Climate Change Prediction on Middle East Using RNN Model and Hybrid LSTM&GRU Model

Ammar Younus Mahmood
Department of computer science, college of science, University of Diyala
Diyala, Iraq
scicompms222309@uodiyala.edu.iq

Khalid Mohammed Saffer
Department of computer science, college of science, University of Diyala
Diyala, Iraq
dr.khaledmoh@uodiyala.edu.iq

Bashar Talib AL-Nuaimi
Department of computer science, college of science, University of Diyala
Diyala, Iraq
alnuaimi_bashar@uodiyala.edu.iq

Dhar Intisar Bakr
Department of computer science, college of science, University of Diyala
Diyala, Iraq
dher@uodiyala.edu.iq

***Abstract*— The issue of climate change is regarded as one of the most significant global concerns of our day due to its unparalleled impact on Earth's natural system. The main effects of climate change are increasing frequency of extreme weather events, fast windfalls and storms, altered rainfall patterns, and dramatic temperature variations. One of the areas most susceptible to climate change is the Middle East. In addition to other detrimental effects of climate change, it is subject to regular heat waves, a shortage of food and water, and other challenges. This area is warming at a rate that is almost twice as fast as the rest of the world, which puts it at danger of significant impacts and increases the risks to food and water security. Similarly, the Middle East's agricultural output is heavily dependent on rainfall since the region is extremely vulnerable to variations in rainfall and temperature brought on by climate change, which has a detrimental impact on the region's economy. Weather forecasts are crucial for providing early notice of how the weather will affect several facets of human life. The objective of meteorologists has always been to forecast meteorological conditions accurately and promptly. Research and applications based on deep learning in a range of domains, including computer vision, speech recognition, translation, and time series prediction, have made it clear how vital deep learning is and how to invest in and use it in many fields. Additionally, it performed incredibly well when it came to removing spatial and temporal aspects from the data. As a result, deep learning-based weather prediction (DLWP) will function well in addition to conventional techniques. Two deep learning models were applied in this work to forecast Middle Eastern climate change. Recurrent neural networks (RNNs) make up the first model, while a mix of gated recurrent units (GRU) and long- and short-term memory (LSTM) makes up the second. There were six climate parameters taken into consideration: humidity, wind speed, sun radiation, maximum and minimum temperatures, and rainfall rate. Nine Middle Eastern capitals were the subject of the investigation. Using statistics from 1981 to 2022, they are: Riyadh (Saudi Arabia), Ankara (Turkey), Baghdad (Iraq), Beirut (Lebanon), Cairo (Egypt), Doha (Qatar), Abu Dhabi (UAE), Sanaa (Yemen), Tehran (Iran). NASA's Modern Era-Retrospective analysis for Research and Applications (MERRA-2) provided the data that was used. To assess how well these deep learning algorithms performed, they were compared using statistical criteria such as mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R2).**

Keywords*— *LSTM, GRU, RNN, Climate Change, Deep learning, Weather forecasts

I. INTRODUCTION

Climate change in the Middle East has huge economic and social costs. Over the past three decades, changing patterns of temperature and rainfall have eroded average per capita income and dramatically changed the sectoral composition of output and jobs. While we see this pattern emerging all over the world, it is particularly true in the Middle East and Central Asia.

Fig. 1. shows the study area

Since the earliest times of humanity, people have always been trying to understand this world and how to predict what will happen the next day. Humans have evolved their forecasting techniques over time in a variety of domains, including weather. Temperature, humidity, wind speed, and other atmospheric factors have a significant influence on numerous facets of human existence [1]. Early warning of weather disasters depends heavily on fast and accurate weather forecasts. As an illustration, consider nowcasting, particularly quantitative precipitation nowcasting (QPN), which is the forecasting of short time spans (a few hours) with high spatial and temporal precision (60–600 s, 100–1000 m)[2]. However, there are still many challenges in numerical weather prediction (NWP) [3].

Therefore, studies have recently turned to machine learning for weather forecasting, especially deep learning. Deep learning (DL) is a method based on teaching a machine so that it has the ability to make decisions similar to those of the human brain. Artificial neural networks, or ANNs, are being created for use in a variety of domains, including image identification, language processing, and future prediction [4].

In recent years, DL research has focused on time series problems [5], where there is a clear yet elusive link between the characteristics. In particular, when a system's behavior is primarily influenced by its spatial or temporal context (like a weather system), standard machine learning (ML) approaches might not be the best. Instead, machine learning approaches that can automatically extract spatial and temporal features are better obtained. More understanding of systems [6]. Predictive accuracy needs further development and improvement, so features must be represented correctly and the correlation of elements must be analyzed clearly. As a result, deep learning is a useful technique for examining time series properties. As a result, DL has been widely used by academics to tackle the multidimensional time series modeling problem of weather forecasting. It is anticipated that data-driven methods will help solve some of the conventional challenges in weather forecasting.

Deep learning has achieved high results with great accuracy, so it is a promising future for machine learning due to its power, flexibility, and smooth extraction of features from time series [7].

A collection of consecutive points listed in chronological order is what we might refer to as a time series. Time series forecasting that works well improves the analysis and decision-making of already-available data. Clinical medicine, financial forecasting, traffic flow forecasting, human behavior forecasting, and other sectors are just a few of its many uses [8].

As we all know, recurrent neural networks (RNNs) are frequently regarded as the most successful method for time series forecasting [9]. RNN-based models are used in [10] and [11] for deterministic wind speed prediction and in [7] for forecasting a variety of time intervals. Examples of these models are Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Gates regulate information during the course of the time series in memory-based RNNs. A feature is saved by the network as a high-weight feature if it is significant; otherwise, it is reset or forgotten [12]. Spatial feature understanding is a limitation of RNN-based architectures, such as LSTM and GRU [13]. CNN is a popular deep structure technique for time series analysis. CNN has shown its inability to capture temporal features particularly effectively in long time series, despite its effective performance in capturing spatial features [14]. Respectively In fact, Artificial Neural Networks (RNNs) The nodes are connected in a loop, and the network can exhibit dynamic timing behavior. However, difficulties commonly occur with RNNs, such as the disappearance of the gradient during training the network, especially with growing length of the time series using typical activation functions such as tanh or sigmoid functions, which reduces the performance accuracy of RNNs. Based on a basic RNN, the Long-Short-Term Memory (LSTM) [15] module addresses memory and forgetting issues by using multi-threshold gates. Thus, to some extent, the restricted capacity to handle long-range dependencies is addressed by LSTM and gated cycle unit (GRU) [16].

To overcome the problems of time series prediction, LSTM was used, imitating cognitive neuroscience. Within the framework of decoding, some researchers have used attention mechanisms [17].

II. Related work

There are many studies conducted on weather forecasting in which machine learning and deep learning techniques were used, among which we mention, Ghamariadyan, Meysam Imteaz, Monzur A[18] in this study, wavelet artificial neural networks (WANN) were used in Queensland, Australia to predict maximum and minimum temperatures as well as to predict precipitation and the Pacific Interdecadal Oscillation (IPO) for ten climate stations, using various sets of variables each from eight groups. On the other hand, QING, TAO FANG, LIU YONG LI, AND DENIS SIDOROV[19]. H. Astsatryan, H. Grigoryan, A. Poghosyan et al [20]. There are also many studies conducted on predicting climate change, including but not limited to them, D. Rolnick, P. Donti, L. Kaack et al [21]. Z.Al Sadeque, F. Bui [22]. Y. Yu, J. Cao, J. Zhu [23]. S. Poornima, M. Pushpalatha [24].

III. Methodology

A. *Recurrent neural networks (RNNs) neural networks with recurrence.*

Neural networks specifically created to handle sequential data. Recurrent neural networks (RNNs) are used in applications such as time series, speech recognition, natural language processing, and weather forecasting. Since LSTM networks are a form of RNN and RNN is a simpler system than LSTM networks, the insight learned from RNN analysis also applies to LSTM networks, so we will start by learning about RNN as the foundation. Furthermore, the recurrent neural network's (RNN) fundamental equations provide as a foundational model that offers a logical route toward ultimately reaching the LSTM system structure [25]. The basic idea is to connect hidden units in neural networks to themselves repeatedly with a time delay [26]. This is because the hidden units can learn some of the features and their representations of the primary inputs, and as a result they return them to themselves. We can interpret this as giving dynamic memory to the network. One of the most important things here is that the same network is used in all time steps in the recurrent neural network. The concept of weight sharing is applicable here, just as it is in CNNs. The same filter might be used in different portions of the input process. More crucially, it generalizes to sequence lengths that were not observed during training, making it possible to train RNNs using sequences of varying lengths. The general layout of the (unregistered) RNN is depicted in Fig. (2).

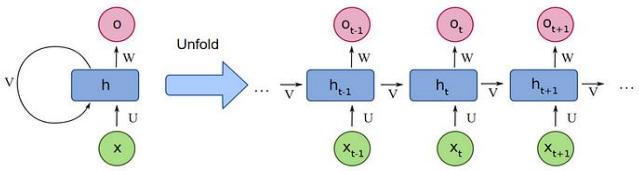

Fig. 2. Structure of the RNN

The corresponding forward propagation formulas for recurrent neural network models can be described in equations (1), (2):

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h \quad (1)$$

$$y_t = g(W_{hy}h_t + b_y) \quad (2)$$

where (f) and (g) stand for the activation functions, g is a function chosen based on the specific task, and bh and by are bias terms. When applied to concealed nodes, f is often a nonlinear, differentiable function.

Since RNNs can be thought of as deep feed-forward networks, back propagation is required after forward propagation in order to train them. The Back propagation through time (BPTT) algorithm is another name for this approach [27].

One of the most popular activation functions is the sigmoid function, sometimes known as the logistic function. A definition of the rising function can be found in (3).

$$\sigma(x) = sigmoid\ (x) = \frac{1}{1+e^{-x}} \tag{3}$$

As seen in Fig. 3, the sigmoid function takes real numbers and converts them to the range between 0 and 1. For output neurons that carry out the classification task, this means that it has a nice interpretation. However, there are some disadvantages to employing the sigmoid function. First, as the gradient levels approach the saturation thresholds, tail 0 or tail 1, the network is denied to learn more and will perform noticeably slowly [28][29].

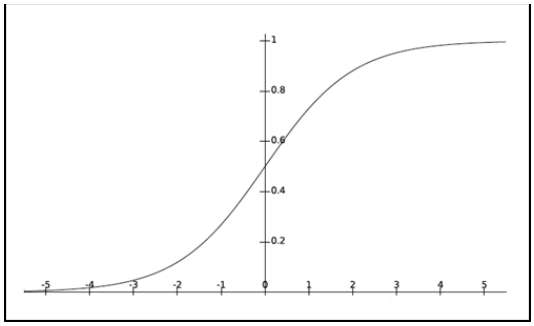

Fig. 3. Sigmoid functions

There is another type of activation function called the hyperbolic tangent function (tanh), which is also represented in the form of S. It is a non-linear function. This function differs from the sigmoid function in that in this (tanh) function the output range is [-1, 1] instead of [0, 1] as shown in Fig. 4.

Therefore, the hyperbolic tangent function is more preferred in practice and is given by equation (4) [28] [29]:

$$\sigma(x) = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

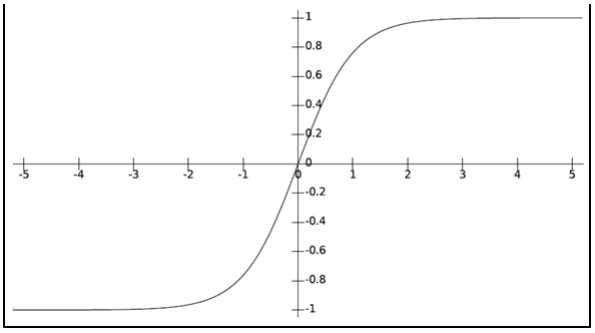

Fig. 4. Tangent (tanh) function

B. LSTM

The Long Short-term Memory Neural Network was proposed by Hochreiter and Schmidhuber in 1997 as a targeted design to prevent long-term dependence. In contrast to a single hidden layer of an RNN, LSTM establishes a new state unit C by storing data in a control unit outside of the RNN's regular flow [30]. The LSTM divides the hidden state of the RNN into two parts: working memory (ht) and memory cells (ct). The memory cell is in charge of preserving the sequence features. The memory of the previous sequence is controlled by the forgetting gate f. Working memory (ht) is the output, and what has to be recorded in the current memory (ct) is decided by the output gate (o). The current input xt and the portion of the current state information ht -1 that will be written to the memory cells are managed by the input gate I. The three types of gates mentioned above are dynamic. Through non-linear activation following linear combination, the present input xt and the previous state information ht -1 are simultaneously determined [23]. The definition of the LSTM architecture is as follows (Fig. 5):

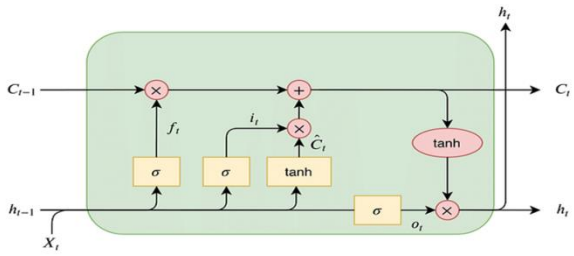

Fig. 5. LSTM architecture

Mathematically, it is given as follows:

$$f_t = \sigma(W_{hf}\, h_{t-1} + W_{xf}\, x_t + b_f) \tag{5}$$

$$i_t = \sigma(W_{hi}\, h_{t-1} + W_{xi}\, x_t + b_i) \tag{6}$$

$$o_t = \sigma(W_{ho}\, h_{t-1} + W_{xo}\, x_t + b_o) \tag{7}$$

$$\widetilde{C_t} = tanh(W_{hc}\, h_{t-1} + W_{xc}\, x_t + b_c) \tag{8}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{9}$$

$$h_t = o_t * tanh(C_t) \tag{10}$$

C. Gated Recurrent Units (GRU)

In 2014 the gated recurrent unit was developed by Cho et al [31]. The performance of the GRU model is similar to the LSTM model in numerous applications such as text translation, natural language processing, time series prediction of a variable, and speech signal modeling. In comparison to the LSTM model, the GRU model has fewer gates. GRU networks frequently outperform LSTM networks, even surpassing them for tiny data sets. The vanishing gradient problem affects RNNs. In order to address this issue, GRU was built with an update and reset gate; nevertheless, in contrast to LSTM, it also has the drawback of not having a dedicated output gate [32]. In numerous applications such as text translation, natural language processing, time series prediction of a variable, voice signal modeling, and natural language processing, the GRU model performs similarly to the LSTM model. There are fewer gates in the GRU model than in the LSTM model. Additionally, the GRU outperforms the LSTM and handles the small data set size better. The GRU, in contrast to the LSTM, uses an update and reset gate in place of a dedicated output gate to get over the standard RNN's vanishing gradient issue [33][34]. In the GRU model, the reset gate is the first gate. This reset gateway is crucial since it is used to determine how

much historical data (h_(t-1)) can be kept. The weight matrix of the reset gate is linearly changed by right-multiplying the data from the previous moment (h_(t-1)) and the present moment (X+). Following the addition of data, the reset gate multiplies the result by the sigmoid function, yielding a value that falls between [0, 1]. This operation is described by the equations below [33]. The GRU architecture can be defined as follows:

$$r_t = \sigma(W_{hr}\ h_{t-1} + W_{xr}\ x_t + b_r) \quad (11)$$

$$z_t = \sigma(W_{hz}\ h_{t-1} + W_{xz}\ x_t + b_z) \quad (12)$$

$$\widetilde{h_t} = tanh(r_t * W_{h\tilde{h}}\ h_{t-1} + W_{x\tilde{h}}\ x_t + b_{\tilde{h}}) \quad (13)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \quad (14)$$

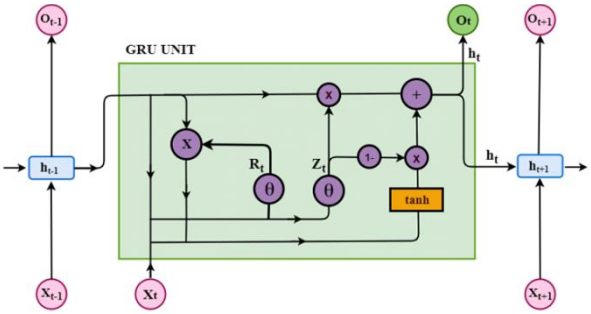

Fig. 6. GRU architecture

IV. Performance Metrics

Several requirements are being used to evaluate various forecasting algorithms for performance. The suggested models' performance was assessed using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for comparing target values and predicted values. To distribute mistakes uniformly, the RMSE and MAE computations are utilized, together with R-Square (R2): R-Squared, which has a range of [0, 1]. The model fits badly if R2 = 0, and it has no mistakes if $R^2 = 1$. The performance of MAE, RMSE and R2 can be calculated in equations (15) and (16) and (17):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |T_i - P_i| \quad (15)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (T_i - P_i)^2} \quad (16)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (T_i - P_i)^2}{\sum_{i=1}^{n} (T_i - P_i)^2} \quad (17)$$

where T_i prediction is P_i is real value and N is the number of testing samples.

A. Dataset Pre-Processing

One of the very important things that must be done before performing the operations is to preprocess. Preprocessing the input data is a very important part of the modeling process. We ensure the quality of the input data and ultimately improve the types of inputs, their configuration, the steps chosen and the time frames. Conducting this process has a significant and direct impact on the forecast results, their accuracy and reliability. Since RNN, GRU and LSTM is sensitive to data scales [23] the data is normalized by Min-Max and transformed to [0, 1], calculated by:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (18)$$

V. Model Implementation

The two proposed systems were applied to ten capitals in the Middle East, and 6 climate variables were used: humidity, wind speed, maximum and minimum temperature, Shortwave Downward Irradiance, and total rainfall.

In the first model, the inputs are six features; two hidden layers each consisting of 1000 RNN nodes, and six outputs, connected as fully connected layers as shown in the Fig. 7.

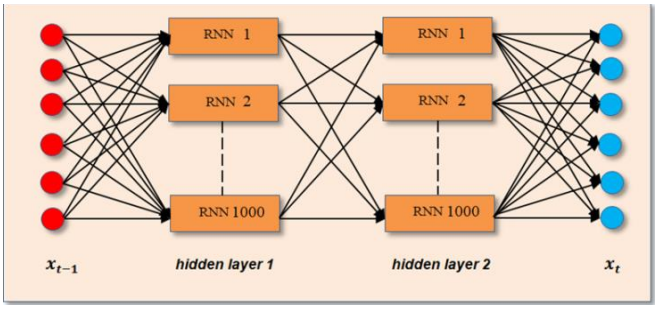

Fig. 7. The first model (RNN)

Six features, two hidden layers made up of 1,000 LSTM nodes each, and six outputs connected as fully connected layers make up the second model's inputs. The second algorithm, GRU, uses these six outputs as inputs. It has two hidden layers, each with 1,000 GRU nodes, and six connected outputs. as completely linked layers, as seen in Fig. 8.

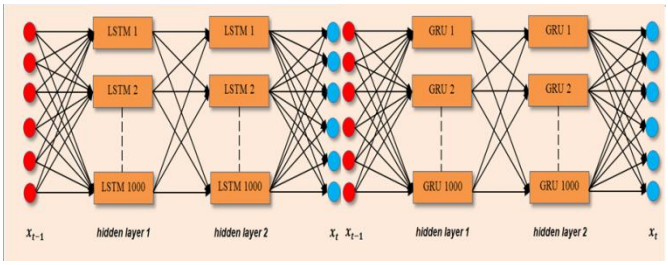

Fig. 8. The Hybrid model (LSTM & GRU)

The inputs are now resampled into a 3D format to work with the appropriate time steps within the prediction models, namely [date, timesteps, features]. For training and testing data, the input format will be [date, 1, characteristics] as long as the model is trained to require 30 days of input to forecast the weather for the following day and the time step is equal to thirty.

A forecast one day and one week in the future represents a short-term forecast. As for forecasting one month in the future, it is considered a medium-term forecast. Here the 3-month forecast was used to create a 5-month weather forecast (142 days represents 30 % of the testing data).

The input x(t-1) is used to predict x(t) as the output. x(t-1) represents one day of climate change for the six variables where forecasting is done by entering feature values for the period (t-1) to predict weather change for the period (t).

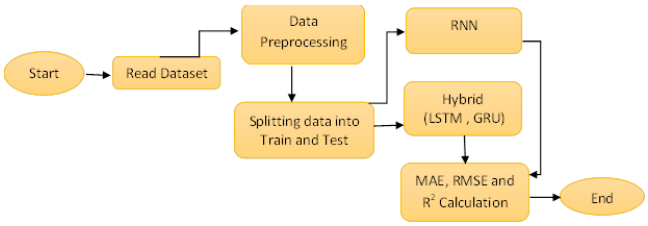

Fig. 9. The Proposed Prediction System Flowchart

VI. RESULTS AND DISCUSSION

The results we obtained from this study were evaluated with three accuracy measures: RMSE, MAE and R2, and the results extracted from applying the two models were as follows:

TABLE I. SHOW THE RESULTS OF THE ERROR PERCENTAGE FOR THE THREE METRICS RMSE, MAE, AND R2 FROM APPLYING THE TWO MODELS

		RNN			LSTM & GRU		
capital Cities	**Variables**	**RMSE**	**MAE**	**R2**	**RMSE**	**MAE**	**R2**
	Relative Humidity	0.098	0.073	0.880	0.093	0.067	0.894
	wind speed	0.094	0.075	0.811	0.088	0.071	0.834
Baghdad (Iraq)	Max_temp	0.070	0.057	0.943	0.054	0.042	0.966
	Min_ temp	0.082	0.067	0.929	0.068	0.054	0.951
	Solar Radiation	0.065	0.052	0.949	0.050	0.039	0.971
	Rainfall	0.137	0.089	0.212	0.136	0.088	0.222
Riyadh (Saudi	Relative Humidity	0.091	0.068	0.808	0.094	0.072	0.795
Arabia)	wind speed	0.147	0.111	0.474	0.153	0.120	0.433
	Max_temp	0.059	0.047	0.956	0.057	0.045	0.960
	Min_ temp	0.070	0.055	0.939	0.063	0.048	0.949
	Solar Radiation	0.061	0.047	0.927	0.057	0.046	0.936
	Rainfall	0.096	0.054	-0.074	0.091	0.046	0.026
Ankara (Turkey)	Relative Humidity	0.099	0.076	0.827	0.094	0.072	0.843
	wind speed	0.121	0.096	0.208	0.109	0.088	0.348
	Max_temp	0.074	0.058	0.917	0.069	0.053	0.926
	Min_ temp	0.082	0.066	0.855	0.077	0.061	0.874
	Solar Radiation	0.060	0.046	0.959	0.057	0.045	0.963
	Rainfall	0.143	0.110	0.081	0.139	0.109	0.131
Beirut (Lebanon)	Relative Humidity	0.136	0.108	0.400	0.133	0.103	0.427
	wind speed	0.131	0.099	0.392	0.128	0.094	0.418
	Max_temp	0.093	0.077	0.822	0.085	0.069	0.854
	Min_ temp	0.093	0.075	0.881	0.086	0.068	0.899
	Solar Radiation	0.070	0.057	0.949	0.055	0.045	0.969
	Rainfall	0.142	0.094	0.563	0.142	0.088	0.561
Cairo (Egypt)	Relative Humidity	0.096	0.076	0.814	0.100	0.077	0.796
	wind speed	0.140	0.111	0.463	0.136	0.109	0.490
	Max_temp	0.083	0.068	0.877	0.074	0.061	0.903
	Min_ temp	0.068	0.057	0.945	0.064	0.051	0.952
	Solar Radiation	0.072	0.052	0.944	0.052	0.044	0.971
	Rainfall	0.122	0.052	-0.087	0.119	0.041	-0.049
Doha (Qatar)	Relative Humidity	0.084	0.065	0.870	0.086	0.070	0.863
	wind speed	0.145	0.115	0.414	0.143	0.113	0.422
	Max_temp	0.071	0.057	0.939	0.062	0.047	0.954
	Min_ temp	0.071	0.056	0.937	0.053	0.043	0.965
	Solar Radiation	0.055	0.043	0.939	0.057	0.045	0.936
	Rainfall	0.118	0.075	-0.099	0.114	0.061	-0.025
Abu Dhabi	Relative Humidity	0.132	0.105	0.421	0.130	0.100	0.445
(UAE)	wind speed	0.142	0.108	0.402	0.141	0.106	0.411
	Max_temp	0.073	0.059	0.925	0.068	0.057	0.936
	Min_ temp	0.066	0.053	0.944	0.063	0.049	0.949
	Solar Radiation	0.073	0.056	0.894	0.071	0.053	0.900
	Rainfall	0.093	0.054	-0.094	0.089	0.049	0.003
Sanaa (Yemen)	Relative Humidity	0.145	0.114	0.357	0.149	0.119	0.319
	wind speed	0.108	0.083	0.565	0.121	0.094	0.451
	Max_temp	0.092	0.073	0.842	0.081	0.065	0.876
	Min_ temp	0.097	0.077	0.819	0.080	0.065	0.875
	Solar Radiation	0.114	0.087	0.638	0.108	0.083	0.678
	Rainfall	0.160	0.085	0.048	0.164	0.084	0.008
Tehran (Iran)	Relative Humidity	0.107	0.083	0.850	0.104	0.081	0.861
	wind speed	0.124	0.092	0.637	0.120	0.090	0.665
	Max_temp	0.066	0.051	0.943	0.064	0.050	0.946
	Min_ temp	0.073	0.054	0.924	0.067	0.049	0.938
	Solar Radiation	0.058	0.044	0.961	0.059	0.047	0.961
	Rainfall	0.218	0.141	0.067	0.216	0.141	0.080

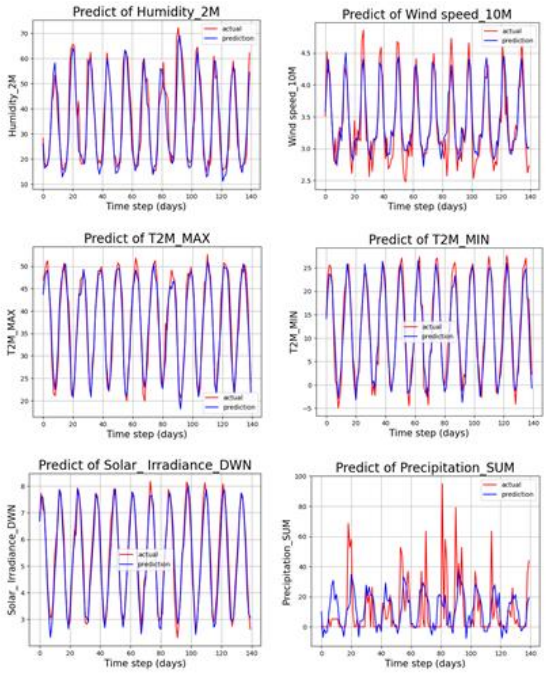

Fig. 10. Shows the Prediction of the Six Variables in the City Of Baghdad (Iraq) Using the First Model (Rnn).

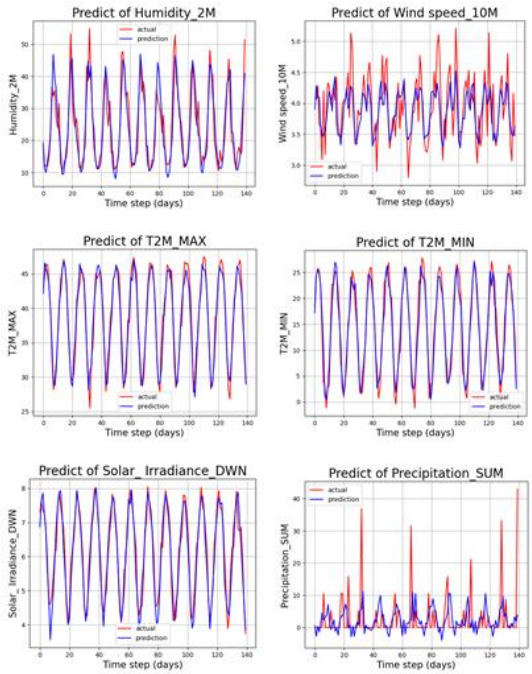

Fig. 11. Shows The Prediction Of The Six Variables In The City Of Riyadh (Saudi Arabia) Using The First Model, Rnn.

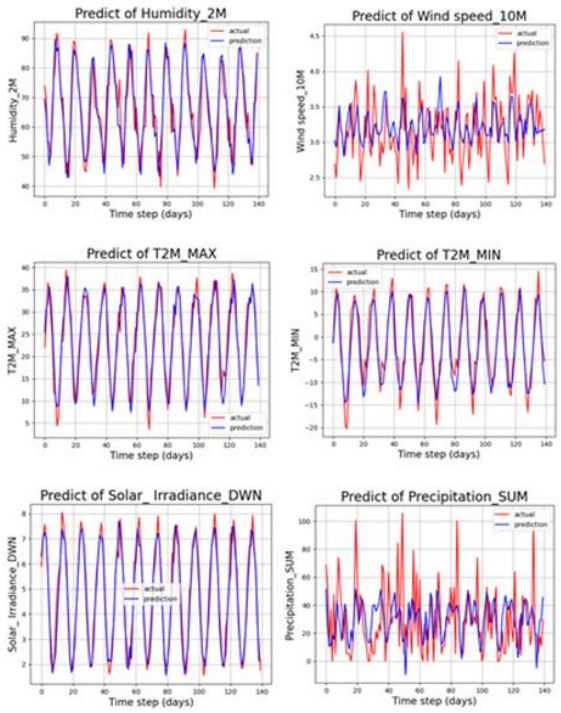

Fig. 12. Shows The Prediction Of The Six Variables In The City Of Ankara (Turkey) Using The First Model, Rnn.

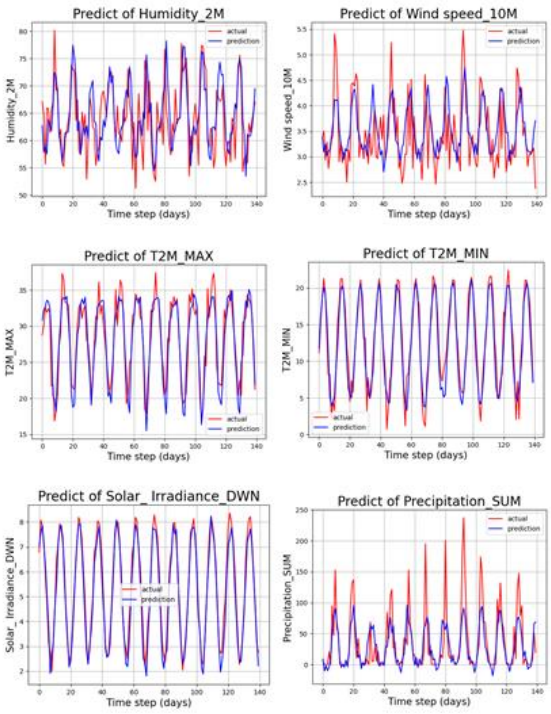

Fig. 13. Shows The Prediction Of The Six Variables In The City Of Beirut (Lebanon) Using The First Model, Rnn.

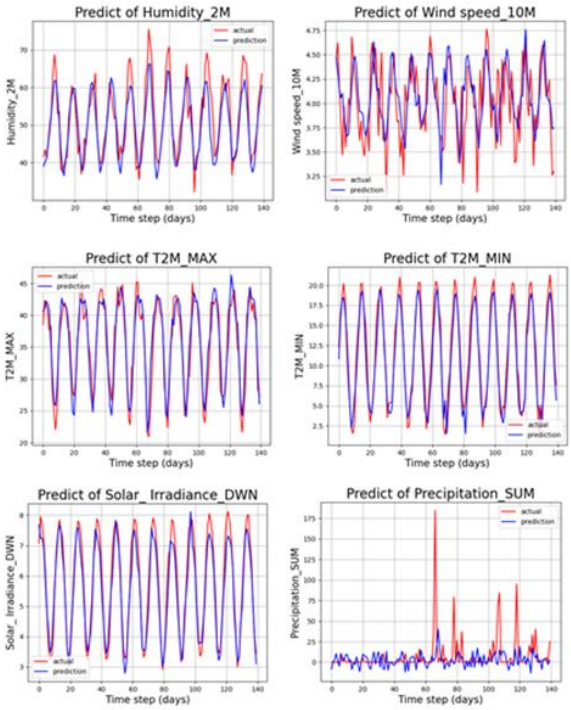

Fig. 14. Shows The Prediction Of The Six Variables In The City Of Cairo (Egypt) Using The First Model, Rnn.

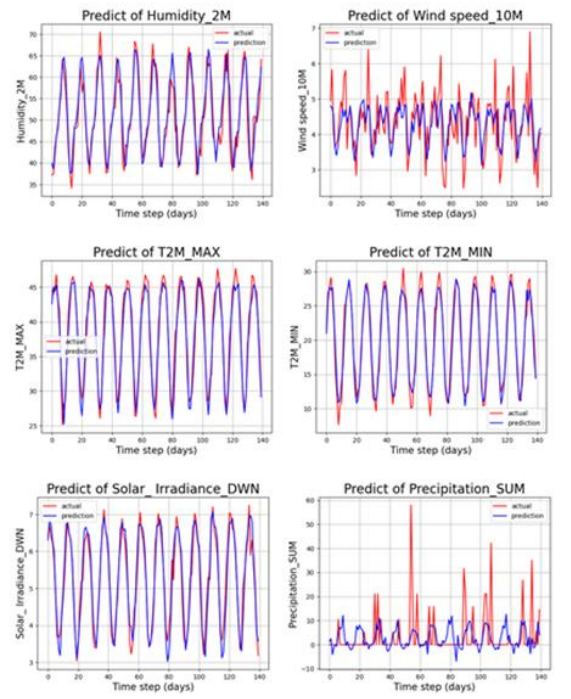

Fig. 15. Shows the Prediction Of The Six Variables In The City Of Doha (Qatar) Using The First Model, Rnn.

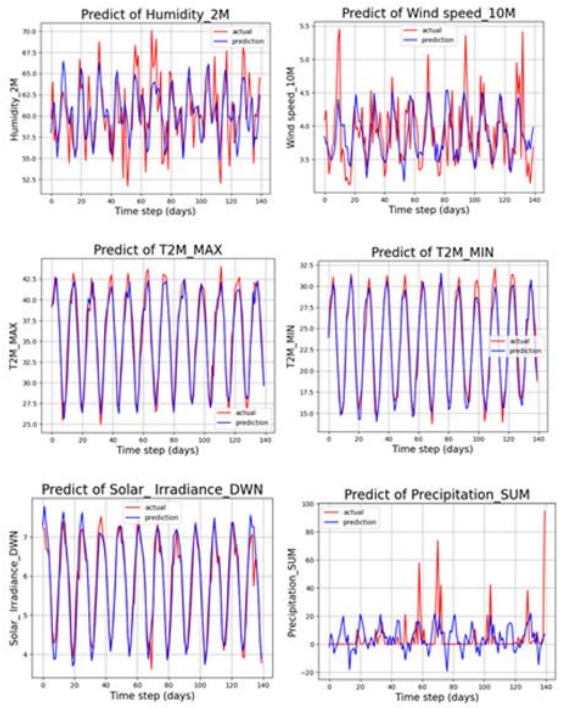

Fig. 16. Shows the Prediction of the Six Variables in the City of Abu Dhabi (UAE) Using The First Model, Rnn.

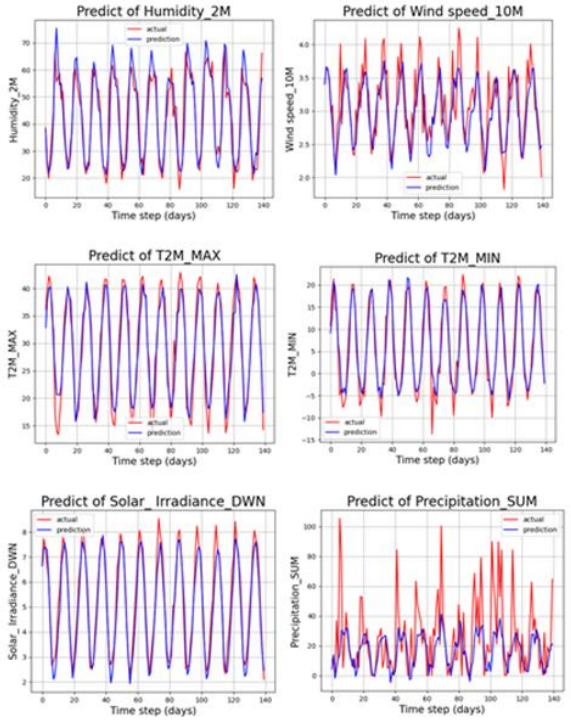

Fig. 17. Shows the prediction of the six variables in the city of Tehran (Iran) using the first model, rnn.

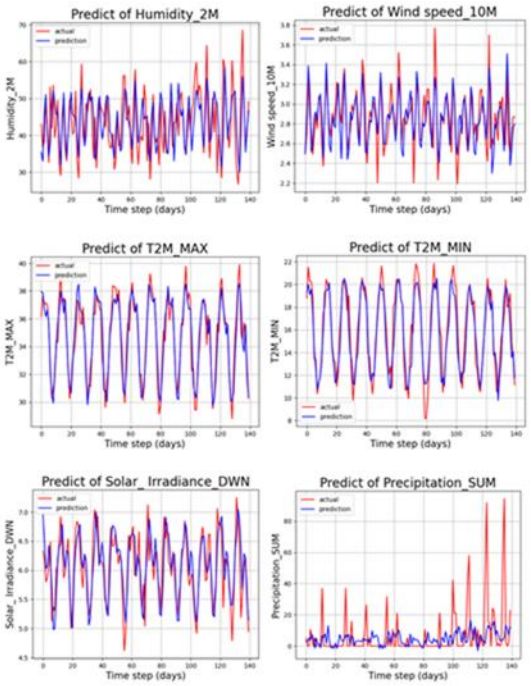

Fig. 18. Shows the Prediction of the Six Variables in The City Of Sanaa (Yemen) Using The First Model, Rnn.

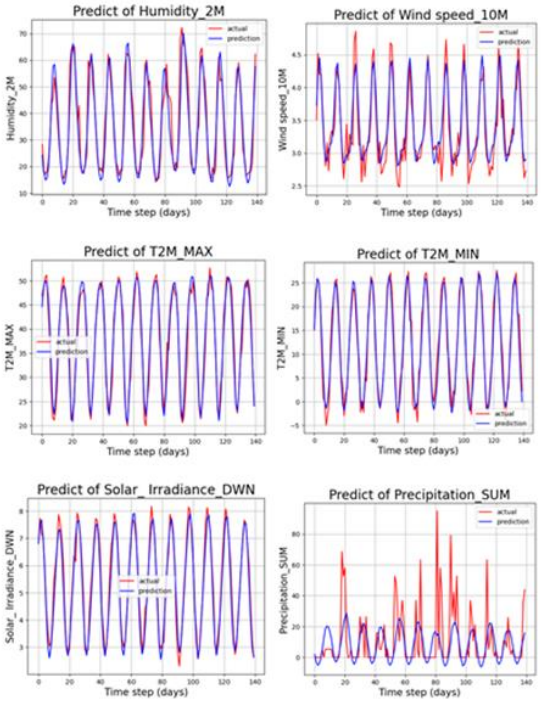

Fig. 19. Shows the prediction of the six variables in the city of baghdad (Iraq) using the second model (lstm&gru).

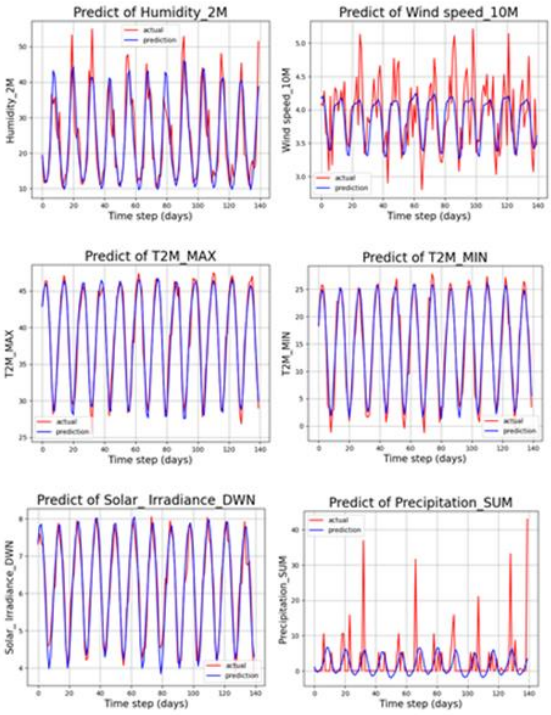

Fig. 20. Shows the prediction of the six variables in the city of riyadh (Saudi Arabia) using the second model (lstm&gru).

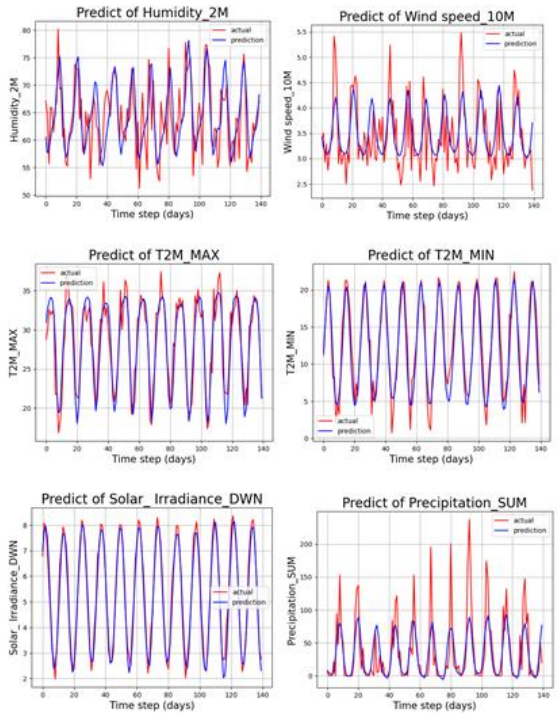

Fig. 21. Shows the prediction of the six variables in the city of Beirut (Lebanon) using the second model (lsm&gru).

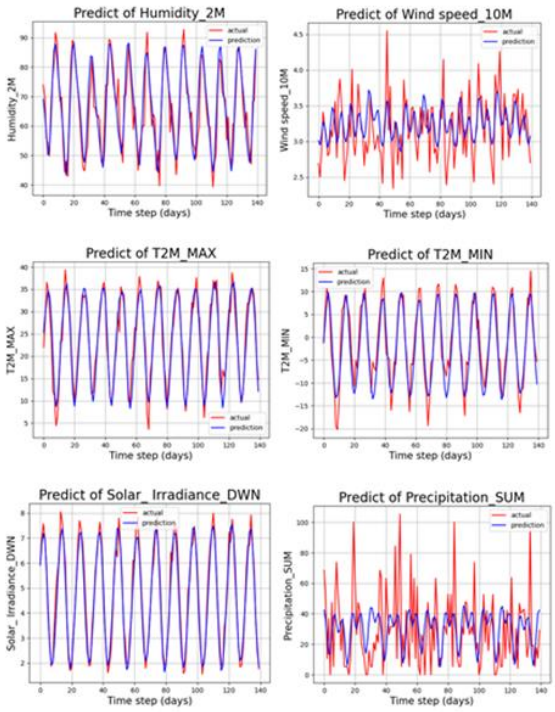

Fig. 22. Shows the prediction of the six variables in the city of ankara (turkey) using the second model (lstm&gru).

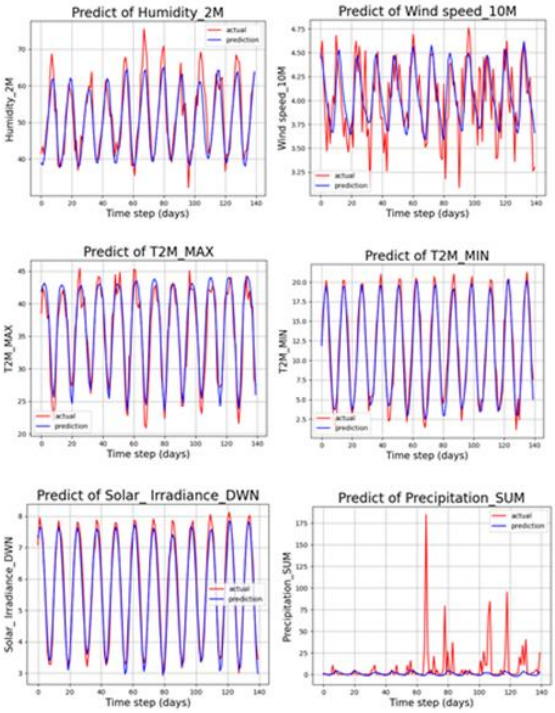

Fig. 23. Shows the prediction of the six variables in the city of Cairo (Egypt) using the second model (lstm&gru).

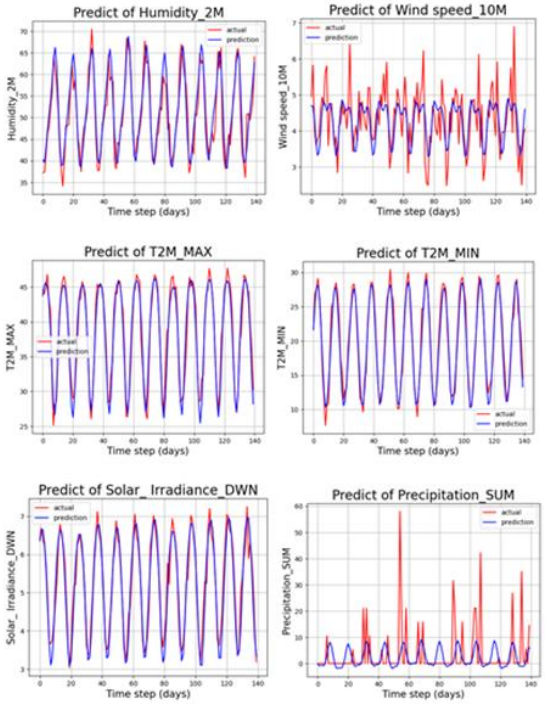

Fig. 24. Shows the prediction of the six variables in the city of Doha (Qatar) using the second model (lstm&gru).

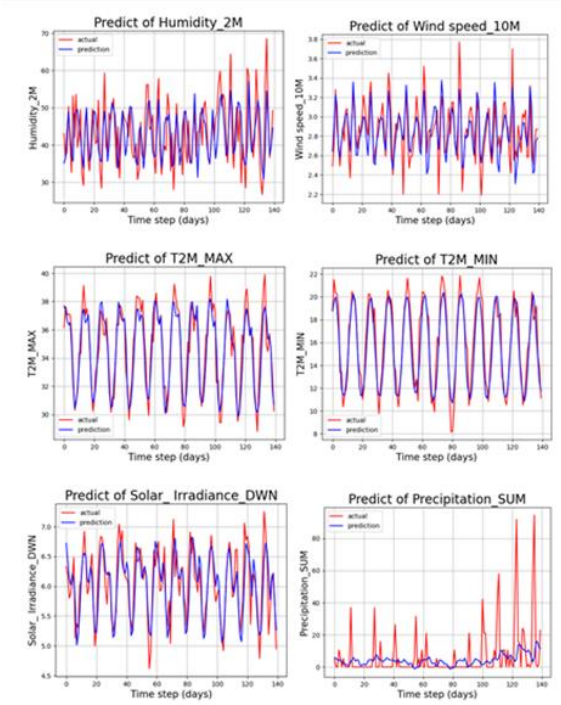

Fig. 25. Shows the prediction of the six variables in the city of sanaa (Yemen) using the second model (lstm&gru).

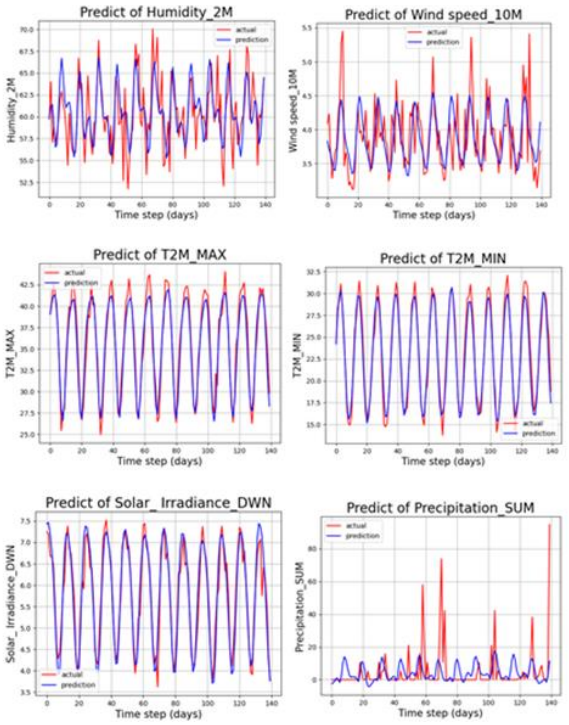

Fig. 26. Shows the prediction of the six variables in the city of Abu Dhabi (UAE) using the second model (lstm&gru).

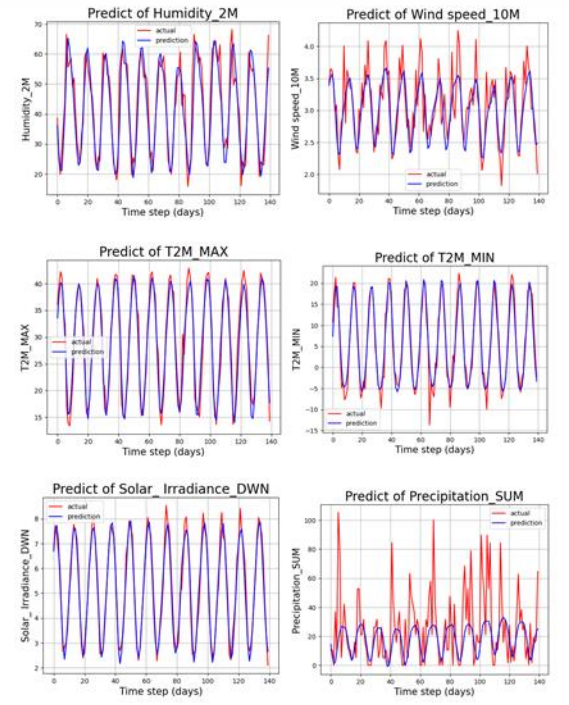

Fig. 27. Shows the prediction of the six variables in the city of Tehran (Iran) using the second model (lstm&gru).

In this work, we applied two deep learning models to predict weather in nine capitals in the Middle East. The first is recurrent neural networks (RNN) and the second is a hybrid model of (LSTM & GRU), and the results were evaluated using three metrics: RMSE, MAE, and R2.

The results shown in Table 1 were reached, where the results of the first model (RNN) showed good predictions regarding humidity, maximum and minimum temperature, and solar radiation for most of the capitals of the study area, as it showed results ranging between (0.091 – 0.145) (0.065 – 0.114) (0.880 – 0.357) for the humidity variable according to the RMSE, MAE and R2 scales respectively.

As for the maximum and minimum temperature variable, respectively, the results ranged between (0.059 – 0.093) (0.047 – 0.075) (0.956 – 0.842) and (0.066 – 0.097) (0.053 – 0.077) (0.945 – 0.819) according to RMSE, MAE and R2, respectively.

Likewise, the results of the solar radiation variable were as follows (0.055 – 0.114) (0.043 – 0.087) (0.961 – 0.638) according to the RMSE, MAE, and R2 metrics, respectively.

The results of the RNN model in predicting wind speed and rainfall were unsatisfactory because rainfall and wind are among the most difficult variables to predict due to the variation in the values of these variables every year according to the data on which the model was trained. Table 1 shows the results of rainfall and wind speed for the three metrics.

The results of the second hybrid model (LSTM & GRU) were better compared to the first model (RNN) Table 1 because the performance of GRUs and LSTMs is better than RNNs, which is consistent with the claim that GRUs and LSTMs have longer-term memories than RNNs and have partially addressed the vanishing gradient problem [35].

VII. Conclusion

In this paper, two deep learning models were applied to predict the weather for nine capitals in the Middle East region for six variables: relative humidity, wind speed, maximum and minimum temperature, solar radiation, and rainfall. The study period was from 1981 – 2022. The first model was the traditional recurrent neural network. (RNN), and the second model is a hybrid model of Long and short-term memory (LSTM) and gated repetitive units (GRU). The results of the second hybrid model are better than the first model. The results of humidity, maximum and minimum temperature, and solar radiation were of good accuracy compared to the two variables. Wind speed and rainfall showed very low results. In both models, predicting wind speed and rainfall was not easy because these variables are among the most difficult variables to predict due to the variation in their monthly and annual values over the years. We suggest for future studies the use of modified algorithms based on LSTM and GRU. To predict the weather and try to improve forecasting accuracy.

References

[1] X. Ren et al., “Deep learning-based weather prediction: a survey,” Big Data Res., vol. 23, p. 100178, 2021.

[2] G. Ayzel, M. Heistermann, and T. Winterrath, “Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0. 1),” Geosci. Model Dev., vol. 12, no. 4, pp. 1387–1402, 2019.

[3] P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” Nature, vol. 525, no. 7567, pp. 47–55, 2015.

[4] D. Shin, E. Ha, T. Kim, and C. Kim, “Short-term photovoltaic power generation predicting by input/output structure of weather forecast using deep learning,” Soft Comput., vol. 25, pp. 771–783, 2021.

[5] J. N. K. Liu and Y. Hu, “Application of feature-weighted support vector regression using grey correlation degree to stock price forecasting,” Neural Comput. Appl., vol. 22, pp. 143–152, 2013.

[6] M. Reichstein et al., “Deep learning and process understanding for data-driven Earth system science,” Nature, vol. 566, no. 7743, pp. 195–204, 2019.

[7] M. Khodayar, O. Kaynak, and M. E. Khodayar, “Rough deep neural architecture for short-term wind speed forecasting,” IEEE Trans. Ind. Informatics, vol. 13, no. 6, pp. 2770–2779, 2017.

[8] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, “EA-LSTM: Evolutionary attention-based LSTM for time series prediction,” Knowledge-Based Syst., vol. 181, p. 104785, 2019.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, 1986.

[10] H. Liu, X. Mi, and Y. Li, “Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM,” Energy Convers. Manag., vol. 159, pp. 54–64, 2018.

[11] [Z. Peng et al., “A novel deep learning ensemble model with data denoising for short-term wind speed forecasting,” Energy Convers. Manag., vol. 207, p. 112524, 2020.

[12] M. Afrasiabi, M. Mohammadi, M. Rastegar, and S. Afrasiabi, “Advanced deep learning approach for probabilistic wind speed forecasting,” IEEE Trans. Ind. Informatics, vol. 17, no. 1, pp. 720–727, 2020.

[13] S. Afrasiabi, M. Afrasiabi, B. Parang, and M. Mohammadi, “Real-time bearing fault diagnosis of induction motors with accelerated deep learning approach,” in 2019 10th international power electronics, drive systems and technologies conference (PEDSTC), 2019, pp. 155–159.

[14] M. Afrasiabi, M. Mohammadi, M. Rastegar, and A. Kargarian, “Multi-agent microgrid energy management based on deep learning forecaster,” Energy, vol. 186, p. 115873, 2019.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” arXiv Prepr. arXiv1409.1259, 2014.

[17] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv Prepr. arXiv1409.0473, 2014.

[18] M. Ghamariadyan and M. A. Imteaz, “Monthly rainfall forecasting using temperature and climate indices through a hybrid method in Queensland, Australia,” J. Hydrometeorol., vol. 22, no. 5, pp. 1259–1273, 2021.

[19] Q. Tao, F. Liu, Y. Li, and D. Sidorov, “Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU,” IEEE access, vol. 7, pp. 76690–76698, 2019.

[20] H. Astsatryan et al., “Air temperature forecasting using artificial neural network for Ararat valley,” Earth Sci. Informatics, vol. 14, pp. 711–722, 2021.

[21] D. Rolnick et al., “Tackling climate change with machine learning,” ACM Comput. Surv., vol. 55, no. 2, pp. 1–96, 2022.

[22] Z. Al Sadeque and F. M. Bui, “A deep learning approach to predict weather data using cascaded LSTM network,” in 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2020, pp. 1–5.

[23] Y. Yu, J. Cao, and J. Zhu, “An LSTM short-term solar irradiance forecasting under complicated weather conditions,” IEEE Access, vol. 7, pp. 145651–145666, 2019.

[24] S. Poornima and M. Pushpalatha, “Prediction of rainfall using intensified LSTM based recurrent neural network with weighted linear units,” Atmosphere (Basel)., vol. 10, no. 11, p. 668, 2019.

[25] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” Phys. D Nonlinear Phenom., vol. 404, p. 132306, 2020.

[26] K. Benidis et al., “Deep learning for time series forecasting: Tutorial and literature survey,” ACM Comput. Surv., vol. 55, no. 6, pp. 1–36, 2022.

[27] R. J. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networks and their computational complexity,” in Backpropagation, Psychology Press, 2013, pp. 433–486.

[28] D. Rysbek, “Sentiment analysis with recurrent neural networks on turkish reviews domain.” Middle East Technical University, 2019.

[29] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” Towar. Data Sci, vol. 6, no. 12, pp. 310–316, 2017.

[30] Y. Huang, L. Shen, and H. Liu, “Grey relational analysis, principal component analysis and forecasting of carbon emissions based on long short-term memory in China,” J. Clean. Prod., vol. 209, pp. 415–423, 2019.

[31] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv Prepr. arXiv1406.1078, 2014.

[32] M. Chhetri, S. Kumar, P. Pratim Roy, and B.-G. Kim, “Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan,” Remote Sens., vol. 12, no. 19, p. 3174, 2020.

[33] B. Wang, W. Kong, H. Guan, and N. N. Xiong, “Air quality forecasting based on gated recurrent long short term memory model in Internet of Things,” IEEE Access, vol. 7, pp. 69524–69534, 2019.

[34] K. U. Kala and M. Nandhini, “Gated recurrent unit architecture for context-aware recommendations with improved similarity measures,” KSII Trans. Internet Inf. Syst., vol. 14, no. 2, pp. 538–561, 2020.

[35] J. Shi, M. Jain, and G. Narasimhan, “Time series forecasting (tsf) using various deep learning models. arXiv 2022,” arXiv Prepr. arXiv2204.11115.