whole dataset) from 1st of January 2019 to 31st of December 2019. The same training and testing percentages were used across all provinces and atmospheric conditions. The number of real (observed) values is represented on the X-axis, while the atmospheric parameters are represented on the Y-axis.

Observed hourly solar radiation (SR) values, training and testing SR data for the models for Karaman province are seen in Figure 5(a). The maximum SR value is 1085.1 Wh/m^2, due to nighttime, while the minimum is of course measured at 0.0 Wh/m^2. The mean of all testing and training data is 213.87 Wh/m^2, with a standard deviation of 303.672 Wh/m^2 for this data set. Training and testing observed hourly atmospheric pressure (AP) data for the methods in Antalya province are shown in Figure 5(b). The highest AP value is 1024.8 hPa, while the lowest AP value is 985.8 hPa. This data group's average value for all training and testing AP data is 1006.59 hPa, with a standard deviation of 5.862 hPa. Figure 5(c) shows the observed hourly soil temperature (ST) values, testing and training ST data for the models for the province of Adana. The maximum ST value is 33.4°C, while the minimum ST is 9.3°C. The average temperature of all testing and training data is 22.26 degrees Celsius, with a standard deviation

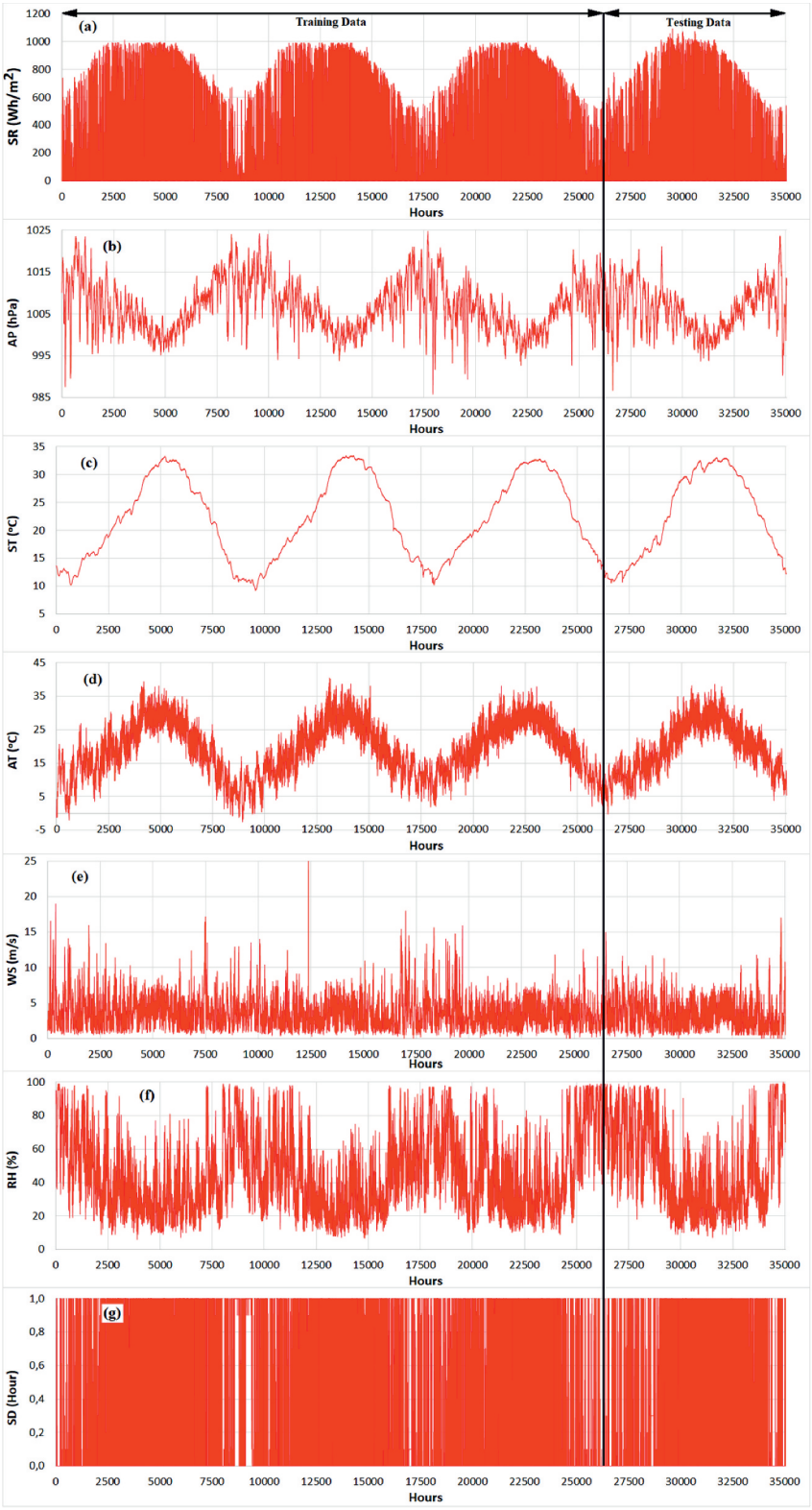

Figure 5. Observed training and testing atmospheric parameters between 2016–2019 years.

of 7.337 degrees Celsius. Figure 5(d) displays the observed hourly atmospheric temperature (AT) values, testing and training AT data for the proposed algorithms for the province of Izmir. The maximum AT value is 40.5 degrees Celsius, and the minimum is −2.6 degrees Celsius. The average temperature of all training and testing data is 19.12 degrees Celsius, with a standard deviation of 8.021 degrees Celsius. Figure 5(e) presents observed hourly wind speed (WS), training, and testing WS data for all approaches. For Canakkale province, the maximum WS value is 24.6 m/s, while the lowest WS value is 0.0 m/s. The average velocity of all training and testing WS data is 3.53 m/s, with a standard deviation of 2.251 m/s. For Sanlıurfa province, Figure 5(f) offers observed hourly relative humidity (RH), training, and testing RH data for the methods. The highest RH value is 100.0%, while the lowest RH value is 6.0%. All training and testing RH data have an average value of 47.26%, with a standard deviation of 24.152% for this data set. The observed hourly sunshine duration (SD), training and testing SD data for all approaches for Mardin province are shown in Figure 5(g). The greatest SD value is 1 hour, while the smallest SD value is 0 hours. The mean value of all testing and training SD data is 0.36 hours, with a standard deviation of 0.453 hours for this data set. According to Figure 5, the largest fluctuation in the data is in SD, while the least fluctuation is in ST.

Figure 6 presents both observed and predicted weather parameters from 1st January 2019 to 31st December 2019 when the LSTM approach was used. Figure 7, on the other hand, shows only the observed and estimated values of February 2019 together to interpret the results more easily. As seen from Figures 6 and 7; the observed and forecasted values in the graphs of AP and ST overlap. This shows that the LSTM analysis for these parameters yields more sensitive results.

The discrepancy between observed and estimated atmospheric parameter values is seen in Figure 8. The error values between real and forecasted SR data varied between − 500 Wh/m^2 and + 500 Wh/m^2 in Karaman province, AP data ranged between − 2.1 hPa and + 3 hPa in Antalya province, ST data ranged between − 1.3°C and + 0.26°C in Adana province, AT data ranged between − 5.7°C and + 4.2°C in Izmir province, WS data ranged between −6.0 m/s and +6.0 m/s in Canakkale province, RH data ranged between − 27.6% and + 65.6% in Sanlıurfa province and SD data ranged between −0.88 hours and +0.93 hours in Mardin province. The figure clearly shows that the least error value was obtained in the WS, RH and SD values in July and August months while error rates in SR and WS increased in the April and May months. However, considering the predictions for the whole of the 2019 year in general, the ST estimation did not change much in the measurement ranges, so it was the smallest value. On the other hand, the SD estimation was found to have the greatest error rate due to cloudiness and night darkness.

The distribution of frequencies of atmospheric parameter variations between estimated and actual values for the LSTM approach is shown in Figure 9 for all provinces. The results for atmospheric parameters in the proposed model seem to be skewed to the positive in all provinces. In LSTM neural network model; more than 89.9% of the model forecasts have an error ≤± 70 Wh/m^2 for SR in Karaman, more than 95.9% of the model forecasts have an error ≤± 0.7 hPa for AP in Antalya, more than 99.9% of the model forecasts have an error ≤± 0.7°C for ST in Adana, more than 94.9% of the model estimations have an error ≤± 1.4°C for AT in Izmir, more than 93.0% of the model forecasts have an error ≤± 1.4 m/s for WS in Canakkale, more than 94.0% of the model forecasts have an error ≤± 7% for RH in Sanlıurfa, more than 92.7% of the model forecasts have an error ≤± 0.35 hour for SD in Mardin as can be seen in this figure.

Figure 10 presents an hourly plot of observed and predicted atmospheric parameters to demonstrate the data distribution, which validates the previous results. This figure also includes the correlation coefficient (R). The observed and estimated data using the LSTM technique nearly agree with ST and AP parameters, with R-values of 0.9999 and 0.9980, respectively. The smallest R-value, 0.9131, was achieved in the SD parameter.

Taylor diagrams (Taylor 2001) were prepared in Figure 11 to evaluate the results obtained from all the methods. As is generally known, a Taylor diagram can be used in several ways to assess the precision of recommended models. The total similarity of the models to the real (measured) data is

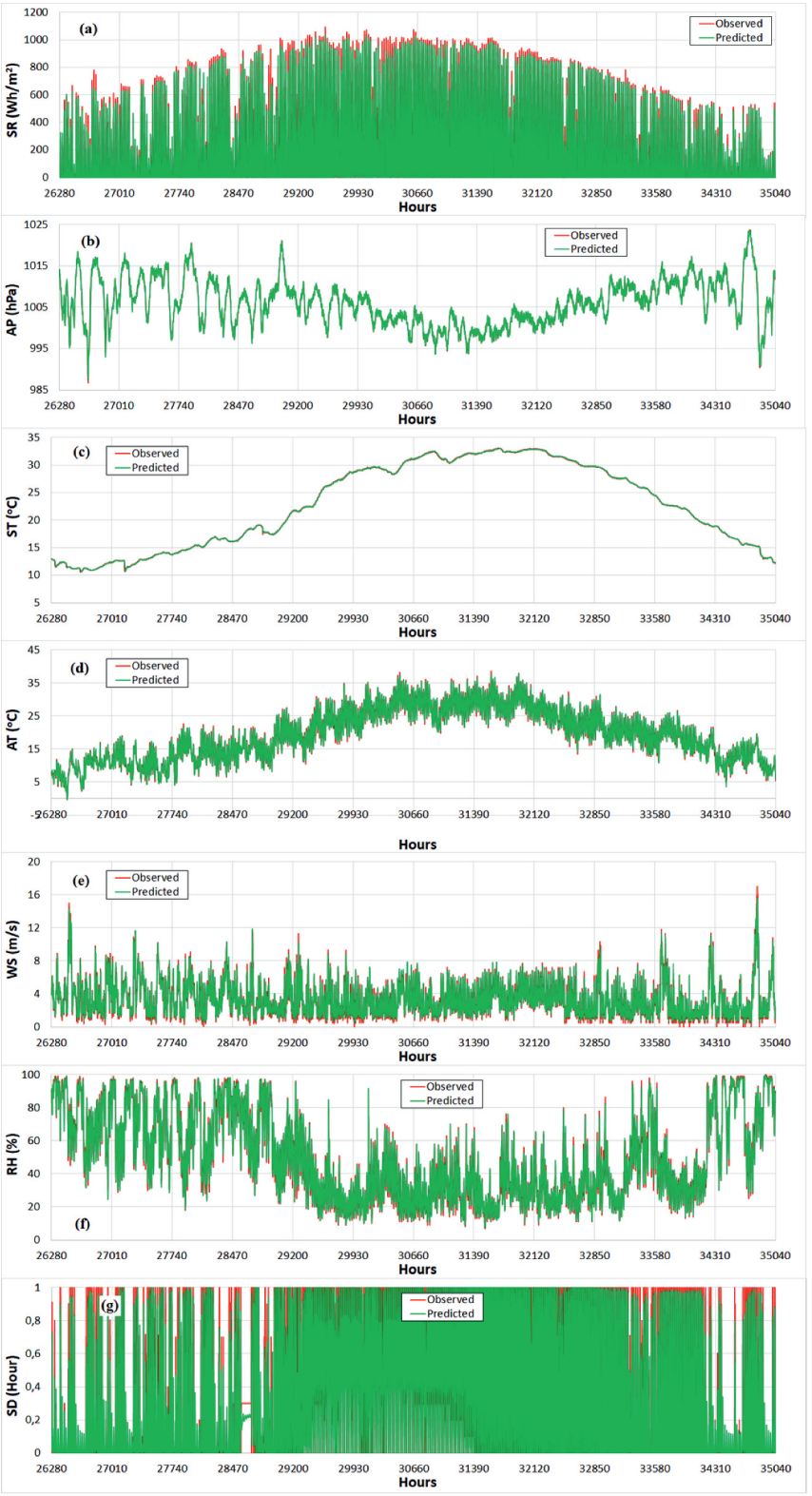

Figure 6. Observed and estimated testing data for the 2019 year and the LSTM method.

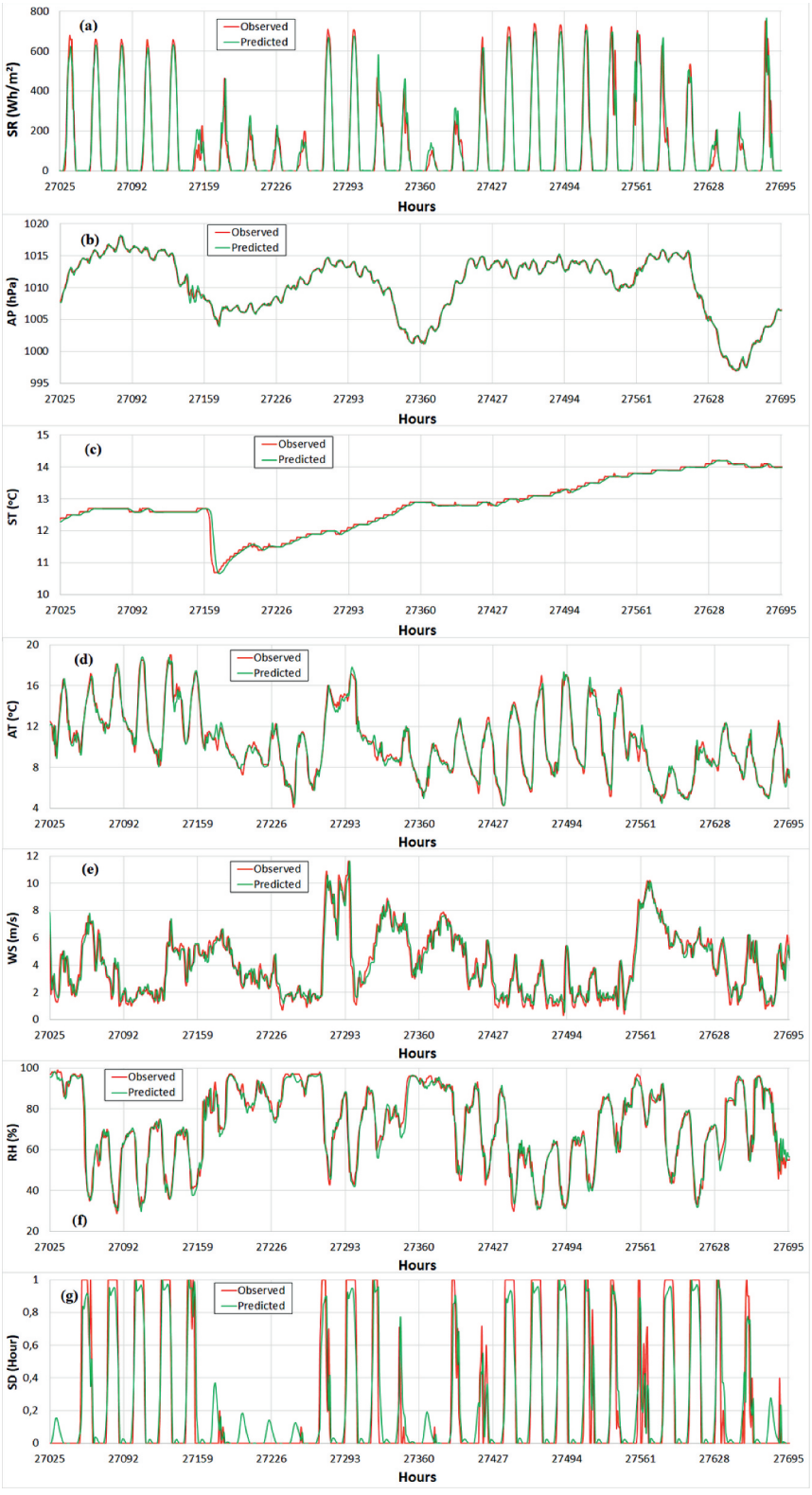

Figure 7. Observed and estimated testing data for February 2019 and LSTM method.

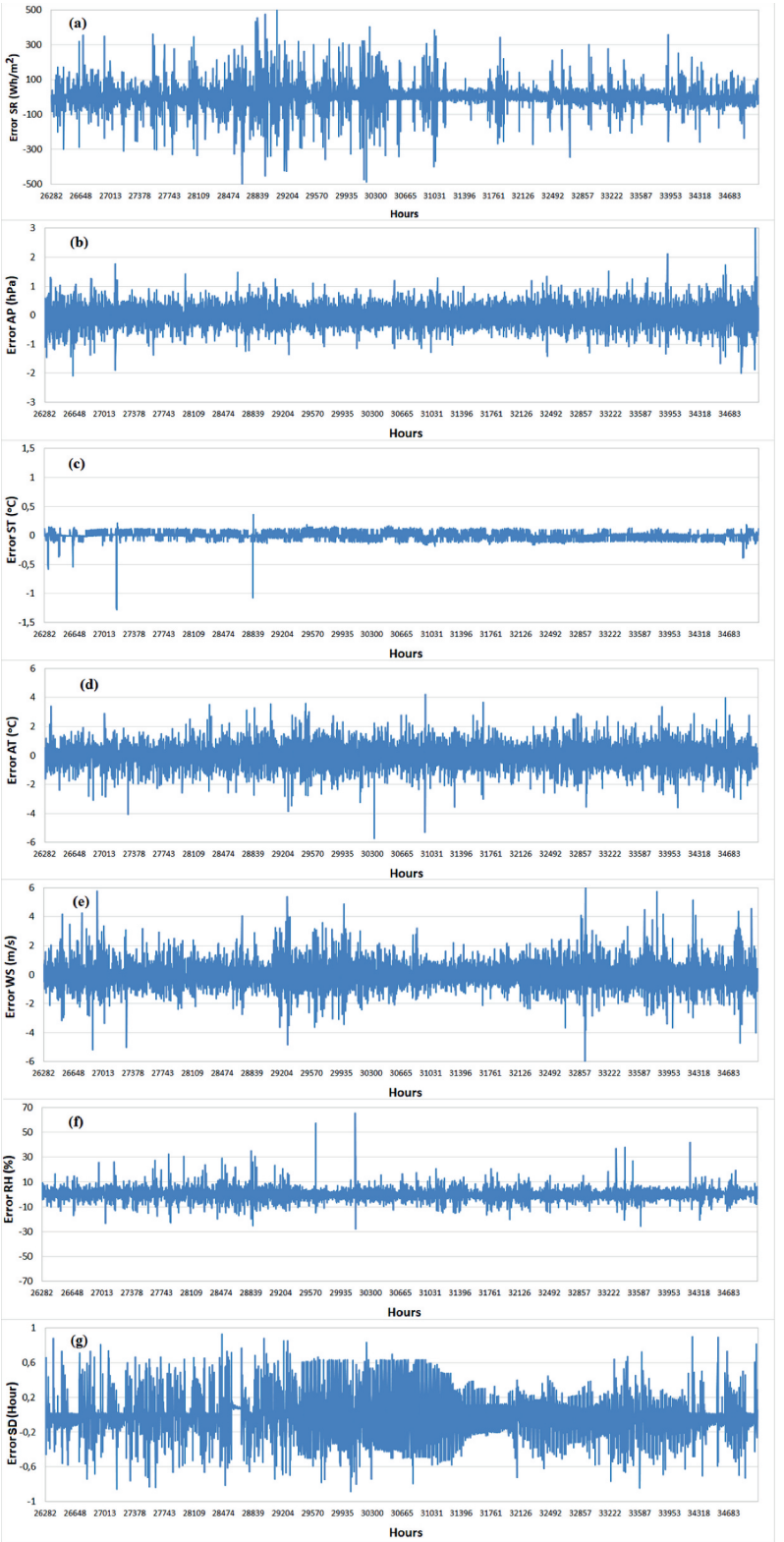

Figure 8. The error value of the observed and forecasted atmospheric parameters for the LSTM model.

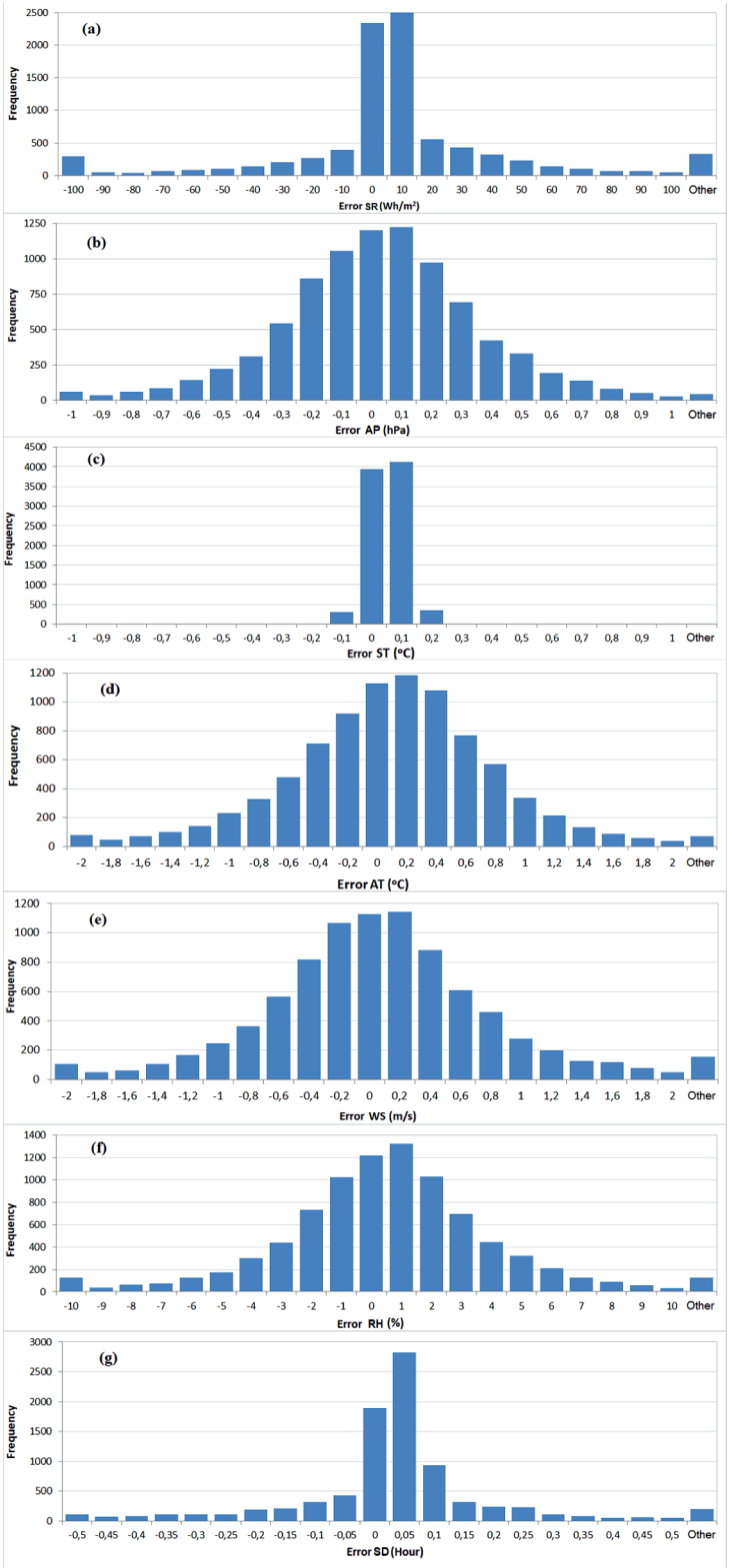

Figure 9. Histogram of estimation errors in all provinces for the LSTM method.

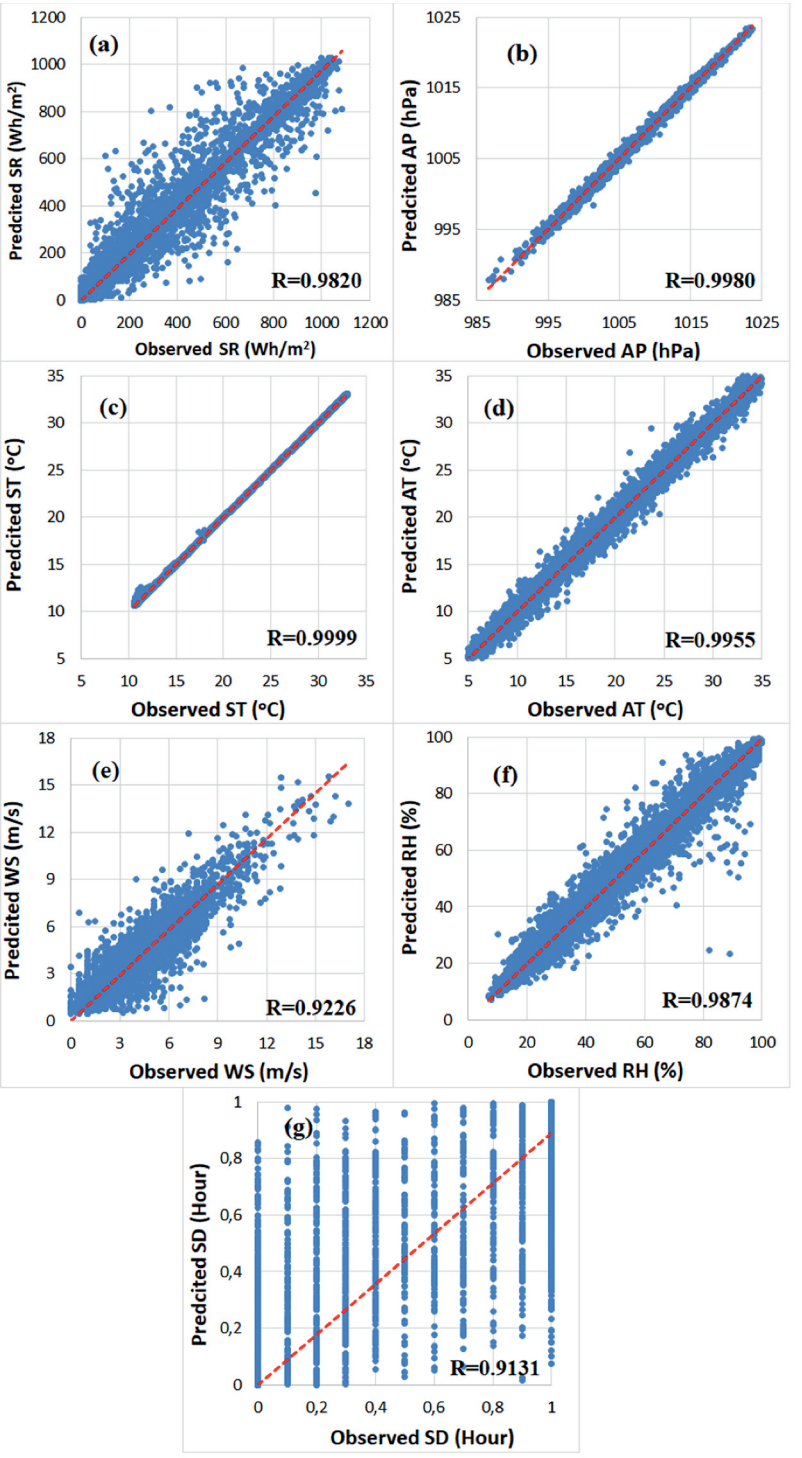

Figure 10. The observed data regression plot and forecasted atmospheric parameters for the LSTM method.

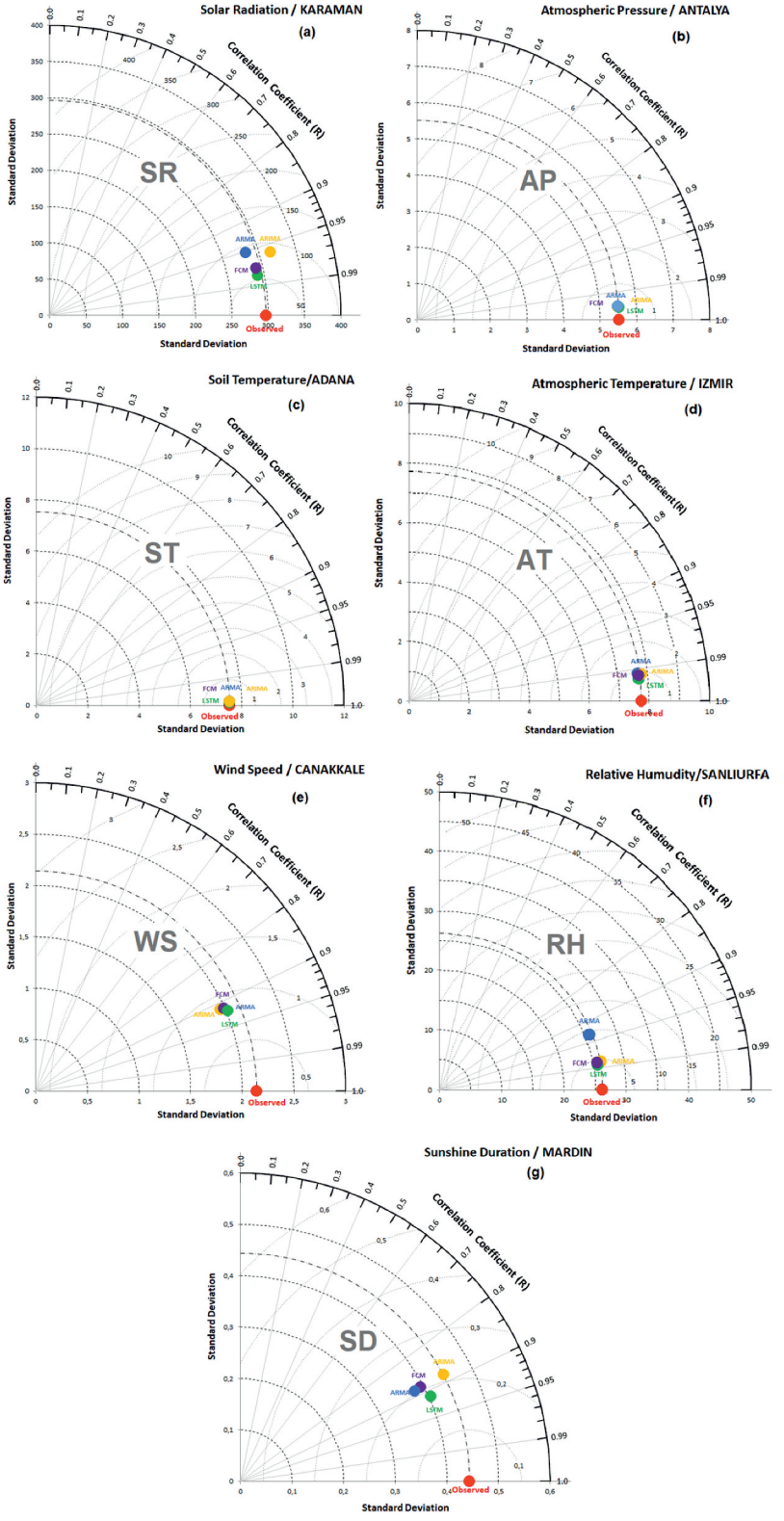

Figure 11. Taylor diagrams of the test data's errors for all suggested models.

Table 2. Results with the highest statistical accuracy of hourly atmospheric parameter forecasts for the models.

Station	Parameter	Method	MAE	RMSE	R	Bias	MAPE	NSE
Karaman	**SR** (Wh/m^2)	ARMA	57.884	91.006	0.9518	0.0210	-	0.9059
		ARIMA	49.017	87.403	0.9610	0.0005	-	0.9132
		LSTM	**25.976**	**56.078**	**0.9820**	**0.3832**	-	**0.9643**
		ANFIS FCM	33.037	66.240	0.9748	0.9939	-	0.9501
Antalya	**AP** (hPa)	ARMA	0.276	0.347	0.9978	−0.0001	0.0274	0.9956
		ARIMA	0.261	0.349	0.9979	−1.8E−05	0.0259	0.9960
		LSTM	**0.256**	**0.345**	**0.9980**	**−0.0160**	**0.0252**	**0.9961**
		ANFIS FCM	0.257	0.346	0.9977	−0.0019	0.0255	0.9952
Adana	**ST** (°C)	ARMA	0.014	0.080	0.9998	2.0E−05	0.0813	0.9999
		ARIMA	0.014	0.080	0.9998	2.2E−05	0.0815	0.9999
		LSTM	**0.016**	**0.078**	**0.9999**	**−1.8E−04**	**0.0805**	**0.9999**
		ANFIS FCM	0.039	0.079	0.9998	−1.7E−05	0.1957	0.9999
Izmir	**AT** (°C)	ARMA	0.695	0.942	0.9925	9.9E−06	-	0.9852
		ARIMA	0.642	0.836	0.9932	−1.3E−04	-	0.9862
		LSTM	**0.555**	**0.753**	**0.9955**	**0.0051**	-	**0.9909**
		ANFIS FCM	0.645	0.786	0.9935	0.0045	-	0.9871
Canakkale	**WS** (m/s)	ARMA	0.616	0.866	0.9143	−0.0002	-	0.8360
		ARIMA	0.615	0.866	0.9144	−0.0018	-	0.8361
		LSTM	**0.584**	**0.827**	**0.9226**	**−0.0238**	-	**0.8507**
		ANFIS FCM	0.614	0.865	0.9148	−0.0378	-	0.8363
Sanlıurfa	**RH** (%)	ARMA	3.116	4.690	0.9838	−0.0011	7.8203	0.9679
		ARIMA	3.013	4.670	0.9842	4.7E−04	7.5082	0.9682
		LSTM	**2.730**	**4.146**	**0.9874**	**0.0696**	**6.6903**	**0.9749**
		ANFIS FCM	3.019	4.562	0.9847	0.1173	7.5466	0.9696
Mardin	**SD** (hour)	ARMA	0.120	0.188	0.8860	0.0010	-	0.7849
		ARIMA	0.128	0.215	0.8851	9.1E−04	-	0.7657
		LSTM	**0.107**	**0.181**	**0.9131**	**0.0094**	-	**0.8333**
		ANFIS FCM	0.122	0.205	0.8879	0.0156	-	0.7863

inversely correlated with the distance between the points denoting the models and the real value. Figure 10 clearly illustrates that the ST and AP forecasts are closer to the observations than the other atmospheric parameters. The LSTM approach yielded the lowest RMSE values for ST and AP in Adana and Antalya provinces at 0.078°C and 0.345 hPa, respectively. In addition, ARMA, ARIMA and ANFIS FCM methods for ST and AP gave sensitive results in these provinces. In Mardin, the R-value value was found to be 0.9131 when the LSTM method was used, while the worst result was obtained when the ARIMA method was used in this province, with 0.8851.

Table 2 presents the different evaluation criteria for the approximations. The LSTM model with 100 hidden layers and 500 epoch numbers presented the best AP, AT, WS, RH, and SD prediction results. But, the LSTM model with 300 hidden layers and 500 epoch numbers provided better MAE, RMSE, R, bias and NSE values for SR prediction as 25.976 Wh/m^2, 56.078 Wh/m^2, 0.9820, 0.3832 Wh/m^2, and 0.9643, respectively. Moreover, MAE, RMSE, R, bias, MAPE and NSE values were obtained as 0.016°C, 0.078°C, 0.9999, −0.00018°C, 0.0805%, and 0.9999 respectively when 300 hidden layers and 700 epoch numbers were used for the prediction of ST. As can be seen in Table 2, the reason why the MAPE value cannot be calculated in the SR, AT, WS and SD parameters is that some values are measured as zero. When it comes to the NSE outcomes, NSE = 1 denotes a model that is perfectly fitted with an estimated error variance of zero. NSE = 0, on the other hand, indicates that the model does not make sense for the pertinent data set. As seen from the table, the highest NSE value was found to be 0.9999 for all methods in ST estimation. In addition, LSTM results gave the highest NSE values in the estimation of parameters other than ST.

As seen from Table 2, the best results in the estimation of all atmospheric parameters were obtained when the LSTM method was used. To determine the effectiveness of the suggested prediction model, a comparison was also done with the outcomes from several models in the literature, as shown in Table 3.

Table 3. Comparison of error values between literature and the LSTM model proposed.

				Evaluation Criteria					
Parameter	Model	Location	Reference	MAE	RMSE	R	Bias	MAPE	NSE
SR	Persistence	China	Qing and Niu (2018)	-	177.03	-	-	-	-
	LR			-	195.86	-	-	-	-
	BPNN			-	150.28	-	-	-	-
	LSTM			-	122.72	-	-	-	-
	Proposed Model (LSTM)			***25.976***	***56.078***	***0.982***	***0.3832***	-	***0.9643***
AP	LSTM	Türkiye	Bilgili, Ilhan, and Ünal (2022)	0.029	0.379	0.998	-	0.029	-
	ANFIS FCM			0.237	0.325	0.998		0.024	-
	ANFIS SC			0.237	0.325	0.998		0.023	-
	ANFIS GP			0.232	0.320	0.998		0.023	-
	Proposed Model (LSTM)			***0.256***	***0.345***	***0.998***	***−0.016***	***0.025***	***0.9961***
ST	ARIMA	Iran	Zeynoddin et al. (2019)	0.522	0.709	0.998	-	-	-
	MLPNN			0.510	0.700	0.998	-	-	-
	Proposed Model (LSTM)			***0.016***	***0.078***	***0.999***	***−1.8E−4***	***0.0805***	***0.9999***
AT	LSTM	Canada	Li et al. (2020)	-	6.54	0.906	-	-	-
	SVM			-	6.748	0.900	-	-	-
	EM				6.966	0.892	-	-	-
	MLR			-	6.954	0.893	-	-	-
	Proposed Model (LSTM)			***0.555***	***0.753***	***0.996***	***0.0051***	-	***0.9909***
WS	LSTM	China	Zhang et al. (2019)	-	0.558	0.992		-	-
	SWLSTM GPR			-	0.579	0.991	-	-	-
	Proposed Model (LSTM)			***0.584***	***0.827***	***0.923***	***−0.0238***	-	***0.8507***
RH	ANFIS FCM	Türkiye	Ozbek, Ş, and Bilgili (2022)	5.95	7.67	0.887	-	-	-
	LSTM			5.76	7.51	0.892	-	-	-
	Proposed Model (LSTM)			***2.730***	***4.146***	***0.987***	***0.0696***	***6.6903***	***0.9749***
SD	ANN	Iran	Rahimikhoob (2014)	-	6.73	0.883	8E−04	-	-
	Proposed Model (LSTM)			***0.107***	***0.181***	***0.913***	***0.0094***	-	***0.8333***

Conclusions

Adaptive Neuro-Fuzzy Inference System (ANFIS) with fuzzy c-means (FCM), Autoregressive Moving Average (ARMA) model, Autoregressive Integrated Moving Average (ARIMA) model, and Long Short-Term Memory (LSTM) neural network approaches were used in this study to estimate seven different weather parameters mentioned above that were measured hourly from seven different provinces of Türkiye by the State Meteorology Department. For a total of 4 years, 35040 hours of 7 different meteorological data were divided into two parts to be used in the LSTM approach: 75% training data and 25% testing data. The results revealed that utilizing the related methodology, 75% of the training data was sufficient for precise hourly prediction. The hourly soil temperature (ST) prediction for the province of Adana yielded the best results when the LSTM method was used, with MAE, RMSE, R, Bias, MAPE and NSE values computed as 0.016°C, 0.078°C, 0.9999, −0.00018°C, 0.0805% and 0.9999 respectively. One of the study's key conclusions is that atmospheric parameters can be modeled without using independent variables. The models proposed can be conveniently used to predict meteorological data for any point without the need for complex calculations. It has been shown that the best result in the estimation of all parameters is obtained with the LSTM method. It can be seen from this study that all atmospheric parameters can be estimated with great precision using only the LSTM method. This also allows us easily analyze the consequences of weather parameters on the environmental ecosystem. Future research can be executed that, using methods such as PCC, CEDMAN, CNN, and BiLSTM, or hybrid deep learning architectures and algorithms that use them together, to increase the precision and accuracy of the estimation results.

Acknowledgment

The State Meteorological Department of Türkiye provided data for which the author is exceedingly appreciative.

Disclosure statement

No potential conflict of interest was reported by the author.

Notes on contributor

Arif Ozbek is an associate professor in Mechanical Engineering Department, Cukurova University, Türkiye. He has been a member of the Chamber of Mechanical Engineers of Turkey since 2000. His research interests are thermodynamics, renewable energy, heat transfer and air-conditioning systems.

ORCID

Arif Ozbek http://orcid.org/0000-0003-1287-9078

References

Al-Shawwa, M., A. A. Al-Absi, S. A. Hassanein, K. A. Baraka, and S. S. Abu-Naser. 2018. Predicting temperature and humidity in the surrounding environment using artificial neural network. *International Journal of Academic Pedagogical Research* 2 (9):1–6.

Badescu, V. 1999. Correlations to estimate monthly mean daily solar global irradiation: application to romania. *Energy* 24 (10):883–93. doi:10.1016/s0360-5442(99)00027-4.

Bemani, A., A. Baghban, S. Shamshirband, A. Mosavi, P. Csiba, and A. R. Varkonyi-Koczy. 2020. Applying ANN, ANFIS, and LSSVM models for estimation of acid solvent solubility in supercritical CO2. *Computers, Materials & Continua*. 63(3):1175–204. doi:10.32604/cmc.2020.07723.

BGE, P., G. M. Jenkins, and G. C. Reinsel. 1994. *Time Series Analysis Forecasting and Control*. Prentice-Hall, Upper Saddle River. N. J.

Bilgili, M., A. Ilhan, and Ş. Ünal. 2022. Time-series prediction of hourly atmospheric pressure using ANFIS and LSTM approaches. *Neural Computing and Applications* 34:15633–48. doi:10.1007/s00521-022-07275-5.

Brown, I. 2012. Influence of seasonal weather and climate variability on crop yields in Scotland. *International Journal of Biometeorology* 57:605–14. doi:10.1007/s00484-012-0588-9.

Chevalier, R. F., G. Hoogenboom, R. W. McClendon, and J. A. Paz. 2010. Support vector regression with reduced training sets for air temperature prediction: A comparison with Artificial Neural Networks. *Neural Computing and Applications* 20:151–59. doi:10.1007/s00521-010-0363-y.

Delbari, M., S. Sharifazari, and E. Mohammadi. 2018. Modeling daily soil temperature over diverse climate conditions in Iran—a comparison of multiple linear regression and support vector regression techniques. *Theoretical and Applied Climatology* 135:991–1001. doi:10.1007/s00704-018-2370-3.

El-Kenawy, E. M., A. Ibrahim, N. Bailek, K. Bouchouicha, M. A. Hassan, M. Jamei, and N. Al-Ansari. 2021. Sunshine duration measurements and predictions in Saharan Algeria region: An improved ensemble learning approach. *Theoretical and Applied Climatology* 147 (3–4):1015–31. doi:10.1007/s00704-021-03843-2.

El-Metwally, M. 2005. Sunshine and global solar radiation estimation at different sites in Egypt. *Journal of Atmospheric and Solar-Terrestrial Physics* 67 (14):1331–42. doi:10.1016/j.jastp.2005.04.004.

Erdil, A., and E. Arcaklioglu. 2013. The prediction of meteorological variables using artificial neural network. *Neural Computing and Applications* 22 (7–8):1677–83. doi:10.1007/s00521-012-1210-0.

FCM. MATLAB & Simulink. Accessed Oct 5 2022. https://www.mathworks.com/help/fuzzy/fuzzy-c-means-clustering.html

Feng, Y., N. Cui, W. Hao, L. Gao, and D. Gong. 2019. Estimation of soil temperature from meteorological data using different machine learning models. *Geoderma* 338:67–77. doi:10.1016/j.geoderma.2018.11.044.

Guo, J., Z. Xie, Y. Qin, L. Jia, and Y. Wang. 2019. Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM. *IEEE Access* 7:42946–55. doi:10.1109/access.2019.2907739.

Hochreiter, S., and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9 (8):1735–80. doi:10.1162/neco.1997.9.8.1735.

Hutapea, M. I., Y. Y. Pratiwi, I. M. Sarkis, I. K. Jaya, and M. Sinambela. 2020. Prediction of relative humidity based on long short-term memory network. AIP Conf Proc 2221(March). doi:10.1063/5.0003171.

Jenkins, G. M., and P. BGE. 1976. *Time series analysis forecasting and control*. Rev. ed. San Francisco: Holden-Day.

Kaba, K., H. M. Kandirmaz, and M. Avci. 2016. Estimation of daily sunshine duration using support vector machines. *International Journal of Green Energy* 14:430–41. doi:10.1080/15435075.2016.1265971.

Kandirmaz, H. M., K. Kaba, and M. Avci. 2014. Estimation of monthly Sunshine duration in Turkey using artificial neural networks. *International Journal of Photoenergy* 1–9. doi:10.1155/2014/680596.

Karakuş, O., E. E. Kuruoğlu, and M. A. Altınkaya. 2017. One-Day ahead wind speed/power prediction based on polynomial autoregressive model. *IET Renewable Power Generation* 11 (11):1430–39. doi:10.1049/iet-rpg.2016.0972.
Khosravi, A., L. Machado, and R. O. Nunes. 2018. Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil. *Applied Energy* 224:550–66. doi:10.1016/j.apenergy.2018.05.043.
Li, X., Z. Li, W. Huang, and P. Zhou. 2020. Performance of statistical and machine learning ensembles for daily temperature downscaling. *Theoretical and Applied Climatology* 140:571–88. doi:10.1007/s00704-020-03098-3.
Liu, Z., and A. Baghban. 2017. Application of LSSVM for biodiesel production using supercritical ethanol solvent. *Energy Sources, Part A: Recovery, Utilization, & Environmental Effects* 39 (17):1869–74. doi:10.1080/15567036.2017.1380732.
Li, Q., Y. Zhu, W. Shangguan, X. Wang, L. Li, and F. Yu. 2022. An attention-aware LSTM model for soil moisture and soil temperature prediction. *Geoderma* 409:115651. doi:10.1016/j.geoderma.2021.115651.
Mateus, P., J. Catalão, V. B. Mendes, and G. Nico. 2020. An ERA5-based hourly global pressure and temperature (HGPT) model. *Remote Sensing* 12:1098. doi:10.3390/rs12071098.
Matuszko, D., and S. Węglarczyk. 2014. Relationship between sunshine duration and air temperature and contemporary global warming. *International Journal of Climatology* 35:3640–53. doi:10.1002/joc.4238.
Ma, T., C. Wang, J. Wang, J. Cheng, and X. Chen. 2019. Particle-swarm optimization of ensemble neural networks with negative correlation learning for forecasting short-term wind speed of wind farms in western China. *Information Sciences* 505:157–82. doi:10.1016/j.ins.2019.07.074.
Mukhtar, M., A. Oluwasanmi, N. Yimen, Z. Qinxiu, C. C. Ukwuoma, B. Ezurike, and O. Bamisile. 2022. Development and comparison of two novel hybrid neural network models for hourly solar radiation prediction. *Applied Sciences*. 12 (3):1435. doi:10.3390/app12031435.
Nash, J. E., and J. V. Sutcliffe. 1970. River flow forecasting through conceptual models part I — a discussion of principles. *Journal of Hydrology* 10 (3):282–90. doi:10.1016/0022-1694(70)90255-6.
Ozbek, A., A. Ilhan, M. Bilgili, and B. Sahin. 2022. One-hour ahead wind speed forecasting using deep learning approach. *Stochastic Environmental Research and Risk Assessment* 36:4311–35. doi:10.1007/s00477-022-02265-4.
Ozbek, A., Ü. Ş, and M. Bilgili. 2022. Daily average relative humidity forecasting with LSTM neural network and ANFIS approaches. *Theoretical and Applied Climatology* 150:697–714. doi:10.1007/s00704-022-04181-7.
Ozbek, A., A. Sekertekin, M. Bilgili, and N. Arslan. 2021. Prediction of 10-min, hourly, and daily atmospheric air temperature: Comparison of LSTM, ANFIS-FCM, and ARMA. *Arabian Journal of Geosciences* 14: doi:10.1007/s12517-021-06982-y.
Pei, S., H. Qin, Z. Zhang, L. Yao, Y. Wang, C. Wang, Y. Liu, Z. Jiang, J. Zhou, T. Yi, et al. 2019. Wind speed prediction method based on empirical wavelet transform and new cell update long short-term memory network. *Energy Conversion and Management* 196:779–92. doi:10.1016/j.enconman.2019.06.041.
Piotrowski, P., D. Baczyński, M. Kopyt, K. Szafranek, P. Helt, and T. Gulczyński. 2019. Analysis of forecasted meteorological data (NWP) for efficient spatial forecasting of wind power generation. *Electric Power Systems Research* 175:105891. doi:10.1016/j.epsr.2019.105891.
Qadeer, K., A. Ahmad, M. A. Qyyum, A. S. Nizami, and M. Lee. 2021. Developing machine learning models for relative humidity prediction in air-based energy systems and environmental management applications. *Journal of Environmental Management* 292:112736. doi:10.1016/j.jenvman.2021.112736.
Qing, X., and Y. Niu. 2018. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* 148:461–68. doi:10.1016/j.energy.2018.01.177.
Rahimikhoob, A. 2014. Estimating sunshine duration from other climatic data by Artificial Neural Network for ET0 estimation in an arid environment. *Theoretical and Applied Climatology* 118 (1–2):1–8. doi:10.1007/s00704-013-1047-1.
Rjoub, G., J. Bentahar, O. Abdel Wahab, and A. Bataineh. (2019) Deep smart scheduling: A deep learning approach for automated big data scheduling over the cloud 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud). doi: 10.1109/ficloud.2019.00034
Sanikhani, H., R. C. Deo, Z. M. Yaseen, O. Eray, and O. Kisi. 2018. Non-tuned data intelligent model for soil temperature estimation: A new approach. *Geoderma* 330:52–64. doi:10.1016/j.geoderma.2018.05.030.
Sarker, I. H., A. S. Kayes, and P. Watters. 2019. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data* 6 (1). doi:10.1186/s40537-019-0219-y.
Sekertekin, A., M. Bilgili, N. Arslan, A. Yıldırım, K. Celebi, and A. Ozbek. 2021. Short-term air temperature prediction by adaptive neuro-fuzzy inference system (ANFIS) and long short-term memory (LSTM) network. *Meteorology and Atmospheric Physics*. 133(3):943–59. doi:10.1007/s00703-021-00791-4.
Solano, E. S., P. Dehghanian, and C. M. Affonso. 2022. Solar radiation forecasting using machine learning and ensemble feature selection. *Energies* 15 (19):7049. doi:10.3390/en15197049.
Taylor, K. E. 2001. Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres* 106 (D7):7183–92. doi:10.1029/2000jd900719.
Uçkan, İ., and K. M. Khudhur. 2022. Improving of global solar radiation forecast by comparing other meteorological parameter models with sunshine duration models. *Environmental Science and Pollution Research* 29 (25):37867–81. doi:10.1007/s11356-022-18781-3.
Wang, S., X. Yu, L. Liu, J. Huang, T. Ho Wong, and C. Jiang. 2020. An approach for radar quantitative precipitation estimation based on spatiotemporal network. *Computers, Materials & Continua* 65:459–79. doi:10.32604/cmc.2020.010627.

Xia, J., X. Ma, W. Wu, B. Huang, and W. Li. 2020. Application of a new information priority accumulated grey model with time power to predict short-term wind turbine capacity. *Journal of Cleaner Production* 244:118573. doi:10.1016/j.jclepro.2019.118573.

Yalçın, S. 2022. Weather parameters forecasting with time series using deep hybrid neural networks. *Concurrency & Computation: Practice & Experience* 34 (21). doi: 10.1002/cpe.7141.

Yao, R., L. Wang, X. Huang, L. Li, J. Sun, X. Wu, and W. Jiang. 2020. Developing a temporally accurate air temperature dataset for Mainland China. *Science of the Total Environment* 706:136037. doi:10.1016/j.scitotenv.2019.136037.

Zeynoddin, M., H. Bonakdari, I. Ebtehaj, F. Esmaeilbeiki, B. Gharabaghi, and D. Z. Haghi. 2019. A reliable linear stochastic daily soil temperature forecast model. *Soil and Tillage Research* 189:73–87. doi:10.1016/j.still.2018.12.023.

Zhang, Z., L. Ye, H. Qin, Y. Liu, C. Wang, X. Yu, X. Yin, and J. Li. 2019. Wind speed prediction method using shared weight long short-term memory network and gaussian process regression. *Applied Energy* 247:270–84. doi:10.1016/j.apenergy.2019.04.047.

Zhang, J., Y. Zhu, X. Zhang, M. Ye, and J. Yang. 2018. Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology* 561:918–29. doi:10.1016/j.jhydrol.2018.04.065.

Appendix

Some parts of the software/code used for implementing the LSTM method:

```
numTimeStepsTrain = floor(0.75*numel(data));
dataTrain = data(1:numTimeStepsTrain +1);
dataTest = data(numTimeStepsTrain +1:end);
mu = mean(dataTrain); sig = std(dataTrain);
dataTrainStandardized = (dataTrain - mu)/sig;
XTrain = dataTrainStandardized(1:end-1);
YTrain = dataTrainStandardized(2:end);
numFeatures = 1;
numResponses = 1;
numHiddenUnits = 100;
    layers = [...
        sequenceInputLayer(numFeatures)
        lstmLayer(numHiddenUnits)
        fullyConnectedLayer(numResponses)
        regressionLayer];
    options = trainingOptions('adam', ...
        'MaxEpochs',500, ...
        'GradientThreshold',1, ...
        'InitialLearnRate',0.005, ...
        'LearnRateSchedule','piecewise', ...
        'LearnRateDropPeriod',125, ...
        'LearnRateDropFactor',0.2, ...
        'Verbose',0, ...
        'Plots','training-progress');
net = trainNetwork(XTrain,YTrain,layers,options);
dataTestStandardized = (dataTest - mu)/sig;
XTest = dataTestStandardized(1:end-1);
net = predictAndUpdateState(net,XTrain);
[net,YPred] = predictAndUpdateState(net,YTrain(end));
numTimeStepsTest = numel(XTest);
for i = 2:numTimeStepsTest
        [net,YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-1),'ExecutionEnvironment','cpu');
end
YPred = sig*YPred + mu;
YTest = dataTest(2:end);
RMSE = sqrt(mean((YPred-YTest).^2))
```