

# Comparison of time series temperature prediction with autoregressive integrated moving average and recurrent neural network

Hamza Jdi, Nouredine Falih

LIMATI Laboratory, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco

## Article Info

### Article history:

Received Feb 4, 2023

Revised Jul 2, 2023

Accepted Nov 29, 2023

### Keywords:

Agriculture

Big data analytics

Deep learning

Machine learning

Temperature forecast

## ABSTRACT

The region of Beni Mellal, Morocco is heavily dependent on the agricultural sector as its primary source of income. Accurate temperature prediction in agriculture has many benefits including improved crop planning, reduced crop damage, optimized irrigation systems and more sustainable agricultural practices. By having a better understanding of the expected temperature patterns, farmers can make informed decisions on planting schedules, protect crops from extreme temperature events, and use resources more efficiently. The lack of data-driven studies in agriculture impedes the digitalization of farming and the advancement of accurate long-term temperature prediction models. This underscores the significance of research to identify the optimal machine learning models for that purpose. A 22-year time series dataset (2000-2022) is used in the study. The machine-learning model autoregressive integrated moving average (ARIMA) and deep learning models simple recurrent neural network (SimpleRNN), gated recurrent unit (GRU), and long short-term memory (LSTM) were applied to the time series. The results are evaluated based on the mean absolute error (MAE). The findings indicate that the deep learning models outperformed the machine-learning model, with the GRU model achieving the lowest MAE.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Hamza Jdi

LIMATI Laboratory, Polydisciplinary Faculty, Sultan Moulay Slimane University

Slimane Mghila BP: 592 Beni Mellal, Morocco

Email: hamzajdi@gmail.com

## 1. INTRODUCTION

Temperature forecasting is critical for early warnings of weather impacts on numerous facets of human living. For example, temperature forecast is critical in agriculture. On one hand, low temperatures restrict water intake due to lower water viscosity and impede the photosynthetic process. On the other hand, warmth promotes vegetation development by increasing water and nutrient intake as well as total plant growth. It is clear that plant development and soil temperature are inextricably linked [1], [2]. Therefore, accurate temperature forecasts can be very helpful in agriculture 4.0 [3].

Traditional theory-driven numerical weather prediction [4] systems face various hurdles [5], such as an insufficient understanding of physical principles and difficulties extracting valuable knowledge from a flood of recorded observations. The successful integration of deep learning techniques, driven by data, has been observed across several industries including natural language processing [6], autonomous vehicles [7] [8], and fraud detection [9], [10]. The proficiency of deep learning in discovering intricate patterns in data and making accurate predictions results in its widespread adoption in these sectors. We wish to extend this

success further in favor of agriculture [11], [12] through weather prediction, temperature forecast in particular, and try to improve the accuracy of the weather forecast obtained in our previous work [13].

The main approaches in the literature range from classical methods to the more advanced machine learning and deep learning approaches [14]. For instance, Yalçın [15] proposes a deep hybrid neural network-based weather forecasting technique using time series. The suggested model trains and predicts the essential parameters for weather forecasting such as temperature using a hybrid long short-term memory (LSTM) convolutional neural network (CNN) model. According to the results obtained with the proposed method, the mean absolute percentage error (MAPE) value for temperature is 1.21. In study [16], the seasonal auto-regressive integrated moving average (SARIMA) and the Prophet model were used to forecast the air temperature. The Prophet model outperforms the SARIMA model in terms of accuracy, with a root mean square percent error of 3.2%. In research [17], LSTM used to estimate the sea surface temperature anomalies based on the previous year's sea surface temperature anomalies. The LSTM model is evaluated using a time series containing 40 years, the model yields an error of 0.036, proves capable of predicting temperature anomalies, and has mean absolute error (MAE) of 0.14. Local weather data is used in another study [18] as input variables in greenhouses. The prediction accuracy of the random forest model is found to be better than the support vector machine and multiple linear regression (MLR) models for minimum temperature predictions. Deep learning models, such as the GRU and LSTM models, had a higher prediction performance than the generalized machine learning models. The GRU model has the best prediction performance overall, with an average R2 value 5.38% higher and root mean squared error (RMSE) 44.34% lower than the MLR model. Study [19] presents a new method for temperature forecasting using a convolutional recurrent neural network (CRNN), a combination of convolutional neural network and recurrent neural network (RNN). The model utilizes historical data to learn time and space correlations of temperature changes. Results of testing the CRNN model with daily temperature data from China between 1952 and 2018 show a prediction error of 0.907 °C. In a different study [20], a hybrid SARIMA-LSTM model is developed to improve air temperature forecasting accuracy. The temperature series is decomposed into trend, seasonal, and residual components using seasonal-trend decomposition. SARIMA is used to predict trend and seasonal components, while LSTM is used to fit residual components. The prediction results of both models are combined to obtain the final prediction result. Evaluation of the model is performed using RMSE, MAE, MAPE, and Kupiec index. Results show that the SARIMA-LSTM hybrid model is more accurate than single models and other combination models, with an improvement of 10% to 27.7%.

In this research, our objective is to evaluate the performance of four distinct algorithms namely SimpleRNN, LSTM, GRU, and ARIMA in forecasting temperature using a daily incremented time series from 2000 to 2022. We aim to determine the most suitable ARIMA model and juxtapose its performance with the well-fitted RNN models. Given the inherent properties of all stated models, we anticipate that RNN models, specifically LSTM and GRU, will surpass the ARIMA model in temperature prediction because of their capacity to capture time series data dependencies.

## 2. METHOD

### 2.1. SimpleRNN

SimpleRNN is the most basic technique for a neural network to retain information over time. The information is saved in the hidden variable  $h$  and is updated each time newer inputs are received. The hidden variable's output can be computed using the time distributed component [21].

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \quad (1)$$

$$y_t = h_t \quad (2)$$

The hidden state  $h_t$  and output  $y_t$  are important variables in RNNs.  $h_t$  refers to the “hidden state” and is the output of the RNN at time step  $t$ . It is computed by taking the dot product of the previous hidden state ( $h_{t-1}$ ), the input at time step  $t$  ( $x_t$ ), and a bias term ( $b$ ) and passing the result through a sigmoid function ( $\sigma$ ). The output  $y_t$  is equal to the hidden state  $h_t$ .

### 2.2. Long short-term memory

Hochreiter and Schmidhuber [22] propose the LSTM cell to address the issue of long-term dependence. They boost the standard recurrent cell's remembering capacity by putting a “gate” into the cell. The LSM model is displayed in (3)-(7).

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} C_{t-1}) \quad (3)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}C_{t-1}) \quad (4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_{t-1}) \quad (5)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1}) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

where  $i_t$  is the “input gate” variable, which controls how much of the new input ( $x_t$ ) to use in updating the cell state,  $o_t$  is the “output gate” variable, which controls how much of the cell state ( $C_t$ ) to use in generating the output,  $C_t$  is the “cell state” variable, which stores the current hidden state of the LSTM,  $h_t$  is the “hidden state” variable, which is the output of the LSTM at time step  $t$ . All these variables are computed from the dot product of input weight matrix  $W_x$ , hidden weight matrix  $W_h$ , and bias term. The variables are updated using the sigmoid function ( $\sigma$ ) and tanh function ( $\tanh$ ).

### 2.3. Gated recurrent unit

The GRU RNN model reduces the gating signals of the LSTM RNN model to two. The update gate  $z_t$  and the reset gate  $r_t$  are the two gates. The GRU RNN model is displayed in (8)-(11) [23].

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r) \quad (8)$$

$$z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + b_z) \quad (9)$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}h}(r_t \cdot h_{t-1}) + W_{\tilde{h}x}x_t + b_r) \quad (10)$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \quad (11)$$

The  $r_t$  and  $z_t$  terms represent the “reset” and “update” gates, respectively. These gates are used to control the flow of information from the previous hidden state  $h_{t-1}$  and the current input  $x_t$  into the current hidden state  $h_t$ . The  $W_{rh}$ ,  $W_{rx}$ ,  $W_{zh}$ , and  $W_{zx}$  terms are weight matrices that are used to transform the previous hidden state and current input.  $b_r$  and  $b_z$  are bias terms. The  $\tilde{h}_t$  term is the candidate hidden state, which is computed by applying a tanh nonlinearity to a linear combination of the current input, the previous hidden state, and the reset and update gates. The final hidden state  $h_t$  is computed by combining the previous hidden state and the candidate hidden state, with the update gate determining the degree to which the candidate hidden state should be used. The GRU is identical to the LSTM in general, but with less external gating signal in the interpolation. This preserves one gating signal and its associated settings.

### 2.4. The ARIMA

ARIMA as its name implies incorporates three methods, which are moving average, integration (differencing), and autoregression. ARIMA models rely on direct correlations and indirect correlations between lags via two functions, which are autocorrelation and partial autocorrelation. This makes it well suited for temperature predictions because temperature is correlated with an instance in a given time series, and a lagged version of itself. The stationarity is a critical aspect because nonstationary time series cannot be forecasted by the ARIMA models.

$$ARIMA(p, d, q) = \sum_{i=1}^p \varphi_i L^i X_t^1 (1 - L)^d \cdot Y_t = \sum_{i=1}^q (1 + \theta_i e_{t-i}) e_t \quad (12)$$

where  $L$  is the lag operator,  $Y_t$  is the time series at time  $t$ ,  $e_t$  is the error term,  $\varphi_i$  are the autoregressive parameters, and  $\theta_i$  are the moving average parameters.

### 2.5. Methodology and data description

In this research paper, our objective is to predict the daily average temperature in Beni Mellal using time series data through four algorithms: ARIMA, SimpleRNN, GRU, and LSTM. After conducting comprehensive analyses and reviewing various approaches found in the literature, it becomes clear that the choice of ARIMA, SimpleRNN, GRU, and LSTM models in the study offers a well-rounded exploration of forecasting techniques. This selection ranges from traditional linear statistical models to advanced neural networks, striking a balance between simplicity, interpretability, and performance. In contrast, CRNNs,

which combine the strengths of CNNs and RNNs, are more suitable for tasks requiring both spatial and temporal feature extraction. However, for temperature time series forecasting, the added complexity of convolutional layers may not provide significant benefits over simpler models like RNNs, GRUs, and LSTMs this is because temperature time series data does not have any spatial information, so the convolutional layers would not be able to extract any useful features. Therefore, for temperature time series forecasting, it is better to use simpler models that are specifically designed for temporal feature extraction. Similarly, CNNs, primarily designed for image recognition [24], may not be the optimal choice for time series temperature forecasting. Although they can be adapted for time series data, they lack the inherent ability to capture temporal dependencies like RNNs, GRUs, and LSTMs. SARIMA, an extension of ARIMA that incorporates seasonality, is designed to handle non-stationary and seasonal data. The data used in this study can be easily transformed into a stationary form after the first differentiation which is the main reason why we choose to work with ARIMA because its models offer greater interpretability than SARIMA models, as they exclude seasonal components. This advantage aids in understanding the underlying patterns and relationships in the data. Additionally, ARIMA models are computationally more efficient due to fewer parameters, which is crucial for large datasets or limited computational resources. Furthermore, model selection is simplified with ARIMA, as it involves choosing optimal values for  $p$ ,  $d$ , and  $q$  parameters, while SARIMA requires additional seasonal parameters, making the process more complex and time-consuming.

The data used is mainly collected from the National Climatic Data Center (NCDC) website. They can be downloaded directly from their website. It concerns station ID 60191099999, which is the local Beni Mellal weather station. However, upon observing missing data, particularly for the year 2016, additional data was required. The missing data span from January 1, 2016 to May 23, 2016, and therefore, this specific period's daily data is acquired from Beni Mellal's local weather station. Eventually, we managed to have more than 22 years' worth of daily data from January 1, 2000, to December 31, 2022. The four algorithms are fed with data in the form of a comma-separated value (CSV) file. The file contains two columns: the first column contains the date in the format YYYY-MM-DD. The dates are incremented daily. The second column represents the temperature observed in Fahrenheit (F). The hardware characteristics is the following: 8<sup>th</sup> gen Intel® Core™ i7, Processor frequency: 2.20 GHz with random access memory (RAM) 8 GB. We used Python 3.9.12 in Jupyter Notebook 3, leveraging Keras and TensorFlow.

In both RNN and ARIMA, the initial step involves preprocessing temperature data. This process entails gathering data from the yearly CSV files downloaded from NCDC, merging them in ascending order from 2000 to 2022, and adding missing information when necessary. When the four models are fine-tuned, the data are split into two distinct sets: the training set covers the period from January 1, 2000, to December 31, 2019, while the testing set encompasses the period from January 1, 2020, to December 31, 2022.

For RNNs, the approach consists of defining the RNN architecture by experimenting with various parameters through grid search, by varying the following parameters: the number of units, embedding size, time lags, number of dense layers, learning rate, and batch size. For each combination we employ cross-validation to determine the number of epochs. Cross-validation consists of splitting the training set into four equally sized, non-overlapping folds, each containing a smaller training set and a validation set. This division ensures the model is evaluated across various time periods, capturing different seasonal patterns and trends. The calculated average metrics across all four-time frames constitute the model's estimated metrics. The hyperparameter combination that yields the best average performance is then used on the entire training data.

For ARIMA, the methodology includes additional steps beyond preprocessing temperature data. These steps involve checking data stationarity, which is easily achieved from the first differentiation, making  $d$  equal to 1. Determining the AR and MA components is done using the results obtained by Akaike information criterion (AIC); these results help to narrow down the scope of the search. As a result, we tried all possible combinations for  $p$  and  $q$  values that vary from 0 to 9. As for  $d$  values, they vary from 0 to 1 until the best-fitting model is found.

### 3. RESULTS AND DISCUSSION

#### 3.1. The proposed RNN models and the parameters used

Upon performing a grid search to determine optimal parameters, by arbitrarily changing the number of units, embedding size, time lags, number of dense layers, learning rate, and batch size, and employing cross-validation to determine the appropriate number of epochs, we observe that the MAE improvement/deterioration is consistent across all three models. The following parameters yield the most favorable outcomes when applied to our time series: number of units: 256, embedding size: 08, time lags: 07, number of dense layers: 32, learning rate: 1e-10, batch size: 32, number of epochs: 1817. We believe that these parameters serve as a valuable foundation for fellow researchers exploring the application of RNNs in

predicting temperature within the Beni Mellal region. These are the results of three deep learning algorithms that based on the chosen parameters have the same number of layers, that is, three layers in total. The first layer corresponds to the algorithm utilized, which can be a SimpleRNN layer, an LSTM layer, or a GRU layer, this first layer's output is a vector (None, 256). Following that comes a temporally distributed dense layer; it takes the form of (None, 32). The RNNs model's architectures are shown in Tables 1, 2, and 3. The SimpleRNN model has the lowest parameter count (67,840) but may struggle with long-range dependencies due to its vanishing and exploding gradient issues. The LSTM model has the highest parameter count (271,360) expected to effectively address the vanishing gradient problem and be suitable for capturing long-range dependencies. The GRU model, with a parameter count of 204,288, offers a balance between the other two models, addressing the vanishing gradient problem with fewer parameters than the LSTM. This balance was the key to the GRU model achieving the lowest MAE in this study with a value of 2.961 F, which is the best among LSTM and SimpleRNN, with MAEs of 2.973 F and 2.986 F respectively. We present the training MAE per epoch for the GRU model in Figure 1.

Table 1. The proposed SimpleRNN architectures

Layer (Type)	Output shape	Parameters
simple_rnn (SimpleRNN)	(None, 256)	67840
dense (Dense)	(None, 32)	8224
dense_1 (Dense)	(None, 1)	33

Table 2. The proposed LSTM architectures

Layer (Type)	Output shape	Parameters
lstm (LSTM)	(None, 256)	271360
dense (Dense)	(None, 32)	8224
dense_1 (Dense)	(None, 1)	33

Table 3. The proposed GRU architectures

Layer (Type)	Output shape	Parameters
gru (GRU)	(None, 256)	204288
dense (Dense)	(None, 32)	8224
dense_1 (Dense)	(None, 1)	33

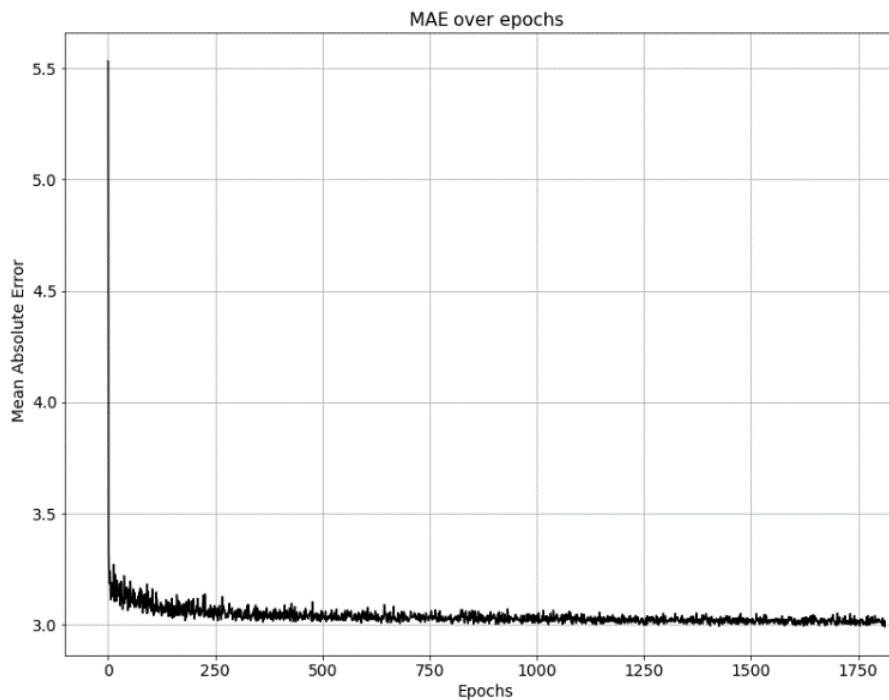


Figure 1. GRU MAE per epoch

### 3.2. The proposed ARIMA model

The methods to determine the stationarity of data vary from simple ones: from looking at the time series graph and observing if it meets the stationarity conditions, to more complex mathematical methods, such as augmented dickey fuller test (ADF) [25]. In this study, we work with ADF, a robust formal test. The results of the ADF test indicate that the data does not exhibit a unit root and is therefore stationary, which is a crucial assumption as it allows us to move forward without having to worry about the issue of non-stationarity. Please note that ARIMA takes three parameters  $p$ ,  $d$ , and  $q$ . These parameters need to be chosen with care as they directly influence the accuracy of the model. For that purpose, we choose AIC, which is a statistical measure used to compare different time series models. It is worth noting that because the AIC is not based on a hypothesis test, it cannot privilege a model's forecasting accuracy in reference to another model. The AIC simply indicates which model fits the available data even if it is by small margins [26]. We use the results obtained by AIC to narrow down the scope of the search. As a result, we try all combinations possible for  $p$  and  $q$  values that vary from 0 to 9. As for  $d$  values, they vary from 0 to 1. Based on the result obtained by testing various parameter combination, we are confident that ARIMA model (5, 1, 9) is the best for forecasting daily average temperature for in Beni Mellal taking the dataset as reference, as shown in Figure 2. The figure shows MAE obtained by varying  $p$ ,  $d$ , and  $q$ . We include only the MAE values under 3.26. The green points represent MAE values for  $d = 0$  with  $p$  and  $q$  ranging from 0 to 9, while the blue points represent MAE values for  $d = 1$  with  $p$  and  $q$  ranging from 0 to 9. The optimum ARIMA model with the lowest MAE of 3.1129 is indicated in red.

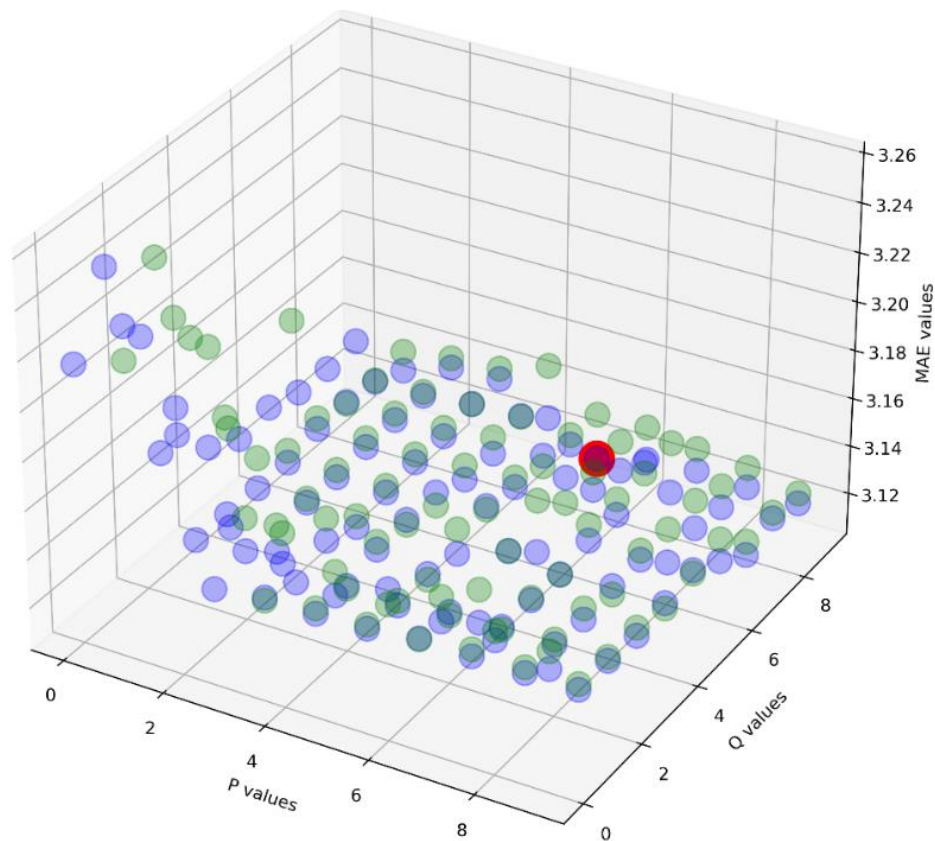


Figure 2. 3D display of the performance of various ARIMA models for forecasting daily average temperature in Beni Mellal

### 3.3. GRU, LSTM, SimpleRNN and ARIMA comparison

In this section, we use the results from the deep learning and machine learning sections for the aim of comparing the performances of the used algorithms. The best results produced by each algorithm, based on the MAE, are reported in Table 4. As demonstrated in Table 4, the MAE of GRU is the smallest when compared with the rest of the deep learning algorithms and the ARIMA model. The MAE of GRU model decreases by 0.40% when compared with the LSTM model, and when compared with the SimpleRNN, we

observe that it is low by 0.83%. As for ARIMA model, its MAE increases by 4.85% in comparison to the GRU's MAE. The experimental results indicate that the GRU has achieved the best predicting performance. The GRU's slightly better performance, in terms of MAE compared to LSTM, suggests that the GRU model is more effective at capturing the data's patterns. However, the small difference in MAE indicates that both models performed similarly. Furthermore, the GRU being slightly better is attributed to its simpler architecture, which has fewer parameters compared to LSTMs. This results in faster training times and a reduced risk of overfitting. While it is true that LSTM is known for its ability to capture long-term dependencies in time series data, the implication that GRU outperforms LSTM suggests that the data may lack significant long-term dependencies, or that the balance seen in GRU architecture and faster training times may have been sufficient to capture the relevant patterns in the data, leading to GRU being slightly better. The good performance seen in deep learning-based methods in forecasting temperature is due to the iterative optimization made possible by the extended version of stochastic gradient descent, formally called Adam optimizer, that solves optimization problems by minimizing the loss function. As a result, the best outcome is discovered.

Even finding the best ARIMA model, the SimpleRNN with known vanishing gradient problem performs better. This makes it clear that the deep-learning algorithms have the upper hand in this case, which is understandable, because RNNs have the ability to learn complex relationships between inputs and outputs through multiple layers of neurons and can handle sequential information effectively. However, this also means RNNs can be more computationally expensive and harder to train. The ARIMA model outperforms the SimpleRNN, LSTM, and GRU in terms of execution time. The large number of parameters and the number of epochs used by the deep learning algorithms can account for the difference in the execution time. The GRU model is employed to perform temperature forecasting over a two-year period, from January 1<sup>st</sup>, 2020 to December 31<sup>st</sup>, 2022. The results, depicted in Figure 3, showcase the model's efficacy in accurately predicting temperature values over the extended time frame, with a strong correlation between the observed and predicted values. Given this high level of accuracy and precision, the model can be relied upon in various industries, such as agriculture, to provide reliable long-term forecasting, enabling effective planning.

Table 4. MAE, MSE, MAPE, and training execution time obtained for each algorithm

	ARIMA model (5, 1, 9)	SimpleRNN	LSTM	GRU
MAE	3.112	2.986	2.973	2.961
MSE	-	16.177	16.027	15.894
MAPE	-	4.471	4.456	4.439
Time (s)	60	10030	10060	9976

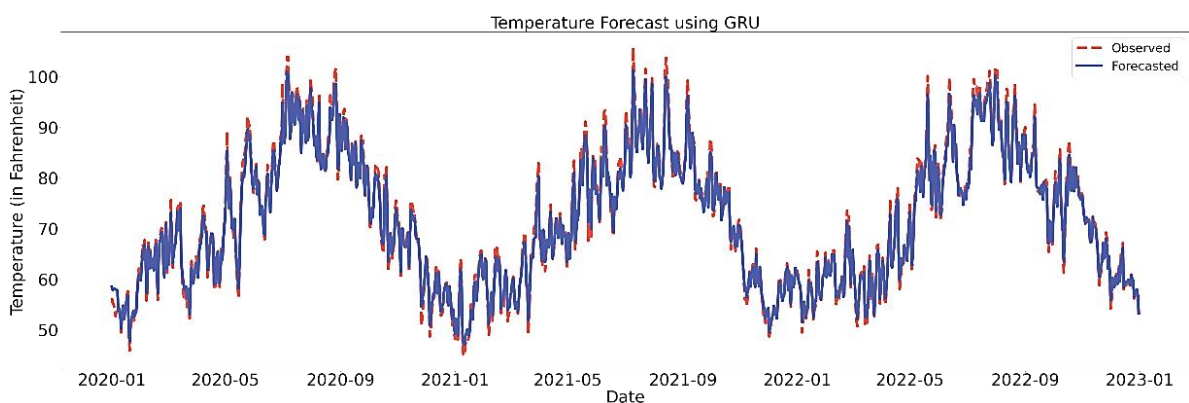


Figure 3. Observed vs predicted temperature values using GRU model for the period of January 1<sup>st</sup>, 2020 to December 31<sup>st</sup>, 2022

#### 4. CONCLUSION

In this study, we compare the forecasting accuracy of four algorithms: ARIMA, SimpleRNN, LSTM, and GRU using MAE as an evaluation metric. The results indicate the success of the GRU algorithm in predicting temperature values in Beni Mellal, making it a viable option for temperature forecasting. The close MAE obtained and decent performance of the four algorithms suggest that any one of them could be utilized for similar forecasting tasks. However, it is important to acknowledge several limitations that may






impact the performance and interpretability of our deep learning models. The dataset size, potential overfitting, and the “black box” nature of deep learning models may pose challenges in comprehending the underlying relationships between input features and predictions. Additionally, ARIMA models have their own limitations, such as the assumption of linearity and stationarity in the time series data, which may not always hold true in real-world scenarios. Future work will involve exploring the application of hybrid approaches for weather parameters’ prediction in Beni Mellal, while addressing the aforementioned limitations to improve model performance and interpretability.

## REFERENCES




- [1] A. Raza *et al.*, “Melatonin-mediated temperature stress tolerance in plants,” *GM Crops and Food*, vol. 13, no. 1, pp. 196–217, Aug. 2022, doi: 10.1080/21645698.2022.2106111.
- [2] A. A. Ambekar, P. Sivaperumal, K. Kamala, P. Kubal, and C. Prakash, “Effect of temperature changes on antioxidant enzymes and oxidative stress in gastropod *Merita oryzae* collected along India’s first Tarapur Atomic Power Plant site,” *Environmental Research*, vol. 216, Jan. 2023, doi: 10.1016/j.envres.2022.114334.
- [3] M. Lezoche, H. Panetto, J. Kacprzyk, J. E. Hernandez, and M. M. E. Alemany Díaz, “Agri-food 4.0: a survey of the supply chains and technologies for the future agriculture,” *Computers in Industry*, vol. 117, May 2020, doi: 10.1016/j.compind.2020.103187.
- [4] P. S. Chang and G. K. Egan, “An implementation of a barotropic numerical weather prediction model in the functional language SISAL,” *ACM SIGPLAN Notices*, vol. 25, no. 3, pp. 109–117, Feb. 1990, doi: 10.1145/99164.99176.
- [5] A. Meque, S. Gamedze, T. Moitlhobogi, P. Booneady, S. Samuel, and L. Mpalang, “Numerical weather prediction and climate modelling: challenges and opportunities for improving climate services delivery in Southern Africa,” *Climate Services*, vol. 23, Aug. 2021, doi: 10.1016/j.cliser.2021.100243.
- [6] O. Maarouf, R. El Ayachi, and M. Biniz, “Amazigh part-of-speech tagging with machine learning and deep learning,” *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 24, no. 3, pp. 1814–1822, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1814-1822.
- [7] A. Sestino, A. M. Peluso, C. Amatulli, and G. Guido, “Let me drive you! the effect of change seeking and behavioral control in the artificial intelligence-based self-driving cars,” *Technology in Society*, vol. 70, Aug. 2022, doi: 10.1016/j.techsoc.2022.102017.
- [8] E. F. Abdelhafid, Y. M. Abdelkader, M. Ahmed, D. Rachid, and E. I. Abdelilah, “Visual and light detection and ranging-based simultaneous localization and mapping for self-driving cars,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 6284–6292, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6284-6292.
- [9] J. M. Arockiam and A. C. S. Pushpanathan, “MapReduce-iterative support vector machine classifier: novel fraud detection systems in healthcare insurance industry,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, pp. 756–769, Feb. 2023, doi: 10.11591/ijece.v13i1.pp756-769.
- [10] V. F. Rodrigues *et al.*, “Fraud detection and prevention in e-commerce: a systematic literature review,” *Electronic Commerce Research and Applications*, vol. 56, Nov. 2022, doi: 10.1016/j.elerap.2022.101207.
- [11] K. El Moutaouakil, B. Jabir, and N. Falih, “A convolutional neural networks-based approach for potato disease classification,” in *Lecture Notes in Business Information Processing*, vol. 449, Springer International Publishing, 2022, pp. 29–40.
- [12] A. Tannouche, A. Gaga, M. Boutalline, and S. Belhouideg, “Weeds detection efficiency through different convolutional neural networks technology,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 1048–1055, Feb. 2022, doi: 10.11591/ijece.v12i1.pp1048-1055.
- [13] H. Jdi and N. Falih, “Weather forecast using sliding window algorithm based on Hadoop and MapReduce,” in *Lecture Notes in Networks and Systems*, vol. 357 LNNS, Springer International Publishing, 2022, pp. 122–132.
- [14] B. Lim and S. Zohren, “Time-series forecasting with deep learning: a survey,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, Feb. 2021, doi: 10.1098/rsta.2020.0209.
- [15] S. Yalçın, “Weather parameters forecasting with time series using deep hybrid neural networks,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 21, Jun. 2022, doi: 10.1002/cpe.7141.
- [16] N. Das, A. Sagar, R. Bhattacharjee, A. K. Agnihotri, A. Ohri, and S. Gaur, “Time series forecasting of temperature and turbidity due to global warming in river Ganga at and around Varanasi, India,” *Environmental Monitoring and Assessment*, vol. 194, no. 9, Jul. 2022, doi: 10.1007/s10661-022-10274-7.
- [17] M. S. Pravalika, S. Vasavi, and S. P. Vigneshwar, “Prediction of temperature anomaly in Indian Ocean based on autoregressive long short-term memory neural network,” *Neural Computing and Applications*, vol. 34, no. 10, pp. 7537–7545, Jan. 2022, doi: 10.1007/s00521-021-06878-8.
- [18] Z. He *et al.*, “Gated recurrent unit models outperform other machine learning models in prediction of minimum temperature in greenhouse based on local weather data,” *Computers and Electronics in Agriculture*, vol. 202, Nov. 2022, doi: 10.1016/j.compag.2022.107416.
- [19] Z. Zhang, Y. Dong, and Y. Yuan, “Temperature forecasting via convolutional recurrent neural networks based on time-series data,” *Complexity*, vol. 2020, pp. 1–8, Mar. 2020, doi: 10.1155/2020/3536572.
- [20] G. Li and N. Yang, “A hybrid SARIMA-LSTM model for air temperature forecasting,” *Advanced Theory and Simulations*, vol. 6, no. 2, Dec. 2023, doi: 10.1002/adts.202200502.
- [21] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: LSTM cells and network architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco\_a\_01199.
- [22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [23] M. Otman, E. A. Rachid, and B. Mohamed, “Amazigh part of speech tagging using gated recurrent units (GRU),” in *2021 7th International Conference on Optimization and Applications (ICOA)*, May 2021, pp. 1–6, doi: 10.1109/ICOA51614.2021.9442662.
- [24] Y. El Madmoune, I. El Ouariachi, K. Zenkour, and A. Zahi, “Robust face recognition using convolutional neural networks combined with Krawtchouk moments,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4052–4067, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4052-4067.
- [25] M. H. Pesaran, “A simple panel unit root test in the presence of cross-section dependence,” *Journal of Applied Econometrics*, vol. 22, no. 2, pp. 265–312, Mar. 2007, doi: 10.1002/jae.951.
- [26] V. A. Profillidis and G. N. Botzoris, “Trend projection and timeseries methods,” in *Modeling of Transport Demand*, Elsevier, 2019, pp. 225–270.



**BIOGRAPHIES OF AUTHORS**

**Hamza Jdi**    has a M.Sc. in business intelligence, received in 2019 from the Faculty of Science and Technologies, University Sultan Moulay Slimane, Beni Mellal, Morocco. Currently, he is a Ph.D. student in the Polydisciplinary Faculty of Beni Mellal. His research interest is about big data analytics, business intelligence, computer science, digital agriculture, and deep learning. He can be contacted by email at: hamzajdi@gmail.com.



**Noureddine Falih**    has a Ph.D. on computer science, received from the Faculty of Sciences and Technologies of Mohammedia, Morocco in 2013. He is an associate professor since 2014 in the Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco. He has 18 years of professional experience in several renowned companies. His research topics are about information system governance, business intelligence, big data analytics and digital agriculture. He can be contacted at email: nourfald@yahoo.fr.