Use of Optimised LSTM Neural Networks Pre-Trained With Synthetic Data to Estimate PV Generation

Miguel Martínez-Comesaña[1]*, Javier Martínez-Torres[2,3], Pablo Eguía-Oller[1], Javier López-Gómez[1]

[1] Department of Mechanical Engineering, Heat Engines and Fluids Mechanics, Industrial Engineering School, CINTECX, University of Vigo (Universidade de Vigo), Maxwell s/n, 36310 Vigo (Spain)
[2] Department of Applied Mathematics I, Telecommunications Engineering School, CINTECX, University of Vigo (Universidade de Vigo), 36310 Vigo (Spain)
[3] Department of Applied Mathematics I, Telecommunications Engineering School, CITMAga, 15782 Santiago de Compostela (Spain)

* Corresponding author: migmartinez@uvigo.gal

Received 13 September 2022 | Accepted 17 October 2023 | Published 17 November 2023

Abstract

Optimising the use of the photovoltaic (PV) energy is essential to reduce fossil fuel emissions by increasing the use of solar power generation. In recent years, research has focused on physical simulations or artifical intelligence models attempting to increase the accuracy of PV generation predictions. The use of simulated data as pre-training for deep learning models has increased in different fields. The reasons are the higher efficiency in the subsequent training with real data and the possibility of not having real data available. This work presents a methodology, based on an deep learning model optimised with specific techniques and pre-trained with synthetic data, to estimate the generation of a PV system. A case study of a photovoltaic installation with 296 PV panels located in northwest Spain is presented. The results show that the model with proper pre-training trains six to seven times faster than a model without pre-training and three to four times faster than a model pre-trained with non-accurate simulated data. In terms of accuracy and considering a homogeneous training process, all models obtained average relative errors around 12%, except the model with incorrect pre-training which performs worse.

Keywords

Genetic Algorithm, LSTM, Optimisation, Pre-Training, PV Power, Synthetic Data.

DOI: 10.9781/ijimai.2023.11.002

I. Introduction

Nowadays, the demand for electric power is growing significantly and the mayor issue is to reduce fossil fuel emissions and thus control global warning [1]. Transport and electricity generation have accounted for 60% of all energy produced in the last few years [2]. In this way, the European Commission has defined new targets for 2030 which include reducing the CO_2 emissions by 40% with respect to 1990 levels [3]. Meeting this target requires reducing the electricity demands and/or increasing the use of renewable energies [4].

Among the renewable energies, solar power generation has proven to be a serious option as a result of its great availability and low production cost [5]. This type of renewable energy generation has two main sources: thermal and photovoltaic (PV). In recent years, solar PV production has expanded considerably, becoming the fastest growing resource for electric power generation with the highest power density among all renewable energy resources [6]–[8]. This resource also has two important barriers: the low efficiency of the PV modules (directly related to meteorological conditions) and the large investment cost [5], [9]. Nevertheless, its potential to feed energy into the grid along with the reduction of transmission losses it provides makes this renewable resource very attractive [10].

Recently, artificial intelligence techniques, more specifically deep learning models, has become widespread as a novel data-driven approach that can be applied to numerous scientific fields such as PV energy analysis or related areas [11], [12]. Deep learning models are famous because they are able to learn complex patterns without requiring in-depth knowledge of the subject under analysis and are characterised for their high performance and easy implementation. In addition, these models have become increasingly more popular due to their ability to better optimise and replicate learning patterns than the more classical machine learning techniques [11]. Some concrete examples of that, in similar studies of the one proposed, are Nabipour et al. [13] show the higher accuracy of DL models prediction stock market trends and Mert. [14] show the better performance of DL models in solar-powered systems production estimations.

Please cite this article in press as:
M.Martínez-Comesaña, J. Martínez-Torres, P. Eguía-Oller, J. López-Gómez. Use of Optimised LSTM Neural Networks Pre-Trained With Synthetic Data to Estimate PV Generation, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 3, pp. 61-70, 2025, http://dx.doi.org/10.9781/ijimai.2023.11.002

Long Short-Term Memory (LSTM) neural networks are a deep learning model, within the group of Recurrent Neural Networks (RNN) [15], which contain a specific hidden layer that considers the existence of connections with past values [16], [17]. In this way, they are suitable for mapping long-term dependencies. These neural networks have been implemented in similar fields such as environment [18], energy efficiency in buildings [16], image processing [17] and PV generation [19], [20]. In particular, they have shown better performance in photovoltaic generation estimations thanks to being able to use the information learned from previous steps [21], [22]. Moreover, most deep learning models leave room for optimisation based on the hyperparameters that defined them. These improvements, which can be achieved in model performance reaching the optimal values for the hyperparameters is demonstrated in several previous studies [23],[24]. In the literature can be found different techniques to efficiently perform this search: univariate dynamic encoding algorithms [25], combination between grid and random searches [26], particle swarm optimisation [27] or genetic algorithms [28]. In particular, Genetic Algorithms (GA) have increased their use in this type of optimisations mainly due to their easy of implementation and the reduction in the number of evaluations and time needed to reach an optimum [24], [29], [30]. Furthermore, multiobjective genetic algorithms such as Non-dominant Sorting Genetic Algorithm (NSGA-II) make it possible to optimise the values of the selected hyperparameters considering more than one objective function [1], [31], [32].

In recent years, feeding machine and deep learning models with simulated data, prior to real data, has been shown to improve their performance. The spread of this technique is due to the fact that several studies have shown that pre-training the model with synthetic data enables subsequent training with real data to be faster and/or more accurate, on the one hand, and that in certain situations collecting real data is very costly or not possible, on the other hand [33], [34]. This methodology has been applied in several fields such as signal denoising [35], pattern recognition [36] and robot perception [37]. The aim of this research is to introduce a methodology to optimise deep learning models architecture and improve their performance using synthetic data. In particular, this study focuses on estimating PV generation and comparing the accuracy of the built models depending on their pre-training. The analysed installation is located on the roof of the School of Mining Engineering in northwest Spain. The available data consist of hourly frequency observations of PV generation together with outdoor temperature and global solar irradiance of the area. Additionally, three temporal variables (month of the year, day of the month and hour of the day) are also considered as model inputs. In this way, taking into account the aforementioned inputs and the variable of interest (PV generation), the optimisation process and the improvement provided by a proper pre-training were analysed. Specifically, both the epochs required to reach a certain error limit and the coefficient of variation of the root mean squared error (CV(RMSE)) and normalised mean bias error (NMBE) are the model evaluation metrics selected.

The novelty of this paper lies in the application of deep learning models, optimised with the NSGA-II algorithm and pre-trained with simulated data, to perform PV generation predictions of an installation consisting of 296 PV modules. Furthermore, the introduced methodology shows the significant improvement of the model behaviour with a correct pre-training process based on synthetic data. Thus, this work contributes with a method that efficiently optimises the deep learning model and improves its training speed in comparison with a model without pre-training or with an incorrect pre-training. In the field of renewable energies, this improvement allows better control and utilisation of photovoltaic energy, optimising, for example, the connection between a house with photovoltaic panels and an electric vehicle.In addition, the presented use of synthetic data allows the implementation of deep learning models in situations where the monitored data is limited or the PV systems have just been installed and there is very few data available to feed the model.

II. Material and Methods

The aim of this research is to analyse the usefulness of synthetic data to pre-train deep learning models and thus study whether they improve their performance in the training process with real data. In this case, the study focuses on a photovoltaic installation, and specifically, on estimating the generation of a PV system based on meteorological and temporal variables. To this end, the deep learning models used are LSTM neural networks optimised with NSGA-II multiobjective genetic algorithm.

A. Long Short-Term Memory (LSTM) Neural Network

The deep learning model used in this study is a Long Short-Term Memory (LSTM) neural network. This type of neural networks are Recurrent Neural Networks (RNN); sequenced-based models that take into account the possible correlations between past and current data [38], [39]. RNN use the backpropagation through time (BPTT) method, which considers that the decision a RNN makes at time step $t-1$ can influence the decision at time step t. However, due to the vanishing gradient problem [40], these models are not good learning relationships in the long run. This problem is described as the gradient norm decays exponentially to zero from long-range dependencies. In this case, LSTM neural networks, having an architecture with a memory cell and a forget gate, are capable of solving the aforementioned problem [41].

The dynamics of RNN can be established with deterministic transitions from previous to current hidden state (h_t^l):

$$RNN: \mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^{l} \rightarrow \mathbf{h}_t^{l} \tag{1}$$

being l the layer and t the time step. In constrast, LSTM neural networks present a more sofisticated structure that enables the memorisation of information for many time steps. The long-term memory is stored in a dedicated vector of memory cells $s_t^l \in \mathbb{R}^k$:

$$LSTM: \mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^{l}, \mathbf{s}_{t-1}^{l} \rightarrow \mathbf{h}_t^{l}, \mathbf{s}_t^{l} \tag{2}$$

As an illustration, we assume an input vector x, where $x_t \in \mathbb{R}^k$ is a k-dimensional vector at time step t. LSTM neural networks maintain an internal memory cell state during the entire process in order to build the temporal connections. The memory cell s_{t-1} interacts with the hidden state h_{t-1} and the specific input x_t to establish the elements of the inner state vector to be deleted, updated or mantained. Furthermore, LSTM neural networks have a forget gate f_t, an input gate i_t, an input node n_t and an output gate ot in their structure (see Fig. 1). The architecture of these models can be defined by the equations 3, 4 and 5 [38], [41]:

$$\mathbf{f}_t = \sigma(W_{fx}\mathbf{x}_t + W_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{3}$$

$$\mathbf{i}_t = \sigma(W_{ix}\mathbf{x}_t + W_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{4}$$

$$\mathbf{o}_t = \sigma(W_{ox}\mathbf{x}_t + W_{oh}\mathbf{h}_{t-1}\mathbf{b}_o) \tag{5}$$

being W weight matrices associated to the activation functions, $\odot$ an element-wise multiplication and σ the representations of the sigmoid function.

In this way, the new current cell state (n_t) can be calculated with Equation 6:

$$\boldsymbol{n}_t = \phi(W_{nx}\boldsymbol{x}_t + W_{nh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_n) \tag{6}$$

where φ represent the tanh activation function. Based on the forget and input gate the state st is updated through Equation 7:

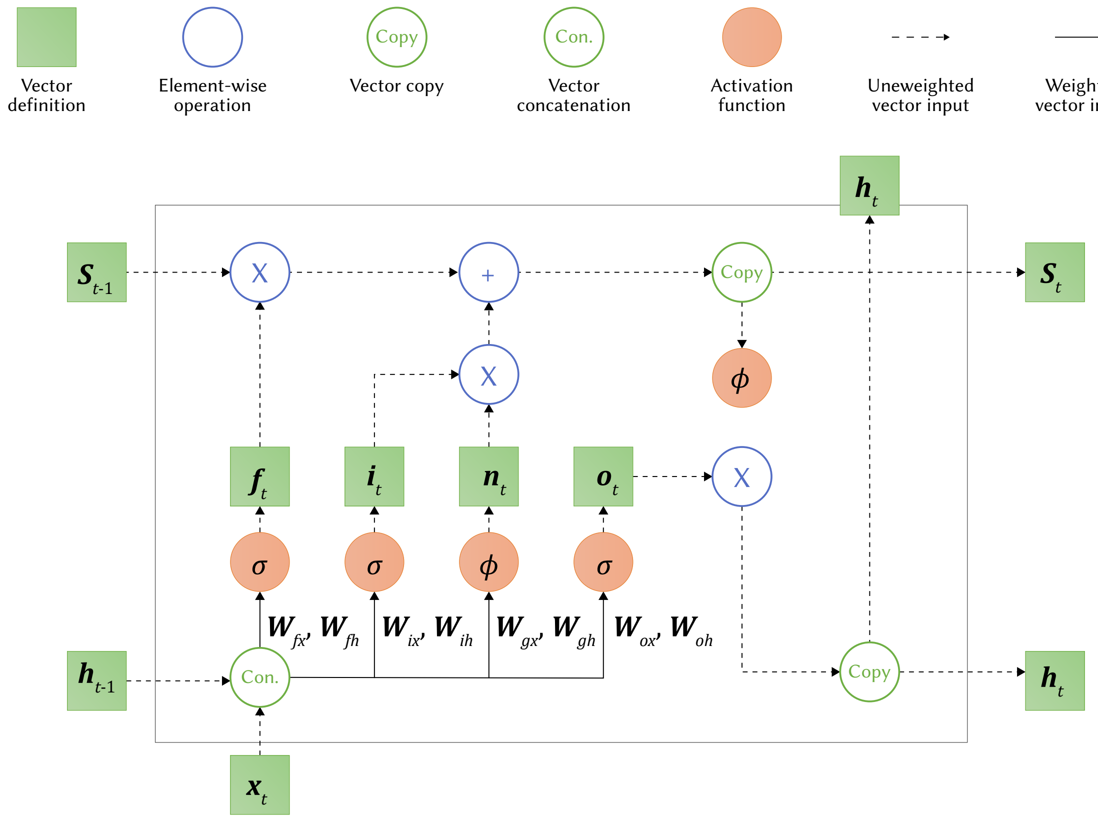

Fig. 1. Internal structure of LSTM block.

$$\boldsymbol{s}_t = \boldsymbol{n}_t \odot \boldsymbol{i}_t + \boldsymbol{s}_{t-1} \odot \boldsymbol{f}_t \tag{7}$$

and the current hidden output using Equation 8:

$$\mathbf{h}_t = \phi(\mathbf{s}_t) \odot \mathbf{o}_t \tag{8}$$

As shown in Fig. 1 there are three sigmoid functions in the LSTM block, which can be 0 or 1 and act as switches to manage which elements pass through the gates. In addition, the present input xt and the past state ht1 affect the decision made at the forget gate f, the input gate i and the output gate o. The forget gate determines which elements of the previous memory cell st1 are forgotten and the input gate selects which elements are kept. Thus, the inner state is updated and the elements of st that move forward as LSTM state ht are selected through the output gate. This process is replicated at every time step [39], [41].

On the one hand, the LSTM neural networks built in this analysis are optimised with a mutiobjective genetic algorithm focusing on the accuracy and the complexity of the model. The parameters adjusted are the number of LSTM layers, the number of Dense layers, the number of neurons in each of them and the number of epochs the model is allowed not to improve (stopping criterion). On the other hand, the built neural networks use the internal optimisation algorithm known as *Adam*, the Rectified Linear Unit (reLU) activation function and a batch size of 24.

B. Model Optimisation

The optimal architecture together with the optimal value for the parameter defining the stopping criterion (number of epochs without improvement) of the built LSTM neural network are obtained with a multiobjective genetic algorithm. Genetic Algorithms (GA) are known for trying to replicate biological evolution to solve optimisation problems. They initiate the process with a random population based on individuals. These individuals are represented by chromosomes consisting of genes which, in turn, are the values of the considered covariates. Thus, this type of algorithms conducts optimisation based on three main operators: crossover, mutation and elitism. Crossover refers to exchanging a portion of a specific chromosome with a portion of another random chromosome. Mutation increases diversity in populations to avoid stagnating at local optima by randomly modifying part of solutions. Elitism is the way in which the selection process is accomplished by choosing the best chromosomes to pass through generations [42], [43].

In this study, the specific algorithm used is the Non-Dominant Sorting Genetic Algorithm (NSGA-II). It is a robust multiobjective algorithm widely implemented in different practical fields that allows the simultaneous optimisation of several parameters. Furthermore, it is characterised by generating a Pareto front betweeen the objectives where the overall optimum is selected and for being an improved version of the original version of the NSGA. These improvements are based on the use of a crowding distance operator, the elitism and a fast nondominated ranking [31], [44].

This algorithm is based on four internal principles that defined its processing [45]:

- Non-dominated sorting: The options considered, which form a population, are ordered by Pareto dominance. In this way, the elements/options with the best rank are separated and the ordering continues with the rest of the options.
- Crowding distance: Between two possible solutions, the one with a larger crowding distance is considered to be in a less crowded area. Thus, the elements in a less crowded region will be selected first. The crowding distance for an element is presented in the Equation 9:

$$\mathrm{CD}(i) = \sum_{i=1}^{k} \frac{F_j^{i+1} - F_j^{i-1}}{F_j^{\max} - F_j^{\min}} \tag{9}$$

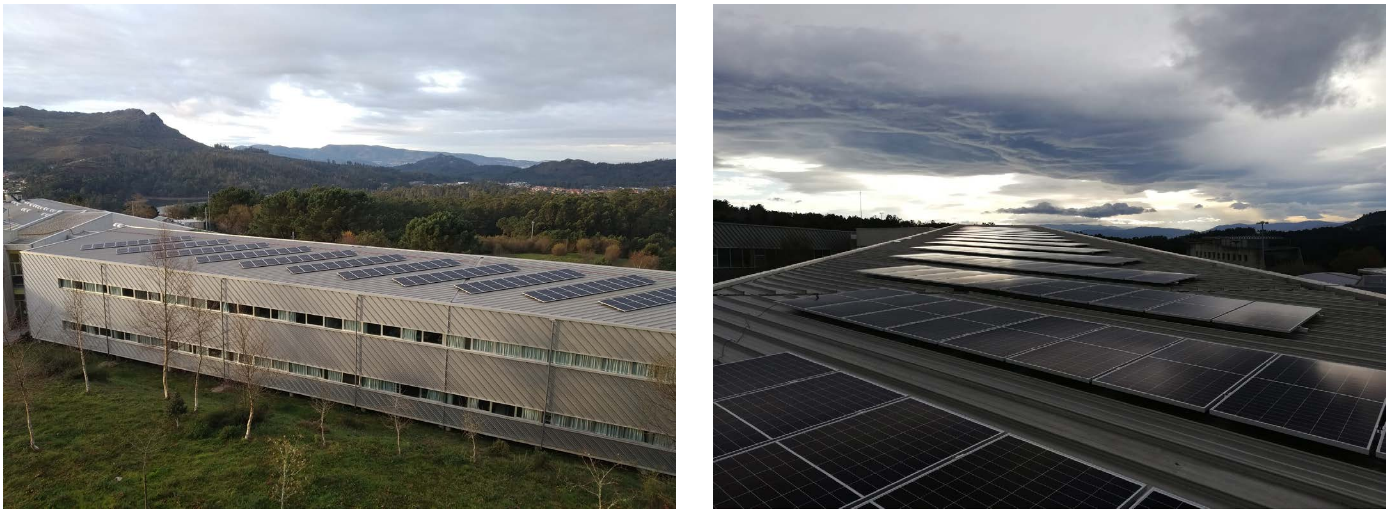

Fig. 2. Pictures of the PV installation analysed.

where k is the number of objectives, F_j^i the value of the i-th element for objective j, and $F_j^{\max}$; $F_j^{\min}$ the maximum and minimum values for objective j.

- Elitism: The best option combinations pass directly pass to next generations of the algorithm. Non-dominated combinations continue until another solution dominate them.
- Selection operator: The selection of elements to be transferred for next generations is based on their rank and their crowding distances.

The objective functions considered to be minimised with NSGA-II are the CV(RMSE) in PV generation predictions and a complexity function that summarises the layers and neurons of the model. This complexity function, already use in [31], [46], relies on the number of layers and neurons in the built neural network:

$$\text{Complexity} = 0.25 \times \frac{l}{L} + 0.75 \times \frac{\sum_{j=1}^{L} n_j}{N} \tag{10}$$

with l and L being the number of layers used and the maximum value allowed (in this analysis, 5). In addition, n_j and N represent the neurons in each layer and the maximum number of neurons allowed (in this analysis, 500). In order to avoid rejecting excessive multilayer architectures, a lower weighting for the number of layers is introduced. In this case, the termination of the optimisation process is based on a specific tolerance value within the space of feasible solutions and the optimal point along the final Pareto front is selected using a decomposition function, known as penalty boundary intersection (PBI) [47].

Further information about the NSGA-II can be found in [48].

C. Validation and Error Assessment

The validation metrics considered in this analysis to evaluate the accuracy of the deep learning models are the the Coefficient of Variation of the Root Mean Square Error (CV(RMSE)), Normalised Mean Biased Error (NMBE) and Mean Absolute Error (MAE):

$$\text{CV(RMSE)} = 100 \times \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 / N}}{\bar{y}} \tag{11}$$

$$\text{NMBE} = 100 \times \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)}{\sum_{i=1}^{N}(y_i)} \tag{12}$$

$$\text{MAE} = \frac{\sum_{i=1}^{N} | y_i - \hat{y}_i |}{N} \tag{13}$$

where y_i represents the real values, $\hat{y}_i$ the estimations and N the number of observations. These metrics are used to compare the performance of the built LSTM neural networks throught a cross-validation process (considering an expanding window) with average results presented in the section IV. They were used in similar studies such as [24], [49], [50]. Moreover, the accuracy of the models is assessed only considering the hours with positive solar irradiance (without irradiance it is known that the panels do not produce).

III. Experimental System

The studied PV system is an installation located on the roof of the School of Mining Engineering in north-western Spain at University of Vigo (see Fig. 2).

This installation is composed by 296 PV modules in parallel, with an azimuth of 72.8º-112.6º, because two groups of modules are considered, and a slope of 2º. In addition, the specific coordinates of the installations are latitude of N 42º 10' 6.1'' and longitude of W 8º 41' 18.44''. The technical information about the inverters and PV modules of the analysed installation is presented in Table I.

TABLE I. PV Inverters and Modules Datasheets

Inverter	$V_{DC,max}$	1000 V
	$V_{DC,MPP}$	500 - 800 V
	$I_{DC,max}$	120 A
	$I_{SC,max}$	30 A
	$V_{AC,nom}$	230 V
	f_{nom}	50 Hz
	$I_{AC,max}$	72.5 A
PV module	P_{MPP}	400 W
	Clasification range	0/+5 W
	Accuracy (P_{MPP})	± 3%
	U_{MPP}	40.32 V
	I_{MPP}	9.92 A
	U_{OC}	400 V
	I_{SC}	10.45 A

A. Synthetic Data

In this study, the available simulated data is generated with the software TRNSYS [51], [52]. The data consist of simulated photovoltaic generation based on physical laws and weather data significantly correlated with PV generation (in this case outdoor temperature and solar irradiance) along one year (see Fig. 3). The aforementioned simulation, considering the same weather conditions, is carried out for different PV installations considering different number of PV modules in parallel, different azimuths and different slopes. The number of PV modules varies between 60 and 740 (60, 89, 178, 296, 414, 562, 740), the azimuth between 0 and 337.5 degrees (22.5 by 22.5) and the slope between 0 and 90 degrees (15 by 15) generating data from 784 different PV instalations. Among this grid of parameters combinations there is the same configuration as the analysed installation.

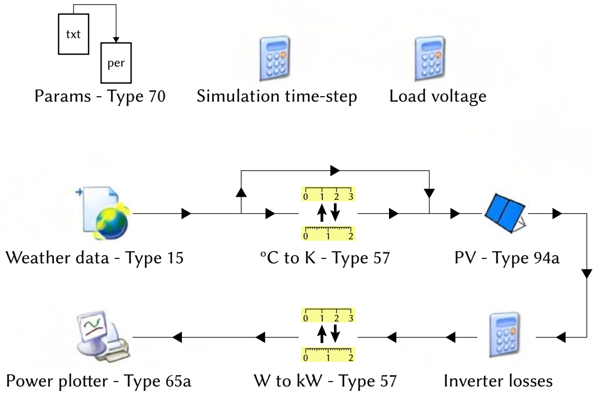

Fig. 3. Simulation process followed by TRNSYS in order to generate PV generation data.

The purpose of these synthetic data is to provide data from different installations (to fit a wide range of possibilities) in order to subsequently pre-train deep learning models and improve their performance on real data. Thus, the deep learning model reaches the training process, with real data, knowing the relationship between the selected inputs and the specific power generation of the installed panels.

B. Weather Data

The meteorological variables considered as model inputs in this analysis are global solar irradiance and outdoor temperature. They have a significant correlation with the generation of the photovoltaic modules [53]. Specifically, the data source used to obtain these data is an automatic weather station belonging to a meteorological agency known as MeteoGalicia [54]. The station is located 250 m northeast of the centre of the PV installation and 35 m higher. For missing or invalid values collected by the station, the Global Forecast System (GFS flux) surface flux model is used. This model generates hourly forecasts on a 13 km resolution grid [55].

C. Data Preprocessing

This research is focused on analising the improvement, on PV generation estimations, that produces pre-training a deep learning model with simulated data (see Fig. 4). In addition to the right installation parameters, the deep learning model is also pre-trained with simulated data based on random parameters (extracted from the list of section A) to consider the case where these data are not available. As mentioned, the aim of the built LSTM neural network is to predict the generation of a PV installation. The data available in this analysis are hourly observations of the PV generation of the studied installation and simulated observations, considering the parameters of that installation and 783 variations of them (Section A), in addition to the solar irradiance and outdoor temperature of the area. The availability of the real data corresponds to the year 2021 (from March to September) and the simulated data corresponds to 2020. In this period of time there is no missing or invalid data. Three complementary variables related with the time (hour of the day, day of the month and month of the year) are also considered as model inputs to improve the accuracy of the model. In order to take into account the existing inertia in the solar irradiance, and thus in PV generation, 24 hourly lags are considered. Moreover, the data set is normalised based on the limits 0 and 1.

As mentioned, the pretaining is conducted with simulated data considering on the one hand the parameters of the studied installations (n_panels: 296, azimuth: 90° and slope: 2°) and, on the other hand, a random set of parameters n_panels: 562, azimuth: 247.5° and slope: 60°). Specifically, these parameters are the number of modules, their azimuth and their slope. The following section presents two analyses: one focused on introducing the process of selecting the optimal

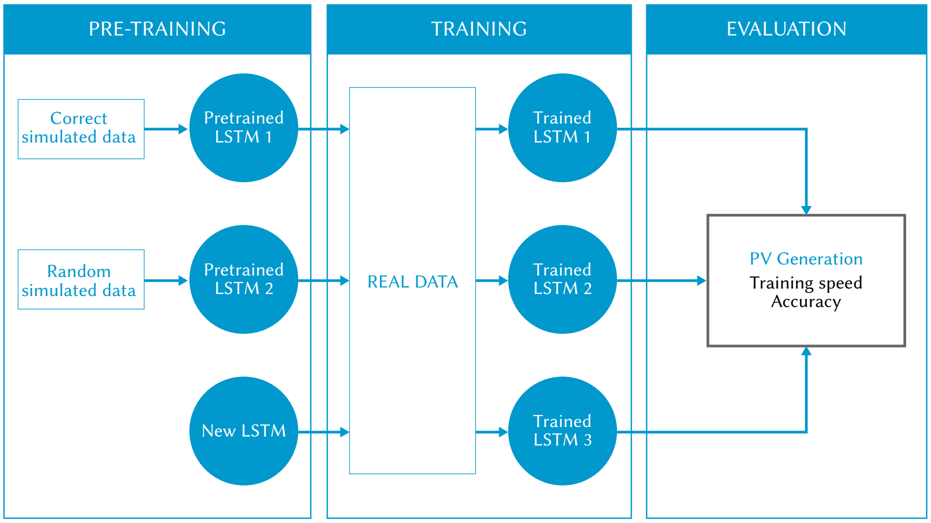

Fig. 4. Research methodology in which the three parts (pre-training with synthetic data, training with real data and the subsequent evaluation) are presented in different ways depending on the pre-training. In addition, the measures selected to compare the performance of the models, which are training speed and accuracy, are shown.

architecture and stopping criterion of the deep learning model and the other focused on showing the improvement in training speed and model accuracy due to pre-training (see Fig. 4).

IV. Results and Discussion

This paper presents a methodology for estimating PV generation optimised with a genetic algorithm that searches for the best LSTM neural network architecture together with the best stopping criterion for training and improved with pre-training based on simulated data. The inputs to produce the PV generation estimations of the built models are solar irradiance, outdoor temperature and three temporal variables. To validate this methodology, monitored data from a photovoltaic installation located in the northwest of Spain are available. On the one hand, section A shows the results and parameters used in the optimisation process of the LSTM neural network architecture and its stopping criterion. On the other hand, section B presents the improvements obtained by pre-training the model with synthetic data based on speed and accuracy. In this section, a comparison between two different pre-training and no pre-training is presented by analysing the number of epochs needed to reach certain error levels (directly related with time) and the accuracy they yield with similar training (same stopping criterion). In addition, the proposed optimisation and method, along with the following results, were implemented using the Python programming language [56].

A. LSTM Neural Network Optimisation

The optimal selection of the LSTM neural network architecture (considering LSTM hidden layers, dense hidden layers and the neurons within them) and the number of epochs without enhancement to stop training is obtained with NSGA-II. The average CV(RMSE), from a cross-validation experiment on the simulated sample corresponding to the studied system, and the complexity of the model (Equation 10) are the objective functions considered. The aim of the multi-objective genetic algorithm is to minimise these functions simultaneously. The optimisation is conducted with simulated data instead of real data, in order to assume the situation where real data is not yet available. The table II shows the specific hyperparameters used in the optimisation process: those that define the option space and those that configure the algorithm termination and selection process.

TABLE II. Parameters and Functions Used in the Optimisation Through NSGA-II, Comprising the General Parameters Related to the Multiobjective Algorithm and the Specific Parameters Related to the Optimal Selection

General parameter	Value	Termination parameter	Value
Neurons options	20 100 (20 by 20)	**Tolerance (*tol*)**	0.1
LSTM layer options	1 3	**Nº max evals (*n_max*)**	5000
Dense layer options	0 2	**Last genes considered (*n_last*)**	40
Patience options	10 or 20 epochs	**Decomposition function**	PBI
Population	50		
Mutation	0.9		
Crossover	0.1		

In this case, considering the parameters presented in Table II, NSGA-II needed 2490 evaluations to find 5 optimal points on the Pareto front (7688 possible options in total). These points correspond to LSTM architectures and epoch limits to stop the model training. Then, the PBI decomposition function taking into account heterogeneous weights (0.75{0.25), respectively for the error and complexity objective functions, is used to select a point on the Pareto front. Although in this case we give more importance to error, the distribution of weights can be adapted to obtain less accurate but simpler models.

The results are an LSTM neural network architecture with an LSTM hidden layer with 80 neurons and a Dense hidden layer with 40 neurons (5 80 40 1), as well as a model patience, measured in epochs, of 20. More information and details of this selection process can be found in [31], [57], [58].

B. LSTM Neural Network Performance

Once the optimal LSTM architecture and the stopping criterion for training the model have been obtained, two different analyses are performed: one based on analysing the improvement in training speed produced by a model pre-training and the other focused on comparing the differences in accuracy between the built models considering the same training process. In this case, the comparison is carried out considering three different models: one without pre-training, one with a random pre-training and one pre-trained with the parameters of the studied installation (number of PV modules, azimuth and slope). In this specific analysis, the values of the random parameters selected are 562 PV modules with an azimuth of 247.5° and a slope of 60°.

On the one hand, Table III, in which the training speed is analysed, shows the average results (30 repetitions) of measuring the number of epochs each model requires to reach certain error limits, also taking into account the time, measured in seconds, required to reach it. Normalised data and the Mean Squared Error (MSE), for error limits, are considered. The pre-trained models use one year of simulated data and all built models are retrained and evaluated with seven months of real data (first 4 for training and the remaining for validation).

TABLE III. Average Results of 30 Repetitions of an Experiment in Which the Number of Epochs Needed by Each Model to Reach the Error Limits Shown Are Analysed. The Average Number of Epochs and Time Each Model Needed to Reach the Limits Are Presented

MSE Limits	Correct pre-training		Random pre-training		No pre-training	
	Epochs [n]	Time [s]	Epochs [n]	Time [s]	Epochs [n]	Time [s]
0.005	1	4.86	1	4.88	3.90	8.00
0.004	1	4.41	1	4.61	5.27	9.51
0.003	1	4.45	1	4.42	8.43	12.54
0.002	3.33	7.41	18	25.51	42.67	46.11

In the case of the first limits (0.005, 0.004, 0.003), both pre-trained models with synthetic data only needed one epoch to reach the limit. The model without pre-trainig is the slowest, needing more than 3, 5 and 8 epochs on average to reach respectively the first mentioned limits. Considering the times spent on training, the pre-trained models reduce it to half at the first limit and to one third at the third limit. With regard to the last error limit, the differences between the three built models become more significant. The model with the correct pre-training requires on average 3.33 epochs to reach the error limit, while the model with a random pre-training requires 18 epochs. Moreover, the model without pre-training remains the slowest, taking, on average, 42.67 epochs. Observing the times the results are similar: the model with a correct pre-training spent, on average, 7.41 seconds (more than three times less than the model pre-train with random parameters (25.51) and more than six times less than the model without pre-training (46.11). In this way, it can be seen that the improvements, in terms of speed, provided by a pre-training with synthetic data are significant considering both correct and incorrect parameters. The information extracted in this pre-training generates models able to adapt faster to real situations, although considering the right pre-training is more efficient.

On the other hand, the results of the study of the accuracy of the built models following a homogeneous training process are presented in Table IV. The training process is based on a cross-validation experiment considering an expanding window; the models are evaluated on the seven months of real data available (one by one) using the remaining previous months for training. In addition, the LSTM architecture and the stopping criterion considered are those obtained in the previous section.

In terms of CV(RMSE), with which the average distance to the real curve is measured, it can be observed in Table IV that the average value of the model with a random pre-training is significantly higher than the others (21.15 %). The standard deviation among all CV(RMSE) values is also the highest (± 5.87), showing a large variability in the results. The average CV(RMSE) yielded by the model with correct pre-training and the one without pre-training are close, with similar variability, but the former is lower (12.84 % and 14.22 % respectively). As for the NMBE results, which measure how close the estimations are on average to reality, the model with incorrect pre-training has the highest average value (0.10 %) and the highest variability in results (0.08) (see Table 4). In this case, the model with no pre-training presents the lowest value (0.07 %) followed by the model pre-trained with correct parameters (0.07 %), both with controlled variances. Regarding the MAE results, a metric that measures the average distance to the real values but in absolute units, the situation is the same as in the previous errors. The model pre-trained with correct simulated data yields the lowest value (3.71 kW ±1.18), followed by the model without pre-training (4.17 kW ± 1.03) and the model with a random pre-training (6.37 % ± 1.96).

TABLE IV. Average Results of a Cross-Validation Experiment Considering an Expanding Window and Considering the Accuracy of the Models. The Average CV(RMSE), the Average NMBE and the Average MAE Are Presented Together With Their Standard Deviations (SD)

Pre-training	CV(RMSE) [%]	SD	NMBE [%]	SD	MAE [kW]	SD
Correct	12.84	3.22	0.07	0.04	3.71	1.18
Random	21.15	5.87	0.10	0.08	6.37	1.96
None	14.22	3.21	0.06	0.04	4.17	1.03

Moreover, Fig. 5 shows the performance of the three built models over an entire week (specifically, from 12 July 2022 to17 July 2022) and considering the similar training process used for the previous accuracy analysis. It can be seen that, as presented in Table IV and considering the CV(RMSE) results, the model with correct pre-training best replicate the real behaviour of the studied PV installation. Although the model with no pre-training exhibits an accuracy not too far from the model just mentioned, the model pre-trained with incorrect simulated data is far from the real data.

In short, it is demonstrated from a speed and accuracy approach that pre-training the deep learning model with synthetic data is an effective way to improve its performance. Furthermore, in order to efficiently get this improvement, it is important to use correct simulated data. Pre-training with appropriate synthetic data allows to reduce the number of epochs, and thus the computational time, required in the training process by more than six times compared with no pre-training (see Table III). However, the use of incorrect simulated data, although faster than the model without pre-training, increases the computational time required in training by more than three times compared with the model correctly pre-trained. With respect to the accuracy of the models based on similar trainings, the use of incorrect simulated data, again, generates a model significantly less accurate than the model with a correct pre-training, but also than the model without pre-training (see Table IV). In this case, focusing only on the final average errors, both the model with no pre-training and the model with the correct pre-training show a similar performance, although the model pre-trained with the correct synthetic data achieved lower errors. These results show that although pre-training with synthetic data can provide more speed adapting to real data, if the data used is not appropriate, the accuracy of the model can stagnate and not reach the levels that would be achieved without pre-training.

Comparing the results of the proposed research with previous similar studies, taking into account the differences between installations and the improvements shown by the pre-trainings, the built models show error values lower or in the same range [59]–[61] and complying with the ASHRAE Guidelines [61], [62].

V. Conclusions

A methodology for optimising deep learning model configurations and improving their performance by means of pre-training based on synthetic data is presented in this paper. In this way, a great time reduction can be obtained, not only considering the reduction in the

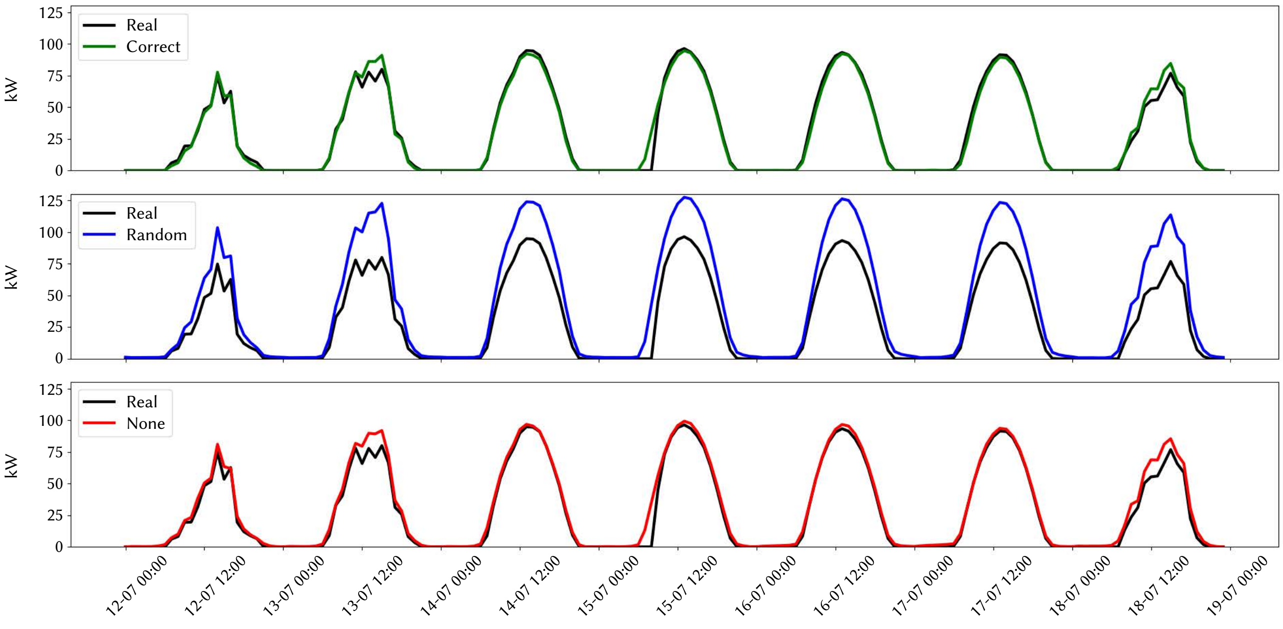

Fig. 5. Results of PV generation estimations in the second week of July 2022. All models built are considered for comparison. The CV(RMSE) values are 10:61 % for the model with correct pre-training, 47:63 % for the model with random pre-training and 13:31 % for the model without pre-training.

model training time but also in the time devoted to the search for the optimal architecture and the optimal training stopping criterion of the deep learning model. The study was conducted with an LSTM neural network built to perform PV generation predictions and pre-trained using synthetic data acquired with TRNSYS software.

On the one hand, the results achieved demonstrate that it is possible to pre-train a deep learning model, both with data simulated using the correct parameters and using random parameters, significantly reducing the time (measured in epochs and seconds) spent in the training process. The computational time required for the model to reach specific training error values is reduced by up to six times. In addition, in relation to the optimisation of DL model configurations, the proposed method (based on a multiobjective genetic algorithm) also reduces to less than half the evaluations needed to search all possible configurations and select an optimal one. On the other hand, the impact of using synthetic data generated with erroneous parameters is also analysed. In this case, an inadequate pre-training not only does not come close to the performance of a correct pre-training, but even can worsens the situation without pre-training. With regard to the accuracy of the built models considering the same training process on real data, it is shown that an incorrect pre-training produces less accurate models when fed with real data than a correct pre-training or a model without pre-training. The former two show a similar final accuracy but the model pre-trained with data simulated considering the correct parameters yields lower average errors. Here is a key insight of the research: although a pre-training with synthetic data may provide higher speed of adaptation to reality, if the data used in this pre-trainig are far from the real situation, it will affect to the final accuracy of the model and even lead to a worse performance compared to a model without pre-training.

The main limitation of this research is the amount of data. The monitoring period could be longer to reach a full year and the availability of data from more PV installations would make this study more consistent. The main outcome of this study is the evidence that the presented methodology can contribute to improve the performance of deep learning models. First, the multiobjective genetic algorithm NSGA-II allows us to use an efficiently optimised LSTM neural network without the need to evaluate all possible hyperparameter options. Second, the use of synthetic data to pre-train the built model allows us to significantly reduce the time spent on training and even slightly improve the final accuracy of the model. In this way, these improvements can be focused on making the use and distribution of photovoltaic energy more efficient. Thus, the fulfilment of the European Commission targets, commented at the beginning of the paper, will be closer. Lastly, this research evidences the importance of selecting adequate datasets for pre-training and generating global models that, once trained with simulated data, are used in real PV installations.

As future lines of research, more installations based on different parameters and different deep learning models could be considered to develop a more complete comparison and analysis. Using more installations to pre-train the deep learning model, or plug it in with a model that estimates the correct installation parameters from monitored data, can generate a global model that can be applied to different installations instead of having to pre-train the model with data specific to the particular installation under study.

Acknowledgments

This research was supported by the Ministry of Science, Innovation and Universities of the Spanish government under the DEEPSMART project (PID2021-126739OB-C21). The authors also want to thank the Ministry of Science, Innovation and Universities for grant FPU19/01187. Funding for open access charge: Universidade de Vigo/CISUG.

References

[1] A. Abdelkader, A. Rabeh, D. Mohamed Ali, J. Mohamed, “Multi-objective genetic algorithm based sizing optimization of a stand-alone wind/pv power supply system with enhanced battery/supercapacitor hybrid energy storage,” *Energy*, vol. 163, pp. 351–363, 2018, doi: https://doi.org/10.1016/j.energy.2018.08.135.

[2] J. Munkhammar, J. D.K. Bishop, J. J. Sarralde, W. Tian, R. Choudhary, “Household electricity use, electric vehicle home-charging and distributed photovoltaic power production in the city of westminster,” *Energy and Buildings*, vol. 86, pp. 439–448, 2015, doi: https://doi.org/10.1016/j.enbuild.2014.10.006.

[3] R. Fachrizal, M. Shepero, D. Van der Meer, J. Munkhammar, J. Widén, “Smart charging of electric vehicles considering photovoltaic power production and electricity consumption: A review,” *eTransportation*, vol. 4, p. 100056, 2020, doi: https://doi.org/10.1016/j.etran.2020.100056.

[4] D. B. Richardson, “Electric vehicles and the electric grid: A review of modeling approaches, impacts, and renewable energy integration,” *Renewable and Sustainable Energy Reviews*, vol. 19, pp. 247–254, 2013, doi: https://doi.org/10.1016/j.rser.2012.11.042.

[5] A. Anand, A. Shukla, H. Panchal, A. Sharma, “Thermal regulation of photovoltaic system for enhanced power production: A review,” *Journal of Energy Storage*, vol. 35, p. 102236, 2021, doi: https://doi.org/10.1016/j.est.2021.102236.

[6] A. R. Jordehi, “Parameter estimation of solar photovoltaic (pv) cells: A review,” *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 354–371, 2016, doi: https://doi.org/10.1016/j.rser.2016.03.049.

[7] S. Theocharides, G. Makrides, A. Livera, M. Theristis, P. Kaimakis, G. E. Georghiou, “Day-ahead photovoltaic power production forecasting methodology based on machine learning and statistical post-processing,” *Applied Energy*, vol. 268, p. 115023, 2020, doi: https://doi.org/10.1016/j.apenergy.2020.115023.

[8] S. Dwyer, S. Teske, “Renewables 2018 global status report,” *Renewables 2018 Global Status Report*, 2018.

[9] I. Renewable Energy Statistics, “International renewable energy agency,” *Renewable Energy Target Setting, Abu Dhabi, UAE*, 2015.

[10] E. Bueno, P. d. S. Vicente, T. C. Pimenta, E. R. Ribeiro, “Photovoltaic array reconfiguration strategy for maximization of energy production,” *International Journal of Photoenergy*, vol. 2015, p. 592383, 2015, doi: 10.1155/2015/592383.

[11] L. Prado Osco, J. Marcato Junior, A. P. Marques Ramos, L. A. de Castro Jorge, S. Narges Fatholahi, J. de Andrade Silva, E. T. Matsubara, H. Pistori, W. N. Gonçalves, J. Li, “A review on deep learning in uav remote sensing,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 102, p. 102456, 2021, doi: https://doi.org/10.1016/j.jag.2021.102456.

[12] M. Martínez-Comesaña, L. Febrero-Garrido, E. Granada-Álvarez, J. Martínez-Torres, S. Martínez-Mariño, “Heat loss coefficient estimation applied to existing buildings through machine learning models,” *Applied Sciences*, vol. 10, no. 24, 2020, doi: 10.3390/app10248968.

[13] M. Nabipour, P. Nayyeri, H. Jabani, S. S., A. Mosavi, “Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis,” *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[14] L. Mert, “Agnostic deep neural network approach to the estimation of hydrogen production for solar-powered systems,” *International Journal of Hydrogen Energy*, vol. 46, no. 9, pp. 6272–6285, 2021, doi: https://doi.org/10.1016/j.ijhydene.2020.11.161.

[15] P. Dhanith, B. Surendiran, S. Raja, “A word embedding based approach for focused web crawling using the recurrent neural network,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 122-132, 2021.

[16] M. Martínez-Comesaña, L. Febrero-Garrido, F. Troncoso-Pastoriza, J. Martínez-Torres, “Prediction of building’s thermal performance using lstm and mlp neural networks,” *Applied Sciences*, vol. 10, no. 21, 2020, doi: 10.3390/app10217439.

[17] R. R. Kumari, V. V. Kumar, K. R. Naidu, “Optimized dwt based digital image watermarking and extraction using rnn-lstm,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 2, pp. 150-162, 2021.

[18] X.-H. Le, H. V. Ho, G. Lee, S. Jung, “Application of long short-term

memory (lstm) neural network for flood forecasting," *Water*, vol. 11, no. 7, 2019, doi: 10.3390/w11071387.

[19] M. Chai, F. Xia, S. Hao, D. Peng, C. Cui, W. Liu, "Pv power prediction based on lstm with adaptive hyperparameter adjustment," *IEEE Access*, vol. 7, pp. 115473–115486, 2019, doi: 10.1109/ACCESS.2019.2936597.

[20] M. Khodayar, M. E. Khodayar, S. Mohammad Jafar Jalali, "Deep learning for pattern recognition of photovoltaic energy generation," *The Electricity Journal*, vol. 34, no. 1, p. 106882, 2021, doi: https://doi.org/10.1016/j.tej.2020.106882.

[21] M. S. Hossain, H. Mahmood, "Short-term photovoltaic power forecasting using an lstm neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172524–172533, 2020, doi: 10.1109/ACCESS.2020.3024901.

[22] M. Abdel-Nasser, K. Mahmoud, "Accurate photovoltaic power forecasting models using deep lstm-rnn," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2727–2740, 2019, doi: 10.1007/s00521-017-3225-z.

[23] L. Yang, A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: https://doi.org/10.1016/j.neucom.2020.07.061.

[24] M. Martínez-Comesaña, P. Eguía-Oller, J. Martínez-Torres, L. Febrero-Garrido, E. Granada-Álvarez, "Optimisation of thermal comfort and indoor air quality estimations applied to in-use buildings combining nsga-iii and xgboost," *Sustainable Cities and Society*, vol. 80, p. 103723, 2022, doi: https://doi.org/10.1016/j.scs.2022.103723.

[25] Y. Yoo, "Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches," *Knowledge-Based Systems*, vol. 178, pp. 74–83, 2019, doi: https://doi.org/10.1016/j.knosys.2019.04.019.

[26] Y. Novaria Kunang, S. Nurmaini, D. Stiawan, B. Yudho Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *Journal of Information Security and Applications*, vol. 58, p. 102804, 2021, doi: https://doi.org/10.1016/j.jisa.2021.102804.

[27] Y. Guo, J.-Y. Li, Z.-H. Zhan, "Efficient hyperparameter optimization for convolution neural networks in deep learning: A distributed particle swarm optimization approach," *Cybernetics and Systems*, vol. 52, no. 1, pp. 36–57, 2021, doi: 10.1080/01969722.2020.1827797.

[28] S. Lee, J. Kim, H. Kang, D.-Y. Kang, J. Park, "Genetic algorithm based deep learning neural network structure and hyperparameter optimization," *Applied Sciences*, vol. 11, no. 2, 2021, doi: 10.3390/app11020744.

[29] H. Chung, K.-S. Shin, "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7897–7914, 2020, doi: 10.1007/s00521-019-04236-3.

[30] Á. A. Domingo, M. A. Ezquerra, "Gpgpu implementation of a genetic algorithm for stereo refinement," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, no. 2, pp. 69–76, 2015.

[31] M. Martínez-Comesaña, A. Ogando-Martínez, F. Troncoso-Pastoriza, J. López-Gómez, L. Febrero-Garrido, E. Granada-Álvarez, "Use of optimised mlp neural networks for spatiotemporal estimation of indoor environmental conditions of existing buildings," *Building and Environment*, vol. 205, p. 108243, 2021, doi: https://doi.org/10.1016/j.buildenv.2021.108243.

[32] R. Damaševičius, M. Gupta, N. Kumar, B. K. Singh, N. Gupta, "Nsga-iii-based deep-learning model for biomedical search engines," *Mathematical Problems in Engineering*, vol. 2021, p. 9935862, 2021, doi: 10.1155/2021/9935862.

[33] P. Trampert, D. Rubinstein, F. Boughorbel, C. Schlinkmann, M. Luschkova, P. Slusallek, T. Dahmen, S. Sandfeld, "Deep neural networks for analysis of microscopy images—synthetic data generation and adaptive sampling," *Crystals*, vol. 11, no. 3, 2021, doi: 10.3390/cryst11030258.

[34] J. Liu, J. Gu, H. Li, K. H. Carlson, "Machine learning and transport simulations for groundwater anomaly detection," *Journal of Computational and Applied Mathematics*, vol. 380, p. 112982, 2020, doi: https://doi.org/10.1016/j.cam.2020.112982.

[35] K. Antczak, "Deep recurrent neural networks for ecg signal denoising," 2018. [Online]. Available: https://arxiv.org/abs/1807.11551, doi: 10.48550/ARXIV.1807.11551.

[36] Q. Wang, J. Gao, W. Lin, Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] J. C. Balloch, I. Agrawal, Varun Essa, S. Chernova, "Unbiasing semantic segmentation for robot perception using synthetic data feature transfer," 2018. [Online]. Available: https://arxiv.org/abs/1809.03676, doi: 10.48550/ARXIV.1809.03676.

[38] Y. Yu, X. Si, C. Hu, J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[39] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020, doi: https://doi.org/10.1016/j.physd.2019.132306.

[40] Y. Hu, A. E. G. Huber, J. Anumula, S. Chii Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," *CoRR*, vol. abs/1801.06105, 2018.

[41] S. Jha, A. Dey, R. Kumar, V. Kumar, "A novel approach on visual question answering by parameter prediction using faster region based convolutional neural network," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 30–37, 2019.

[42] A. H. Elkasem, S. Kamel, A. Rashad, F. J. Melguizo, "Optimal performance of doubly fed induction generator wind farm using multi-objective genetic algorithm," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 48–53, 2019.

[43] S. Katoch, S. S. Chauhan, V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.

[44] A. Vukadinović, J. Radosavljević, A. Đorđević, M. Protić, N. Petrović, "Multi-objective optimization of energy performance for a detached residential building with a sunspace using the nsga-ii genetic algorithm," *Solar Energy*, vol. 224, pp. 1426–1444, 2021, doi: https://doi.org/10.1016/j.solener.2021.06.082.

[45] S. Wang, D. Zhao, J. Yuan, H. Li, Y. Gao, "Application of nsga-ii algorithm for fault diagnosis in power system," *Electric Power Systems Research*, vol. 175, p. 105893, 2019, doi: https://doi.org/10.1016/j.epsr.2019.105893.

[46] M. A. J. Idrissi, H. Ramchoun, Y. Ghanou, M. Ettaouil, "Genetic algorithm for neural network architecture optimization," in *2016 3rd International Conference on Logistics Operations Management (GOL)*, 2016, pp. 1–4.

[47] Y. Wu, J. Wei, W. Ying, Y. Lan, Z. Cui, Z. Wang, "A collaborative decomposition-based evolutionary algorithm integrating normal and penalty-based boundary intersection methods for many-objective optimization," *Information Sciences*, vol. 616, pp. 505–525, 2022, doi: https://doi.org/10.1016/j.ins.2022.10.136.

[48] S. Verma, M. Pant, V. Snasel, "A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems," *IEEE Access*, vol. 9, pp. 57757–57791, 2021, doi: 10.1109/ACCESS.2021.3070634.

[49] F. Troncoso-Pastoriza, M. Martínez-Comesaña, A. Ogando-Martínez, J. López-Gómez, P. Eguía-Oller, L. Febrero-Garrido, "Iot-based platform for automated ieq spatio-temporal analysis in buildings using machine learning techniques," *Automation in Construction*, vol. 139, p. 104261, 2022, doi: https://doi.org/10.1016/j.autcon.2022.104261.

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

[51] U. O. W.-M. Solar Energy Laboratory, *TRNSYS, a transient simulation program*. Madison, Wis. : The Laboratory, 1975., 1975.

[52] A. Remlaoui, D. Nehari, M. Laissaoui, A. M. Sandid, "Performance evaluation of a solar thermal and photovoltaic hybrid system powering a direct contact membrane distillation: Trnsys simulation," *Desalin. Water Treat*, vol. 194, pp. 37–51, 2020.

[53] J. López Gómez, A. Ogando Martínez, F. Troncoso Pastoriza, L. Febrero Garrido, E. Granada Álvarez, J. A. Orosa García, "Photovoltaic power prediction using artificial neural networks and numerical weather data," *Sustainability*, vol. 12, no. 24, 2020, doi: 10.3390/su122410295.

[54] X. de Galicia, "Meteogalicia," 2021. [Online]. Available: https://www.meteogalicia.gal/web/RSS/ rssIndex.action, Last access: 12 September 2022.

[55] NOAA, "The global forecast system (gfs) - global spectral model (gsm)," 2021. [Online]. Available: https://www. emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gfs/documentation.php, Last access: 12 September 2022.

[56] G. Van Rossum, F. L. Drake Jr, "Python/c api reference manual," *Python Software Foundation*, 2002.

[57] S. Martínez, E. Pérez, P. Eguía, A. Erkoreka, E. Granada, "Model calibration and exergoeconomic optimization with nsga-ii applied to a residential cogeneration," *Applied Thermal Engineering*, vol. 169, p. 114916, 2020, doi: https://doi.org/10.1016/j.applthermaleng.2020.114916.

[58] A. E. I. Brownlee, J. A. Wright, "Constrained, mixed-integer and multi-objective optimisation of building designs by nsga-ii with fitness approximation," *Applied Soft Computing*, vol. 33, pp. 114–126, 2015, doi: https://doi.org/10.1016/j.asoc.2015.04.010.

[59] M. K. Park, J. M. Lee, W. H. Kang, J. M. Choi, K. H. Lee, "Predictive model for pv power generation using rnn (lstm)," *Journal of Mechanical Science and Technology*, vol. 35, no. 2, pp. 795–803, 2021, doi: 10.1007/s12206-021-0140-0.

[60] B. Kim, D. Suh, "Solar photovoltaic generation forecasting using machine learning methods," *The Journal of Contents Computing*, vol. 2, no. 1, pp. 105–112, 2020.

[61] B. Kim, D. Suh, "A hybrid spatio-temporal prediction model for solar photovoltaic generation using numerical weather data and satellite images," *Remote Sensing*, vol. 12, no. 22, 2020, doi: 10.3390/rs12223706.

[62] ANSI/ASHRAE, "Measurement of energy and demand savings," in *ASHRAE Guideline 14-2002*, vol. 8400, 2002, p. 170.

Miguel Martínez-Comesaña

Miguel Martínez Comesaña was born in Vigo (Spain). He holds a degree in Economics since 2017. He obtained his Master in Statistics from the University of Santiago de Compostela in 2019. He received his PhD in Artificial Intelligence from the University of Vigo in 2023. He is the author of several articles specialized in the application of AI in energy efficiency analysis. Currently, he is Data Scientist and Engineer at the University of Vigo, being part of the research group GTE (Energy Technology Group).

Javier Martínez Torres

Javier Martínez Torres is a Mathematician and Engineering PhD from the University of Vigo. He is currently an Assistant Professor at the University of Vigo and has participated in more than 20 research projects as principal investigator. He has published more than 60 papers in JCR indexed journals and participate in more than 35 international conferences.

Pablo Eguía Oller

Pablo Eguía Oller is an engineer and Engineering PhD from the University of Vigo. Main researcher in the Energy Technology research group (GTE) and has been working for 10 years on indoor air quality and energy efficiency in buildings in both European and national projects. More than 70 papers in JCR indexed journals.

Javier López-Gómez

Javier López-Gómez holds a PhD in Energy Efficiency from the University of Vigo. Specialized in the collection, processing and analysis of data applied to multiple fields of engineering (energy generation and consumption, meteorological phenomena, forest fires, environmental quality in buildings, or coastal marine flows).