# Photovoltaic Power Forecasting Using Neural Networks for Short and Medium-Term Dependencies

Raaid Kabir, Abdellatif Elmouatamid, Hamza Elkhoukhi, Philip Pong

# Photovoltaic Power Forecasting Using Neural Networks for short and Medium-Term Dependencies

Raaid Kabir[1], Abdellatif Elmouatamid[1][0000-0001-5499-5865], Hamza Elkhoukhi[2][0000-0002-7881-4325], Philip W. T. Pong[1][0000-0002-5554-8386]

[1] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
[2] IA Laboratory, Moulay University Ismail of Meknes, Faculty of science, 11201 Zitoune, Meknès – Morocco;
abdellatif.elmouatamid@njit.edu
philip.pong@njit.edu

**Abstract.** Renewable energy sources present a main factor in the development of microgrid systems. The end consumer can produce, consume, and store locally the electrical power through the installation of sustainable sources and energy storage systems. However, the random nature of these sources requires the use of new technologies (e.g., machine learning, internet of things) for the control, maintenance, and seamless integration into the actual electrical grid infrastructure. The development of an accurate power forecaster, particularly for photovoltaic (PV) systems, is the main challenge to smooth the integration of such sources (e.g., solar panels, wind turbines) into the grid. In this work, long short-term memory (LSTM) algorithm is proposed for multistep-ahead photovoltaic power forecast. LSTM and convolutional LSTM (Conv-LSTM) models are adapted for short and medium-term PV power forecasts. The objective is to predict the PV power and extend the forecasting time horizon as much as possible at the same time keeping acceptable errors and processing time. A set of criteria are analyzed to show the proposed algorithms' usefulness. In fact, the required processing time and the machine re-sources for the training and testing are considered to select suitable neural network parameters. The obtained forecasting results show the utility of the proposed LSTM model for short and Medium-term PV power forecasting.

**Keywords:** Artificial Neural, LSTM, Machine Learning, Model Accuracy, Model Error, Power Forecast, Processing Time,

## 1 Introduction

The variability and the stochastic nature of renewable energy sources (RESs) power production are the main challenges to the seamless integration of these in the actual grid infrastructure. In order to deal with these issues, researchers are more focused on the development of new predictive control and energy management strategies for maintenance scheduling, trading in the market, and future energy stability and availa-

bility [1]. In this way, the development of new machine learning algorithms for power forecasting is the main key to reach the objectives [2, 3]. Photovoltaic (PV) panels are one of the RESs increasingly integrated into the grid as well as into the microgrid systems. Weather conditions, such as temperature, solar irradiance, humidity, and wind speed influence directly and indirectly the PV power generation. Therefore, an accurate machine learning algorithm for PV power forecasting should be capable to predict the solar power output with minimal errors for a given time horizon. On another hand, the required processing performances for the model training is the main factor for the model selection [4]. In fact, several parameters influence on the model output such as the sampling time of the training dataset, the number of input/output neural networks, the time horizon as well as the epochs number, and the batch size in the algorithm [5]. A set of criteria is considered for the algorithm parameters selection depending on how diverse the training dataset. Statistical metrics are usually implemented specifically mean absolute percentage error (MAPE), root-mean-square error (RMSE), R-Squared ($R^2$), mean absolute scaled error (MASE), and symmetric mean absolute percentage error (SMAPE) [6, 7]. These metrics are calculated based on mathematical expressions between the actual and predicted values. Depending on the power forecasting objectives, the power forecasting algorithm parameters are selected in order to minimize a specified error to reach the application objectives. Mainly, the power forecasting objective is the first step that should be selected before designating the machine learning algorithm and its parameters. In this way, three categories are considered in the literature depending on the time scale. i) Short-term: the time horizon varies from a few milliseconds to a few minutes and it is used generally for control action regulation and services management [8]. This application class can be limited by the processing time. For that other predictive methods are used, instead machine learning algorithms, such as general predictive and model predictive control strategies [9, 10]. ii) Medium-term: the time horizon varies from a few hours to few days and it is adapted for applications such as electricity trading in the market, energy management, fault detection and diagnostic. iii) long-term: the time horizon varies from the medium-term limit to one year or more and it is deployed for the power plan dimensioning, predictive maintenance, and future energy statistics and security for a country [11]. Different datasets size and variable measurements are required for the algorithm training depending on the forecasting time scale. Therefore, increasing the time horizon requires the use of a large dataset including the different probable events and data variability that we can have in the future. Though, the use of a large dataset increases the training time and the required machine performances used for the algorithm deployment. For that an optimal dataset size and parameters configuration should be selected for the machine learning algorithm deployment [12]. Several works are presented in the literature investigating the use of machine learning for power prediction, especially for short and long terms PV power prediction focusing more on the common errors as metrics [5, 13, 14]. By the analysis of the existing works, deep learning family, particularly recurrent neural networks (RNN) algorithms present strong performance for both short- and long-term PV power forecasting. In this work seq2seq-LSTM and conv-LSTM models are tested and compared for short and medium-term PV power forecasting. The SMAPE, MASE, RMSE, and $R^2$ errors

are calculated together with the processing time in order to select the suitable deep learning algorithm. Univariable models are used and the data is fitted and pre-processed in way to keep minimal errors and acceptable processing time. Several sampling times are used for the algorithms training and testing to show the robustness of the proposed algorithm for medium-term PV power forecasting.
The remainder of this paper is summarized as follows: A brief state of the art is presented in the introduction in order to compare the existing work with the paper results. Section 2 investigates the proposed deep learning method and its adaptation for power forecasting implementation. Section 3 reviews a brief survey of the forecasting errors. Results and discussions are presented in section 4 while conclusions and perspectives are summarized in section 5.

## 2 LSTM model for PV power forecasting

PV power generation depends essentially on the weather conditions (e.g., temperature, solar irradiance). For that, the PV power data is a stochastic time series signal that varies during the hours of the day and the seasons of the year. Mainly, neural networks model work with any type of sequential data and they can be used for each specific type of time series forecasting problems, especially the models which kept a context of memory within their pipeline. The traditional neural network models have the ability to analyze input sequences which correlate on each other on the context and information to predict the next output sequence. This simple model does not have concept of memory and it takes a standalone data vector for each timestep making it inappropriate for tasks that need memory. In this way, RNN models are developed using feedback approach in which the output is relied to the input in order to deliver context on the last delivered output. However, RNN models suffer from two immense problems which are the vanishing and exploding gradient problems [15]. In fact, the RNN model finds the optimal setup for the network by the calculation of the minimum of the cost function. The information travels in the RNN from the input to the output through time and the information generated from the previous points is used as input for the next point. During this process, the cost function and the errors are calculated at each time point and the outputs are compared to the desired outputs that are used for the training. The cost function and the error are calculated and propagated back to the neural network in order to update the weights. At the start of the neural network, weights are generated randomly with values close to zero. During the training, the output functions of the neurons are multiplied by the weight recurring. This weight recurring is the source of the two problems (vanishing and exploding gradient problems) depending on the starting value multiplied the different neurons. When the weight recurring is close to zero, the gradient becomes less with each multiplication and it will be hard for the network to update the neurons weights and to reach the final output values. Accordingly, less weight recurring values generate the vanishing gradient problem and the RNN takes more time to train the weights and the large weight recurring values generate the exploding gradient problem. For these raisons, other models are developed to deal with these problems. One of the proposed solutions is

the LSTM neural networks which aim to recall long-term information from sequential data due to its capability of automatically learning features and long-term dependencies [16, 17]. LSTM neurons contain input, output, and forget layers combined by pointwise operations acting as gates (see Figure 1).

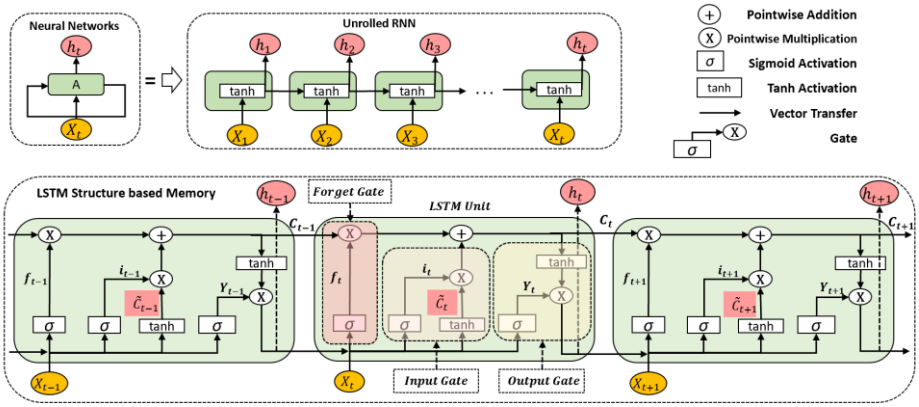


**Fig. 1.** Recurrent neural networks and long short-term memory information flow.

The sigmoid layer named Forget-gate-layer decides which information is going to throw away based on the previous hidden state $h_{t-1}$ and the current input $X_t$ at time $t$. The output from this layer is a number between 0 and 1 generated for the different cell state numbers $C_{t-1}$. When the neural obtains new information, the previous information is forgotten by the generation of output 0 to indicate that the learned value is completely excluded from the memory, while 1 indicates "completely keep the memory". The output is generated by equation (1) where, $f_t$ is the forget gate, $\sigma$ is the sigmoid activation, $W_f$ and $b_f$ are the initial LSTM unit parameters:

$$f_t = \sigma(W_f . [h_{t-1}, X_t] + b_f) \tag{1}$$

The second step concerns the information that will be stored in the cell state. The input gate layer decides which values should be update depending on the forget gate output and a $tanh$ layer generates a new vector of the candidate values $\tilde{C}_t$ to be integrated in the cell state. The input gate and $tanh$ layers are combined together to prepare the new value for the state. The input gate is expressed by equation (2) where, $i_t$ is the input gate, $b_i$ and $W_i$ are LSTM parameters:

$$i_t = \sigma(W_i . [h_{t-1}, X_t] + b_i) \tag{2}$$

Now, the current cell state $C_t$ will be update based on the new $\tilde{C}_t$ and the previous cell state $C_{t-1}$ multiplied by the forget state $f_t$ in order to keep or to get rid the actual memory. The new cell and the current cell states are presented respectively by equation (3) and (4):

$$\tilde{C}_t = tanh(W_c . [h_{t-1}, X_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

This is the simple LSTM process which integrates the memory concept to resolve the classical recurrent neural networks problems. The $C_t$ memory cell acts as an accumulator to learn and store long and short-term sequence of related information to avoid the long-term dependency problem. The $C_t$ propagates to the next LSTM unit at the same time a sigmoid layer is used to decide which part of the cell can circulates to the output. A hyperbolic tangent function $tanh$ is used to put the state values between 0 and 1 and multiplying the value with the output-gate output to decide which part of the memory cell can propagate to the hidden state $h_t$. The output gate $o_t$ presented by equation (5) decides which output can flow into making the hidden state $h_t$ expressed by equation (6):

$$o_t = \sigma(W_o.[h_{t-1}, X_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

The memory cell and the hidden cell are transferred repeatedly to the next LSTM unit for each timestep. The model calculates the cost function and the error and updates the weights and the biases in order to minimize the difference between the actual and the predicted samples. Accordingly, several variants are proposed [18, 19] for the LSTM structure compared to the first model presented by [17]. Specifying the forecasting application is the first step to identify the constrains and the equivalent machine learning model that can deal with these constraints. Accordingly, predictive control and fault detection in microgrid systems require high accuracy and short processing time. Especially, for the processing time and the computational resources required for the algorithm training, in this work Conv-LSTM model is proposed. This model is more suitable for timeseries forecasting with efficient computational resources. Conv-LSTM model has some simple instruction compared to the LSTM model; therefore, the combination of both architectures adds more peepholes to cell. The new forget gate, input gate, and the output gate are presented respectively by equation (7), (8), and (9). In Conv-LSTM model, at each timestep, there is an equivalent hidden calculated by the combination the previous timestep hidden unit and the new input information. This concept can improve the decision of the memory reset decision generated by the forget layer and consequently reducing the information propagation time and avoiding the classical RNN problems.

$$f_t = \sigma(W_f.[C_{t-1}, h_{t-1}, X_t] + b_f) \quad (7)$$

$$i_t = \sigma(W_i.[C_{t-1}, h_{t-1}, X_t] + b_i) \quad (8)$$

$$o_t = \sigma(W_o.[[C_t, h_{t-1}, X_t] + b_o) \quad (9)$$

The main objective of this work is the deployment of an LSTM model that can forecast the PV power with a high accuracy and for a long-time horizon has much as it is possible. Some existing works talk about proposed method for long-term power forecasting, however, by the analysis of the deployed dataset, the data is collected by a long sampling time (e.g., day, week, month). In fact, the algorithms generate a set of

future predicted steps depending on the time horizon without having the concept of the time units. For example, if the dataset has a sampling time of one hour, the generation of ten steps meaning that the model predicts ten hours ahead. Though, if the same model is deployed using a new dataset with a sampling time of one minute, the prediction of ten values means that we can predict only ten minutes ahead. In this case, it is difficult to forecast the output in the scale of hours and days. In this work, a dataset with a sampling time of one second is used and the objective is to predict the PV power with high accuracy and for a long-time horizon compared to the input data sampling time. This forecasting model is more robust for the control and for faults detection in the power systems.

## 3 Performance evaluation and errors calculation

The mathematical equations of the different errors are presented in the literature [6, 7]. This section focuses more on the physical significance of the errors and the equivalent improvement that can be obtained for each metric. A set of errors are calculated based on the percentage errors due to their scale-independent propriety and their efficiency. The MAPE measures the accuracy of the forecasting algorithm based on the absolute percentage errors of each entry in the dataset. It should be minimal as much as possible while the values around 50% maiming that the obtained results are totally ignored. It is a comprehensive method to compare the algorithms output and to present the results for an end user. The formula for MAPE calculation is as follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{10}$$

where $n$ is the number of data points in the testing dataset, $A_t$ is the actual values, and $F_t$ is the forecasting values.

The second error considered in this study is the RMSE which presents tells you how concentrated the data is around the line of the best fit. For the power forecasting models, we don't care about the data units. For the algorithms, it is an input timeseries signal used to train the model and to identify the heuristic to help us decrease the error with each iteration. We consider only the relative size of the error from one step to the next, not the absolute size of the error. Since the unit is not considered, the obtained RMSE values cannot be a metric to analyze the models if it is not compared to another algorithm RMSE or to the same algorithm with different parameters. The RMSE is presented by the following equation:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}} \tag{11}$$

Another interesting statistical measure of fit is the $R^2$ that calculate the variation of dependent variable depending on the independent variable in a linear regressive model to identify how well the model fits the data. It presents the relationship between the model and the dependent variable on a fitting from 0 to 100% scale. However, the $R^2$ high value is not always an indicator of a robust model. An analyze of the graph with

the calculation of the other metrics is necessary to identify the usefulness of the model. it is calculated by the following equation:

$$R^2 = 1 - \frac{\text{regression sum of squares}}{\text{error sum of squares}} \quad (12)$$

These errors are used to evaluate the performances of the deployed machine learning algorithm for PV power forecasting. Depending on the application requirements, several type of errors can be interpreted. In some cases, a model presents a good accuracy by the analysis of a specific error while other metrics are not robust. For each study, a given metric is the main objective to establish.

## 4 Results and discussions

Depending on the forecasting application requirements, several type of errors can be interpreted. In some cases, a model presents a good accuracy by the analysis of a specific error while other metrics are not robust. For each study, a given metric is the main objective to establish. In this way, different scenarios are realized to specify the accurate parameters for the LSTM model and for the neural network (Table 1). A dataset of five days is used with a sampling time of one minute to train the model, while two other days are used for the testing. An optimal number of the input and output neutrals is fixed together with the initial dropout of 10%. The selected parameters are used to train the LSTM and Conv-LSTM models. The objective is to select the accurate algorithm that can forecast the PV power with long time horizon and minimal errors as much as possible. In this case, the sampling time is one second, therefore, predicting an hour is a long term.

**Table 1.** Scenarios for the optimal LSTM parameters selection.

| | *Input* | | | *Output* | |
|---|---|---|---|---|---|
| *Parameter* / *Scenario* | Epochs | Batch | Time Horizon | Training Time | Loss |
| *1* | 15 | 10 | 300 | 348 | 63.48 |
| *2* | 15 | 30 | 300 | 246 | 62.83 |
| *3* | 15 | 100 | 300 | 88 | 62.23 |
| *4* | 10 | 100 | 300 | 59 | 63.32 |
| *5* | 5 | 100 | 300 | 29 | 66.08 |
| *6* | 1 | 100 | 300 | 6 | 273.65 |

For the control and energy management applications, a dataset of one hour collected with a sampling time of one second is long-term and the forecast of few minute can be considered as a medium term. Table 2 presents the errors calculated for both LSTM and Conv-LSTM together with the processing time in order to select the suitable algorithm for such application. In fact, a time horizon of five minutes is forecasted which presents a medium term for the predictive control application.

Fig. 2 presents the Actual and the forecasting PV power during two days and for a time horizon of five minutes. The blue curve is the actual PV power, while the green and the red curve are respectively the LSTM and Conv-LSTM power forecasting. The

two algorithms present an acceptable accuracy and errors. In this case, the machine performances and processing time is an interesting metric to focus on.

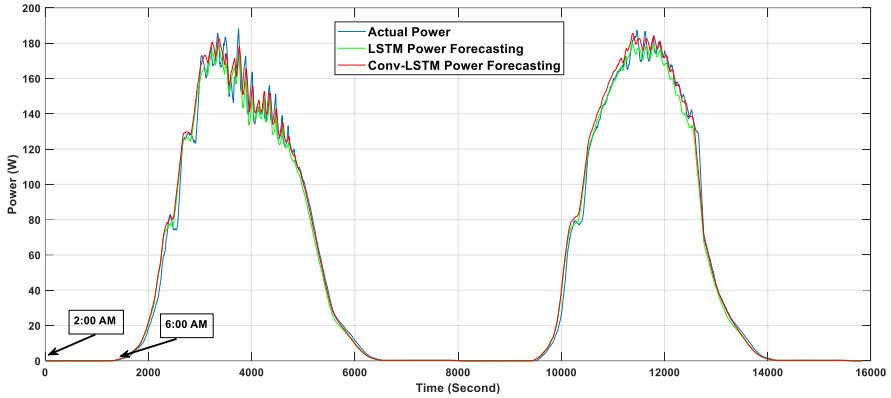


**Fig. 2.** Actual and forecasted PV power using LSTM and Conv-LSTM.

**Table 2.** Machine learning algorithms errors calculation

| | $R^2$ | RMSE | SMAPE | MAPE | rsq | Training Time |
|---|---|---|---|---|---|---|
| ***LSTM*** | 0.99 | 8.47 | 22.24% | 0.36 | **98.57%** | **62.12s** |
| ***Conv-LSTM*** | 0.99 | **8.13** | **17.98%** | **0.23** | 98.61% | 72.45s |

By the analysis of the calculated parameters (Table 2), the Conv-LSTM presents robust performances compared to the LSTM, however, the LSTM has an acceptable processing time and it is more suitable for the application limited by the processing performance. Another scenario is tested using the data collected during a cloudy day. The power generated from the PV panels is not stable and varies depending on the perturbances on the weather conditions (e.g., solar irradiance, temperature).

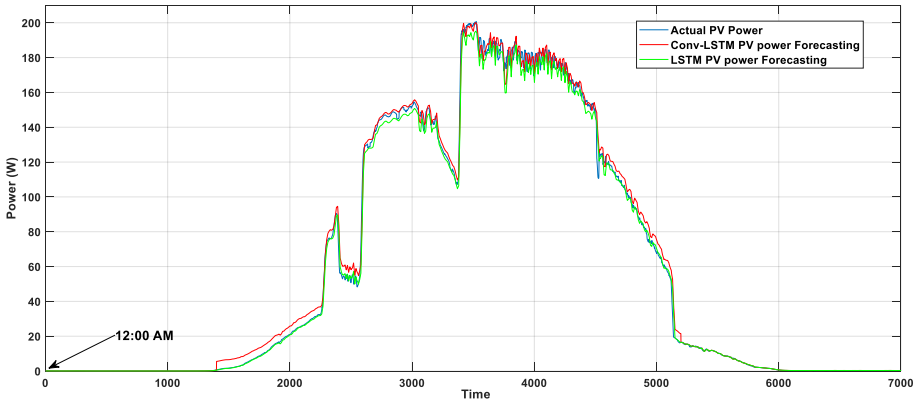


**Fig. 3.** Actual and forecasted PV power using LSTM and Conv-LSTM for a cloudy day.

In order to test the performance of the proposed machine learning algorithm, a dataset of a cloudy day is used to test the algorithm (Fig.3). As is mentioned in Table 3, the two algorithms keep acceptable errors values with some fluctuation on the LSTM output. The variability of the PV power production can be followed up by the deployed machine learning algorithms.

**Table 3.** Errors calculation for cloudy day scenario.

| | $R^2$ | RMSE | SMAPE | MAPE | rsq | Training Time |
|---|---|---|---|---|---|---|
| *LSTM* | 0.92 | 9.32 | 27.45% | 0.46 | **92.57%** | **65.08s** |
| *Conv-LSTM* | 0.96 | **8.65** | **18.72%** | **0.31** | 96.61% | 79.12s |

The dataset used for the two scenarios has some missing values due to the short sampling time used for the data collection from the sensors. However, the number of measurements collected during the testing period presents significant interest to the control and energy management in the microgrid system. In order to manage the energy flows, some actual predictive control approaches are based on the power forecasting, while the obtained results can be deployed for such control strategies with acceptable errors.

## 5 Conclusions and perspectives

Due to the robust performances of neural networks family of machine learning, two algorithms, LSTM and Conv-LSTM, are tested for the PV power forecasting. A set of metrics are considered to select the suitable algorithm (e.g., errors, processing time). A dataset of five days is used for the training and the testing with a short sampling time. The objective is the use of a short sampling time to forecast a long-time horizon with acceptable errors. In this case, the predictive control for energy management is considered, therefore, five minutes are forecasted with high accuracy which present a medium-term for the control applications. The algorithm is tested for power prediction in a cloudy day. In the future works, a set of artificial intelligence algorithms will be compared and tested for short, medium and long-term power forecasting in microgrid systems. The algorithms will be tested for different context and data variety. The objective of long-term power forecasting is for the power stability of the grid including renewable energy sources as well as the electricity trading in the market. Moreover, a medium-term of one day to one week can help for the predictive maintenance and for the electricity operator to make the system diagnostics.

## References

1. Nair, U.R. and R. Costa-Castelló, *A model predictive control-based energy management scheme for hybrid storage system in islanded microgrids.* IEEE access, 2020. **8**: p. 97809-97822.

2. Markovics, D. and M.J. Mayer, *Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction.* Renewable and Sustainable Energy Reviews, 2022. **161**: p. 112364.
3. Hossain, M.S. and H. Mahmood, *Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast.* Ieee Access, 2020. **8**: p. 172524-172533.
4. Zhou, Y., et al., *Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine.* Energy, 2020. **204**: p. 117894.
5. Mellit, A., A.M. Pavan, and V. Lughi, *Deep learning neural networks for short-term photovoltaic power forecasting.* Renewable Energy, 2021. **172**: p. 276-288.
6. Nour-eddine, I.O., et al., *Power forecasting of three silicon-based PV technologies using actual field measurements.* Sustainable Energy Technologies and Assessments, 2021. **43**: p. 100915.
7. Shcherbakov, M.V., et al., *A survey of forecast error measures.* World applied sciences journal, 2013. **24**(24): p. 171-176.
8. Li, P., et al., *A hybrid deep learning model for short-term PV power forecasting.* Applied Energy, 2020. **259**: p. 114216.
9. Elmouatamid, A., et al., *A predictive control strategy for energy management in micro-grid systems.* Electronics, 2021. **10**(14): p. 1666.
10. Khosravi, M., et al., *Review of model predictive control strategies for matrix converters.* IET Power Electronics, 2019. **12**(12): p. 3021-3032.
11. Kazem, H.A., et al., *Long-term power forecasting using FRNN and PCA models for calculating output parameters in solar photovoltaic generation.* Heliyon, 2022. **8**(1): p. e08803.
12. Bouktif, S., et al., *Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches.* Energies, 2018. **11**(7): p. 1636.
13. Wang, K., X. Qi, and H. Liu, *Photovoltaic power forecasting based LSTM-Convolutional Network.* Energy, 2019. **189**: p. 116225.
14. Elmouatamid, A., et al. *A control strategy based on power forecasting for micro-grid systems.* in *2019 IEEE International Smart Cities Conference (ISC2)*. 2019. IEEE.
15. Ribeiro, A.H., et al. *Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness*. in *International Conference on Artificial Intelligence and Statistics*. 2020. PMLR.
16. Livieris, I.E., E. Pintelas, and P. Pintelas, *A CNN–LSTM model for gold price time-series forecasting.* Neural computing and applications, 2020. **32**(23): p. 17351-17360.
17. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation, 1997. **9**(8): p. 1735-1780.
18. Wang, H., et al., *A review of deep learning for renewable energy forecasting.* Energy Conversion and Management, 2019. **198**: p. 111799.
19. Khalil, M., et al., *Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption—A systematic review.* Engineering Applications of Artificial Intelligence, 2022. **115**: p. 105287.