Deep Learning Using Genetic Algorithm Optimization for Short Term Solar Irradiance Forecasting

Wadie Bendali
Inovative Technologies Laboratory
Univérsité Sidi Mohamed Ben
Abdellah Fez, Morocco
wadiebendali20@gmail.com

Ikram Saber
PERE Laboratory
Univérsité Sidi Mohamed Ben
Abdellah Fez, Morocco
ikram.saber@usmba.ac.ma

Bensalem Bourachdi
2EMI 2EMI Team,L2MC laboratory
ENSAM,Moulay Ismail University
Mekenes, Morocco
bourachdi.bensalem@gmail.com

Mohammed Boussetta
Inovative Technologies Laboratory
Univérsité Sidi Mohamed Ben
Abdellah Fez, Morocco
https://orcid.org/0000-0002-0379-4454

Youssef Mourad
Inovative Technologies Laboratory
Univérsité Sidi Mohamed Ben
Abdellah Fez, Morocco

***Abstract*— The increase in the use of renewable energy sources (RES) has been remarkable in recent years, especially photovoltaic energy, which is one of the most widely used renewable energy sources for electricity generation. In fact, the world has known the installation of a large number of autonomous or grid-connected photovoltaic systems. However the problem with the introduction of PV is the improvised nature of solar energy that can influence the stability of electricity grids and reliability of the grid, Accurate solar forecasting makes it easier to integrate solar generation into the grid by reducing the integration and operating costs associated with intermittent solar power.**

In this context, the objective of this work is to improve the development of appropriate forecasting models for the prediction of photovoltaic energy production. For that reason, this article presents new hybrid methods to optimize deep learning forecasting by using genetic algorithm based Deep Neural Network. The model is employed to forecast time series of solar irradiation output. Comparisons are made between the performances of three types of neural networks: long short-term memory *(*LSTM), Gate recurrent unit (GRU), and Recurrent Neural Network (RNN). GA is exploited to find the most appropriate number of window size, and number of units (neurons) in each and all three hidden layers.

Keywords— solar irradiation, *Recurrent Neural Network, Long short-term memory*, deep learning, *Gate recurrent unit,* genetic algorithm.

Introduction:

The production of solar energy has increased today significantly in comparison with other types of renewable energy. It is used to replace the traditional form of generation, such as coal-based electricity generation, etc. On the other hand, the production of solar energy is not reliable due to natural variability. This can degrade the reliability and stability of the power system [1]. The integration of such new forms of power plants into the power grid represents one of the significant challenges for the industry today. Without proper management of this volatile energy, there will be problems with power reserve and frequency regulation [29]. In order to deal with those problems , sophisticated forecasting methods has been made , in raison to enable us to prepare sufficient energy and keeping up with demand at the same time, and also allowing the system stability of the frequencies. Forecasting also concern power plant operators, the energy trading market and grid managers. By knowing the future development of power production on the grid, both technical and financial uncertainties are minimized for all market parties.

In the last few years, a wide range of research around the world has been conducted on solar power forecasting. Generally, three types of methods for predicting exist: physical, statistical and artificial intelligence methods [2]. Artificial Intelligent prediction models are capable of finding complex relationships between variables in the forecast without using complex mathematics, so they are favored over the other models mentioned above [3]. Machine learning methods as intelligent methods have a good result in terms of time series forecasting , by using historical data through training model [4]. Among different methods of machine learning using to forecast time series, there are artificial neural network (ANN), which have a good performance. This method has an effective use due to the limitation of statistical techniques in dealing with non-linear data as solar energy. ANN has different architectures, that includes Multilayer Feed Forward NN (MLFFNN) [5], Multilayer Perceptron NN (MLPNN) [6], Radial Base Function NN (RBFNN) [8], Recurrent NN (RNN) [5], Adaptive Neuro-Fuzzy Inference System (ANFIS) [5, 7].

Achieving accurate forecasts depends on many factors, such as choosing the right architecture to handle dynamic data modeling, selecting the right algorithm, selecting the input variables and adjusting the model's hypermeters, etc. Moreover, the recurrent neural network allows us to obtain all these factors, thanks to its different structures of units and hyper-parameters. However, this kind of deep learning architecture is developed by specialist using trial and error, which is very time-consuming to obtain accurate models. Today, the performance of graphics processing units (GPUs) has considerably increased, leading to the extended use of reinforcement learning and scalable algorithms to enable the automation of the search for the best model structure. As a result, coupling deep neural network model with some optimization algorithms is a one way to avoid trial and error procedure and possibly obtain a big increase in the performance of model. Genetic algorithms has ability to select model input. It is also choose best architecture by finding appropriate number of deep neural network unites (neuron) and hidden layers. In addition it can select best weight values between different unites to obtain good accuracy and performance. [9] - [10] .

978-1-7281-8084-7/20/$31.00 ©2020 IEEE

In order to optimize forecasting model accuracy performance, this paper aim to combine deep learning units and genetic algorithm by using three architecture of RNN, which is simple RNN, GRU and LSTM as deep learning models, and GA as optimization model, that can select the appropriate input windows size and most suitable number of unites in each layer (as shown in Figure 6) . The proposed method will based on hourly solar irradiation data of Fes Moroccan city.

The body of this article is organized as follows: after the introduction, Section 1 explains Data Selection and preprocessing. Section 2, describing of genetic algorithm and his architecture. Section 3, we highlight different deep neural network architectures we used and their parameters. Section 4, we represent experiment and results .Finally, discussion of the results and conclusion.

I. DATA AND PREPROCESSING

A. TIME SERIES

A time series is defined as a series of measurements or observations over time representing a phenomenon [11, 12, and 13]. In this study, the observation is global solar irradiation of Fes Moroccan city .In theory, two important assumptions are necessary for a more accurate analysis:

- The time interval of the observations is constant,
- The method of data preprocessing shall be the same during the observation/measurement time.

In order to use the formalism of time series, it is necessary to establish certain definitions in advance. Therefore, the current value in t of the time series X is noted X_n where t is the time between 1 and n, n being the total number of measurements. The number of values to be predicted from the time series is noted k. Forecasting of the time series from (n+1) to (n+k), having the history from X_1 to X_n, is called the horizon of the prediction (horizon 1... horizon k). By taking horizon 1, the general model of the forecast can be shown in Equation 1.

$$X_{t+1} = F_n(X_t, X_{t-1}, \dots, X_{t-p+1}) + \varepsilon(t+1) \qquad (1)$$

Where ε is the error between predicted value and measured value, F_n is the mathematical model to be estimated and t is the time parameter which takes the (n-p) following values: n, n-1... p+1, p; p being the number of samples considered by the model, under the assumption that: n >> p.

B. DATA DESCRIPTION

In this study, time series of Global Horizontal Irradiance (GHI) of Fes (33,3 °N , -5,0 °E, Altitude = 579 m) , from 2016 to 2019 using as training and testing data .These GHI data were acquired using good quality from satellite data using metronome 7 platform which allows access to historical time series of irradiation and temperature. Fes has a complex climate with strong volatility due to her geographical position.

The high environmental volatility would cause the solar radiation sequence to be non-stationary and, ultimately, would make it difficult to establish an accurate forecast model. Consequently, the 4 years data of solar radiation sequence have used to verify the performance of the proposed method.

Figure 1 shows the Global Horizontal Irradiance (GHI) series of hourly values, with a total of 8760 sample data. There are 24 sampling points per day and each sample data represents a mean value of the 1-hour of GHI. Fig 1,2,3 and 4 illustrate that the seasonal factor has a strong impact on GHI. In order to minimize the impact of the seasonal factor on the model's forecasting performance, data are divided into 4 data sets (Figure 1,2,3 and 4). (Winter from December 21 to March 19, spring from March 20 to July 19, summer from July 20 to September 21 and autumn from September 22 to December 20).

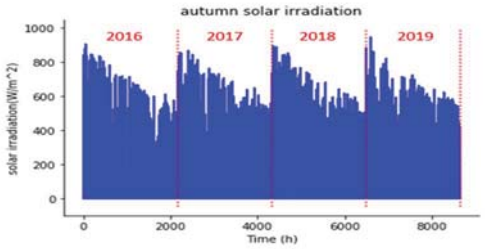

Fig.1 Time series data of hourly GHI for autumn collected on Fes

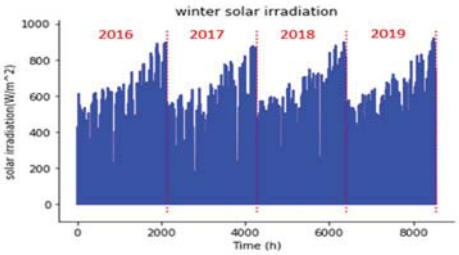

Fig.2 Time series data of hourly GHI for winter collected on Fes

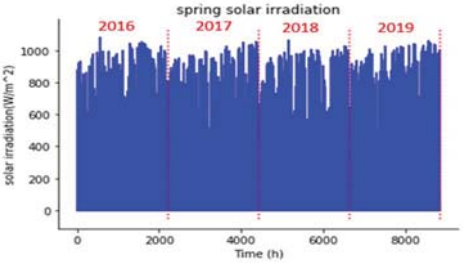

Fig.3 Time series data of hourly GHI for spring collected on Fes

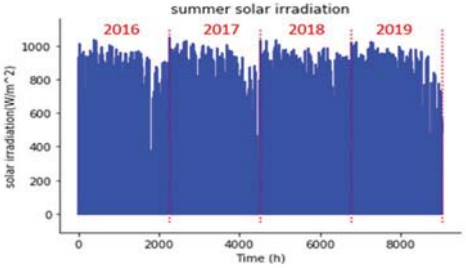

Fig.4 Time series data of hourly GHI for summer collected on Fes

C. DATA PREPROCESSING

The pre-processing of input data can significantly reduce the problem of inadequate training and the cost of calculation by learning the historical pattern correctly. As a result, the accuracy of the forecast model can be considerably improved by pre-processing the input data.in this case, a number of methods have been employed to pre-process the input data for the forecast models. Stationary, historical lag identification [14],trend-free time series [17], normalization [15], wavelet transform (WT) [16], and self-organizing map (SOM) [18] are useful methods to pre-process the input data. The most usable method is normalization ,which is widely used for pre-processing the input data of the forecasting solar irradiation model. In this article, the normalization technique has been choose to be used. In this method, a smaller range of input data is treated from a larger range of data. Thus, it is possible to limit the data to a range between 0 and 1 to minimize regression error, improve accuracy, and maintain correlation between data sets.

$$I_{norm} = \frac{I_{now} - I_{min}}{I_{max} - I_{min}} \qquad (2)$$

Where I_{norm} is the normalized data; I_{now} is the original input data; I_{min} and I_{max} are the minimum and maximum values of the input data, respectively. When the predicted values are obtained, post-processing is necessary upstream or downstream of the analysis for the performance of the forecast model. In this case post-processing of the normalized values must be anti-normalizing to extract the expected actual solar irradiation and analyze the model performance.

II. GENETIQUE ALGORITHME

Nature has always been a great source of inspiration for all humankind. Genetic Algorithms (GA) are search algorithms based on the concepts of natural selection and genetics. GAs are a subset of a much larger branch of computing known as evolutionary computing. It is frequently used to find optimal or near-optimal solutions to difficult problems that would otherwise take a lifetime to solve. It is frequently used to solve optimization problems, in research and in machine learning [19]. The GA process is divided into two parts:

- identify the appropriate chromosomal encoding to be used as solutions or parameters to be tested,
- determine the fitness function to be used to test the solutions obtained from the GA to determine if they are suitable to be used as next-generation solutions,

Genetic algorithms are carried out in different steps as shown in Fig. 5 above.

Firstly, it is important to initialize a population, which is a group of individuals who can solve the problem in question. An individual is characterized by a set of factors called genes and when these genes are linked together, they are called a chromosome and a group of chromosomes makes a population. After selection of population, test the solutions got from GA whether it is appropriate to be used as a next-generation solutions or not by using fitness functions. In this study, Fitness Function is Mean Square Error (MSE).

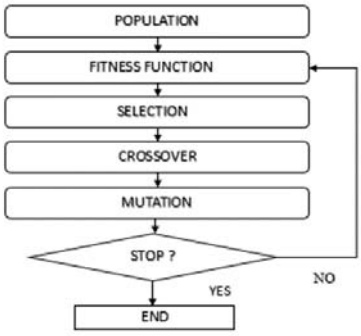

Fig. 5: Flowchart of genetic algorithm.

Crossover create a new solution from the existing solutions based on the process of mating between the two-selected individual, that makes it possible to obtain a lot of solutions, with reason to choose the best result. Finally, mutation: A method allows the variety of solutions to be increased after the crossing process has been completed by creating new solutions; this introduces some unpredictability into the algorithm and ensures that the algorithm is not stuck in an endless loop of crossovers and selections.

Table. 1 Parameter of Genetic algorithm

Generation number	Genes	Population number
4	22	4

III. DEEP LEARNING MODELS

A. Deep neural network

Deep neural network (DNN) is an artificial neural network, having several hidden layers between the input and output layers. The DNN is capable to find the correct mathematical manipulation to transform the input into the output by using forward and backward propagation, whether it is a linear or non-linear relationship [20]. It is applied in diverse applications such as image recognition and time series. Training of neural networks, it essentially means minimizing a loss function. The value of this loss function provides a measure of the gap between the performance of our network and perfection for given set of data. In order to minimize loss function it is necessary to use gradient descent, which is an optimization algorithm used to minimize certain functions by iteratively running in the direction of the steepest descent as defined by the negative of the gradient. In deep learning, we use gradient descent to update the parameters of model. The parameters refer to neural network weights. Deep neural network having also many hyper-parameters that can optimize forecasting performance:

- Learning rate: The amount that the weights are updated during training is referred to as the step size or the learning rate. It is a highly configurable hyperparameter

used in neural network training that has a small positive value, often between 0 and 1.

- Mini-Bache size: When a gradient descent is performed with a loss function that is created by summing all individual losses, the gradient of the individual losses can be calculated in parallel, whereas it must be calculated sequentially step by step in the case of a stochastic gradient descent. Rather than using the entire data set or a single example to construct our loss function, we use fixed number of examples (for example 16, 32 or 128), to form what is called a mini-batch. That make gradient descent to achieve sufficient stochasticity to avoid local minima, as well as making sufficient use of the computing speed of parallel processing.

- Dropout rate: Generally, Dropout is a regularization technique for dealing with neural network model overfitting problem, by selecting randomly some neurons and ignoring them during training. In other words, their contribution to the activation of downstream neurons is temporarily suppressed during the forward propagation and no weight update is applied to the neuron during the backward propagation. Therefore, dropout rate is the probability of ignoring neuron during each weight update cycle.

In support of improvement of model, we will add ADAM optimization, which is an adaptive learning rate methods, that means it calculates individual learning rates for different parameters. Its name is derived from the adaptive moment estimate, and the reason it is so called is that Adam uses estimates of the first and second gradient moments to adapt the learning rate for each neural network weight.

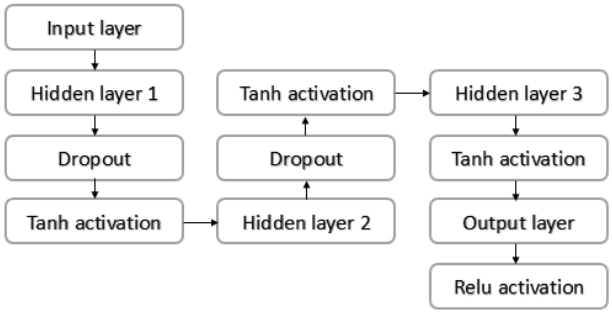

Fig 6: Proposed deep neural network architecture

TABLE.2 HYPERPARAMETRES OF PROPOSED DEEP NEURAL NETWORK

Learning rate	Mini-Bach size	Dropout rate	Out-put activation
0,2	128	0,01	Relu

However, the deep neural network model has limitations. Due to the unavailability of a memory unit, it cannot memorize the previous state to model the data sequence. This restriction is a major problem for sequence data such as text and time series.

B. Recurrent neural network (RNN)

Recurrent Neural Networks, or RNNs, is a type of deep learning unites; it is designed to work with sequence prediction problems. Due to their internal hidden states, RNN unites have ability to memorize information from the previous hidden state, and use it to calculate the current hidden state, that is also used to calculate the next hidden state. From equation 3 and 4, RNN layers contain a hidden state $h_t$and an out-put o_t, it is described as follow in Fig 7:

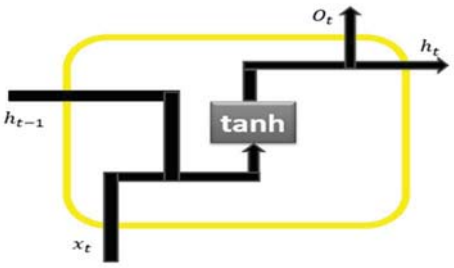

Fig. 7 RNN unit architecture

$$h_t = tanh(ux_t + wh_{t-1}) \quad (3)$$
$$o_t = Sig\ (vh_t) \quad (4)$$

With u is the input weight matrix, and w is the previous hidden state weight matrix, and v the output weight. Several works observed that it is two problems can perturbs RNN training [21]:

- Vanishing gradient means that the Gradients of the loss function approach zero, making the network difficult to train as additional layers using certain activation functions are added to neural networks.
- Exploding gradients which is opposite of vanishing, it refer to the problem that the gradients get too large in training, making the model unstable.

In fact, it is easy to deal with exploding gradients; by using Gradient clipping [26]. However, it is difficult to stop vanishing problem just with the use of a simple method. To solve this problem, it is necessary to use another unit that it can face against this kind of problem. In this case, the best solution is to use LSTM or GRU unites which are extension of RNN unit perhaps simple RNN unites [22].

C. Long short term memory (LSTM)

LSTM is specifically designed to avoid the problem of long-term dependency. Remembering information for long periods is practically their behavior. LSTM uses the hidden state and the cell state to save the data and pass it on to the process in the next period during which the different gates assess how much data will be saved in the hidden state and in the cell state. Those gates represent an input gate, an output gate and a forget gate as described in Fig 8. LSTM remember

their inputs over a long period. This is because LSTM contain their information in a memory, by using these three gates. LSTM can read, write and delete information from its memory. As a result, long-distance information transmission can be easily achieved by detecting the property of an input sequence in an advanced state, thus capturing the long distance dependency. This helps reduce the Instance of Vanishing Gradient.

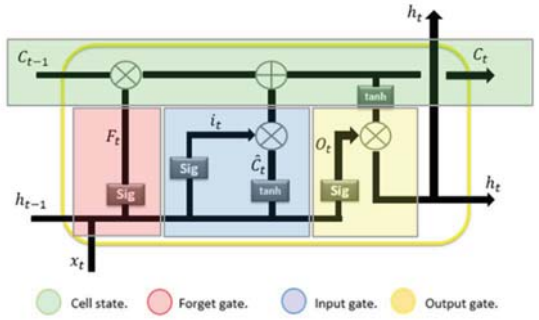

Fig. 8: LSTM unit architecture

As a first step in processing LSTM, a decision is made on what information can be deleted from the state of the cell; this is made by using the input gate, shown in equation (5). Next, choose the information that must be forgotten in the previous state using forget gate, according to the following equation (6). The output port in the final state determines the internal state information that must be transmitted, using equation (7).

$$i_t = Sig(h_{t-1}w^i + x_t u^i) \qquad (5)$$

$$f_t = Sig(h_{t-1}w^f + x_t u^f) \qquad (6)$$

$$o_t = Sig(h_{t-1}w^o + x_t u^o) \qquad (7)$$

Memorization is based on the cell state or intern memory, which has two steps to update, the first step, the sigmoid of input gate help to know which values can be updated, then, tanh equation generates a new candidate value vector $\tilde{c}_t$, eventually added to the state. In the second step, it links the input layer with the vector of new values to make an update of the state, according to the following equations.

$$\tilde{c}_t = tanh(h_{t-1}w^g + x_t u^g) \qquad (8)$$

$$c_t = \tilde{c}_t \times i_t + c_{t-1} \times f_t \qquad (9)$$

With w and u are the weight matrices of the hidden state and input respectively. Following the update of the cell state c_t , it enabled to get the hidden state of the next step by using this equation.

$$h_t = tanh(c_t) \times o_t \qquad (10)$$

D. Gate recurrent unit (GRU)

The gated recurring unit has two main gates; these gates are the reset and update gates respectively as shown in Fig 9. They have a special characteristic of adapting the information flow through the unit without requiring separate memory cells. These two equations represent the rest and the update gate.

$$r_t = Sig(h_{t-1}w^r + x_t u^r) \qquad (11)$$

$$z_t = Sig(h_{t-1}w^z + x_t u^z) \qquad (12)$$

Updating GRU's hidden state required two steps, at the beginning it is necessary to calculate $\widetilde{h_t}$, which represents the output of the hidden state of the reset gate. The following step h_t is obtained by using the output of the update gate zt. These steps represent as follows:

$$\widetilde{h_t} = tan\left((h_{t-1} \times r_t)w^h + x_t u^h\right) \qquad (13)$$

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

There is a similarity between the GRU and LSTM with regard to the memorization of information and the long distance transport of this information, which helps to solve the problem of long-term dependency. The GRU has a special feature, namely low complexity and high efficiency.

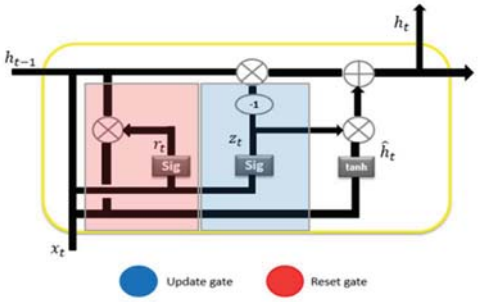

Fig. 9: GRU unit architecture

The advantage of GRU over LSTM is as follows:

- The calculation of GRU is more powerful than that of LSTM because it does not require an internal memory unit, which makes the structure less complicated.
- GRU can be trained faster than LSTM. It also needs less data for generalization because there are fewer parameters (u and w are smaller), helping the calculation as there are only two gates.
- GRU exposes the complete memory contrary to LSTM.

The advantage of LSTM over GRU is as follows:

- If the work needs a model that can treat Long-distance relationship sequence data (more feedback on data), LSTM is better than GRU because LSTM is built with the capability to remember sequence data from long distance relationships.

IV. EXPERIMENTS AND RESULTS

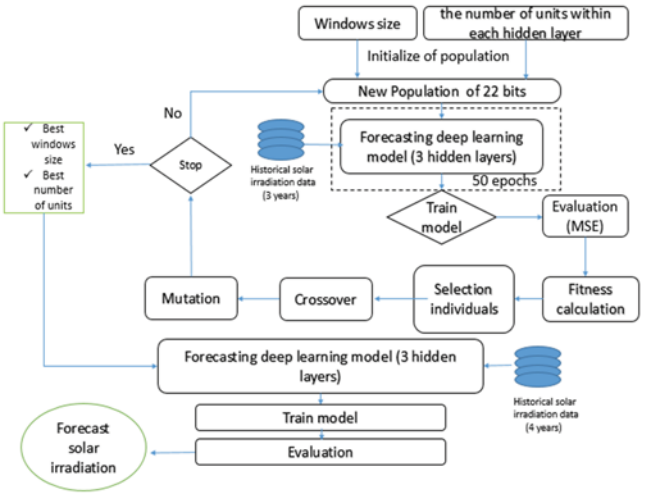

Fig. 10: Flowchart of proposed method

In this work, we create our model with Python environment by using KERAS as a deep learning library, which supports CPU and GPU [23]. In order to facilitate population generation, crossing and mutation operations, we will use distributed evolutionary Python (DEAP) library [24]. To build our model we divide the data into four datasets, each of them represent one season in four years.

To obtain the window size and the number of units in each hidden layer, the solution of GA is decoded as a binary array into decimal form. after that we Prepare and normalize the dataset depending upon the window size obtained from GA, in this step we split 3 years of a season data (2016,2017,2018) into 67 % (2 years) for training and 33 %(1 years) for validation . We obtain Fitness function (MSE) of GA solutions in a current generation, by training deep learning model (fig.10).

The architecture of the model consists of three hidden layers that contain a number of units (Fig.6), chosen by GA. The input layer has a shape that depends on the size of the widows, which be chosen by GA as well, hidden layers 1 and 2 have been dropped with a dropout rate of 20%, and activation function of each hidden layers is Tanh with Rectified Linear Unit (ReLU) as activation function of the output layer . The model is trained during 50 iterations, which are the number of times the training algorithm will go through all samples using Bach gradient descent with mini-Bache size of 128 samples (Tab.1). In order to optimize training, we use ADAM optimization with a learning rate of 1%. in terms of GA , We use a random Bernoulli distribution to initialize the initial values of a 22 bit binary array, which respectively represent 4 bits for the number of windows and 6 bits for the number of units in each hidden layer (3 layers × 6 bits + 4 bits = 22 bits). Bernoulli distribution also used for random initialization of crossover, mutation, and selection. Regarding the parameters of GA, we use four population, four generations, and twenty-two genes (Tab.1), the choice of this parameter depend the nature of the problem. Finally, the structured model that provides the lowest MSE value will be used in the training set and Testing set, by 4 years of a season data (2016, 2017, 2018, 2019). This data are split into 75% (3 years) for training and 25% (1 years) for testing. In order to evaluate the models we propose, we have compared them to the DNN models without GA , from which we have chosen windows of size 1 and 60 units in each hidden layer.

Performance index to evaluate the results of this study is Mean Square Error (MSE) and Mean Absolute Error (MAE) with the following equation (12) [25]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(G_{forecast}(t) - G_{true}(t))^2 \quad (15)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|G_{forecast}(t) - G_{true}(t)\right| \quad (16)$$

With N is the number of samples, $G_{forecast}(t)$ is the forecast solar irradiation at time t, and $G_{true}(t)$ is the true solar irradiation at time t.

Tab. 3 Models MSE and MAE of all season

		MSE	MAE
Autumn	RNN	0.0087	0.0599
	GRU	0.0083	0.0577
	LSTM	0.0081	0.055
	RNN-GA	0.0044	0.0492
	GRU-GA	0.0018	0.022
	LSTM-GA	0.0019	0.0215
Winter	RNN	0.00915	0.0625
	GRU	0.00911	0,0622
	LSTM	0.00908	0.061
	RNN-GA	0.00341	0.0345
	GRU-GA	0.00293	0.0311
	LSTM-GA	0.00301	0.0375
Spring	RNN	0.0119	0.075
	GRU	0.0105	0.069
	LSTM	0.0109	0.073
	RNN –GA	0.0033	0.0315
	GRU-GA	0.00306	0.028
	LSTM-GA	0.00322	0.031
Summer	RNN	0.01015	0.0763
	GRU	0.0101	0.074
	LSTM	0.0098	0.073
	RNN-GA	0.0022	0.037
	GRU-GA	0.0017	0.029
	LSTM -GA	0.0015	0.027

Tab. 4 Best windows size and bests number of units in each hidden layers chosen by GA for 3 models

		Windows size	H1	H2	H3
Autumn	RNN-GA	2	40	20	61
	GRU-GA	14	24	34	43
	LSTM-GA	12	30	8	7
Winter	RNN-GA	12	29	7	49
	GRU-GA	5	62	37	6
	LSTM-GA	15	29	54	46
Spring	RNN-GA	15	31	52	36
	GRU-GA	3	33	44	61
	LSTM-GA	11	47	45	29
Summer	RNN-GA	8	51	8	29
	GRU-GA	2	45	8	4
	LSTM-GA	4	30	26	11

Tab. 5 Time training of each DNN model with and without GA

Models	RNN	LSTM	GRU
Time training(s)	29	58	57
Models	RNN-GA	LSTM-GA	GRU-GA
Time training(s)	226	603	540

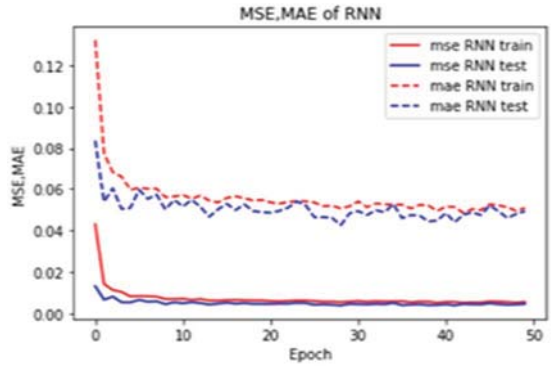

Fig. 11 MSE and MAE of RNN-GA in autumn

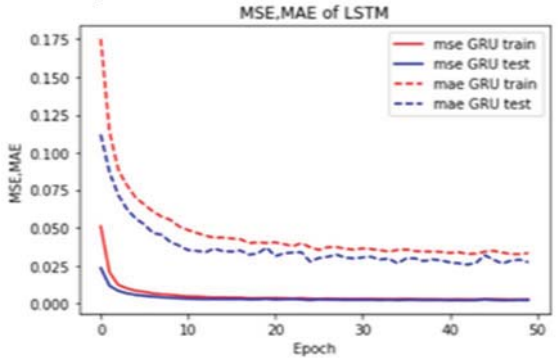

Fig. 12 MSE and MAE of LSTM-GA in autumn

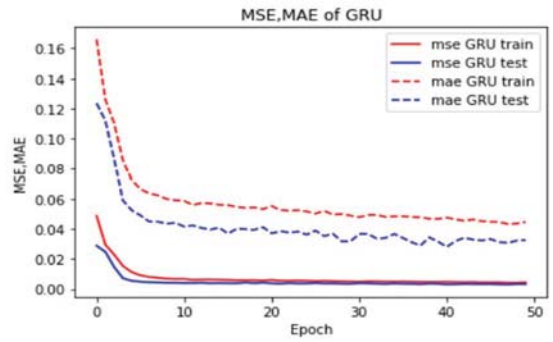

Fig. 13 MSE and MAE of GRU-GA in autumn

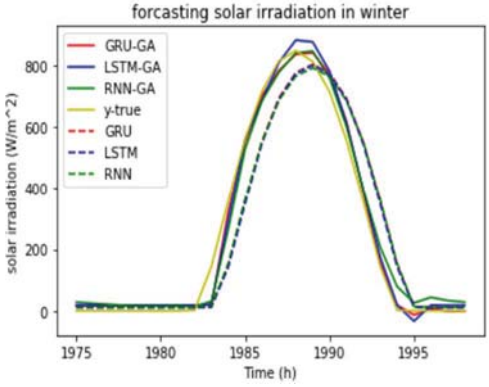

Fig. 14 Comparison of true solar irradiation with the three models with and without GA in winter

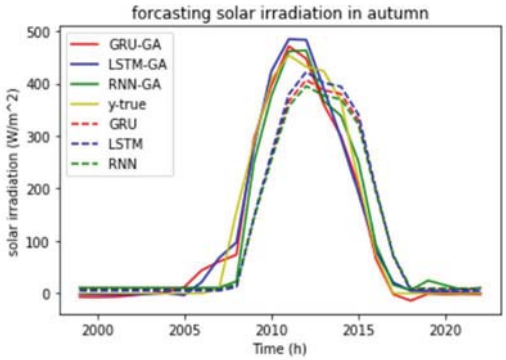

Fig. 16 Comparison of true solar irradiation with the three models in autumn

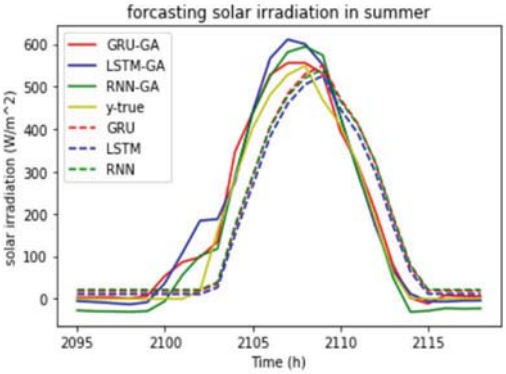

Fig. 17 Comparison of true solar irradiation with the three models with and without GA in summer

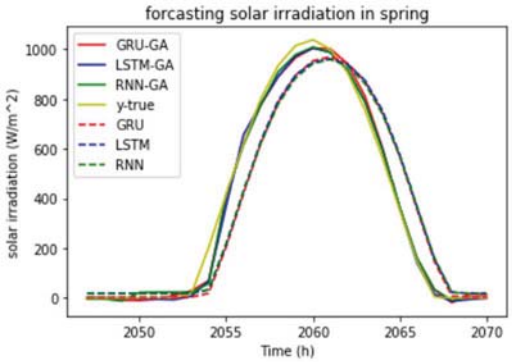

Fig. 15 Comparison of true solar irradiation with the three models with and without GA in spring

V. Discussion and Conclusion

To verify the accuracy and feasibility of the propose method, simulation experiment are evaluated under different solar irradiation data. As seen in Table 3, in summer, the MSE of LSTM-GA and GRU-GA is 0.0015 and 0;0017 respectively and 0;0022 for simple RNN-GA. The MAE of LSTM-GA and GRU-GA for are 0;027 and 0;029 respectively ,which are suitable compared with simple RNN-GA that has 0;037. The LSTM-GA and GRU-GA have good MSE and MAE than RNN-GA, due to their characteristics of

carrying important information over a long distance. In summer, we can observe the advantage of LSTM-GA model, because of his great ability to deal with vanishing gradient. As a result, it can deal with large data size better than GRU-GA that can be seen in result of summer data forecasting from Fig 17 and Table 3. GRU-GA has features like LSTM-GA, but it has a special advantage in high efficiency and less complexity. From these observations, we find that GRU-GA had better than LSTM-GA in spring and autumn seasons (Fig 15, 16 and Table 3). However, RNN-GA still has good performance spatially in term of calculation speed comparing with LSTM-GA and GRU-GA as seen in Table 5. Generally, in terms of precision, proposed models can forecast summer and spring season data better than other seasons, due to the huge variety of weather in other seasons. The use of dropout support models to solve the problem of over-fitting, as a result, we can observe from Fig 11, 12 and 13; that there is a small difference between the train and the test. In terms of optimization, it can be seen from table 3 that DNN modules with GA algorithms for solar irradiation forecasting have a higher accuracy in MSE and MAE compared to standard DNN models, due to best choices of window size and number of units in each hidden layer by GA (Table 4). From these statistical results, the using of deep learning methods with genetic algorithm can improve the performance and accuracy of short-term solar irradiation forecasting. In addition, proposed model is firstly used for short-term solar irradiation forecasting, that make it has reached a new state of the art in term of solar irradiation forecasting precision. Hopefully, our work can be extended to different types of applications such as simulation design, installation and planning of renewable energy in micro-grids.

REFERENCES

[1] Boussetta, M., El Bachtiri, R., Khanfara, M., & El Hammoumi, K. (2017). Assessing the potential of hybrid PV–Wind systems to cover public facilities loads under different Moroccan climate conditions. Sustainable Energy Technologies and Assessments, 22(2017), 74–82. https://doi.org/10.1016/j.seta.2017.07.005

[2] Akhter, M. N., Mekhilef, S., Mokhlis, H., & Shah, N. M. (2019). Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. IET Renewable Power Generation, 13(7), 1009–1023. https://doi.org/10.1049/iet-rpg.2018.5649

[3] Lawan, S. M., Abidin, W. A. W. Z., Chai, W. Y., Baharun, A., & Masri, T. (2014). Different Models of Wind Speed Prediction; A Comprehensive Review. International Journal of Scientific & Engineering Research, 5(1), 1760–1768. Retrieved from http://www.ijser.org/researchpaper/Different-Models-of-Wind-Speed-Prediction-A-Comprehensive-Review.pdf

[4] Chow SKH, Lee EWM, Li DHW, "Short-term prediction of photovoltaic energy generation by intelligent approach", Energy Build.2012; 55: 660–667.

[5] Das, U. K., Tey, K. S., Seyedmahmoudian, M., Mekhilef, S., Idris, M. Y. I., Van Deventer, W., … Stojcevski, A. (2018). Forecasting of photovoltaic power generation and model optimization: A review. Renewable and Sustainable Energy Reviews,

[6] Ahmadi, N., & Akbarizadeh, G. (2018). Hybrid robust iris recognition approach using iris image pre-processing, two-dimensional gabor features and multi-layer perceptron neural network/PSO. IET Biometrics, 7(2), 153–162. https://doi.org/10.1049/iet-bmt.2017.0041

[7] Zaman, M. H. M., Marzuki Mustafa, M., & Hussain, A. (2018). Estimation of voltage regulator stable region using radial basis function neural network. Journal of Telecommunication, Electronic and Computer Engineering, 10(2–8), 63–66.

[8] Manogaran, G., Varatharajan, R., & Priyan, M. K. (2018). Hybrid Recommendation System for Heart Disease Diagnosis based on Multiple Kernel Learning with Adaptive Neuro-Fuzzy Inference System. Multimedia Tools and Applications, 77(4), 4379–4399. https://doi.org/10.1007/s11042-017-5515-y

[9] Y. Tao and Y. Chen, "Distributed PV power forecasting using genetic algorithm based neural network approach," Proceedings of the 2014 International Conference on Advanced Mechatronic Systems, Kumamoto, 2014, pp. 557-560.

[10] Harendra Kumar Yadav, Yash Pal & Madan Mohan Tripathi, "A novel GA-ANFIS hybrid model for short-term solar PV power forecasting in Indian electricity market, Journal of Information and Optimization Sciences", 40:2, 377-395, 2019.

[11] Bourbonnais R., Terraza M., « Analyse de séries temporelles en économie », Paris, Dunod, 318 pages, 2008.

[12] Bourbonnais R., « Analyse des séries temporelles en économie », Presses Universitaires de France – PUF, 1998.

[13] Simon G., « Méthodes non linéaires pour série temporelles », Thèse de doctorat, Université Catholique de Louvain, 2007.

[14] Raza MQ, Nadarajah M, Ekanayake C. On recent advances in PV output power forecast. Sol Energy 2016;136:125–44

[15] Shi, J., Lee, W. J., Liu, Y., Yang, Y., & Wang, P. (2012). Forecasting power output of photovoltaic systems based on weather classification and support vector machines. IEEE Transactions on Industry Applications, 48(3), 1064–1069. https://doi.org/10.1109/TIA.2012.2190816

[16] Al-Hakeem, D., Mandal, P., Haque, A. U., Yona, A., Senjyu, T., & Tseng, T. (2015). 07286233. IEEE Power Energy Society General Meeting, 1–5.

[17] Azadeh, A., Ghaderi, S. F., & Sohrabkhani, S. (2007). Forecasting electrical consumption by integration of Neural Network, time series and ANOVA. Applied Mathematics and Computation, 186(2), 1753–1761. https://doi.org/10.1016/j.amc.2006.08.094

[18] Huang, C. M. T., Huang, Y. C., & Huang, K. Y. (2014). A hybrid method for one-day ahead hourly forecasting of PV power output. Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014, 5(3), 526–531. https://doi.org/10.1109/ICIEA.2014.6931220

[19] Azadeh, A., Ghaderi, S. F., & Sohrabkhani, S. (2007). Forecasting electrical consumption by integration of Neural Network, time series and ANOVA. Applied Mathematics and Computation, 186(2), 1753–1761. https://doi.org/10.1016/j.amc.2006.08.094

[20] Huang, C. M. T., Huang, Y. C., & Huang, K. Y. (2014). A hybrid method for one-day ahead hourly forecasting of PV power output. Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014, 5(3), 526–531. https://doi.org/10.1109/ICIEA.2014.6931220

[21] Gen, M. and Cheng, R., "Genetic Algorithm and Engineering Design."John & Wiley Sons, New York. 1997.

[22] L. Deng, G. Hinton and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013, pp. 8599-8603, doi: 10.1109/ICASSP.2013.6639344.

[23] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8609-8613. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6639346&isnumber=6637585

[24] Ruiz, C. W., Perapoch, J., Castillo, F., Salcedo, S., & Gratacós, E. (2006). Bessons monocoriònics afectes de transfusió fetofetal: Conseqüències a curt i llarg termini. Pediatria Catalana, 66(2), 53–61.

[25] F. Chollet., "Keras: Deep learning library for theano and tensorflow. [Online].", Available: https://keras. io/k, 2018.

[26] François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau and Christian Gagné, "DEAP: A Python Framework for Evolutionary Algorithms", in EvoSoft Workshop, Companion proc. of the Genetic and Evolutionary Computation Conference (GECCO 2012), July 07-11, 2012.

[27] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 6, pp. 1109-1121, December 1984, doi: 10.1109/TASSP.1984.1164453.

[28] V. H. QUINTANA & E. J. DAVISON (1974) Clipping-off gradient algorithms to compute optimal controls with constrained magnitude, International Journal of Control, 20:2, 243-255, DOI: 10.1080/00207177408932734

[29] Chaibi, Y., Allouhi, A., Salhi, M., & El-jouni, A. (2019). Annual performance analysis of different maximum power point tracking techniques used in photovoltaic systems. Protection and Control of Modern Power Systems, 4(1). https://doi.org/10.1186/s41601-019-0129-1