Engineering Applications of Artificial Intelligence 116 (2022) 105458

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Convolutional-LSTM networks and generalization in forecasting of household photovoltaic generation

Rogério Luís de C. Costa

Computer Science and Communication Research Centre (CIIC), Polytechnic of Leiria, Leiria 2411-901, Portugal

ARTICLE INFO

Keywords:
Time series forecasting
Photovoltaic power generation
Deep learning
LSTM
Convolutional neural networks

ABSTRACT

Solar panels can generate energy to meet almost all of the energy needs of a house. Batteries store energy generated during daylight hours for future use. Also, it may be possible to sell extra electricity back to distribution companies. However, the efficiency of photovoltaic systems varies according to several factors, such as the solar exposition at ground levels, atmospheric temperature, and relative humidity, and predicting the energy generated by such a system is not easy.

This work is on the use of deep learning to predict the generation of photovoltaic energy by residential systems. We use real-world data to evaluate the performance of LSTM, Convolutional, and hybrid Convolutional-LSTM networks in predicting photovoltaic power generation at different forecasting horizons. We also assess the generalizability of the solutions, evaluating the use of models trained with data aggregated by geographic areas to predict the energy generation by individual systems.

We compare the performance of deep networks with Prophet in terms of MAE, RMSE, and NRMSE, and in most cases, Convolutional and Convolutional-LSTM networks achieve the best results. Using models trained with region-based data to predict the power generation of individual systems is confirmed to be a promising approach.

1. Introduction

Renewable energy sources are becoming more relevant in the energy mix (i.e., the proportion of individual energy sources used for electricity generation Ahn et al., 2021) of many countries. Renewable sources of electricity, like hydropower, wind, and solar power, generate more than a quarter of the generated electricity globally in 2021 (IEA, 2021).

Household solar energy systems generate electricity that may be used immediately to power a house, stored in batteries for later use, or even sold to grid systems. Solar energy generation forecasting on multiple scales has several applications, like power scheduling and grid balancing, which may reduce costs related to weather dependency. However, the performance of household solar energy systems is highly impacted by the solar exposition at ground levels, which depends on cloud cover variability, atmospheric aerosol levels, and indirectly and to a lesser extent, participating gases in the atmosphere (Inman et al., 2013). Wind speed, relative humidity, and the temperature of the ambient and solar cell are some other parameters that impact the performance of photovoltaic systems (Sharadga et al., 2020). Therefore, forecasting energy generation by household systems is relevant and challenging.

Most of the works on photovoltaic (PV) energy generation forecasting focus on solar radiation prediction, but this problem may be studied in the context of time series forecasting (Sharadga et al., 2020). There are various conventional methods for time series forecasting, but using machine learning over traditional ones that employ statistical techniques has shown promising results in recent years with reduced complexity.

Recurrent Neural Networks (RNNs) are the most widely used neural network type to exploit time-series data (Hewamalage et al., 2021a). They can transfer information across time steps, as their cells use a hidden state to capture the dependencies that exist in the data (Sharadga et al., 2020; Yunpeng et al., 2017). Long–short Term Memory (LSTM) is a kind of RNN with the ability to learn long dependencies within sequential data (chao Miao et al., 2020).

Convolutional neural networks (CNNs) are another type of deep network and have gained prominence for their use in image processing and pattern recognition. It has applications in time series forecasting in several domains, including energy and fuels, financial, environmental, industry, and health (Torres et al., 2021).

CNNs and LSTM are complementary in their capability to model time series (Xue et al., 2019). While LSTM is qualified for modeling the sequence temporally, CNN reduces frequency variation (Xue et al., 2019; Sainath et al., 2015). There exist several ways to combine CNNs, recurrent neural networks (e.g., LSTM), and deep fully connected neural networks (Deng and Platt, 2014), and their ensemble use in

E-mail address: rogerio.l.costa@ipleiria.pt.

https://doi.org/10.1016/j.engappai.2022.105458
Received 30 March 2022; Received in revised form 14 September 2022; Accepted 15 September 2022
Available online xxxx

Nomenclature

ANN	Artificial Neural Network
$ARIMA$	Autoregressive Integrated Moving Average
CNN	Convolutional neural network
DoE	Design of experiments
FNN	Feedforward neural network
$LSTM$	Long short-term memory
MAE	Mean Absolute Error
$MAPE$	Mean absolute percentage error
MLP	Multilayer Perceptron
$NRMSE$	Normalized Root Mean Square Error
$PReLU$	Parametric Rectified Linear Unit
PV	Photovoltaic
$ReLU$	Rectified Linear Unit
$RMSE$	Root Mean Square Error
RNN	Recurrent neural network
$SARIMA$	Seasonal Autoregressive Integrated Moving Average
SVM	Support Vector Machine
SVR	Support Vector Regression

the same framework may improve the performance of their individual use (Sainath et al., 2015). In this work, we evaluate the performance of LSTM, CNNs, and hybrid Convolutional-LSTM (Conv-LSTM) networks in photovoltaic energy generation forecasting.

A common drawback of deep learning models is the considerable effort to train them, which may turn it prohibitive to train a distinct model for each household energy generation system. On the other hand, adding the data of thousands of domestic systems, creating a region-based time series on energy generation, and then training the model on such aggregated data would require much less effort. However, it is crucial to know how a model built this way would perform when predicting the individual values of different domestic systems.

The questions this work is going to answer are: (a) Would convolutional and LSTM networks provide good predictions on real-world data on photovoltaic energy generation by individual systems and at the region level considering different forecasting horizons? (b) Would the use of convolutional layers and LSTM together in a hybrid model boost the prediction performance? (c) How would the performance of models trained with aggregated region-level data be when predicting the PV power generation of individual household systems?

We build models of three deep architectures composed of LSTM layers, convolutional layers, and fully connected neurons. We train those models and predict values on several series we generate based on real-world data on household energy generation in Sydney, Australia. The series contains data on power generation by individual household systems every half hour, day, and month, but also data on power generation in an entire region every half hour, day, and month. Models trained with the region data are used to forecast energy generation by individual systems and have performance compared with ones obtained when training the models with each system data. We also compare the performance of the deep models with the ones of Prophet (Taylor and Letham, 2018).

In the following section, we review some concepts and related work. Then, in Section 3, we describe our base methods, time series, and performance metrics. Section 4 presents experimental results. Finally, Section 5 contains the main conclusions and outlines future works.

2. Background and related work

Time series forecasting has been studied for several decades. The Autoregressive Integrated Moving Average (ARIMA) is one of the most used *classic* methods for univariate time series forecasting. ARIMA is a combination of an auto-regressive term, a moving-average term, and an integrating term. SARIMA (Seasonal Autoregressive Integrated Moving Average) is a variation of ARIMA that considers seasonal components. Other classic methods include Simple Moving Average and Exponential Smoothing. More recently, Prophet (Taylor and Letham, 2018) was proposed as a forecasting method that deals with seasonal effects and non-linear trends. In this work, we use Prophet as a baseline method.

2.1. Neural networks and deep learning

Artificial Neural Networks (ANNs) are computational systems inspired by human nervous systems. Such networks are made by interconnected processor elements called *neurons*. Neurons receive one or more inputs, process them and generate an output.

Fig. 1 presents a representation of a neuron. It receives one or more inputs (X_i) that may be data fed into the network or even the outputs from other neurons. To each input, there is an associate weight (W_i), which represents the strength of the connection between an input and a neuron. Inputs and weight are multiplied, summed, and passed to an activation function (φ), as represented in Eq. (1) (Dongare et al., 2012).

$$f(x_i, w_i) = \varphi(\sum_i (x_i * w_i)) \tag{1}$$

In ANNs, nodes are usually organized in layers, like the *input* and *output* layers. The first is activated through sensors perceiving the environment and thus gets the initial data from the environment into the network (Schmidhuber, 2015). The output layer generates the network's output. In some classes of ANNs, there are one or more layers between the input and output layers, which are called *hidden* layers. They receive weighted inputs and use an activation function to generate outputs. A Multilayer Perceptron (MLP) is an ANN composed of an input layer, an output layer, and at least one hidden layer, as represented in Fig. 2. This architecture would be capable of approximating any function when using a non-linear activation function (Hornik et al., 1989).

There are several classes and architectures of ANN with distinct types of neurons and interconnections. In *feedforward* neural networks (FNNs), the information flows in just one direction and there is no cycle or loops formed by node interconnections. In *recurrent neural networks* (RNNs), node connections form a direct graph. The information may flow in any direction through the use of loops in the network, and the network is capable of maintaining a state.

In *Deep learning*, layers are used to hierarchically learn complex features from simple ones. Deep neural networks have been around for decades. They became somewhat feasible in the 1990s, but it was in the 2000s that they became used in a wide number of applications outperforming AI methods based on kernel machines (Schmidhuber, 2015).

The *Long Short-Term Memory* (LSTM) is an RNN proposed in 1997 (Hochreiter and Schmidhuber, 1997) that became widely used as deep learning advanced. The basic unit of an LSTM network is the memory block (Gers et al., 2000). Originally, a memory block was composed by one or more *memory cells*, an *input gate* and an *output gate*. Memory cells 'remember' the temporal state of the network. Each memory cell contains a state cell that is responsible for storing the state value and a recurrent self-connection. The gates control the access to the cell. The input gate protects the cell from irrelevant input values (e.g., noise), while the output gate protects the network from being perturbed by the cell state. In Gers et al. (2000), the authors introduce the *forget gate*, which gradually resets memory blocks when their contents are out of date. Several applications use LSTM networks, including anomaly detection, speech recognition and time series forecasting (e.g., Han et al. (2017), Li et al. (2015), Malhotra et al. (2015), Ashraf et al. (2021), Pinho et al. (2019), Chimmula and Zhang (2020) and chao Miao et al. (2020)).

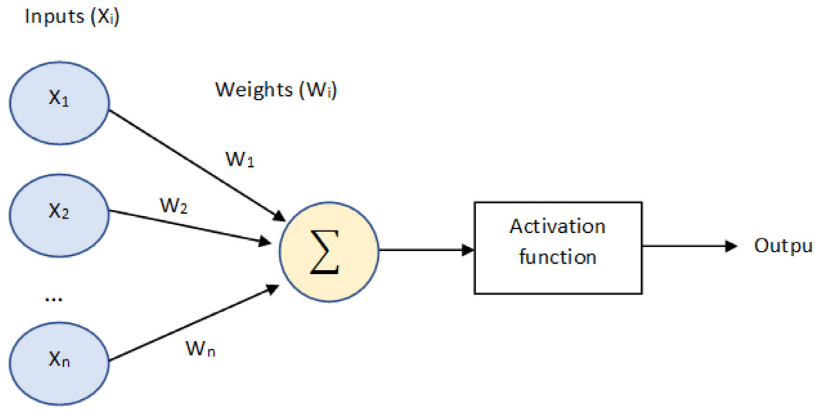


Fig. 1. Neuron representation.

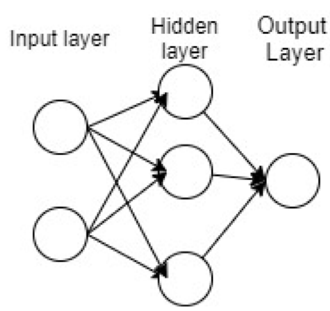


Fig. 2. Typical structure of an MLP.

Convolutional neural networks (CNNs) are another widely used deep learning network. CNNs are feedforward networks whose input is a tensor. They are composed of several hidden convolutional layers, which may be combined with layers of other types, like *pooling* layers. Convolutional networks are widely used in image and signal processing, and audio classification (e.g., Caldeira et al. (2020), Liang et al. (2019), Hershey et al. (2017) and Kiranyaz et al. (2019)).

2.2. Prophet

Prophet is a decomposable time series forecasting model inspired by time series forecasting done at Meta (former Facebook) (Taylor and Letham, 2018). Its components are combined as in Eq. (2), on which $g(t)$ is a trend function to model nonperiodic changes, $s(t)$ models the seasonal changes, $h(t)$ models the effects of holidays, and ϵ_t represents changes which are not represented by the other components.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2)$$

Prophet relies on a logistic function to model the trends, on a Fourier series to model seasonality, and on parameterization to implement the holiday's effects modeling. It deals with missing values and non-linear trends, having the ability to automatically find the inflection points and trend changes.

2.3. Time-series and power generation forecasting

In Pinho et al. (2019), authors use LSTM, ARIMA, Prophet, and an MLP network to forecast trends in real-world telecom data. LSTM achieved the best results in terms of mean absolute percentage error (MAPE) when forecasting internet data consumption, but did not provide good forecasts in terms of mobile phone card recharges. Costa and Moreira (2022) use Prophet to model the evolution of 2D regions and detect outliers on data series. chao Miao et al. (2020) use LSTM, K-Nearest Neighbor, AdaBoost, and convolutional neural network (CNN) in terms of their performance when forecasting fog formation. Authors use data on fog status (i.e., visibility level) to create four datasets with time series and forecast fog status one, two, three and four hours ahead. LSTM was the method that achieved the best results in short-term forecasting. In Chimmula and Zhang (2020) authors achieve a reasonable accuracy when using LSTM to forecast COVID-19 infections in Canada.

Ensafi et al. (2022) evaluate the use of several algorithms and networks (including Prophet, LSTM, and convolutional networks) in forecasting seasonal item sales. Authors conclude that Prophet, LSTM, and CNNs are some of the solutions that may forecast the sales of furniture within the acceptable error, but more complex LSTM and convolutional networks should be evaluated to improve their performance.

In Yunpeng et al. (2017), authors compare the performance of ARIMA and LSTM in multi-step ahead time series forecasting. Authors argue that LSTM networks can fit a wider range of data patterns than ARIMA, achieving higher accuracy, and also requires less modeling time.

Sharadga et al. (2020) assess the use of classic methods (e.g., ARIMA and SARIMA) and neural networks (e.g., LSTM and MLP) in forecasting PV generation on large-scale plants. The authors conclude that neural networks are more accurate than classic methods but can only be used to predict the produced power of PV plants one hour ahead otherwise there is a need for knowing weather parameters, like solar irradiance measures.

In Zhang et al. (2018), authors use Support Vector Regression (SVR) modeling to forecast energy consumption in the individual household with hourly and daily granularity. In González-Vidal et al. (2019), authors study energy consumption forecasting using Support Vector Machines (SVM) and Gaussian Processes. Deb et al. (2017) and Wei et al. (2019) present reviews on forecasting energy consumption methods.

Xue et al. (2019) compare the performance of SARIMA and CNN-LSTM on forecasting the inventory for perishable food in a food shop. They evaluate the use of two meta-heuristics (Particle Swarm Optimisation and Differential Evolution) to tune the design of the CNN-LSTM network architectures and make these outperform SARIMA. In Livieris et al. (2020), authors compare the use of CNN-LSTM, LSTM, a feedforward network, and SVR for forecasting gold price on different horizons. Authors conclude that the use of CNN together with LSTM increases the forecasting performance of LSTM. Zaouali et al. (2018) uses LSTM as part of a home solar power system for predicting solar generation. Authors show that LSTM outperforms statistical (Auto-Regressive Integrated Moving Average — ARIMA) and traditional machine learning (Support Vector Machine — SVM) methods when considering the prediction cost and performance ensemble.

In Abdulkarim and Engelbrecht (2021), authors use neural networks trained with particle swarm optimization algorithms in forecasting ten distinct scenarios, which include sunspot times, airline passengers, wine

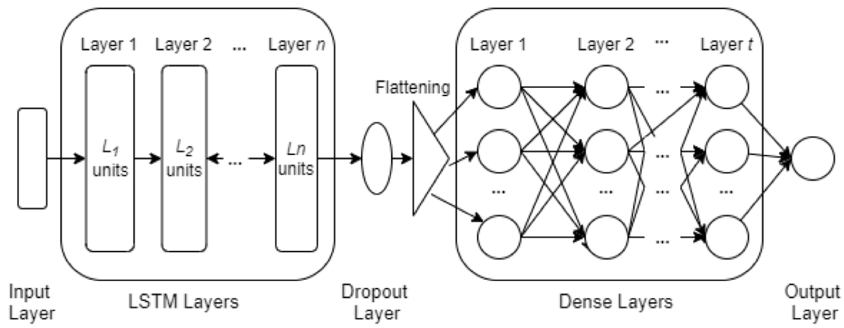


Fig. 3. LSTM network architecture.

sales, and internet traffic. In Hewamalage et al. (2021b), the authors present a survey on the use of RNN for time series forecasting.

Some years ago, Inman et al. (2013) reviewed solar forecasting methods for power generation by solar plants. In that work, the authors review a wide number of works that use several methods and techniques, including regressive methods, Artificial Intelligence-based techniques, remote sensing models, and numerical weather prediction models. Authors conclude that artificial neural networks can achieve good results and are not temporally limited. But at that time, deep learning-based models were not evaluated. In Gensler et al. (2017), authors use Deep Belief Networks, AutoEncoders, and LSTM to forecast energy generation in solar power plants. Authors argue that, in future work, the using LSTM combined with other networks should be studied in order to improve forecasting performance.

In van der Meer et al. (2018), the authors review the use of probabilistic methods to forecast solar power generation and electricity consumption. The authors argue that specialized models outperform general models, even though the models have their parameters fine-tuned. Das et al. (2018) review photovoltaic generation forecasting methods. Authors analyze statistical and machine-learning models, but the work does not cover the use of deep models with LSTM and Convolutional nodes. Still, the authors conclude that machine learning-based forecasting solutions (using ANNs and SVM) perform well when predicting energy generation in varying environmental conditions. Sobri et al. (2018) review several solar photovoltaic power forecasting methods. The authors conclude that Artificial Intelligence-based methods achieve better results when compared to statistical approaches. More recently, Wang et al. (2019) reviewed deep learning-based for renewable energy generation forecasting. The authors state that establishing the most appropriate deep learning model for each dataset is an open challenge. In Torres et al. (2021), a recent survey on using deep learning for time series forecasting exemplifies using LSTM and convolutional networks for forecasting in several application areas.

In Zhen et al. (2020), the authors study the use of sky images to calculate surface solar irradiance values to use in ultra-short-term solar PV generation forecasting. Kong et al. (2020) also study sky image-based short-term solar forecasting. The authors argue their proposal achieves good performance, especially on partially cloudy days. Moreira et al. (2021) describe the use of ANNs and the Design of Experiments (DoE) approach (i.e., a statistical approach that considers each experimental run as a test) in PV generation forecasting. The DoE supports the definition of some ANNs hyperparameters and factors related to the dataset (e.g., the correlation between climate variables and the number of points to use). Sangrody et al. (2020) use similarity-based forecasting models for day-ahead PV generation forecasting. Authors use distinct models depending on the available weather variables (e.g., temperature, sky cover, humidity, and wind speed) and evaluate their proposals using data from solar panels on a commercial building.

In this work, we evaluate using deep convolutional and LSTM networks and their combination to forecast photovoltaic energy generation on several forecasting horizons. We train our models using aggregated data on historical PV generation by hundreds of household systems in a region and assess their performance in predicting energy generation in the entire region and by individual systems.

3. Learning architectures and photovoltaic generation series

To evaluate the use of deep learning in forecasting household photovoltaic generation data, we use three network architectures over several series based on real-world data.

3.1. Deep models

We use three neural network architectures: LSTM networks, Convolutional neural networks, and hybrid Convolutional-LSTM networks.

LSTM networks — Fig. 3 presents the base architecture of our LSTM network. The first hidden layers are based on LSTM units. The number of layers and the units per layer are some of the hyperparameters that we define experimentally, as described in Section 4.2.

After the LSTM layers, there is a Dropout layer and a flattening. The Dropout layer is one of the measures we took to prevent overfitting. Dropout is a regularization technique that randomly drops some neurons during training, which changes the network configuration and improves generalization during training (thus reducing overfitting). The flattening changes the dimensionality of data, converting it into a 1-dimension array. Then, there is a set of dense layers composed of fully connected neurons. The number of neurons per layer and the number of layers are some of the hyperparameters we experimentally define.

Convolutional network — these networks start with a set of temporal convolutional layers, as represented in Fig. 4. These are 1D-convolution layers, each having its own filter and kernel sizes. Then, there is a Dropout layer (to prevent overfitting) and a flattening. The last block of layers is composed of fully connected neurons, like in the LSTM networks.

Convolutional-LSTM (Conv-LSTM) networks — this architecture combines layers from the previous architectures, as represented in Fig. 5. It starts with a set of convolutional layers, followed by a Dropout layer and a set of LSTM layers. Then, there is another Dropout layer and a flattening. Following the flattening, there is a set of dense layers with fully-connected neurons.

In the proposed base-model architectures, we use the Rectified Linear Unit (ReLU) as the LSTM layer activation function and the Parametric Rectified Linear Unit (PReLU) as the activation function for Convolutional and Dense layers. We use Bayesian Optimization (Pelikan et al., 1999) to define some layer-dependent hyperparameters (care is taken to avoid overfitting) and evaluate if the obtained models may be generalized for other time series.

3.2. Time series on residential photovoltaic generation

The *Solar home electricity dataset* (Ratnam et al., 2017) was made available by Ausgrid, an electricity distributor on Australia. Available data includes the postal code, installed PV generation capacity, PV production (recorded at each half-hour), and electricity consumption for each customer from 1 July 2010 to 30 June 2013. The residential

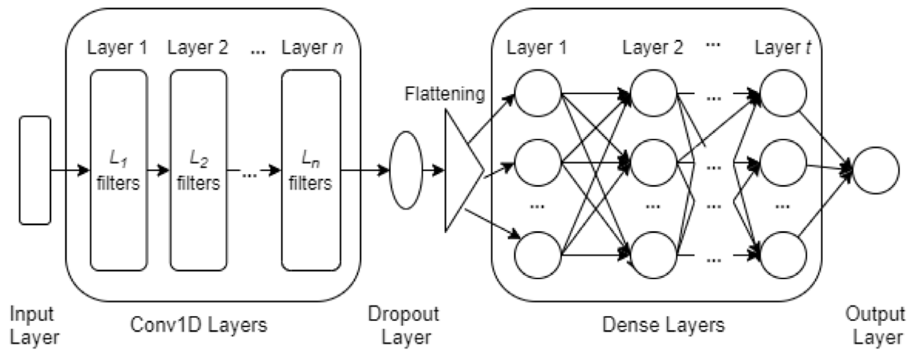


Fig. 4. Convolutional network.

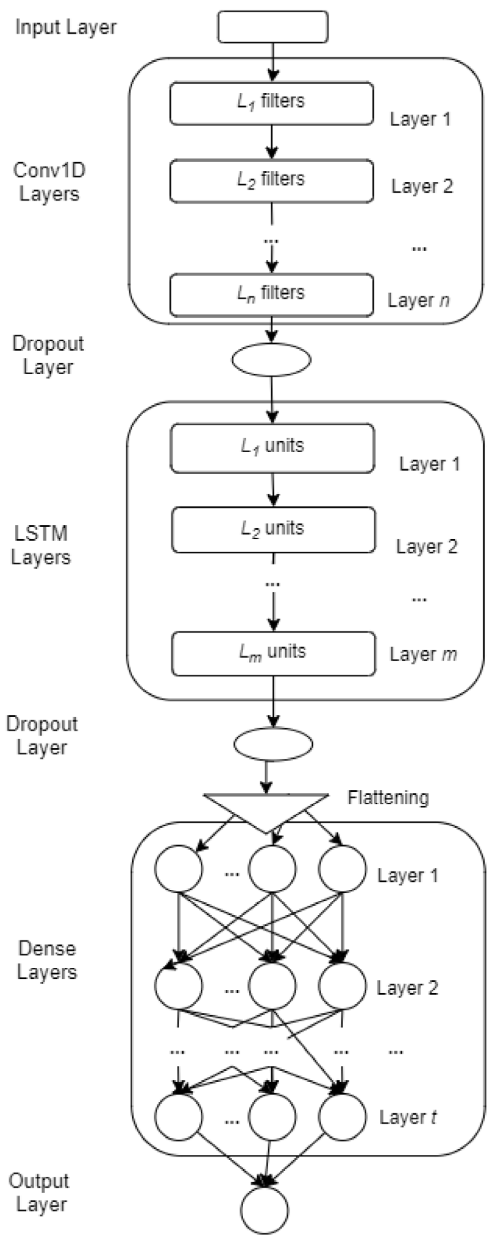


Fig. 5. Proposed hybrid Conv-LSTM architecture.

photovoltaic generation data is referred to as *Gross Generation* (GG) in the original dataset.

One issue we want to assess is the use of region-based data in forecasting. We will evaluate the models' capability to predict the values of the entire region. But we will also compare the performance of models trained with the region-based data in predicting the photovoltaic generation of individual systems with the one obtained if we forecast each system's data generation using models trained with the system's historical data. So, we built region-based series with the aggregated data of the different customers.

Consider that the residential photovoltaic energy generated (V) by each system s_i ($1 \leq i \leq number\ of\ household\ systems$) at a timestamp t is represented by $V(s_i, t)$. The energy generated (V) by the set of systems ($S = \{s_1, s_2, \ldots, s_j\}$) at a timestamp t is $V(S,t) = \sum_{i=1}^{j} V(s_i, t)$.

Furthermore, we want to assess the ability of models to make predictions in terms of several time horizons. Hence, we build new time series (for each household system and at the region level) by aggregating data by day and month. Note that by grouping the data by different time windows, we modify the size of the train and test datasets. Hence, we have the following time series:

- $\boldsymbol{S_i^h}$: Residential photovoltaic energy generated at each half hour (t) by each household s_i ($s_i \in S$)

$$S_i^h : V(s_i, t) \tag{3}$$

- $\boldsymbol{S_i^d}$: Residential photovoltaic energy generated at each day (d) each household s_i ($s_i \in S$)

$$S_i^d : V(s_i, d) = \sum_{t \in d} V(s_i, t) \tag{4}$$

- $\boldsymbol{S_i^m}$: Residential photovoltaic energy generated at each month (m) by each household s_i ($s_i \in S$)

$$S_i^m : V(S_i, m) = \sum_{t \in m} V(s_i, t) \tag{5}$$

- $\boldsymbol{R^h}$: Region-level energy generated at each half hour (t)

$$R^h : V(t) = \sum_{i=1}^{j} V(s_i, t) \tag{6}$$

- $\boldsymbol{R^d}$: Region-level energy generated at each day (d)

$$R^d : V(d) = \sum_{i=1}^{j} \sum_{t \in d} V(s_i, t) \tag{7}$$

- $\boldsymbol{R^m}$: Region-level energy generated at each month (m)

$$R^m : V(m) = \sum_{i=1}^{j} \sum_{t \in m} V(s_i, t) \tag{8}$$

3.3. Performance metrics

There are several performance metrics in the literature, some of which are more suitable for specific contexts and objectives.

Table 1
Hyperparameters defined using Bayesian Optimization.

LSTM network

Time series	LSTM layers	Units per layer	Dropout	Dense layers	Neurons per layer
R^m	3	469;469;338	0.20	1	18
R^d	2	650;650	0.15	1	28
R^h	4	521;521; 418;335	0.41	1	28

Convolutional network

Time series	Conv layers	Filters per layer	Kernel size	Dropout	Dense layers	Neurons per layer
R^m	7	650 at all layers	7	0.15	1	62
R^d	2	299;136	9	0.20	1	6
R^h	3	514;57;3	5	0.27	2	77;76

Conv-LSTM network

Time series	Conv layers	Filters/layer	LSTM layers	Units/layer	Dropout	Dense layers	Neurons per layer
R^m	1	6	1	650	0.15	1	42
R^d	4	635;447 314;221	1	6	0.18	1	8
R^h	1	14	3	213; 70;23	0.27	3	64;21;6

The Mean Absolute Error (MAE) is the mean of the differences between two observations of the same phenomenon, which in this work are the real (observed) and predict values at a certain time. MAE is calculated as represented in Eq. (9), where n is the number of predictions, Y are the real (observed) values and $\hat{Y}$ are the predicted values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{Y}_i| \tag{9}$$

The Root Mean Square Error (RMSE) is a non-negative metric based on prediction errors (i.e., residuals). RMSE is computed as the square root of the mean of squares of errors between real values (Y) and predicted values ($\hat{Y}$), as represented in Eq. (10).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{10}$$

We use the Normalized Root Mean Square Error (NRMSE) to compare the quality of predictions with different scales. NRMSE is calculated by dividing the RMSE by the difference between the highest observed value (Y_{max}) and the lowest observed value (Y_{min}), as represented in Eq. (11).

$$NRMSE = \frac{RMSE}{(Y_{max} - Y_{min})} \tag{11}$$

All three metrics are non-negative (lower is better) and indifferent to the direction of errors. When computing MAE, the individual errors have the same weight, while the RMSE metric is more sensitive to large errors. We use the RMSE as the control metric in hyperparameter optimization. NRMSE is a metric more adequate to compare models that have different scales. Hence, we use the NRMSE metric to compare the performance of models when forecasting for different forecasting horizons.

4. Experimental evaluation

We made several experiments to evaluate the performance and generalization of considered solutions. We implemented the deep models using Python, Jupyter Notebooks, and Tensorflow (Abadi et al., 2016). We used Meta's Prophet implementation in Python as documented in Meta (2022). Experiments were executed using an NVIDIA GeForce RTX 2060 GPU.

Fig. 6 presents an overview of our methodology. First, we built the data series and the training, validation, and testing datasets. Section 4.1 describes such processes. Then, we used Bayesian Optimization and region-level data to define the models' hyperparameters, as described in Section 4.2. Next, we access the use of the Prophet tool and the deep models to forecast region-level data at distinct time scales. Section 4.3 presents the obtained results. The fourth stage of our evaluation methodology comprises two steps: (a) evaluating the use of models trained with region-level data to forecast PV generation by individual systems, and (b) training new models with individual systems' historical data and evaluating the performance of such models and Prophet on forecasting individual systems PV power generation at three forecasting horizons. Section 4.4 presents the results of such stage.

4.1. Training, validation and testing datasets

We use a subset of the *Solar home electricity dataset* (Ratnam et al., 2017) available in Ausgrid (2015) on the residential photovoltaic generation of 300 customers randomly selected in Sydney, Australia, and nearby areas. Available data include customers' postal code, installed PV generation capacity, and PV production recorded at each half-hour from 1 July 2010 to 30 June 2013.

The residential photovoltaic generation data is referred to as *Gross Generation* (GG) in the original dataset. We used the *Gross Generation* to build the region-level R^h, R^d, and R^m series, as described in Section 3.2. Figs. 7(a) and 7(b) presents the values of the region-level PV generation series by day (R^d) and month (R^m), respectively.

We also selected 30 customers to build the series S_i^h, S_i^d, and S_i^m for each customer. Hence, we have three region-level series (with distinct timescales) and 90 individual household system series (30 for each of the considered timescales). Selected systems are from different postal addresses to remove bias related to geographic conditions.

We use data from 2010 to 2012 as training and validation data (75% of data for training and 25% for validation). We used the algorithms to estimate the gross generation for the first semester of 2013, i.e., the test dataset contains data for the first semester of 2013.

4.2. Model configuration and selection of hyperparameters

We used Bayesian Optimization over the region-based data series to define several hyperparameters of considered models, including the number of layers, number of neurons per layer, dropout rate, filter size in each Conv1D layer, and number of units in each LSTM layer.

We split each of the region-level datasets (corresponding to R^h, R^d, and R^m series) into training, validation, and testing. We used Early Stopping (i.e., model training stops even when a monitored metric has stopped improving) to avoid overfitting. We monitored the value of the mean squared error regression loss for the validation data, using 0.001 as the minimum change to qualify as an improvement, and with the training stopping after ten epochs with no improvement. We also used dropout layers to avoid overfitting.

Table 1 presents the hyperparameters of the LSTM, Convolutional and proposed Conv-LSTM models. At all series, the number of LSTM layers in the LSTM models was higher than the number of LSTM layers in the corresponding Conv-LSTM models. Also, the number of

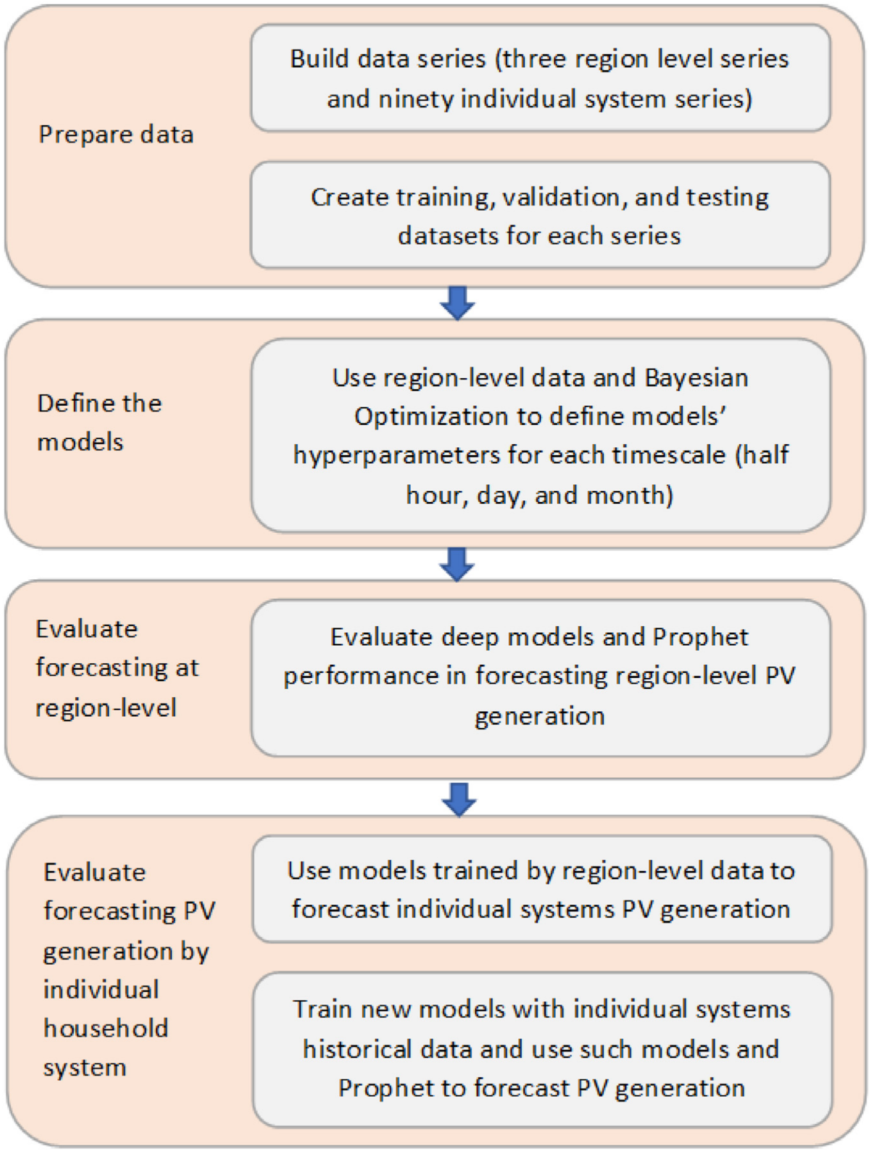
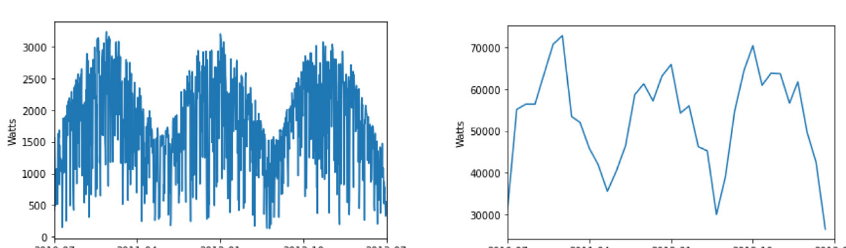


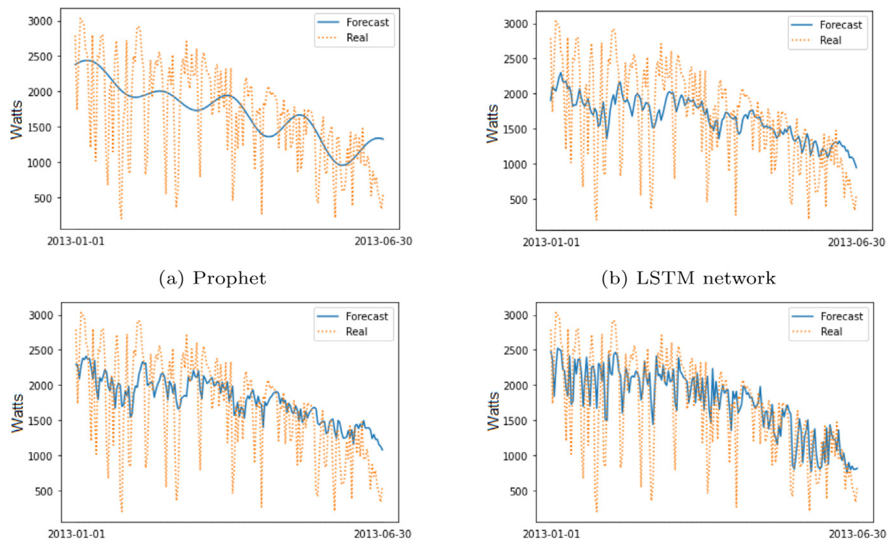
Fig. 6. Evaluation methodology.



(a) Region-level generation by day (R^d)

(b) Region-level generation by month (R^m)

Fig. 7. Region-level PV generation.



(a) Prophet (b) LSTM network

(c) Convolutional network (d) Conv-LSTM network

Fig. 8. Forecasting total photovoltaic generation per day (R^m series) - testing dataset.

Table 2
Total generation forecasting — Series: R^m, R^d and R^h.

Time interval	Algorithm/Network architecture	MAE	RMSE	NRMSE
Month (R^m)	Prophet	16 730.3	21 753.2	0.59
	LSTM	9719.4	12 271.3	0.33
	Convolutional	4737.9	6842.2	0.19
	Conv-LSTM	5517.1	7824.2	0.21
Day (R^d)	Prophet	487.1	605.2	0.21
	LSTM	479.3	599.2	0.21
	Convolutional	458.1	575.6	0.20
	Conv-LSTM	426.0	540.8	0.19
Half hour (R^h)	Prophet	43.8	67.8	0.35
	LSTM	31.6	38.5	0.19
	Convolutional	4.0	7.0	0.04
	Conv-LSTM	2.9	5.2	0.03

convolutional layers in the Convolutional models was higher than in the corresponding Conv-LSTM models. Hence, in the Conv-LSTM models, the learning related to temporal behavior is distributed by specialized layers of both types (i.e., LSTM and Convolutional). All configurations contain a small number of dense layers.

4.3. Sydney area photovoltaic generation forecasting

Initially, we used the models to predict the household energy generation using region-based aggregated data (containing data from all domestic systems). We executed the predictions 25 times for each model and time series. Table 2 presents the average values obtained with each model and using the Prophet tool.

Deep networks performed better than Prophet in all scenarios. In terms of MAE, the LSTM networks obtained the worst results among considered deep networks, and the Conv-LSTM networks achieved the best results. However, in terms of RMSE, convolutional networks got performance close to the ones of the Conv-LSTM networks, even being better for the S^m series (which is the series with the lowest number of points).

Fig. 8 presents the real and foreseen household total energy generation on a daily basis. Through the analysis of the graphs, it is possible to verify that the Prophet's predictions are closer to a trend line than to the actual values, and the Conv-LSTM network is the one that makes the predictions that are closest to the actual values.

4.4. Forecasting individual household energy generation

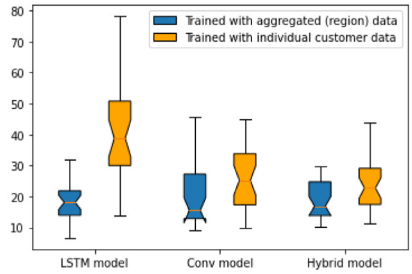
We trained models of each architecture for each system's series. Then we used these models to predict the energy generation by each system. To evaluate the generalization of previously trained models, we use the models trained with the (region-level) aggregated data to forecast the values of photovoltaic generation by each of the selected domestic systems.

Fig. 9 presents the MAE and RMSE obtained when using the deep models for each forecasting horizon. The figure presents the metrics when forecasting individual energy generation using the models trained with the aggregated sets (i.e., R^d, R^m and R^h), with the ones obtained when training the models with each system's data (i.e., S_i^d, S_i^m and S_i^h). Each box plot represents the metric computed using the forecasting on the 30 series built for each time scale and individual system.

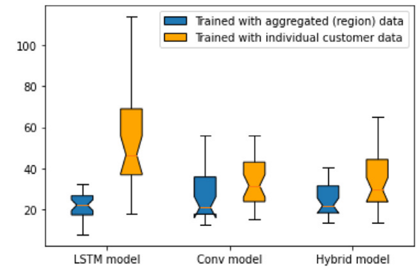
The models trained with the series composed of the data aggregated by region showed similar or even better performance than those obtained by training models with the data of the domestic system for which they were making the forecasting. Those models obtained comparatively better results in the LSTM architecture for the series S_i^m and S_i^h, which contain energy generation aggregated by month and every half hour, respectively. The monthly series (S_i^n) contains relatively few values that reduce the model's capacity to learn the series patterns. The S_i^h series, on the other hand, contemplates a more erratic variation for each system due to specific issues and noise that can also make it difficult for the model to learn data patterns. In both series, the performance of the Conv-LSTM models proved to be equal or superior to that of the other models. In both series, the performance of the Conv-LSTM models proved to be equal or superior to that of the other models. In the series with aggregated data per day (S_i^d), the models trained with each domestic system data performed close to models trained with region-level aggregated data.

4.5. Discussion

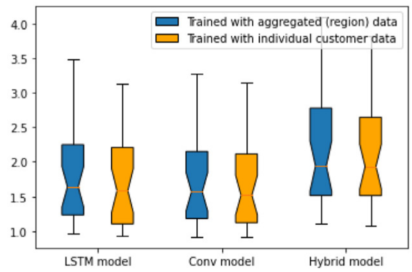
We used the NRMSE to compare the performance that the evaluated solutions had when predicting in different time scales. Tables 3, 4, and 5 summarizes the average values of NRMSE for month, day and half hour prediction horizons respectively. The deep learning models obtained good results (e.g., 0.03 and 0.04), with higher performance than the reference forecasting tool in all the evaluated situations.



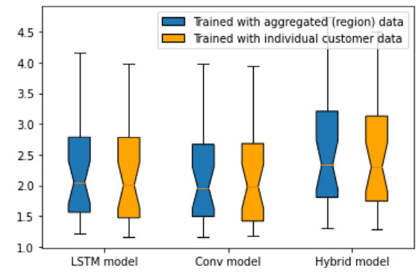
(a) MAE - forecasting energy generation per month - S_i^m



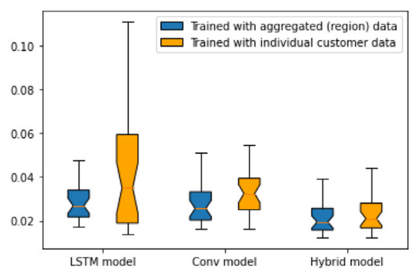
(b) RMSE - forecasting energy generation per month - S_i^m



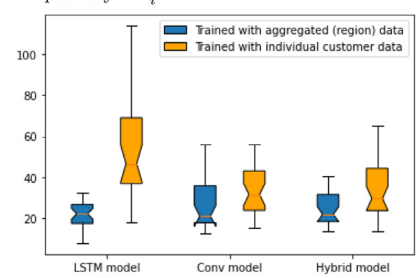
(c) MAE - forecasting energy generation per day - S_i^d



(d) RSME - forecasting energy generation per day - S_i^d



(e) MAE - forecasting energy generation per half hour - S_i^h



(f) RMSE - forecasting energy generation per half hour - S_i^h

Fig. 9. Forecasting photovoltaic generation for each household system s_i.

Table 3
Normalized RMSE — Data summarized by month (R^m and S_i^m).

	Forecasting PV generation at region level (R^*)	Forecasting individual system PV generation (S_i^*) - training using region-level historical data (R^*)	Forecasting individual system PV generation (S_i^*) - training with each system's historical data (S_i^*)
Prophet	0.59	–	0.56
LSTM	0.33	0.21	0.51
Convolutional	0.19	0.23	0.28
Conv-LSTM	0.21	0.23	0.28

The use of the Conv-LSTM network in forecasting values of the R^h series leads to the best performance in terms of RMSE of all configurations. This series has the highest number of points among those that summarize energy generation in the entire region. On the other hand, the LSTM networks were the ones that had the worst results among the deep models. The results of the LSTM networks were particularly inferior to the other networks in the series with few points.

The Convolutional and Conv-LSTM networks obtained very similar performances in terms of NRMSE, with the Conv-LSTM network presenting a slightly superior performance in most cases. Both architectures had little performance difference when comparing the monthly and daily series. But they got considerably higher performance in the data series (R^h and S_i^h) for each half-hour (the one with the highest amount of data).

The models trained with the summarized data at the region level (R^m, R^d, and R^h) reached a good level of generalization. These models made efficient predictions on series based on data from specific domestic systems, achieving performance close to or even superior to those obtained using models trained with the historical data of each energy generation system. The models trained with the data summarized by region performed well in the forecasts of domestic systems with historical series where there were erratic and noisy data since noise in data of individual systems has a lower impact on the region summarized data.

5. Conclusions

The photovoltaic energy generation by domestic systems is relevant and challenging, as energy generation at the ground level depends on several factors, like the intensity of luminosity and temperature. We evaluate the use of deep learning and Prophet in forecasting PV energy generation at the region-level and by individual systems using real-world data and considering three timescales.

Overall, convolutional layers better identified the features in time series than LSTM. The Convolutional and Conv-LSTM networks achieved the best performance in all timescales when predicting region-level PV generation. Those also outperformed the other solutions when

Table 4
Normalized RMSE — Data summarized by day (R^d and S_i^d).

	Forecasting PV generation at region level (R^*)	Forecasting individual system PV generation (S_i^*) - training using region-level historical data (R^*)	Forecasting individual system PV generation (S_i^*) - training with each system's historical data (S_i^*)
Prophet	0.21	–	0.23
LSTM	0.21	0.23	0.22
Convolutional	0.20	0.22	0.21
Conv-LSTM	0.19	0.27	0.26

Table 5
Normalized RMSE — Half hour data (R^h and S_i^h).

	Forecasting PV generation at region level (R^*)	Forecasting individual system PV generation (S_i^*) - training using region-level historical data (R^*)	Forecasting individual system PV generation (S_i^*) - training with each system's historical data (S_i^*)
Prophet	0.35	–	0.37
LSTM	0.19	0.09	0.11
Convolutional	0.04	0.08	0.08
Conv-LSTM	0.03	0.07	0.07

training the algorithms with individual systems data and using the half-hour and month timescales. The highest performance difference between the methods occurred in half-hour region-level forecasting when the NRMSE of LSTM was approximately sixfold of the Convolutional and Conv-LSTM networks, and the Prophet's NRMSE was tenfold of the best ones. Conv-LSTM was slightly superior to the convolutional network in most scenarios. Thus, the ensemble use of convolutional and LSTM layers improved performance but to a limited extent.

The evaluated networks also showed good generalization capacity, i.e., when forecasting PV generation by individual systems, models trained with region-level historical data achieved performance similar (and sometimes even superior) to models trained with each system's data.

In future work, we intend to explore the study of model generalization and the application of Conv-LSTM networks in other time series applications related to renewable energies.

CRediT authorship contribution statement

Rogério Luís de C. Costa: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., Portugal, under the Scientific Employment Stimulus - Institutional Call - CEECINST/00051/2018 and in the context of the project UIDB/04524/2020.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp. 265–283.

Abdulkarim, S.A., Engelbrecht, A.P., 2021. Time series forecasting with feedforward neural networks trained using particle swarm optimizers for dynamic environments. Neural Comput. Appl. 33 (7), 2667–2683.

Ahn, K., Chu, Z., Lee, D., 2021. Effects of renewable energy use in the energy mix on social welfare. Energy Econ. 96, 105174.

Ashraf, J., Bakhshi, A.D., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A., 2021. Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. 22 (7), 4507–4518. http://dx.doi.org/10.1109/TITS.2020.3017882.

Ausgrid, 2015. Solar home electricity data. Last accessed in 2022-01-12. URL https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data.

Caldeira, M., Martins, P., Costa, R.L.C., Furtado, P., 2020. Image classification benchmark (ICB). Expert Syst. Appl. 142, 112998. http://dx.doi.org/10.1016/j.eswa.2019.112998.

Chimmula, V.K.R., Zhang, L., 2020. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos Solitons Fractals 135, 109864.

Costa, R.L.d.C., Moreira, J., 2022. Automatic quality improvement of data on the evolution of 2d regions. In: Advanced Data Mining and Applications. Springer International Publishing, pp. 288–300. http://dx.doi.org/10.1007/978-3-030-95408-6_22.

Das, U.K., Tey, K.S., Seyedmahmoudian, M., Mekhilef, S., Idris, M.Y.I., Van Deventer, W., Horan, B., Stojcevski, A., 2018. Forecasting of photovoltaic power generation and model optimization: A review. Renew. Sustain. Energy Rev. 81, 912–928.

Deb, C., Zhang, F., Yang, J., Lee, S.E., Shah, K.W., 2017. A review on time series forecasting techniques for building energy consumption. Renew. Sustain. Energy Rev. 74 (February), 902–924. http://dx.doi.org/10.1016/j.rser.2017.02.085.

Deng, L., Platt, J., 2014. Ensemble deep learning for speech recognition. In: Proc. Interspeech.

Dongare, A., Kharde, R., Kachare, A.D., et al., 2012. Introduction to artificial neural network. Int. J. Eng. Innov. Technol. (IJEIT) 2 (1), 189–194.

Ensafi, Y., Amin, S.H., Zhang, G., Shah, B., 2022. Time-series forecasting of seasonal items sales using machine learning–A comparative analysis. Int. J. Inf. Manag. Data Insights 2 (1), 100058.

Gensler, A., Henze, J., Sick, B., Raabe, N., 2017. Deep learning for solar power forecasting - an approach using AutoEncoder and LSTM neural networks. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings. pp. 2858–2865. http://dx.doi.org/10.1109/SMC.2016.7844673.

Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. Neural Comput. 12 (10), 2451–2471. http://dx.doi.org/10.1162/089976600300015015.

González-Vidal, A., Jiménez, F., Gómez-Skarmeta, A.F., 2019. A methodology for energy multivariate time series forecasting in smart buildings based on feature selection. Energy Build. 196, 71–82. http://dx.doi.org/10.1016/j.enbuild.2019.05.021.

Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y., et al., 2017. Ese: Efficient speech recognition engine with sparse lstm on fpga. In: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. pp. 75–84.

Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al., 2017. CNN architectures for large-scale audio classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 131–135.

Hewamalage, H., Bergmeir, C., Bandara, K., 2021a. Recurrent neural networks for time series forecasting: Current status and future directions. Int. J. Forecast. 37 (1), 388–427. http://dx.doi.org/10.1016/j.ijforecast.2020.06.008, URL https://www.sciencedirect.com/science/article/pii/S0169207020300996.

Hewamalage, H., Bergmeir, C., Bandara, K., 2021b. Recurrent neural networks for time series forecasting: Current status and future directions. Int. J. Forecast. 37 (1), 388–427. http://dx.doi.org/10.1016/j.ijforecast.2020.06.008, arXiv:1909.00590.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2 (5), 359–366. http://dx.doi.org/10.1016/0893-6080(89)90020-8.

IEA, 2021. Global Energy Review. Tech. rep., International Energy Agency (IEA).

Inman, R.H., Pedro, H.T., Coimbra, C.F., 2013. Solar forecasting methods for renewable energy integration. Prog. Energy Combust. Sci. 39 (6), 535–576. http://dx.doi.org/10.1016/j.pecs.2013.06.002.

Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., Gabbouj, M., 2019. 1-D convolutional neural networks for signal processing applications. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8360–8364. http://dx.doi.org/10.1109/ICASSP.2019.8682194.

Kong, W., Jia, Y., Dong, Z.Y., Meng, K., Chai, S., 2020. Hybrid approaches based on deep whole-sky-image learning to photovoltaic generation forecasting. Appl. Energy 280, 115875.

Li, J., Mohamed, A., Zweig, G., Gong, Y., 2015. LSTM time and frequency recurrence for automatic speech recognition. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, pp. 187–191.

Liang, W.-j., Zhang, H., Zhang, G.-f., Cao, H.-x., 2019. Rice blast disease recognition using a deep convolutional neural network. Sci. Rep. 9 (1), 1–10.

Livieris, I.E., Pintelas, E., Pintelas, P., 2020. A CNN – LSTM model for gold price time-series forecasting. Neural Comput. Appl. 32 (23), 17351–17360. http://dx.doi.org/10.1007/s00521-020-04867-x.

Malhotra, P., Vig, L., Shroff, G., Agarwal, P., 2015. Long short term memory networks for anomaly detection in time series, In: Proceedings, Vol. 89. pp. 89–94.

Meta, 2022. Prophet installation. Last accessed in January 14, 2022. URL https://facebook.github.io/prophet/docs/installation.html.

chao Miao, K., ting Han, T., qing Yao, Y., Lu, H., Chen, P., Wang, B., Zhang, J., 2020. Application of LSTM for short term fog forecasting based on meteorological elements. Neurocomputing 408, 285–291. http://dx.doi.org/10.1016/j.neucom.2019.12.129.

Moreira, M., Balestrassi, P., Paiva, A., Ribeiro, P., Bonatto, B., 2021. Design of experiments using artificial neural network ensemble for photovoltaic generation forecasting. Renew. Sustain. Energy Rev. 135, 110450.

Pelikan, M., Goldberg, D.E., Cantú-Paz, E., et al., 1999. BOA: The Bayesian optimization algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Vol. 1. Citeseer, pp. 525–532.

Pinho, A., Costa, R.L.d.C., Silva, H., Furtado, P., 2019. Comparing time series prediction approaches for telecom analysis. In: Theory and Applications of Time Series Analysis. Springer International Publishing, Cham, pp. 331–345. http://dx.doi.org/10.1007/978-3-030-26036-1_23.

Ratnam, E.L., Weller, S.R., Kellett, C.M., Murray, A.T., 2017. Residential load and rooftop PV generation: an Australian distribution network dataset. Int. J. Sustain. Energy 36 (8), 787–806.

Sainath, T.N., Vinyals, O., Senior, A., Sak, H., 2015. Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4580–4584.

Sangrody, H., Zhou, N., Zhang, Z., 2020. Similarity-based models for day-ahead solar PV generation forecasting. IEEE Access 8, 104469–104478.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Netw. 61, 85–117.

Sharadga, H., Hajimirza, S., Balog, R.S., 2020. Time series forecasting of solar power generation for large-scale photovoltaic plants. Renew. Energy 150, 797–807.

Sobri, S., Koohi-Kamali, S., Rahim, N.A., 2018. Solar photovoltaic generation forecasting methods: A review. Energy Convers. Manage. 156 (May 2017), 459–497. http://dx.doi.org/10.1016/j.enconman.2017.11.019.

Taylor, S.J., Letham, B., 2018. Forecasting at scale. Amer. Statist. 72 (1), 37–45.

Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A., 2021. Deep learning for time series forecasting: a survey. Big Data 9 (1), 3–21.

van der Meer, D., Widén, J., Munkhammar, J., 2018. Review on probabilistic forecasting of photovoltaic power production and electricity consumption. Renew. Sustain. Energy Rev. 81, 1484–1512. http://dx.doi.org/10.1016/j.rser.2017.05.212.

Wang, H., Lei, Z., Zhang, X., Zhou, B., Peng, J., 2019. A review of deep learning for renewable energy forecasting. Energy Convers. Manage. 198 (July), 111799. http://dx.doi.org/10.1016/j.enconman.2019.111799.

Wei, N., Li, C., Peng, X., Zeng, F., Lu, X., 2019. Conventional models and artificial intelligence-based models for energy consumption forecasting: A review. J. Pet. Sci. Eng. 181 (March), 106187. http://dx.doi.org/10.1016/j.petrol.2019.106187.

Xue, N., Triguero, I., Figueredo, G.P., Landa-Silva, D., 2019. Evolving deep CNN-LSTMs for inventory time series prediction. In: 2019 IEEE Congress on Evolutionary Computation (CEC). pp. 1517–1524. http://dx.doi.org/10.1109/CEC.2019.8789957.

Yunpeng, L., Di, H., Junpeng, B., Yong, Q., 2017. Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In: 2017 14th Web Information Systems and Applications Conference (WISA). IEEE, pp. 305–310.

Zaouali, K., Rekik, R., Bouallegue, R., 2018. Deep learning forecasting based on auto-lstm model for home solar power systems. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (Hpcc/Smartcity/Dss). IEEE, pp. 235–242.

Zhang, X.M., Grolinger, K., Capretz, M.A.M., Seewald, L., 2018. Forecasting residential energy consumption: Single household perspective. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 110–117. http://dx.doi.org/10.1109/ICMLA.2018.00024.

Zhen, Z., Liu, J., Zhang, Z., Wang, F., Chai, H., Yu, Y., Lu, X., Wang, T., Lin, Y., 2020. Deep learning based surface irradiance mapping model for solar PV power forecasting using sky image. IEEE Trans. Ind. Appl. 56 (4), 3385–3396.