Comparative analysis of Elman and Jordan Recurrent Neural Networks for Solar Radiation and Air Temperature Prediction Using Backpropagation Variants

Rachid Ed-daoudi[1*], Mourad Azhari[2], Badia Ettaki[1], Jamal Zerouaoui[3]
[1]LyRICA: Laboratory of Research in Informatics, Data Sciences and Artificial Intelligence, School of Information Sciences, B.P. 604, Rabat-Instituts, Rabat, Morocco
[2]Center of Guidance and Educational Planning, B.P. 6222, Rabat-Instituts, Rabat, Morocco
[3]Laboratory of Engineering Sciences and Modeling, Faculty of Sciences- Ibn Tofail University, Campus Universitaire, BP 133, Kenitra, Morocco
E-mail : rachid.ed-daoudi@uit.ac.ma, azharimourad@yahoo.fr, bettaki@esi.ac.ma, jamal.zerouaoui@uit.ac.ma.
[*]Corresponding author

Keywords: recurrent neural networks, prediction, backpropagation, solar radiation, temperature

Received: January 29, 2025

This study compares the effectiveness of Elman Recurrent Neural Networks (ERNN) and Jordan Recurrent Neural Networks (JRNN) for predicting solar radiation and ambient temperature in Ouarzazate, Morocco. Data collected at 10-minute intervals from three meteorological stations (OUA_001, OUA_002, and OUA_003) over a 3–4-year period (2018-2022) were analyzed. The dataset was split into training (75%) and validation (25%) subsets to develop and test the models. The research systematically explored different network architectures with 1-2 hidden layers containing 4-12 neurons, applying three learning algorithms: BackPropagation (BP), BackPropagation with Momentum (BPM), and Resilient Backpropagation (Rprop). Key hyperparameters were optimized within specific ranges, including learning rates (0.00001-0.4) for BP/BPM and weight decay exponents (0.00001-4) for Rprop. Input variables included date, temperature, solar radiation, wind speed, relative humidity, precipitation, and atmospheric pressure in various combinations. Performance evaluation using Nash-Sutcliffe Efficiency (NSE) and Index of Agreement (d) revealed high prediction accuracy for both model types, with values exceeding 0.9 during validation. JRNN with BPM performed best at station OUA_001 (NSE: 0.909 for radiation, 0.971 for temperature), while ERNN with BPM demonstrated superior performance at station OUA_002 (NSE: 0.978 for radiation, 0.976 for temperature). At station OUA_003, both models showed comparable results when using BP. Despite the high overall accuracy, both models exhibited limitations in predicting extreme solar radiation values, particularly during nighttime hours. The study concludes that ERNN and JRNN are effective tools for short-term prediction of solar radiation and temperature in arid regions like Ouarzazate, though further refinement is needed to better capture extreme values and improve prediction accuracy during transitional periods.

Povzetek: Elmanove in Jordanove povratne nevronske mreže so uporabljene za 48-urnem napovedovanju sončnega sevanja in temperature, pri čemer JRNN izkazuje boljšo računsko učinkovitost in primerljivo točnost.

1 Introduction

The rapid growth of solar energy generation technologies requires increasingly sophisticated processing techniques to understand the variability of solar resources over short time intervals [1]. Photovoltaic solar energy is an intermittent renewable source that can be considered a non-stationary time series [2]. Accurate forecasting of solar radiation and temperature is crucial for efficient energy management, grid stability, and optimization of solar power plants, particularly in regions with high solar potential like Ouarzazate, Morocco.

Traditional time series prediction methods include statistical approaches of the AutoRegressive (AR) type and their variants, such as the AutoRegressive Moving Average (ARMA) method or the AutoRegressive Conditional Heteroskedasticity (ARCH) [3]. These methodologies are univariate and generally perform well with stationary time series. However, they often struggle with the non-stationary, intermittent nature of solar radiation data and the complex relationship between meteorological variables [4].

Machine learning methods in AI, like neural networks, allow for the incorporation of other relevant variables into the model when predicting stationary or non-stationary time series. Among the different architectures of Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs) have shown promise in pattern classification and prediction involving multiple variables. RNNs incorporate feedback mechanisms that enable them to capture temporal dependencies in time series data, making them potentially suitable for solar radiation and temperature prediction [5].

The Elman Recurrent Neural Network (ERNN) and Jordan Recurrent Neural Network (JRNN) represent two distinct architectures of RNNs with different feedback mechanisms [6]. While both have been applied to various forecasting problems, their comparative performance in predicting solar radiation and temperature presents an opportunity for further research

This study aims to address the following specific research questions:

- How do ERNN and JRNN compare in terms of accuracy for short-term (48-hour) prediction of solar radiation and ambient temperature?
- Which combination of network architecture, learning algorithm (BP, BPM, or Rprop), and hyperparameters gives the optimal prediction performance for each meteorological station?
- How does the inclusion of different meteorological variables (wind speed, relative humidity, precipitation, and atmospheric pressure) as inputs affect the prediction accuracy of the models?
- What are the specific strengths and limitations of ERNN and JRNN in capturing the temporal patterns of solar radiation and temperature, particularly during extreme values and transitional periods?

The article is organized as follows: Section 2 reviews related work in solar and temperature prediction. Section 3 explains the methodology used for predicting temperature and radiation time series using ERNN and JRNN. Section 4 presents the results obtained, including figures and comparative statistical tables of the two prediction methods. Section 5 discusses the obtained results in terms of performance comparison. Finally, Section 6 presents the conclusions and recommendations of the study.

2 Literature review

2.1 Neural networks for time series prediction

ANNs have gained significant attention in environmental and renewable energy forecasting due to their ability to model complex non-linear relationships without requiring explicit knowledge of the underlying physical processes. As noted by Krogh [7], ANNs are computational models inspired by the functioning of biological neurons, consisting of a series of processors (neurons) organized into interconnected layers. These neurons process input signals and generate output through weighted connections. RNNs represent a specialized class of neural networks designed to capture temporal dependencies in sequential data. RNNs are composed of neuronal units where information travels from the input layer to the output layer with feedback or memory of past events [8]. This feedback mechanism allows RNNs to represent dynamic systems, such as non-stationary time series, making them particularly suitable for solar radiation and temperature forecasting.

The fundamental difference between an ERNN and a JRNN lies in their feedback mechanisms: in Elman networks, feedback goes from the output of the hidden layer to the context layer, whereas in Jordan networks, feedback occurs from the output layer to the neurons of the context layer. This architectural distinction affects both computational requirements and predictive capabilities.

2.2 Solar radiation and temperature prediction

Solar radiation and temperature forecasting have been approached using various techniques ranging from statistical models to complex machine learning architectures. Table 1 presents a comparative summary of recent studies focusing on solar radiation and temperature prediction, highlighting their methodologies, key metrics, and findings.

Table 1: Comparative summary of recent studies on solar radiation and temperature prediction

Study	Region/Dataset	Method	Prediction Horizon	Input Variables	Performance Metrics	Key Findings	Limitations
Sudharshan et al. [1]	Multiple regions	Review of irradiance forecasting	Various	Various	Various	Combining methods (ensemble/hybrid) works best. Deep learning excels with images and long-term predictions. Simpler models suffice for short-term forecasts. Each approach has strengths: time series models handle patterns well, and physical models establish reliable baselines.	Key challenges include location-specific performance, unpredictable weather, computational demands, inconsistent evaluation standards, and data quality issues.

Yuan Gao et al. [9]	Tokyo and Okinawa, Japan (hourly meteorological data from 2020-2022)	Adversarial Discriminative Domain Adaptation (ADDA) - a zero-label transfer learning approach using domain adaptation and GAN training techniques	One hour ahead	historical weather data and temporal features	MSE: 0.0857 (ADDA model with Okinawa as source) R^2: 0.8870 (ADDA model with Okinawa as source) MAE: 2.42 (ADDA model with Okinawa as source predicting for Tokyo in January)	4-16% improved accuracy over baseline models without requiring target domain labels	Requires identical network architectures between domains, same input feature structures, and limited testing to only two Japanese regions.
Muhammad Samee Sevas et al. [10]	Bangladesh	Ensemble machine learning approaches, hybridized clustering techniques with LightGBM, and Explainable AI (XAI)	-	Multiple variables including "Sunshine (Hours)" identified as most important	R^2: 0.91 (best performance from LightGBM and CatBoost)	LightGBM and CatBoost ensemble methods achieved R^2=0.91 for solar irradiance prediction. Sunshine (Hours) identified as most influential predictor, with hybridized clustering approach performing best for "Very Cloudy" conditions	Not explicitly stated
Wassila Tercha et al.[11]	Algeria 7-month period data (Jan-July) 212 daily entries with year, month, day, temperature, and irradiance	Decision Tree Random Forest Support Vector Machine XGBoost	Daily forecasting	Temporal features Historical temperature data Historical solar irradiance data	MAE : 47.49, MSE : 5289.67, RMSE : 72.73	Decision Tree performed best for temperature forecasting with perfect fit. Random Forest achieved best results for solar irradiance Decision Tree and XGBoost models had fastest prediction speeds	Small dataset size (7-month period only) Climate variability and extreme weather events
Chengliang Fan et al. [12]	Pearl River New Town (PRNT), Guangzhou, China	Long Short-Term Memory neural network model and SHAP	Hourly air temperature prediction, using 3-hour historical data as optimal time step	Five urban morphology factors Historical air temperature data Land cover and urban spatial form classifications	R^2 : 0.975 RMSE: 0.344°C MAE : 0.256°C	LSTM outperformed CNN and FCN in microclimate prediction	Limited data samples for training the model Predicts with slight underestimation during day and overestimation at night
Faouzi Didi et al. [13]	Algeria, Meteorological data	fuzzy logic controller (FLC) based on the Mamdani method	real-time control and optimization of the greenhouse microclimate	Temperature, Relative Humidity,Solar Radiation, Wind Speed and Direction, CO_2 Concentration	-	The fuzzy logic controller effectively managed the greenhouse microclimate	Complexity and Regional performance and limitations
Alaa Sahl Gaafar et al. [14]	Antarctic Automatic Weather Stations (1980–2021) data and Kaggle standard dataset	Fuzzy AHP, Fuzzy logic control	nowcasting focus, 6-hourly intervals	Wind directionhumidity, air temperature, pressure, wind speed	MSE : 0.1563 RMSE : 0.3953 MAPE : 00.2104	FAHP-optimized FLC reduced error rates significantly vs. unoptimized FLC and Model performed better on standardized datasets	Performance variability across datasets and Computational constraints due to hardware limitations

The literature review indicates that most studies predict solar radiation or temperature separately rather than jointly with unified models. Comparative analyses of different RNN architectures (particularly ERNN and JRNN) are limited for meteorological prediction in arid regions, with no investigation into how various learning algorithms and input combinations affect prediction accuracy. This study addresses these limitations by comparing ERNN and JRNN models for joint prediction of solar radiation and temperature in Ouarzazate, Morocco, evaluating different architectures, learning algorithms, and input variables to enhance understanding of the applications of the two RNN variants for prediction.

3 Materials and methods

The methodology of this work is divided into five sections: the Working Mechanism of a Neural Network model, Data and materials, identification of outliers in the analysis data, application of ERNN and JRNN in prediction, and selection criteria for the prediction method.

3.1 How Neural Networks models work?

A neural network is composed of interconnected layers of neurons. Mathematically, these components can be described as follows [15]:

- Inputs: represented as a vector $x = x_1$, $x_2, x_3, \ldots\ldots x_i, \ldots. x_n)$ where x_i are the features of the input data;
- Weights: each input is multiplied by a weight w_{ij}, where i denotes the input and j denotes the neuron in the layer. Weights are stored in matrices for efficient computation;
- Bias: a scalar b_j added to the weighted sum to adjust the output, allowing flexibility in the model;
- Output of a Neuron: for a single neuron (equation 1) ;

$$z_j = \sum w_{ij} x_i + b_j \tag{1}$$

- This is often represented in vector by equation 2;

$$z = wx + b \tag{2}$$

- Activation Function: a non-linear function $f(z_j)$applied to the output z_j, producing the neuron's final output (equation 3).

$$y_j = f(z_j) \tag{3}$$

Figure 1 illustrates the general structure of an artificial neuron [16].

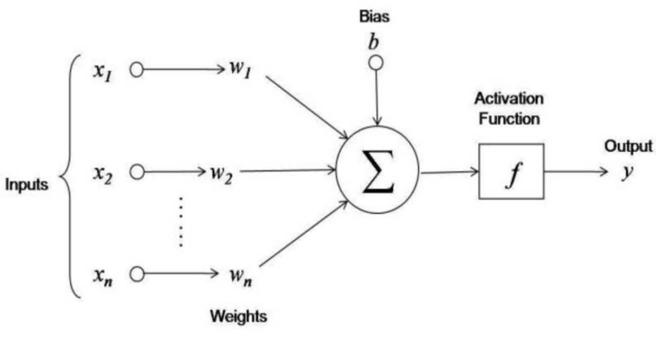

Figure 1: Structure of an artificial neuron

3.2 Data and materials

The data used come from three meteorological stations located in the city of Ouarzazate, Morocco. The details of the stations' locations are shown in Table 2.

Table 2: Geographical location of the meteorological stations (WGS84)

Station	Latitude (°N)	Longitude (°W)	Altitude (m)
OUA_001	30.9335	-6.9094	1,160
OUA_002	30.9250	-6.9130	1,150
OUA_003	30.9420	-6.9000	1,170

The key variables were normalized, and outliers removed to enhance data quality.
In Table 3, the period of data analysis for the meteorological stations is shown.

Table 3: Data analysis period of the meteorological stations

Station	Analysis Period
OUA_001	01-01-2020 to 31-12-2022
OUA_002	01-06-2019 to 31-12-2021
OUA_003	01-01-2018 to 31-12-2021

From the analysis period used, the dataset was split into training (75%) and validation (25%) set.
The specifications of the meteorological stations in Table 1 are detailed in Table 4.

Table 4: Meteorological stations specifications

Description	Model	Unit of Measure
Barometer	BMP280	hPa
Anemometer	WindSonic	m/s
Wind Vane	WindSonic	degrees
Thermometer	PT100	°C
Pyranometer	CMP11	kW/m²
Hygrometer	HMP60	%RH
Pluviometer	Tipping Bucket Rain Gauge (0.2mm resolution)	mm
Data Logger	Campbell CR1000	--
Solar Charge Controller	MPPT 30A	--
Battery	Deep Cycle AGM	--

The data recording frequency is every ten minutes, resulting in 144 samples obtained in a full day of measurement. The behavior of a time series of solar radiation and ambient temperature for meteorological station OUA_001 over one month of data is shown in Figure 2 and Figure 3.

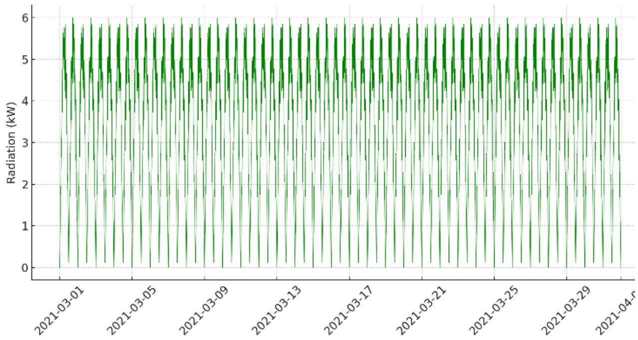

Figure 2: Time series of solar radiation for station OUA_001, period 01/03/2021 – 31/03/2021

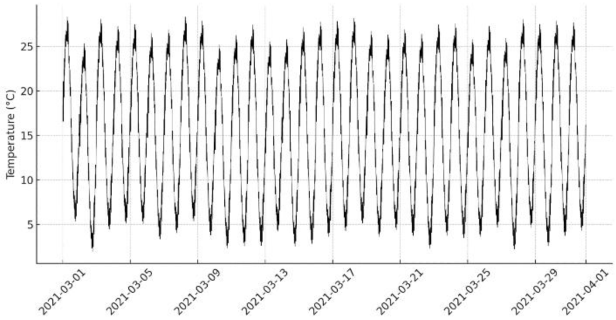

Figure 3: Time series of temperature for station OUA_001, period 01/03/2021 – 31/03/2021

For the analysis of data from the meteorological stations and for the prediction of solar radiation and temperature, Python (Version 3.8) was used. Additionally, the simulation of the ERNN and JRNN networks utilized the TensorFlow and Keras libraries [17].

3.3 Identifying outliers

Data from the meteorological stations undergo processing in which the quality of the measurements recorded by the sensors is evaluated.

Outliers were identified by applying a range-based filtering method, removing data points that fell outside the predefined measurement limits for each sensor type (as detailed in Table 5). This boundary-based approach effectively eliminated measurements beyond the expected physical ranges, ensuring data quality by excluding values potentially resulting from sensor errors or anomalous readings.

Table 5: Measurement ranges for each sensor of the meteorological station

Sensor	**Unit**	**Measurement Range**
Anemometer	m/s	0 to 60
Pyranometer	kW/m²	0 to 1,500
Thermometer	°C	-10 to 50
Hygrometer	%RH	0 to 100
Pluviometer	mm/h	0 to 300
Barometer	hPa	800 to 1,050

Additionally, values that were not recorded due to sensor failure, power loss, or maintenance at the station were also considered.

3.4 Application of ERNN and JRNN in prediction

A RNN is characterized by having a context layer where part of the information is fed back as a new input; this allows the network to have greater learning capacity by recognizing and generating patterns [18]. The difference between an ERNN and a JRNN is that the feedback in the Elman network goes from the output of the hidden layer to the context layer, whereas in the Jordan network, the feedback occurs from the output layer to the neurons of the context layer. The basic architectures of an ERNN and a JRNN are shown in Figure 4 and Figure 5, respectively.

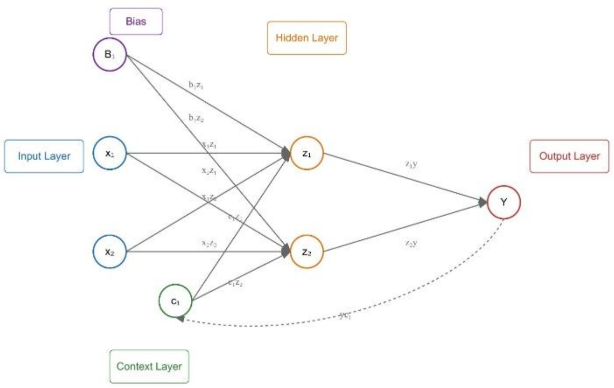

Figure 4: Basic architecture of an Elman RNN

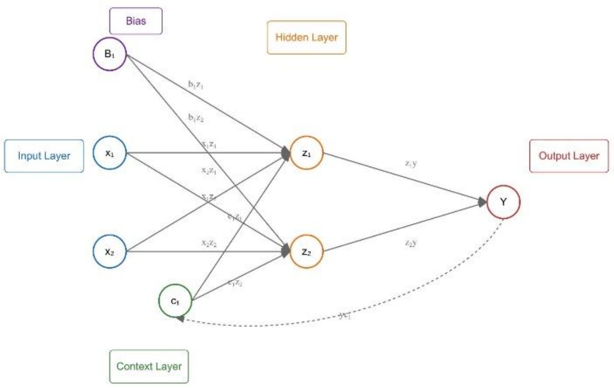

Figure 5: Basic architecture of a Jordan RNN

ERNNs have in their context cell the same number of neurons as the hidden layer, whereas JRNNs have in the context layer the same number of neurons as the output layer.

Due to the architecture of the Elman-type RNN, the computational time required for training is greater than that of a Jordan-type RNN. This is because the recurrence in the ERNN is taken from the outputs of the neurons in the hidden layers and not from the output layers as in a JRNN, where the number of neurons in the hidden layers was always greater than the two neurons in the output layer.

The number of inputs of the RNN varied between three and seven neurons; among which date (F), temperature (T), and solar radiation (I) were fixed variables in the analysis. Additionally, variables such as wind speed (V), relative humidity (H), precipitation (R), and atmospheric pressure (P) were tested in the prediction model for ambient temperature and solar radiation. The prediction of the data was made for two future days, that is, for the next 288 ten-minute intervals. This time frame provides a near-term horizon that is relevant for operational planning in solar energy systems, as it captures daily and potential seasonal variations without becoming overly complex or computationally intensive

Station OUA_003 presented anomalous readings in the relative humidity time series, so in this case, the variable H was not used.

The meteorological variables used for the analysis were scaled to values between 0 and 1, as shown in equation (4), due to the learning processes of the network. When the output values of the neural network are obtained, the data undergo the inverse normalization process.

$$x_{scaled} = \frac{x-\mu}{\sigma} \quad (4)$$

Where,

- x : Original value;
- μ : Mean of the data;
- σ : Standard deviation of the data.

For the ERNN models, experiments were conducted with either one or two hidden layers. In the one-hidden-layer configuration, variations with 4, 6, 8, 10, and 12 neurons were tested. For the two-hidden-layer configuration, various combinations were examined where each layer contained between 4 and 12 neurons. For the JRNN models, only a single hidden layer structure was used in all experiments, with the number of neurons varying between 4 and 12.

Additionally, three different learning algorithms were used: BP, BPM, and Rprop. The selection of these algorithms was based on their demonstrated effectiveness with modest-sized meteorological datasets and recurrent architectures [19]. While newer optimizers like Adam or RMSprop offer faster convergence, traditional BP-based methods perform competitively for time-series prediction when the network isn't excessively deep. Rprop was specifically included for its robustness to different input scales without extensive hyperparameter tuning, an important property for meteorological data with varying units.

The BP algorithm propagates the error signal backward, allowing the calculation of changes in weight values in previous layers based on minimizing the cost function, in this case through gradient descent of the error function. In BPM, a momentum term is introduced to reduce oscillations in the gradient descent. The difference between Rprop and backpropagation algorithms is that in Rprop, the derivative of the error function is used to determine the direction in which the weights should be corrected, not the magnitude of their change [20].

Each learning function has specific hyperparameters that were used within the intervals presented in Table 6.

Table 6: Hyperparameters used in the learning functions

Hyperparameters	**Learning Function**	**Values Tested**
Learning Rate (LR)	BP, BPM	[0.00001; 0.4]
Maximum Tolerated Error	BP, BPM	0
Momentum Term	BPM	0.1
Flat Spot Elimination	BPM	0.3
Initial Update Value of Weights	Rprop	0.1
Limit of Update Variation	Rprop	30
Weight Decay Exponent	Rprop	[0.00001; 4]

In BP and BPM, the learning rate values were varied, while in Rprop, the values of the weight decay exponent were varied; the other hyperparameters remained constant. In Table 7, the initialization parameters of the network are shown.

Table 7: Initialization parameters for the ERNN and JRNN networks

Parameter	**Value**
Initial weights of feedforward connections	[-0.5, 0.5]
Initial weights of connections to recurrent cells	0
Initial weights of connections from recurrent cells	0.5
Initial activation of context units	0.5

The feedforward connections were initialized with weights randomly distributed in the range [-0.5, 0.5], a approach that helps prevent initial saturations and provides a symmetric starting point for the neural network. While not employing specialized initialization techniques like Xavier or He methods, this uniform random initialization strategy ensures small, varied initial weights that can help break symmetry and facilitate initial learning across the network's neurons.

From the tests performed, twelve RNNs were selected, six of the Elman type and six of the Jordan type. The training parameters used for each network are shown in Table 8. The labels of the twelve RNNs, located in the first column, are written as follows: F. Learning_NetworkType_Station;

Table 8: Training parameters of the ERNN and JRNN

Learning_Network_Station	Neurons in Layers	Iterations	Inputs
BP_E1_001	(5, 4)	800	F, I, T, V, R, P
Rprop_E1_001	-6	450	F, I, T, V, P
BP_J1_001	-12	500	F, I, T, V, P
BPM_J1_001	-6	450	F, I, T, V, R, P
BPM_E1_002	-12	500	F, I, T, P
Rprop_E2_002	-12	250	F, I, T, V, P
BPM_J2_002	-11	500	F, I, T, P
Rprop_J1_002	-8	150	F, I, T
BP_E2_003	-11	500	F, I, T, P
BPM_E2_003	-12	400	F, I, T, V, R, P
BP_J2_003	-6	500	F, I, T
Rprop_J2_003	-8	300	F, I, T

The prediction, training, and validation process for an ERNN or JRNN is shown in Figure 6, Figure 7, and Figure 8, respectively.

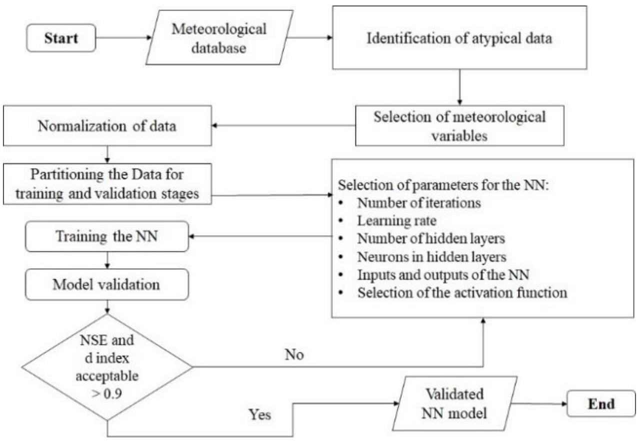

Figure 6: Flowchart of the application of RNN in prediction

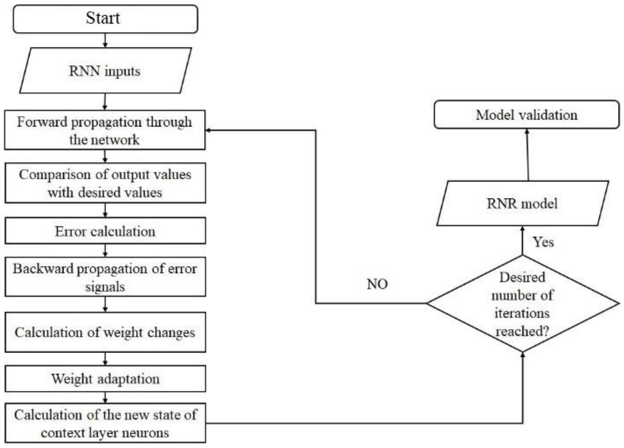

Figure 7: Flowchart of the training of the RNN

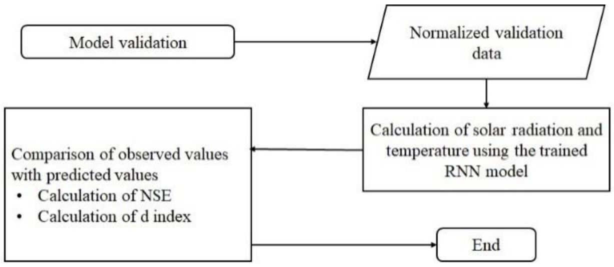

Figure 8: Diagram of the validation stage of the RNN

Once the parameters and hyperparameters of the RNN were defined, the network was trained for the desired number of iterations until obtaining the final weights for the model. During each iteration of the network, signals are propagated from the input layer to the hidden layers and then to the output layer. Afterwards, a synchronous update of the context units is performed. At the end of each iteration, the Sum of Squared Errors is calculated, which can be considered as the first performance indicator of the network before its validation stage.

With the synaptic weights established in the trained network model, the outputs of radiation and temperature are obtained using 20% of the data from each station for the validation stage. The output data, or predicted data, are compared with the theoretical outputs using the performance indicators described in the following subsection.

3.5 Criteria for selecting the prediction method using ANN

To evaluate the performance of the solar radiation and temperature prediction in this study for Ouarzazate, two indicators shown in equations (5) and (6) were used: the NSE [21] and the d index [22], comparing the measured values with the predicted values obtained by the model.

$$NSE = 1 - \frac{\sum_{i=1}^{N}(y_i - x_i)^2}{\sum_{i=1}^{N}(x_i - \bar{x}_i)^2} \tag{5}$$

$$d = 1 - \frac{\sum_{i=1}^{N}(y_i - x_i)^2}{\sum_{i=1}^{N}(|y_i - \bar{x}| + |x_i - \bar{x}|)^2} \tag{6}$$

Where x_i is the value measured by the sensor, y_i is the value predicted by the model, N is the number of data points, and $\bar{x}$ is the arithmetic mean of the measured values x.

These two indicators were chosen because they provide a comprehensive evaluation of model performance, especially in environmental and hydrological studies. The NSE assesses the predictive power of the model by comparing the magnitude of the residual variance (the noise) to the variance of the observed data (the information). An NSE value closer to 1 indicates a more accurate model. The d index is a standardized measure that reflects the degree to which the observed data are accurately estimated by the model. It varies between 0 (no agreement) and 1 (perfect agreement), providing a sensitive and reliable assessment of predictive accuracy. These indicators account for both systematic and random errors, making them suitable for evaluating the ANN models' ability to predict solar radiation and temperature in Ouarzazate.

4 Results analysis

Each of ERNN and JRNN models shown in Table 7 underwent a validation stage and a prediction stage. In both stages, the models were evaluated using NSE and the d index, comparing the measured values with the predicted values obtained by the model.

In Table 9, the NSE and d index values of the models evaluated during the validation stage are presented.

Table 9: Performance indicators during the validation stage for ERNN and JRNN

Validation stage				
Learning_Label_Station	**NSE Radiation**	**d Radiation**	**NSE Temperature**	**d Temperature**
BP_E1_OUA_001	0.970	0.980	0.980	0.985
Rprop_E1_OUA_001	0.972	0.982	0.982	0.987
BP_J1_OUA_001	0.963	0.973	0.978	0.983
BPM_J1_OUA_001	0.969	0.979	0.979	0.984
BPM_E1_OUA_002	0.968	0.978	0.977	0.982
Rprop_E2_OUA_002	0.978	0.988	0.983	0.988
BPM_J2_OUA_002	0.970	0.980	0.981	0.986
Rprop_J1_OUA_002	0.962	0.972	0.976	0.981
BP_E2_OUA_003	0.959	0.969	0.975	0.980
BPM_E2_OUA_003	0.948	0.958	0.970	0.975
BP_J2_OUA_003	0.950	0.960	0.971	0.976
Rprop_J2_OUA_003	0.922	0.932	0.960	0.965

The NSE values for radiation and temperature exceed 0.91, and the d index values are above 0.95, indicating strong model performance during the validation stage. The highest NSE and d index values for both solar radiation and ambient temperature were achieved by the model BPM_J1_OUA_001.

In Table 10, the NSE and d index values during the prediction stage for two subsequent days are presented.

Table 10: Performance indicators during the prediction stage for ERNN and JRNN

Prediction stage				
Learning_Label_Station	**NSE Radiation**	**d Radiation**	**NSE Temperature**	**d Temperature**
BP_E1_OUA_001	0.905	0.915	0.974	0.979
Rprop_E1_OUA_001	0.902	0.912	0.976	0.981
BP_J1_OUA_001	0.897	0.907	0.961	0.971
BPM_J1_OUA_001	0.909	0.919	0.971	0.976
BPM_E1_OUA_002	0.978	0.988	0.976	0.981
Rprop_E2_OUA_002	0.970	0.980	0.973	0.978
BPM_J2_OUA_002	0.958	0.968	0.970	0.975
Rprop_J1_OUA_002	0.934	0.944	0.917	0.927
BP_E2_OUA_003	0.935	0.945	0.930	0.940
BPM_E2_OUA_003	0.929	0.939	0.929	0.939
BP_J2_OUA_003	0.914	0.924	0.949	0.959
Rprop_J2_OUA_003	0.885	0.895	0.830	0.840

The highest NSE and d index values for stations OUA_001, OUA_002, and OUA_003 are achieved by models BPM_J1_OUA_001, BPM_E1_OUA_002, and BP_E2_OUA_003, respectively, as shown in Table 9. The selected architecture for prediction in the three meteorological stations depends on the behavior of the analyzed time series.

Overall, the highest NSE and d index values during the prediction stage for both solar radiation and ambient temperature were obtained by the model BPM_E1_OUA_002.

Figure 9 shows the prediction graph of solar radiation.

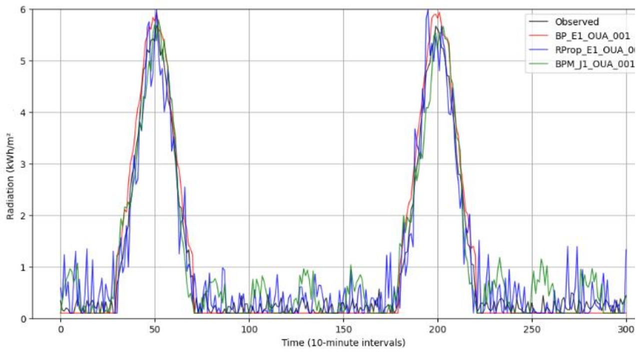

Figure 9: Solar radiation prediction (JRNN) for Station OUA_001

In the behavior of the solar radiation predictions, the values do not reach 0 as the minimum radiation value, which is adjusted based on the behavior of the time series (radiation = 0 during nighttime hours). A limitation is observed in the methods to capture the minimum and maximum of the observed values. Despite high NSE and d index values, the JRNN model shows some difficulty in accurately predicting the extreme values of solar radiation.

Figure 10 shows the ambient temperature prediction graph, both for station OUA_001.

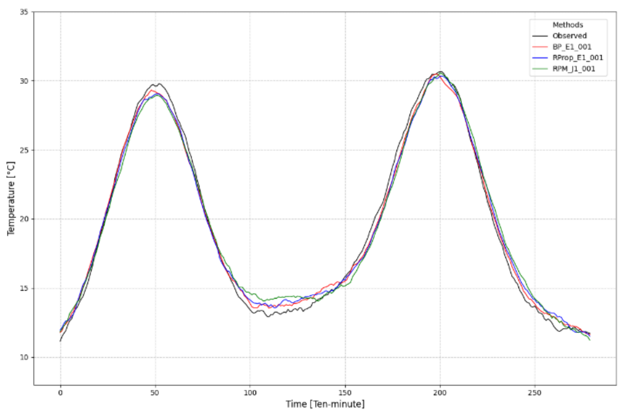

Figure 10: Prediction of ambient temperature (JRNN) for Station OUA_001

The behavior of ambient temperature predictions aligns closely with the measurements taken by the sensor. The model effectively captures the temperature trends, with high NSE and d index values indicating strong predictive performance.

In Figure 11, the solar radiation prediction graph is presented.

In the behavior of the solar radiation predictions, a similar pattern to Station OUA_001 is observed; the values do not reach the minimum of 0 in radiation nor the maximum observed values. This suggests that while the ERNN model captures the overall trend, there are limitations in predicting the extremes of the solar radiation data.

Figure 12 presented the ambient temperature prediction graph, both for Station OUA_002 using model BPM_E1_OUA_002.

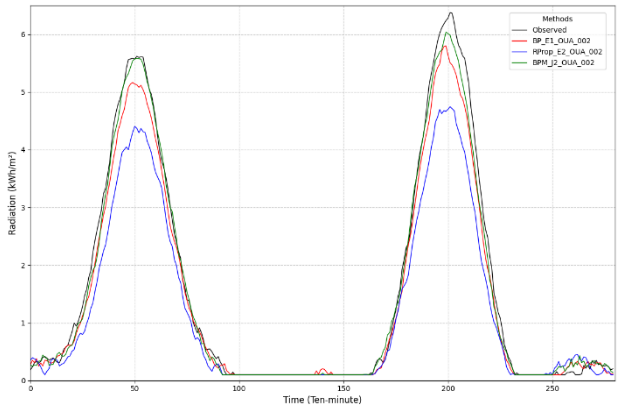

Figure 11: Prediction of solar radiation (ERNN) for Station OUA_002

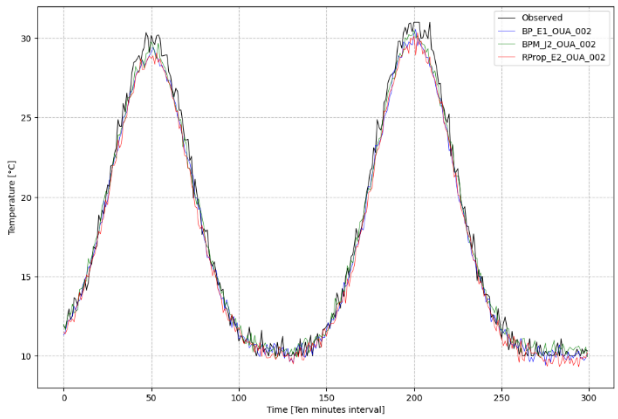

Figure 12: Prediction of ambient temperature (ERNN) for Station OUA_002

The behavior of ambient temperature predictions aligns with the measurements taken by the sensor; however, limitations are observed in reaching the maximum and minimum temperature values. The high NSE and d index values indicate good model performance, but the discrepancies at the extremes suggest areas for model improvement.

In table 11, the computational analysis is presented. It reveals consistent efficiency advantages for JRNN architectures, requiring 26-43% less training time than equivalent ERNN configurations while using 19-27% less memory.

Table 11: Computational efficiency comparison

Model	**Learning Algorithm**	**Avg Training Time (min)**	**Avg Iterations to Convergence**	**Peak Memory (GB)**
ERNN	BP	92 ± 7	580	3.8 ± 0.2
ERNN	BPM	78 ± 6	520	3.6 ± 0.3
ERNN	Rprop	51 ± 4	310	3.4 ± 0.2
JRNN	BP	68 ± 5	540	2.9 ± 0.2
JRNN	BPM	58 ± 4	490	2.7 ± 0.3
JRNN	Rprop	37 ± 3	280	2.5 ± 0.2

The experiment was conducted on hardware: Intel Core i7-8750H CPU, 32GB RAM.

5 Discussion

The performance variations between ERNN and JRNN models can be attributed to their distinct architectural designs and feedback mechanisms. ERNN models demonstrated superior performance at station OUA_002 (NSE: 0.978 for radiation, 0.976 for temperature) primarily due to their context layer receiving feedback directly from hidden layers, which enables them to better capture the complex non-linear relationships in meteorological data with high variability. This architectural advantage allows ERNNs to maintain a more comprehensive "memory" of hidden state representations across time steps, which is particularly beneficial for parameters with gradual temporal transitions like temperature.

Conversely, JRNNs exhibited better performance at station OUA_001 (NSE: 0.909 for radiation, 0.971 for temperature) when implemented with the BPM algorithm. This superiority can be explained by the JRNN's feedback mechanism originating from the output layer, making it more responsive to recent prediction errors and thus more suitable for datasets with distinct daily periodicity. Furthermore, the computational efficiency of JRNNs, requiring fewer neurons in the context layer than ERNNs, allowed for more extensive hyperparameter optimization during training, potentially contributing to their improved performance.

6 Conclusions and recommendations

The Elman and Jordan RNN models showed convergence with a number of iterations between 300 and 600. The computational time employed in training each model is directly related to the size of the dataset and the number of iterations during network training; as these variables increase, the computational time also increases. It was observed that if the iterations exceed 600, the results of the evaluation metrics vary by approximately ±2%.

The results in the validation stage for the three meteorological stations in Ouarzazate showed NSE values greater than 0.92 for both temperature and solar radiation using ERNN and JRNN models. This demonstrates that the training of the networks fits well with the behavior of the time series data.

For the prediction stage:

- At Station OUA_001, the best result was obtained with a JRNN using BPM learning;
- At Station OUA_002, the best result was achieved with an ERNN using BPM learning;
- At Station OUA_003, two similar results were obtained, one with a JRNN and the other with an ERNN, both using the BP learning algorithm.

In the solar radiation prediction graphs, positive radiation values are observed during nighttime hours. These values are not high but affect the expected behavior of the prediction. Therefore, it is important to perform a prior adjustment or correction before demonstrating the results. This is proposed when implementing the methods in a solar resource monitoring situation to ensure the accuracy and reliability of the predictions.

The highest NSE and index of agreement values under the training parameters of the ERNN and JRNN in the validation stage resulted in the best performance in the prediction stage. This allows us to affirm that the parameterization with the best results in the validation stage should be selected for the predictive model. Selecting the optimal hyperparameters during validation is crucial for improving model performance during prediction.

The findings of this study contribute directly to solar energy planning in Ouarzazate and similar regions by providing a 48-hour prediction of solar resources. Grid operators can select RNN architectures based on local conditions (JRNN or ERNN) to anticipate resource variability, storage management, and grid stability during Morocco's renewable energy transition.

References

[1] Sudharshan, K., Naveen, C., Vishnuram, P., Krishna Rao Kasagani, D.V.S., & Nastasi, B. 2022. Systematic review on impact of different irradiance forecasting techniques for solar energy prediction. Energies, 15(17), 6267. https://doi.org/10.3390/en15176267

[2] Bosak, A., Matushkin, D., Davydenko, L., Kulakovskyi, L., & Bronytskyi, V. 2022. Short-term forecasting of photovoltaic solar power generation based on time series: Application for ensuring the efficient operation of the integrated energy system of Ukraine. In Power Systems Research and Operation: Selected Problems II (pp. 159-179). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-17554-1_8

[3] Prema, V., & Rao, K. U. 2015. Development of statistical time series models for solar power prediction. Renewable energy, 83, 100-109. https://doi.org/10.1016/j.renene.2015.03.038

[4] Atique, S., Noureen, S., Roy, V., Bayne, S., & Macfie, J. (2020, April). Time series forecasting of total daily solar energy generation: A comparative analysis between ARIMA and machine learning techniques. In 2020 IEEE green technologies conference (GreenTech) (pp. 175-180). IEEE. https://doi.org/10.1109/GreenTech46478.2020.9289796

[5] Pang, Z., Niu, F., & O'Neill, Z. 2020. Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons. Renewable Energy, 156, 279-289. https://doi.org/10.1016/j.renene.2020.04.042

[6] Kumari, S., Aravindakshan, S., & Srinivasa Chakravarthy, V. 2021. Elman and jordan recurrence in convolutional neural networks using attention window. In International Conference on Innovative Computing and Communications: Proceedings of

ICICC 2020, Volume 1 (pp. 983-993). Springer Singapore.

[7] KROGH, Anders. What are artificial neural networks? Nature biotechnology, 2008, vol. 26, no 2, p. 195-197. https://doi.org/10.1038/nbt1386

[8] Kaur, M., & Mohta, A. 2019. A review of deep learning with recurrent neural network. In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 460-465). IEEE.https://doi.org/10.1109/ICSSIT46314.2019.8987837

[9] Gao, Y., Hu, Z., Shi, S., Chen, W. A., & Liu, M. 2024. Adversarial discriminative domain adaptation for solar radiation prediction: A cross-regional study for zero-label transfer learning in Japan. Applied Energy, 359, 122685. https://doi.org/10.1016/j.apenergy.2024.122685

[10] Sevas, M. S., Sharmin, N., Santona, C. F. T., & Sagor, S. R. 2024. Advanced ensemble machine-learning and explainable ai with hybridized clustering for solar irradiation prediction in Bangladesh. Theoretical and Applied Climatology, 155(7), 5695-5725. https://doi.org/10.1007/s00704-024-04951-5

[11] Tercha, W., Tadjer, S. A., Chekired, F., & Canale, L. 2024. Machine learning-based forecasting of temperature and solar irradiance for photovoltaic systems. Energies, 17(5), 1124. https://doi.org/10.3390/en17051124

[12] [Fan, C., Zou, B., Li, J., Wang, M., Liao, Y., & Zhou, X. 2024. Exploring the relationship between air temperature and urban morphology factors using machine learning under local climate zones. Case Studies in Thermal Engineering, 55, 104151. https://doi.org/10.1016/j.csite.2024.104151

[13] Didi, F., Bibi-Triki, N., Mohamed, B., & Abène, A. 2017. Optimization, Modeling and Simulation of Microclimate and Eenergy Management of the Greenhouse by Modeling the Associated Heating and Cooling Systems and Implemented by a Fuzzy Logic Controller using Artificial Intelligence. Informatica, 41(3).

[14] Gaafar, A. S., Dahr, J. M., & Hamoud, A. K. 2025. Optimizing Fuzzy Logic Control-Based Weather Forecasting through Optimal Antecedent Selection Using the Fuzzy Analytical Hierarchy Process Model. Informatica, 49(12).

[15] Bengio, Y., Goodfellow, I., & Courville, A. 2017. Deep learning (Vol. 1). Cambridge, MA, USA: MIT press. https://doi.org/10.4258/hir.2016.22.4.351

[16] Krenker, A., Bešter, J., & Kos, A. 2011. Introduction to the artificial neural networks. Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech, 1-18. https://doi.org/10.5772/15751

[17] Tufail, S., Riggs, H., Tariq, M., & Sarwat, A.I. 2023. Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. Electronics, 12(8), 1789. https://doi.org/10.3390/electronics12081789

[18] Krauss, P. 2024. Recurrent neural networks. In Artificial Intelligence and Brain Research: Neural Networks, Deep Learning and the Future of Cognition (pp. 131-137). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-030-91234-0_15

[19] Putro, S.S., Syakur, M.A., Rochman, E.M.S., & Rachmad, A. 2022. Comparison of backpropagation and ERNN methods in predicting corn production. Communications in Mathematical Biology and Neuroscience, 2022, Article-ID. https://doi.org/10.28919/cmbn/7562

[20] Zaras, A., Passalis, N., & Tefas, A. 2022. Neural networks and backpropagation. In Deep Learning for Robot Perception and Cognition (pp. 17-34). Academic Press. https://doi.org/10.1016/B978-0-12-823600-6.00002-1

[21] Şen, Z., Şişman, E., & Kızılöz, B. 2022. A new innovative method for model efficiency performance. Water Supply, 22(1), 589-601. https://doi.org/10.2166/ws.2021.368

[22] Pereira, H.R., Meschiatti, M.C., Pires, R.C.D.M., & Blain, G.C. 2018. On the performance of three indices of agreement: An easy-to-use R-code for calculating the Willmott indices. Bragantia, 77(2), 394-403. https://doi.org/10.1590/1678-4499.201