REGULAR PAPER

A comparative study of LSTM-ED architectures in forecasting day-ahead solar photovoltaic energy using Weather Data

Ekin Ekinci[1]

Abstract

Solar photovoltaic (PV) energy, with its clean, local, and renewable features, is an effective complement to traditional energy sources today. However, the photovoltaic power system is highly weather-dependent and therefore has unstable and intermittent characteristics. Despite the negative impact of these features on solar sources, the increase in worldwide installed PV capacity has made solar energy prediction an important research topic. This study compares three encoder-decoder (ED) networks for day-ahead solar PV energy prediction: Long Short-Term Memory ED (LSTM-ED), Convolutional LSTM ED (Conv-LSTM-ED), and Convolutional Neural Network and LSTM ED (CNN-LSTM-ED). The models are tested using 1741-day-long datasets from 26 PV panels in Istanbul, Turkey, considering both power and energy output of the panels and meteorological features. The results show that the Conv-LSTM-ED with 50 iterations is the most successful model, achieving an average prediction score of up to 0.88 over R-square (R^2). Evaluation of the iteration counts' effect reveals that the Conv-LSTM-ED with 50 iterations also yields the lowest Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values, confirming its success. In addition, the fitness and effectiveness of the models are evaluated, with the Conv-LSTM-ED achieving the lowest Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values for each iteration. The findings of this work can help researchers build the best data-driven methods for forecasting PV solar energy based on PV features and meteorological features.

✉ Ekin Ekinci
  ekinekinci@subu.edu.tr

1   Faculty of Technology, Computer Engineering Department, Sakarya University of Applied Sciences, Sakarya, Turkey

🖄 Springer

1 Introduction

One of the most significant effects of human activity is global warming. This occurs when greenhouse gases, such as CO_2, CH_4, N_2O, and water vapor, are produced in greater quantities in the atmosphere due to the excessive use of fossil fuels as an energy source [2]. To prevent global warming, transitioning from fossil fuels to renewable energy sources—such as hydro, geothermal, solar, wind, wood, plant residues, biomass, tidal, and wave—is important [35]. One of the most promising renewable energy sources for producing enormous amounts of energy is solar energy. However, solar energy is impacted by its limited availability during daytime and its seasonal nature, making it unstable and unpredictable. As a result, predicting their output is challenging and requires sophisticated methodologies. In the literature, the four categories of methodologies employed for this purpose include physical methods, statistical models, artificial intelligence techniques, and their hybrid and ensemble approaches [3, 27].

Physical methods use atmospheric physical and mechanical data such as wind speed, temperature, rainfall, humidity, and day duration and are based on numerical weather prediction (NWP) to simulate atmospheric dynamics (Lorenz, Scheidsteger, Hurka, Heinemann, & Kurz, 30; Urquhart et al., [47]. Statistical approaches are used to regress unknown constants and to determine the mathematical relationship between inputs and outputs (Ye, Yang, Han, & Chen, 51). They are widely used in the literature, but their performance did not meet expectations because they are unsuitable for modeling nonlinear relationships (De Giorgi, Congedo, & Malvoni, 10; Sheng, Xiao, Cheng, Ni, & Wang, 43; Yadav, Kannan, Meraj, & Masaoud, 49). The capacity of artificial intelligence algorithms to model nonlinear relationships, such as machine learning and more recently deep learning, has caught the attention of researchers. A wide range of machine learning algorithms are employed (AlShafeey & Csaki, 2021; Belmahdi, Louzazni, & El Bouardi, 2022; De Leone, Pietrini, & Giovannelli, 11; Frederiksen & Cai, [16] Malvoni, De Giorgi, & Congedo, 33; Markovics & Mayer, [34] Yagli, Yang, & Srinivasan, 50; Zhao et al., [56]. The future of deep learning in solar energy prediction is promising due to its capacity for generalization. Feed Forward Neural Network (FFN) (Rodriguez, Azcarate, Vadillo, & Galarza, 2022; Rodriguez, Galarza, Vasquez, & Guerrero, 2022), Long Short-Term Memory (LSTM)/ Recurrent Neural Network (RNN) [14]; Hafiz, Awal, de Queiroz, & Husain, 20; Hossain & Mahmood, [21, 29] X. Luo, Zhang, & Zhu, 32; Park, Lee, Kang, Choi, & Lee, 38; Wang et al., [48], Convolutional Neural Network (CNN) (C. Zhang, Li, Jiang, Luo, & Xu, 54), Gated Recurrent Unit (GRU) (Qu, Xu, Sun, & Liu, 40), ED (Chang, Bai, & Hsu, 8). Data decomposition, feature extraction, or hybridization of algorithms are used to compose hybrid and ensemble algorithms (Ishaq, Kwon, et al., 2022; Khan, Walker, & Zeiler, 23; Lai, Zhong, Pan, Ng, & Lai, [25] P. Li, Zhou, Lu, & Yang, 26; Lin et al., [28] Ospina, Newaz, & Faruque, 37; Tang, Yang, Zhang, & Zhang, 46; J. Zhang, Tan, & Wei, 55).

In this study, we present a thorough performance comparison among LSTM-ED architectures—namely, LSTM-ED, Conv-LSTM-ED, and CNN-LSTM-ED—to

predict daily solar PV energy output. Due to the challenges of handling instability and unpredictability in continuous historical data with only LSTM, we prefer to use several ED-based LSTM architectures. To our knowledge, there are no previous studies that compare solar energy prediction using these specific network architectures. The evaluation is done against a 1741-day-long 1-day interval historical dataset. This comprehensive dataset allows us to fully characterize the effectiveness of these methods and to compare them fairly. The panel data are provided by 26 different panels connected to an inverter on the roof of the Hidayet Türkoğlu Sports Complex in Istanbul, Turkey. In addition to the panel data, meteorological data are also used for the prediction step. Three error metrics are used to evaluate the performance of the models: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-square (R^2)) are used to evaluate the performances of the models. The experimental findings reveal that Conv-LSTM-ED has an overall good efficiency in the accuracy of the predictions. Additionally, measurements using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) indicate that Conv-LSTM-ED is superior to the other two models in terms of both fitness and complexity.

The remainder of this paper is organized as follows. The materials and methods used in this study are introduced in Sect. 2, starting with the studies and data collections and moving on to the LSTM-ED architectures that are used. Beginning with the pre- processing of the data, error metrics utilized for evaluation, and hyper-parameter settings of the models, Sect. 3 represents the case study. Results and discussion are found in Sect. 4, commencing with an evaluation of model performance and effectiveness using error metrics, hyper-parameter settings, and figures. Section 5 provides a summary of the findings and suggested further research.

2 Materials and methods

2.1 Studied area and data collection

In order to develop reliable and successful deep learning applications, high-quality, long-term historical data is essential. However, acquiring such data can be challenging because data providers may not record it or may be unwilling to share it. The solar PV panel and meteorological data used in this study are entirely real, having been obtained directly from their providers. The panel data were collected from 26 different panels connected to an inverter on the roof of the Hidayet Türkoğlu Sports Complex in Istanbul, Turkey, at one-day intervals. Data were collected separately from each panel, and experiments were conducted individually for each one. The information obtained from the panels includes the power (P) output of each panel and the energy (E) produced by it. The technical information about the panels is detailed specifically in Table 1.

Solar PV panels exhibit varying generation characteristics depending on the time of year. As a result, meteorological data significantly influence production characteristics and are also used as input features in this study. Due to the annual periodicity in weather patterns, at least one year of data is necessary for training

Table 1 Technical Panel Information

Panel Parameters	Information
Location	Turkey
Inverter Size/Type	25/8 kW Solaredge SE8K
Maximum Efficiency of Inverter	98%
Number of PV Panel	26
PV Panel Type	260 W Solarturk
Panel Current at Maximum Power IMPP	8.53 A
Efficiency	15.92%

algorithms effectively. Consequently, all datasets used in this study encompass panel and meteorological data spanning from February 24, 2017, to November 30, 2021. Each panel is represented by 1741 data points, amounting to a total of 45,266 data points across all panels. The meteorological data were obtained from the Güngören/Davutpaşa Marmara Automatic Meteorology Observation Station, which is operated by the 1st Regional Directorate of the Republic of Turkey Ministry of Environment, Urbanization and Climate Change, Turkish State Meteorological Service, and is located near the Hidayet Türkoğlu Sports Complex. The meteorological variables used in this study are listed in Table 2.

In this study, we predicted the value of energy (E in kWh) for the next day $(t+1)$ using daily input features including power (P in kWe), maximum apparent power (MaxAP), minimum apparent power (MinAP), mean apparent power (MeanAP), maximum relative humidity (MaxRH), minimum relative humidity (MinRH), mean relative humidity (MeanRH), maximum temperature (MaxT), minimum temperature (MinT), mean temperature (MeanT), maximum wind direction (MaxWD), mean wind direction (MeanWD), maximum wind speed

Table 2 Meteorological Features

| Abbreviation	Feature Name	Unit
MaxAP	Maximum actual pressure	hPa
MinAP	Minimum actual pressure	hPa
MeanAP	Mean actual pressure	hPa
MaxRH	Maximum relative humidity	%
MinRH	Maximum relative humidity	%
MeanRH	Mean relative humidity	%
MaxT	Maximum temperature	C°
MinT	Minimum temperature	C°
MeanT	Mean temperature	C°
MaxWD	Maximum wind direction	°
MeanWD	Mean wind direction	°
MaxWS	Maximum wind speed	m/s
MeanWS	Mean wind speed	m/s
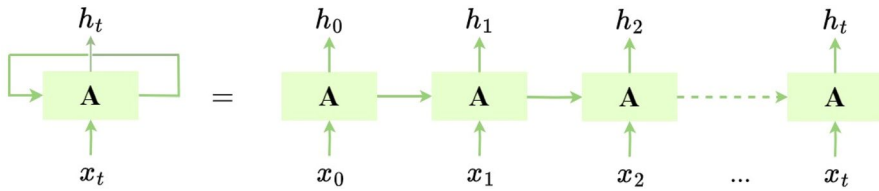
Fig. 1 Unrolled RNN cell
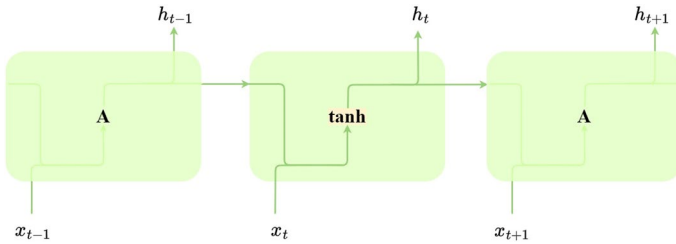
Fig. 2 RNN architecture

(MaxWS), mean wind speed (MeanWS), and the energy (E in kWh) produced on the current day (t).

2.2 Methodology

This study proposes LSTM-ED, Conv-LSTM-ED, and CNN-LSTM-ED models to measure and compare the reliability and accuracy of one-step-ahead PV forecasting. The methods used in this study are summarized as follows.

2.2.1 LSTM

Recurrent Neural Networks (RNNs) are a specialized version of traditional FFNs with cyclic connections to hidden neurons that can handle sequential data processing. Unlike traditional FFNs, which are unable to accept sequential inputs and require all of their inputs (and outputs) to be independent of one another, RNN models realize a sequence-to-sequence mapping between input and output. As a result, the output is derived based on previous computations. For lengthy sequences, RNNs can utilize past sequential information. In practice, however, the duration of the sequential information they can remember is limited to only a few steps back due to RNN memory constraints. Figure 1 illustrates the unrolled RNN, and Fig. 2 depicts the RNN architecture.

x_t and h_t represents the input of the cell and recurrent information of the cell at time t, respectively. tanh is a hyperbolic tangent activation function.

LSTM models effectively extend the memory of RNNs, enabling them to learn and maintain dependencies over long input sequences. This extended memory can retain information for prolonged periods, with capabilities to read, write, and delete

data from its memory cells. The memory unit within an LSTM is known as a 'gated cell,' where 'gate' refers to the mechanism that determines whether to preserve or discard information in the memory. An LSTM model identifies critical features from the input and retains them for extended durations. During the training phase, the significance of the information is controlled by the weight values assigned to it, dictating whether it should be kept or removed. Consequently, an LSTM model learns to discern which information is crucial to keep and which to discard.

An LSTM network generally consists of three gates: the forget gate, the input gate, and the output gate. The forget gate decides whether to retain or discard existing information in the cell state. This decision is made using a sigmoid function that processes information from the previous output and the current input to produce the forget gate's output. Mathematically, this can be expressed as follows:

$$f_t = \sigma\left(W_f \bullet \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

where σ is a sigmoid activation function ranging from value 0 to 1. f_t is the forget gate and W_f is the weight of the forget gate. While h_{t-1} represents the recurrent information of the cell at time $t-1$, x_t represents the input of the cell at time t. And b_f is the bias of the forget gate.

The input gate determines whether the new information should be added to the cell. To do this, a sigmoid layer decides which values will be updated. Subsequently, a tanh layer generates a vector of candidates for the state. The outputs of the sigmoid and tanh layers are written as follows:

$$i_t = \sigma\left(W_i \bullet \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$\tilde{C}_t = tanh\left(W_C \bullet \left[h_{t-1}, x_t\right] + b_C\right) \tag{3}$$

where i_t is the input gate, W_i is the weight of the input gate and b_i is the bias of the input gate. C_t represents the candidate vectors and tanh is a hyperbolic tangent activation function ranging from value -1 to 1. While W_C is the weight of the candidate state, b_C is the bias of the cell state. By multiplying the output of these two layers the cell state (C_t) is updated as below:

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

The output gate determines which parts of the current cell state will contribute to the final output. To achieve this, the sigmoid function takes the previous state and the current input into account. The new cell state is then obtained by multiplying the tanh function's output with the sigmoid layer's output. This new state is passed on to the next state. The mathematical equations for the output gate are as follows:

$$o_t = \sigma\left(W_o \bullet \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t * tanh\left(C_t\right) \tag{6}$$

$$f_t = \sigma\left(W_f \bullet \left[C_{t-1}, h_{t-1}, x_t\right] + b_f\right) \tag{7}$$
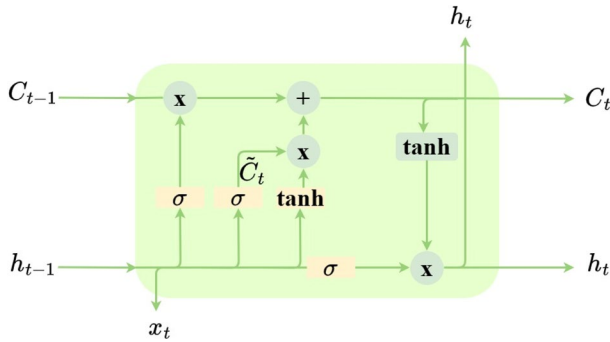
Fig. 3 Inner structure of LSTM unit
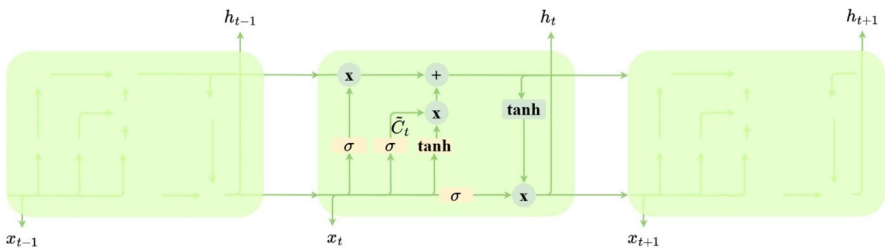
Fig. 4 LSTM architecture

$$o_t = \sigma\left(W_o \bullet \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$i_t = \sigma\left(W_i \bullet \left[C_{t-1}, h_{t-1}, x_t\right] + b_i\right) \tag{8}$$

$$o_t = \sigma\left(W_o \bullet \left[C_{t-1}, h_{t-1}, x_t\right] + b_o\right) \tag{9}$$

where o_t is the output gate, W_t is the weight of the output gate and b_o is the bias of the output gate. In the equations above while '•' represents the connection of two vectors, element-wise multiplication is represented by '*'. Inner structure of the LSTM unit according to all equations mentioned above is given in Fig. 3.

The LSTM architecture is given with Fig. 4.

2.2.2 Conv-LSTM

Conv-LSTM is a variation of LSTM that was initially presented by Shi et al. [44] to address spatio-temporal prediction challenges in precipitation forecasting. In a Conv-LSTM network, the input features are 3D spatio-temporal tensors, with the first two dimensions representing the spatial dimension. Unlike the classic LSTM model, the Conv-LSTM cell's input-to-state and state-to-state transitions use convolutional operations that produce 3D tensors instead of matrix multiplication. The following equations can be used to construct this model, where convolution is indicated by the '*' symbol and the Hadamard product is shown by the '∘' symbol:

$$i_t = \sigma\left(W_{xi} - X_t + W_{hi} - H_{t-1} + W_{Ci}\circ C_{t-1} + b_i\right) \tag{10}$$

$$f_t = \sigma\left(W_{xf} - X_t + W_{hf} - H_{t-1} + W_{Cf}\circ C_{t-1} + b_f\right) \tag{11}$$

$$C_t = f_t\circ C_{t-1} + i_t\circ\tanh\left(W_{xC} * X_t + W_{hC} * H_{t-1} + b_C\right) \tag{12}$$

$$o_t = \sigma\left(W_{xo} - X_t + W_{ho} - H_{t-1} + W_{Co}\circ C_{t-1} + b_o\right) \tag{13}$$

$$H_t = o_t\circ\tanh\left(c_t\right) \tag{14}$$

where $X_1, \ldots, X_t$ are the inputs, $C_1, \ldots, C_t$ are the output of cells and $H_1, \ldots, H_t$ are the hidden states. i_t, f_t and o_t are input, forget and output gates, respectively. The inner structure of the Conv-LSTM unit, as described by Eqs. (10–14), is depicted in Fig. 5.

2.2.3 CNN-LSTM

CNN is frequently used in feature engineering—because it has the ability to focus on the most important features of the data—and in time series forecasting as well. Many time-series tasks that feed-forward networks with fixed-size time windows cannot handle can be addressed using LSTM. CNN-LSTM, on the other hand, combines the advantages of both CNN and LSTM networks. In this sense, it can be considered a hybrid network that allows for various combinations of CNN and LSTM architectures. The main architecture of this network typically includes a one-dimensional convolutional layer, a pooling layer, an LSTM hidden layer, and a fully connected layer. The architecture of the CNN-LSTM is illustrated in Fig. 6.

2.2.4 Encoder-decoder models

In many sequence-to-sequence applications, where the goal is to construct a model that predicts future outputs from a given set of historical or past data, the encoder-decoder paradigm is widely used in computer science and related fields (Ekinci,
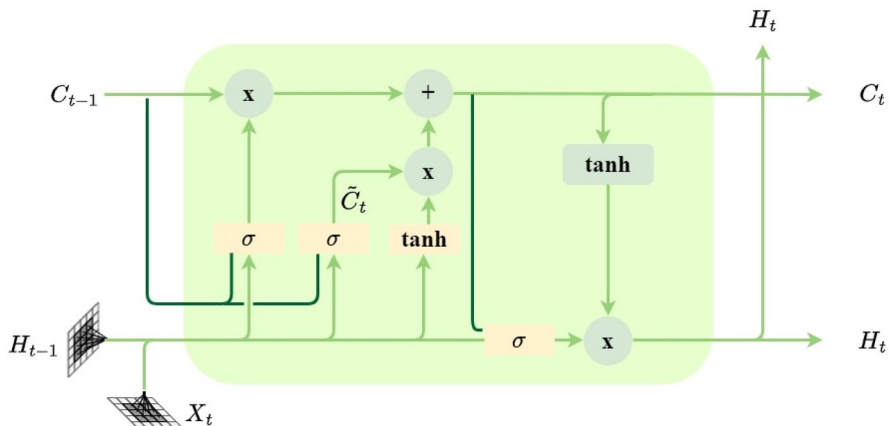

Fig. 5 The inner structure of Conv-LSTM

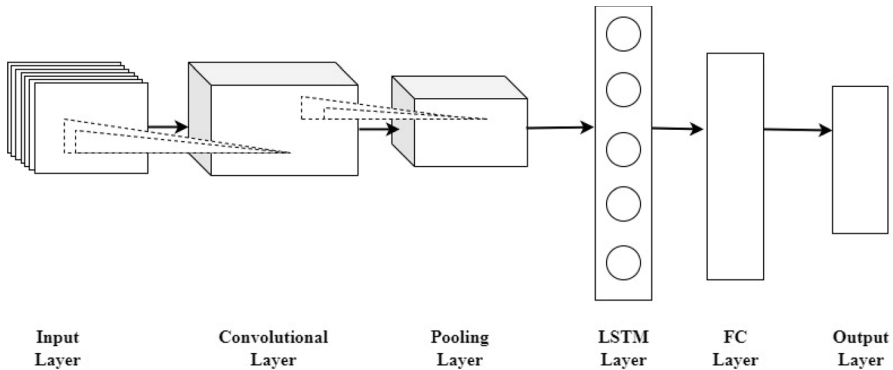| Input | Convolutional | Pooling | LSTM | FC | Output |
| Layer | Layer | Layer | Layer | Layer | Layer |

Fig. 6 Architecture of CNN-LSTM

Omurca, & Özbay, 2021; Gürses-Tran, Körner, & Monti, 2022; Qiu, Tang, Liu, Teng, & Yao, 39; B. Zhang et al., [53] Zhu, Li, Mao, Li, & Yan, 57). However, LSTM architectures are not always effective for sequence-to-sequence prediction due to difficulties arising from fixed input and output shapes (Zdravkovic, Ciric, & Ignjatovic, 2022). To address this issue, Cho et al. [9] proposed using RNNs with an encoder-decoder model. In this model, one RNN network serves as the encoder, extracting timing characteristics from historical data, while another RNN network acts as the decoder, predicting the time series. The encoder compresses the data from the variable-length input sequence into a vector derived from the sequence of RNN hidden states. The final hidden state of the encoder provides a fixed-dimensional representation of the input sequence. The decoder uses one RNN network to generate the variable-length output sequence [15]; Gangopadhyay, Tan, Huang, & Sarkar, 17; Gupta, Malhotra, Narwariya, Vig, & Shroff, 18). The encoder-decoder model is a generic approach for learning a conditional distribution of variable-length sequences based on the assumption of another variable-length sequence, where the input and output may not be the same length [31].

Sutskever et al. [45] proposed replacing the RNNs in the ED model with LSTMs for natural language processing (NLP) tasks. The first LSTM layer reads the input language sentences and obtains their vector representation of the highest dimension. The second LSTM layer then generates sentences in the output language from this vector. This model is referred to as LSTM-ED.

Unlike the traditional LSTM model, LSTM-ED reads the entire input sequence into a fixed-length vector (Nguyen, Kalra, & Kim, 36). This internal representation is both the initial hidden state of the decoder and the final hidden state of the encoder. Its built-in "memory" allows it to remember long input sequences, which is why the model is also known as sequence embedding and has been shown to perform well with very long input sequences.

In the Conv-LSTM-ED model, Conv-LSTM is used for encoding, and LSTM is used for decoding. The incorporation of Conv-LSTM as the encoder enhances the model's feature extraction capabilities. This aligns with our hypothesis that enhancing the model to extract data features will improve the prediction

accuracy. The Conv-LSTM model reads a sequential input and passes it directly to a decoder LSTM through convolutional operations.

Another variation of the LSTM-ED model incorporates a CNN in the encoder layer. While CNNs alone cannot directly process sequential data, they are used to read the data and extract key features. An LSTM then decodes these features and forecasts a sequential series. This model is called CNN-LSTM-ED because it uses a CNN as the encoder and an LSTM as the decoder.

3 Case Study

3.1 Data preparation

Missing data is a problem that must be addressed in the context of data science applications. The presence of missing data not only results in a loss of information but also hinders the ability of models to extract hidden patterns from datasets (T. Liu, Jin, Zhong, & Xue, 2022). Consequently, rows with missing data are either removed or the gaps are filled using various methods. However, deleting rows with missing data is not an advisable approach, especially for time series data. Instead, missing data should be filled using time interpolation. Interpolating missing values is a crucial step in data preprocessing in data science. In the dataset we are examining, missing values are present only in the P and E features. It has been observed that there are a total of 4,350 missing values across 26 panels in the dataset. These missing values have been filled using the padding method. Padding interpolation involves filling in missing values with the last available value above the missing entry [5].

The experimental power and energy data used in this study range from 0 to 157.1 and from 0 to 3463, respectively. Model accuracy can be adversely affected by imbalances in data ranges. To address this issue, min-max normalization is employed, which maps the range to [0, 1]. Min-max normalization facilitates accurate predictions by preserving the relationships between data instances. The formula for min-max normalization is as follows:

$$v_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{15}$$

where x_i denotes the i_{th} data point, x_{min} denotes the data point with the lowest value, and x_{max} denotes the data point with the highest value. The normalized version of x_i, denoted by v_i, lies between [0, 1].

Training and test sets must be properly defined for accurate evaluation of models. However, there is no fixed rule for splitting a dataset into training and test sets. In this case, the dataset is divided into a 75% training set and a 25% test set. The training set contains 1307 observations, while the test set comprises 434 observations.

3.2 Error metrics

We employ RMSE, MAE, R^2, and loss value as error metrics in our experiments to evaluate the performance of the experimental algorithms. Additionally, we use the AIC and BIC criteria to assess the architectures.

RMSE, which estimates the standard deviation of the residuals, is the square root of the average squared difference between the observed and predicted values. A higher RMSE indicates a worse fit, with greater dispersion around the mean, while a lower RMSE signifies a better fit, with smaller discrepancies between the observed and predicted values. Therefore, a lower RMSE value is preferable. The formula for calculating RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_i - \widehat{y}_i} \tag{16}$$

where y_i is the expected value, $\widehat{y}_i$ is the predicted value of ith observation. n is the total number of observations in the test set.

MAE represents the average absolute difference between the observed and predicted values. A lower MAE indicates that the prediction error of the model is smaller, which corresponds to more accurate predictions. The formula for calculating MAE is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \widehat{y}_i| \tag{17}$$

R^2, for instance, is the square of the correlation coefficient between the observed and predicted values, and it assesses how well the model explains the variance in the data [12]. A high R^2 value indicates a strong correlation between the observed and predicted values, whereas a low R^2 value suggests a weak correlation. R^2 values range from [0, 1]. If R^2 is greater than 0.8, the model is considered to be reliable; if it is lower, the model is considered to be less reliable [24]. R^2 is calculated using the following formula:

$$R^2 = \frac{\sum_{i=1}^{n} \left(y_i - \widehat{y}_i \right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y} \right)^2} \tag{18}$$

where $\bar{y}$ is the mean of expected values of variable y.

When working with different models, it is very important to perform calculations related to statistical model fitting, as model selection is based on these results. The most popular techniques for choosing between models are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). While R-squared (R²) is used solely to evaluate how well the model fits the data, AIC and BIC also take into account model complexity [7, 13].

The AIC, which assesses the fit of the model, is calculated using the following formula:

$$AIC = L \times \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i - \widehat{y}_i\right) + 2 \times F \tag{19}$$

The effectiveness of the model for data prediction is assessed using the Bayesian Information Criterion (BIC), which is calculated as follows:

$$BIC = L \times \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i - \widehat{y}_i\right) + F \times \log(F) \tag{20}$$

In Eqs. 19 and 20, 'L' represents the number of observations in the dataset, and 'F' denotes the number of features. A model is considered optimal when both the AIC and BIC values are at their lowest. The fundamental distinction between these two criteria is that the BIC tends to prefer simpler models, while the AIC may favor more complex models [1].

3.3 Hyper-parameters of ED-based LSTM architectures

In the Jupyter IDE, we construct our environment using the Python programming language along with the sklearn and Keras libraries.

In this study, we have suggested comparing ED based LSTM architectures to determine which model and which hyper-parameters are better for forecasting energy (E).

In the LSTM-ED architecture, the number of neurons in the input layer is set to 15, corresponding to the number of features in the datasets. There is no general rule for determining the optimal number of hidden neurons; however, our LSTM layers have 128 hidden neurons. The LSTM encoder receives input sequences, and a repeat vector is used to feed the compressed vector from the LSTM encoder to the LSTM decoder, which also has 128 hidden neurons. The output states of the LSTM encoder serve as inputs to the LSTM decoder. Finally, two time-distributed layers transform the output of the LSTM decoder into the predicted output. A dense layer applies within the time-distributed layer, acting as a wrapper layer (Wolf & Yang, 2020). The first dense layer has 100 units with the Rectified Linear Unit (ReLU) activation function, followed by a second dense layer with a single unit. An activation function is essential for assessing the deep learning model's performance. To conclude, a dropout layer set at 0.2 acts as a regularizer to reduce overfitting, and a flatten layer is included.

In the Conv-LSTM-ED model, a two-dimensional Conv-LSTM is used as an encoder to handle one-dimensional, univariate time series, while an LSTM is used as a decoder, similar to the previous architecture. The ConvLSTM2d, which supports two-dimensional Conv-LSTM models, is now capable of handling one-dimensional univariate time-series data. A two-dimensional convolutional layer extracts 128 feature mappings from the input sequence. These feature maps are then amplified. Following the convolutional layer, a flatten layer creates a vector of 128 values by flattening the output feature map values. A repeat vector feeds the compressed vector from the flatten layer to the LSTM decoder with 128 hidden neurons. ReLU

is used as an activation function in both the encoder and decoder. Finally, two time-distributed layers transform the output of the LSTM decoder into the predicted output. The first dense layer with 100 units uses ReLU as the activation function, followed by a second dense layer with a single unit. A dropout layer of 0.2 is added at the end.

We have also developed a CNN-LSTM-ED model for a single variable that uses two convolutional layers as the encoder and LSTM as the decoder. The sequence from the previous day serves as the model's input. With a kernel size of one timestamp, the first convolutional layer extracts 128 feature mappings from the input sequence. The second layer performs the same function on the feature maps created by the first layer and amplifies the features. This second convolutional layer is followed by a max-pooling layer, which reduces the feature maps by half while retaining the maximum values of the features. Next, a long vector comprising 128 values is created by flattening the output feature map values from the max-pooling layer. A repeat vector feeds the compressed vector from the flatten layer to the LSTM decoder with 125 hidden neurons. ReLU is used as the activation function in both the encoder and decoder. Finally, two time-distributed layers transform the output of the LSTM decoder into the predicted output. The first dense layer with 100 units uses ReLU as the activation function, followed by a second dense layer with a single unit. A dropout layer of 0.2 is included at the end.

Optimizing the loss function is crucial for developing deep learning models and fine-tuning parameters. The optimization of the loss function has gained significant importance, especially in recent times. Throughout the training phase of all models, we employ the Adam optimizer, which uses a stochastic gradient descent algorithm to compute first and second-order moments. Adam is known for its quick convergence, which is the primary reason for its use in this study. All models are trained with a batch size of 72, over 50 and 100 epochs, respectively.

The proposed architectures are illustrated in Fig. 7.

4 Results and discussions

Five evaluation metrics were used to assess the performance of the ED-based LSTM, Conv-LSTM, and CNN-LSTM models. To confirm the models' reliability, the study evaluated their performance across 26 panels.

4.1 Evaluation of model performances

It is important to note that the training algorithms and structural compositions of the ED-based LSTM, Conv-LSTM, and CNN-LSTM models differ from one another. Consequently, this study investigated the performance and stability of each model. Using the training datasets, all three models underwent 50 and 100 iterations of training. Subsequently, their performances were evaluated using the test datasets. The model that produced the highest average R^2 value over a single time step in the test rounds was selected as the best model for each experiment. The top-performing
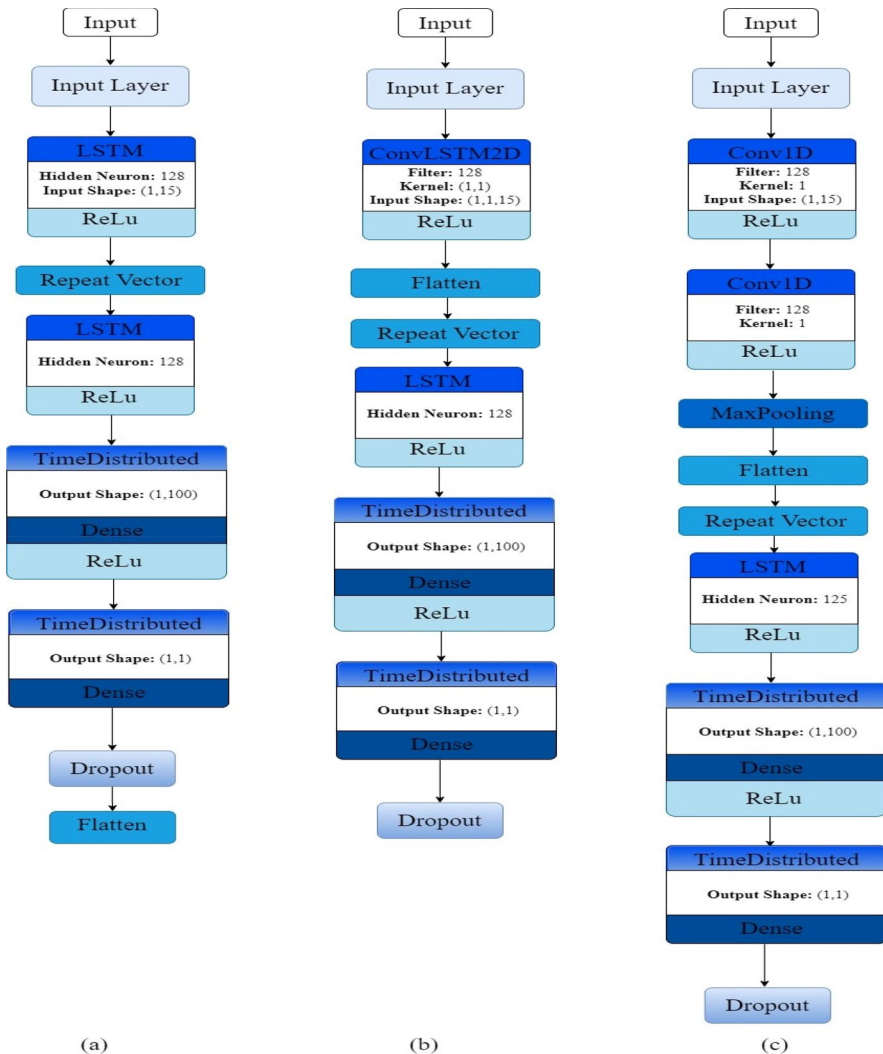
Fig. 7 (a) LSTM-ED (b) Conv-LSTM-ED (c) CNN-LSTM-ED

ED-based LSTM models were then compared. Finally, the findings from the LSTM-ED models are presented.

The results for each iteration of the LSTM-ED models during the test phases are displayed in Tables 3 and 4. With low RMSE and MAE values, and exceptionally high R^2 values in the test phases, it appears that for 50 iterations, all models could be adequately trained for nearly all panels on average. However, a problem with all models was observed concerning panel 1 (p1) in terms of R^2. Consequently, the success of time series prediction is as dependent on the data quality as it is on the algorithms used. When the error metrics are averaged over 100 iterations, it is

Table 3 Performances of the LSTM-ED models for 50 iterations for test dataset

Panel	LSTM-ED			Conv-LSTM-ED			CNN-LSTM-ED		
	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2
p1	32.69	24.53	0.65	31.69	23.83	0.71	32.8	25.06	0.64
p2	20.66	15.95	0.88	20.01	15.24	0.88	20.62	15.87	0.83
p3	21.45	16.39	0.87	21.07	16.2	0.88	21.73	16.97	0.83
p4	21.73	16.42	0.87	21.41	16.04	0.88	21.86	16.75	0.83
p5	21.88	16.9	0.86	21.41	16.38	0.88	22.98	17.61	0.82
p6	22.04	17.2	0.86	21.76	16.74	0.87	23.26	18.17	0.81
p7	15.79	12.19	0.89	15.62	11.73	0.91	17.36	12.94	0.85
p8	16.28	12.67	0.89	16.2	12.26	0.9	17.51	13.09	0.85
p9	16.93	12.82	0.89	16.28	12.67	0.89	17.68	13.41	0.84
p10	**14.39**	**10.74**	**0.94**	**14.17**	10.81	**0.96**	**14.74**	**10.97**	**0.89**
p11	17.22	13.05	0.88	16.92	13.09	0.89	17.97	13.69	0.84
p12	17.31	13.23	0.88	17.18	13.35	0.88	18.16	14.12	0.84
p13	17.74	13.34	0.88	17.54	12.59	0.88	18.79	14.41	0.84
p14	18.28	14.1	0.88	17.81	13.72	0.88	18.83	14.87	0.84
p15	14.66	11.26	**0.94**	14.39	**10.72**	0.95	15.4	11.96	**0.89**
p16	14.84	11.44	0.93	14.83	11.01	0.93	15.42	11.96	**0.89**
p17	14.96	11.5	0.92	14.92	11.45	0.93	15.51	12	0.88
p18	15.37	11.83	0.92	15.29	11.91	0.92	15.56	12.05	0.86
p19	15.68	11.95	0.9	15.4	11.87	0.92	15.94	12.4	0.86
p20	22.04	17.26	0.86	21.81	16.75	0.87	24.35	18.18	0.81
p21	23.03	17.49	0.86	23.23	17.74	0.87	25.24	18.68	0.81
p22	24.15	17.47	0.85	23.99	17.46	0.86	25.62	19.62	0.8
p23	25.28	17.96	0.85	24.44	18.82	0.86	25.68	19.74	0.77
p24	25.89	19.73	0.85	25.11	17.58	0.86	26.8	19.8	0.76
p25	26.98	20.96	0.85	26.34	20.64	0.86	27.63	21.24	0.76
p26	27.69	21.21	0.84	27.4	20.33	0.86	28.29	21.67	0.74
Average	20.19	15.37	0.87	19.85	15.04	0.88	20.99	16.05	0.82

observed that the models fail more frequently at 100 iterations than at 50 iterations across all metrics. Overall, the performance results indicate that the Conv-LSTM-ED model surpasses the LSTM-ED and CNN-LSTM-ED models in the test phases by achieving higher R^2 values and lower RMSE and MAE values. The CNN-LSTM-ED model exhibits much larger error ranges across all three metrics than the other two models for all iterations. These findings suggest that the Conv-LSTM-ED model excels in terms of model stability, reliability, and accuracy.

The scatter diagrams for the LSTM-ED, Conv-LSTM-ED, and CNN-LSTM-ED models for p1 (the worst) and p10 (the best) predictions after 50 and 100 iterations are shown in Figs. 8 and 9, respectively.

When examining the spread of the scatter plots, it is observed that the spread for p10 is smaller than that for p1, and the spread obtained after 50 iterations is also

Table 4 Performances of the LSTM-ED models for 100 iterations for test dataset

Panel	LSTM-ED			Conv-LSTM-ED			CNN-LSTM-ED		
	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2
p1	33.43	25.31	0.55	32.94	24.92	0.71	36.28	27.38	0.44
p2	17.57	13.72	0.85	17.15	13.45	0.88	18.19	14.3	0.75
p3	18.04	14.11	0.85	17.27	13.17	0.87	19.46	15.59	0.75
p4	18.12	12.97	0.85	18.03	12.98	0.86	20.09	15.45	0.75
p5	18.76	14.75	0.84	18.46	14.38	0.86	20.41	14.72	0.75
p6	21.43	16.221	0.84	20.46	15.8	0.86	22.29	16.96	0.74
p7	21.83	16.94	0.84	21.21	16.13	0.85	22.85	17.96	0.74
p8	21.88	16.54	0.84	21.6	16.35	0.85	23.15	18	0.73
p9	21.88	16.54	0.83	22.04	17.15	0.85	23.17	17.97	0.72
p10	**14.71**	**10.93**	**0.93**	**14.55**	**10.93**	**0.94**	**15.42**	**11.66**	**0.87**
p11	15.2	11.82	0.89	14.68	11.49	0.92	16.02	12.44	0.83
p12	15.21	11.87	0.89	14.75	11.43	0.9	16.25	12.57	0.82
p13	15.26	11.82	0.89	15.11	11.63	0.89	16.64	12.4	0.8
p14	15.65	12.2	0.88	15.23	11.71	0.89	17.31	13.45	0.8
p15	15.92	12.34	0.87	15.33	11.38	0.89	17.4	13.48	0.77
p16	16	12.02	0.86	15.8	12.33	0.88	17.91	13.57	0.77
p17	16.88	12.85	0.85	16.74	12.64	0.88	17.98	13.55	0.76
p18	17.27	13.65	0.85	17.07	13.13	0.88	18.18	14.18	0.76
p19	23.01	16.77	0.83	22.49	17.53	0.85	24	18.82	0.71
p20	23.13	17.96	0.81	23.23	17.67	0.84	24.16	18.22	0.71
p21	23.22	17.78	0.81	23.57	18.17	0.84	26.26	19.45	0.71
p22	23.89	18.51	0.8	24.6	17.64	0.83	26.71	18.86	0.7
p23	25.02	17.2	0.8	24.63	18.98	0.82	26.88	20.57	0.69
p24	25.57	19.89	0.79	25.27	17.41	0.81	27.24	21.25	0.68
p25	26.35	20.48	0.79	26.28	20.04	0.8	27.76	21.92	0.67
p26	28.57	21.55	0.76	27.6	20.44	0.8	28.16	21.27	0.65
Average	20.53	15.64	0.83	20.23	15.34	0.86	21.93	16.77	0.73

smaller than the spread obtained after 100 iterations. From this, we can infer that the differences between the expected and predicted values for p10 and after 50 iterations are smaller, respectively.

To assess the generalization capacity of each model on the test dataset, we examined the loss plot for panel p10 shown in Fig. 10. The relatively low test loss for each model suggests that their generalization capacity is satisfactory, indicating that the models perform successfully.

4.2 Evaluation of model efficiencies

When selecting a model, the one that minimizes the information criterion is typically chosen. In the literature, two common information criteria are the Akaike
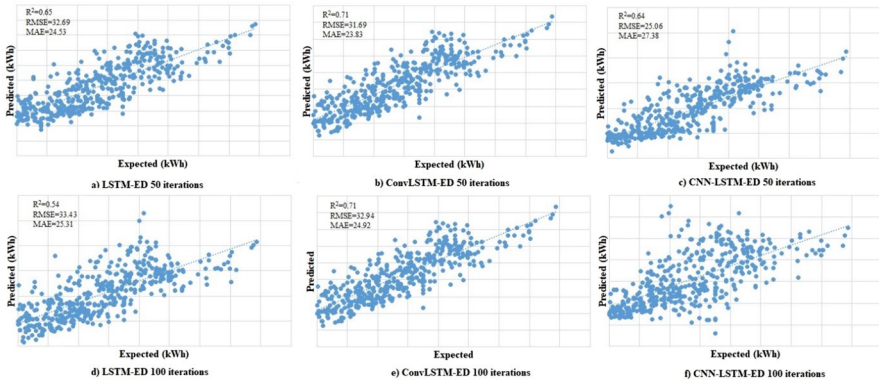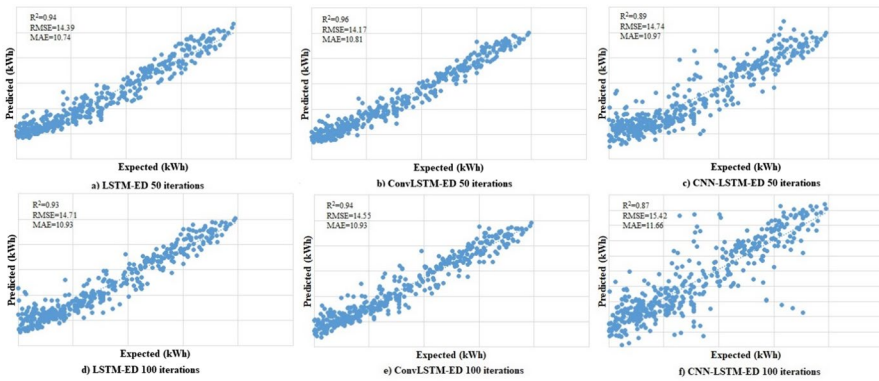
Fig. 8 Scatter diagrams of the p1

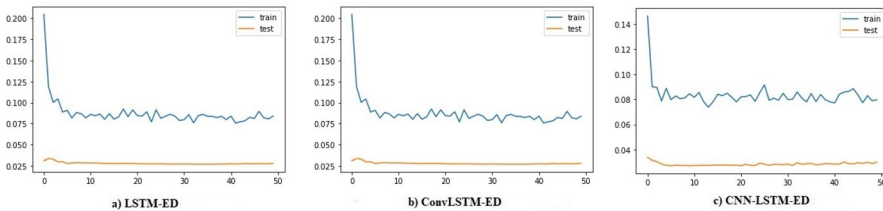**Fig. 9** Scatter diagrams of the p10

Fig. 10 Loss values of the p10

Information Criterion (AIC) and the Bayesian Information Criterion (BIC). AIC tends to favor larger models due to its lighter penalty, which can lead to inconsistency. In contrast, BIC includes a larger penalty term, which promotes consistency but can also be less effective.

To determine the correct model order, the magnitudes of these information criteria are compared across several models. Box plots are often utilized to assess model quality and to identify the model that best fits the data. Figure 11 presents the AIC and BIC box plots for each iteration count.
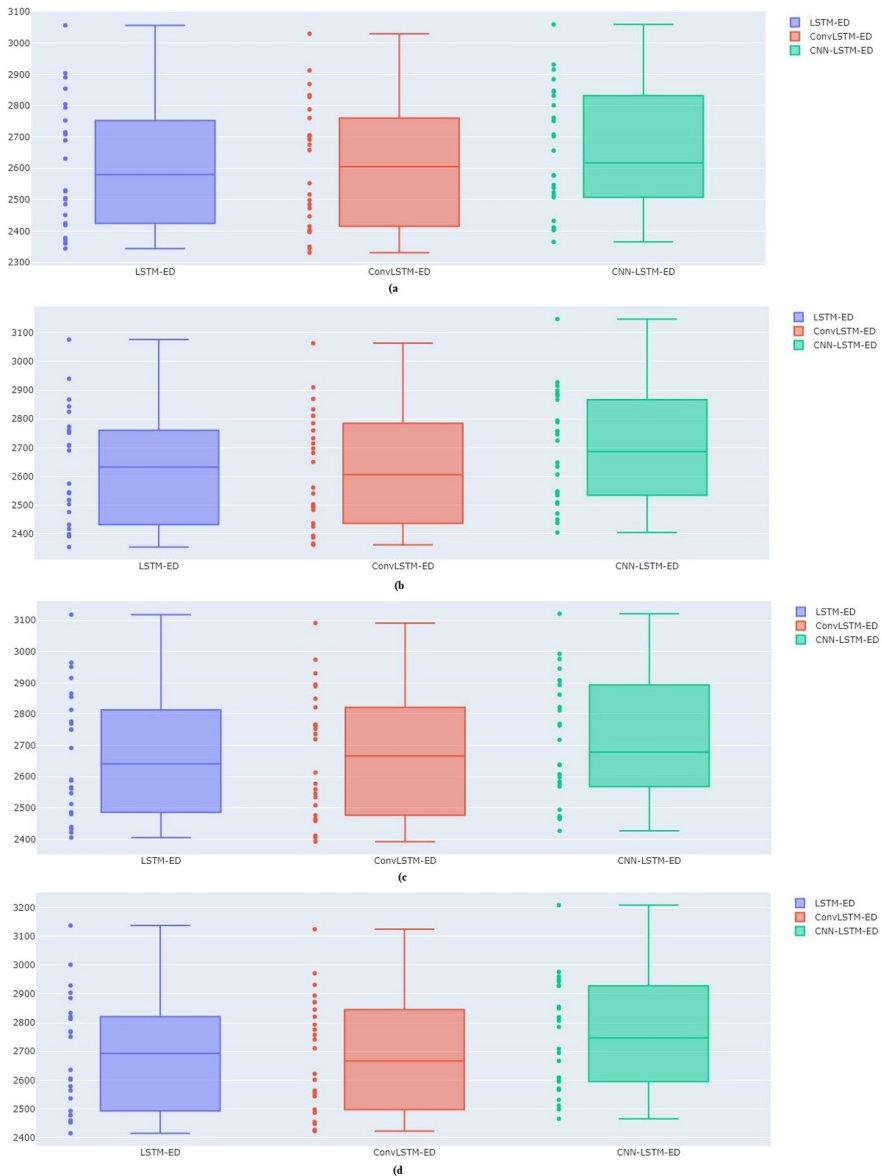
Fig. 11 (a) AIC for 50 iterations (b) AIC for 100 iterations (c) BIC for 50 iterations (d) BIC for 100 iterations

Upon examining Fig. 11, it is clear that the AIC and BIC values are in agreement with the R^2, RMSE, and MAE values. The lowest AIC and BIC values are achieved with the Conv-LSTM-ED model at 50 iterations. Furthermore, a comparison between the models and the number of epochs indicates that neither the specific model nor the number of epochs provides a distinct advantage for the current dataset. Combining all the results, it becomes apparent that the Conv-LSTM-ED model demonstrates robust capabilities in forecasting solar photovoltaic energy.

5 Conclusions

This study examines the performance of ED based LSTM models for predicting the solar energy output of PV panels. To demonstrate the viability and effectiveness of the compared models, several case studies are presented as examples. The findings reveal that the proposed models have successfully predicted the energy output of solar PV panels with precise accuracy and reasonable prediction intervals.

Initially, the ED-based LSTM models are tested on 26 PV panels connected to a single inverter to facilitate comparison. Subsequently, the relationship between the models' performance and their complexity is evaluated. The experimental dataset was collected from an inverter connected to 26 distinct panels installed on the roof of the Hidayet Türkoğlu Sports Complex in Istanbul, Turkey. The study also considers meteorological factors that significantly influence the panels' performance for energy prediction. The results show that the Conv-LSTM-ED model outperforms the LSTM-ED and CNN-LSTM-ED models, although all three networks exhibit similar accuracy and are highly effective for making one-day-ahead predictions.

This work compares application-ready deep learning models from a high-level programming library, with a focus on solar PV panel energy prediction. As such, it serves as a guide for researchers seeking to develop practical solutions. Future research may explore the impact of varying look-ahead times on model performance. Additionally, an investigation could be conducted to determine which subset of features plays a more significant role in prediction accuracy through feature selection.

Declarations

Competing interests The authors declare no competing interests.

Conflict of 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.

Ethical approval Not applicable as no human and/or animal data is involved.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Agiakloglou C, Tsimpanos A (2021) Evaluating information criteria for selecting spatial processes. Ann Reg Sci 66(3):677–697
2. Al-Ghussain L (2019) Global warming: review on driving forces and mitigation. Environ Prog Sustain Energy 38(1):13–21
3. Alkhayat G, Mehmood R (2021) A review and taxonomy of wind and solar energy forecasting methods based on deep learning. Energy and AI 4:100060
4. AlShafeey M, Csáki C (2021) Evaluating neural network and linear regression photovoltaic power forecasting models based on different input methods. Energy Rep 7:7601–7614
5. Ay S, Ekinci E (2022) Comparison of machine learning and deep learning meth- ods for modeling ozone concentrations. J Intell Systems: Theory Appl 5(2):106–118
6. Belmahdi B, Louazzani M, Bouardi E, A (2022) Comparative optimization of global solar radiation forecasting using machine learning and time series models. Environ- mental Science and Pollution Research 29(10):14871–14888
7. Bode NW, Ronchi E (2019) Statistical model fitting and model selection in pedestrian dynamics research. Collective Dynamics 4:1–32
8. Chang R, Bai L, Hsu C-H (2021) Solar power generation prediction based on deep learning. Sustain Energy Technol Assess 47:101354
9. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*
10. De Giorgi MG, Congedo PM, Malvoni M (2014) Photovoltaic power forecasting using statistical methods: impact of weather data. IET Sci Meas Technol 8(3):90–97
11. De Leone R, Pietrini M, Giovannelli A (2015) Photovoltaic energy production forecast using support vector regression. Neural Comput Appl 26(8):1955–1962
12. De Luca D, Fiano F, Mazzenga F, Monti C, Ridolfi S, Vallone F (2007) Outdoor path loss models for ieee 802.16 in suburban and campus-like environments. In *2007 ieee international conference on communications* (pp. 4902–4906)
13. Ekinci E, Omurca S, İ., Ö zbay B (2021) Comparative assessment of modeling deep learning networks for modeling ground-level ozone concentrations of pandemic lock- down period. Ecol Model 457:109676
14. Elsaraiti M, Merabet A (2022) Solar power forecasting using deep learning techniques. IEEE Access 10:31692–31698
15. Fang X, Yuan Z (2019) Performance enhancing techniques for deep learning models in time series forecasting. Eng Appl Artif Intell 85:533–542
16. Frederiksen CAF, Cai Z (2022) Novel machine learning approach for solar photovoltaic energy out- put forecast using extra-terrestrial solar irradiance. Appl Energy 306:118152

17   Gangopadhyay T, Tan SY, Huang G, Sarkar S (2018) Temporal attention and stacked lstms for multivariate time series prediction

18   Gupta P, Malhotra P, Narwariya J, Vig L, Shroff G (2020) Transfer learning for clinical time series analysis using deep neural networks. J Healthc Inf Res 4(2):112–137

19   Gu¨rses-Tran G, K¨orner TA, Monti A (2022) Introducing explainability in sequence- to-sequence learning for short-term load forecasting. Electr Power Syst Res 212:108366. https://doi.org/10.1016/j.epsr.2022.108366

20   Hafiz F, Awal M, de Queiroz AR, Husain I (2020) Real-time stochastic optimization of energy storage management using deep learning-based forecasts for residential pv applications. IEEE Trans Ind Appl 56(3):2216–2226

21   Hossain MS, Mahmood H (2020) Short-term photovoltaic power forecasting using an lstm neural network and synthetic weather forecast. Ieee Access 8:172524–172533

22   Ishaq M, Kwon S et al (2022) A cnn-assisted deep echo state network using multiple time-scale dynamic learning reservoirs for generating short-term solar energy forecasting. Sustain Energy Technol Assess 52:102275

23   Khan W, Walker S, Zeiler W (2022) Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach. Energy 240:122812

24   Kumar HA, Ramana VV (2020) Influence of tool parameters on the tensile properties of friction stir welded aluminium 5083 and 6082 alloys. Mater Today: Proc 27:951–957

25   Lai CS, Zhong C, Pan K, Ng WW, Lai LL (2021) A deep learning based hybrid method for hourly solar radiation forecasting. Expert Syst Appl 177:114941

26   Li P, Zhou K, Lu X, Yang S (2020) A hybrid deep learning model for short-term pv power forecasting. Appl Energy 259:114216

27   Li Z, Xu R, Luo X, Cao X, Du S, Sun H (2022) Short-term photovoltaic power prediction based on modal reconstruction and hybrid deep learning model. Energy Rep 8:9919–9932

28   Lin P, Peng Z, Lai Y, Cheng S, Chen Z, Wu L (2018) Short-term power prediction for photovoltaic power plants using a hybrid improved kmeans-gra-elman model based on multivariate meteorological factors and historical power datasets. Energy Conv Manag 177:704–717

29   Liu R, Wei J, Sun G, Muyeen S, Lin S, Li F (2022) A short-term probabilistic photovoltaic power prediction method based on feature selection and improved lstm neural network. Electr Power Syst Res 210:108069

30   Lorenz E, Scheidsteger T, Hurka J, Heinemann D, Kurz C (2011) Regional Pv power prediction for improved grid integration. Prog Photovoltaics Res Appl 19(7):757–771

31   Luo T, Cao X, Li J, Dong K, Zhang R, Wei X (2021) Multi-task prediction model based on convlstm and encoder-decoder. Intell Data Anal 25(2):359–382

32   Luo X, Zhang D, Zhu X (2021) Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge. Energy 225:120240

33   Malvoni M, De Giorgi MG, Congedo PM (2017) Forecasting of pv power generation using weather input data-preprocessing techniques. Energy Procedia 126:651–658

34   Markovics D, Mayer MJ (2022) Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction. Renew Sustain Energy Rev 161:112364

35   Mutezo G, Mulopo J (2021) A review of Africa's transition from fossil fuels to renewable energy using circular economy principles. Renew Sustain Energy Rev 137:110609

36   Nguyen HM, Kalra G, Kim D (2019) Host load prediction in cloud computing using long short-term memory encoder–decoder. J Supercomputing 75(11):7592–7605

37   Ospina J, Newaz A, Faruque MO (2019) Forecasting of pv plant output using hybrid wavelet-based lstm-dnn structure model. IET Renew Power Gener 13(7):1087–1095

38   Park MK, Lee JM, Kang WH, Choi JM, Lee KH (2021) Predictive model for pv power generation using rnn (lstm). J Mech Sci Technol 35(2):795–803

39   Qiu W, Tang Q, Liu J, Teng Z, Yao W (2019) Power quality disturbances recognition using modified s transform and parallel stack sparse auto-encoder. Electr Power Syst Res 174:105876

40   Qu Y, Xu J, Sun Y, Liu D (2021) A temporal distributed hybrid deep learning model for day-ahead distributed pv power forecasting. Appl Energy 304:117704

41   Rodr´ıguez F, Azc´arate I, Vadillo J, Galarza A (2022) Forecasting intra-hour solar photovoltaic energy by assembling wavelet based time-frequency analysis with deep learning neural networks. Int J Electr Power Energy Syst 137:107777

42   Rodr´ıguez F, Galarza A, Vasquez JC, Guerrero JM (2022) Using deep learn- ing and meteorological parameters to forecast the photovoltaic generators intra-hour output power interval for smart grid control. Energy 239:122116

43   Sheng H, Xiao J, Cheng Y, Ni Q, Wang S (2017) Short-term solar power forecasting based on weighted gaussian process regression. IEEE Trans Industr Electron 65(1):300–308

44   Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W- (2015) c. Con- volutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*

45   Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst, *27*

46   Tang Y, Yang K, Zhang S, Zhang Z (2022) Photovoltaic power forecasting: a hybrid deep learning model incorporating transfer learning strategy. Renew Sustain Energy Rev 162:112473

47   Urquhart B, Ghonima M, Nguyen D, Kurtz B, Chow CW, Kleissl J (2013) Sky-imaging systems for short-term forecasting. Solar Energy Forecasting and Resource Assessment, 195–232

48   Wang F, Xuan Z, Zhen Z, Li K, Wang T, Shi M (2020) A day-ahead pv power forecasting method based on lstm-rnn model and time correlation modification under partial daily pattern prediction framework. Energy Conv Manag 212:112766

49   Yadav O, Kannan R, Meraj ST, Masaoud A (2022) Machine learning based pre- diction of output pv power in india and malaysia with the use of statistical regression. *Mathematical Problems in Engineering*, *2022*

50   Yagli GM, Yang D, Srinivasan D (2019) Automatic hourly solar forecasting using machine learning models. Renew Sustain Energy Rev 105:487–498

51   Ye H, Yang B, Han Y, Chen N (2022) State-of-the-art solar energy forecasting approaches: critical potentials and challenges. Front Energy Res 10:1–5

52   Zdravkovi´c M, C´iri´c I, Ignjatovi´c M (2022) Explainable heat demand forecasting for the novel control strategies of district heating systems. Annu Rev Control.

53   Zhang B, Zou G, Qin D, Lu Y, Jin Y, Wang H (2021) A novel encoder-decoder model based on read-first lstm for air pollutant prediction. Sci Total Environ 765:144507

54   Zhang C, Li Z, Jiang H, Luo Y, Xu S (2021) Deep learning method for evaluating photovoltaic potential of urban land-use: a case study of Wuhan, China. Appl Energy 283:116329

55   Zhang J, Tan Z, Wei Y (2020) An adaptive hybrid model for day-ahead photovoltaic output power prediction. J Clean Prod 244:118858

56   Zhao W, Zhang H, Zheng J, Dai Y, Huang L, Shang W, Liang Y (2021) A point prediction method based automatic machine learning for day-ahead power output of multi-region photovoltaic plants. Energy 223:120026

57   Zhu K, Li Y, Mao W, Li F, Yan J (2022) Lstm enhanced by dual-attention-based encoder-decoder for daily peak load forecasting. Electr Power Syst Res 208:107860