

Received October 26, 2021, accepted November 3, 2021, date of publication November 8, 2021, date of current version November 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125895

Enhanced Deep Belief Network Based on Ensemble Learning and Tree-Structured of Parzen Estimators: An Optimal Photovoltaic Power Forecasting Method

MOHAMED MASSAOUDI^{1,2}, (Member, IEEE), HAITHAM ABU-RUB¹, (Fellow, IEEE), SHADY S. REFAAT¹, (Senior Member, IEEE), MOHAMED TRABELSI³, (Senior Member, IEEE), INES CHIH^{4,5,6}, AND FAKHREDDINE S. OUESLATI^{7,8}

¹Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar

²Laboratoire Matériaux Molécules et Applications (LMMA) à l'IPEST, Carthage University, Tunis 2036, Tunisia

³Department of Electronic and Communications Engineering, Kuwait College of Science and Technology, Safat, Kuwait City 13133, Kuwait

⁴Laboratory of Energy Applications and Renewable Energy Efficiency (LAPER), El Manar University, Tunis 1068, Tunisia

⁵Department of Engineering, Faculty of Science, Technology, and Medicine, University of Luxembourg, 4365 Luxembourg, Luxembourg

⁶National Engineering School of Bizerta, Carthage University, Tunis 2070, Tunisia

⁷National Engineering School of Carthage, Carthage University, Tunis 1054, Tunisia

⁸Laboratoire d'Energétique et des Transferts Thermique et Massique (LETTM), Physics Department, Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis 2092, Tunisia

Corresponding author: Mohamed Massaoudi (mohamed.massaoudi@qatar.tamu.edu)

This work was supported in part by the National Priorities Research Program (NPRP) from the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP12C-33905-SP-213, and in part by Qatar National Library.

ABSTRACT The random fluctuation and non-uniformity of Photovoltaic (PV) power generation greatly affect the power grids' stability and operation. This paper addresses the high volatility of PV power by proposing a precise and reliable ensemble learning model for short-term PV power generation forecasting. The proposed forecasting tool incorporates a base model and meta-model layers. The first-layer base learner combines extreme learning machines, extremely randomized trees, k-nearest neighbor, and mondrian forest models. The meta-model layer exploits deep belief network to generate the final outputs. The hyper-parameters of the proposed stacking ensemble are carefully tuned using the tree-structured of parzen estimators algorithm to achieve top-notch predictive performance. The proposed model is thoroughly assessed through an empirical study using a real data set from Australia. The simulation results confirm the performance superiority of the proposed model over the existing forecasting models with the lowest average root mean square error and mean absolute percentage error of 3.88kW and 2.30%, respectively.

INDEX TERMS Deep belief network, PV power forecasting, stacking ensemble, smart grid, power generation planning.

NOMENCLATURE

Functions:

| | |
|------------|------------------------------|
| σ | Sigmoid function. |
| σ_i | Gaussian standard deviation. |
| D | Euclidean distance. |
| g_i | Activation function. |
| H | Hidden layer matrix. |
| RPP | Random Position Probability. |
| SD | Standard Deviation. |

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano^{1b}.

Variables:

| | |
|----------------|---|
| \aleph | Normalizing constant. |
| β_i | Output weight. |
| λ | Timestep. |
| Π | Random repartition. |
| Θ | network parameter set. |
| θ | Optimal hyperparameter value. |
| c_i | Bias. |
| W_{ij} | Weight between visible and hidden unit. |
| Γ_{TPE} | Optimizer classification coefficient. |

Abbreviations:

| | |
|-------|---|
| BM | Boltzmann Machine. |
| CNN | Convolutional Neural Network. |
| DBN | Deep Belief Network. |
| DKASC | Desert Knowledge Alice Springs Center. |
| DT | Decision Trees . |
| ELM | Extreme Learning Machine. |
| ET | Extremely Randomized Trees. |
| GRU | Gated Recurrent Unit. |
| HO | Hyperparameter Optimization. |
| HP | Hyperparameters. |
| IMFO | Improved Moth-Flame Optimization algorithm. |
| KNN | K-Nearest Neighbors. |
| LSTM | Long-Short Term Memory. |
| MAE | Mean Absolute Error. |
| MAPE | Mean Absolute Percentage Error. |
| MLP | Multi-Layer Perceptron. |
| PSO | Particle Swarm Optimization. |
| PVPF | Photovoltaic power forecasting. |
| RBFNN | Radial Basis Function Neural Network. |
| RBM | Restricted Boltzmann Machine. |
| RMSE | Root Mean Squared Error. |
| TPE | Tree-structured of Parzen Estimators. |
| WPD | Wavelet Packet Decomposition. |

I. INTRODUCTION

With the rapid growth of Photovoltaic (PV) capacity, PV Power Forecasting (PVPF) presents an effective solution to cope with the unexpected changes of weather conditions [1]. The PV forecasts allow the compensation of the deficit in PV power generation from alternative sources. A precise expectation for future PV power generation guarantees a secure and effective system commitment by improving the energy stability and ensuring the grid's reliability [2]. More precisely, the PVPF copes with the weather outliers and provides information integrity to customers and energy suppliers. In addition, accurate forecasting of PV power output is crucial for energy control and management in smart grid systems, especially when the well-known concept adopted by customers is the "fit and forget" approach [1]. Therefore, PVPF overcomes the lack of coordination between the load and its variant suppliers. The conditional hierarchical relations between heterogeneous generation sources and demand need an accurate forecasting model to prevent blackouts and system failures [1]. However, the intermittency and randomness of meteorological conditions pose great challenges to the accuracy of PV power production forecasts, especially during rainy, cloudy, or extreme weather conditions [3]. Consequently, PVPF remains in a theoretical exploration phase due to the unsatisfactory performance in particular case scenarios where the Root Mean Squared Error (RMSE) can exceed 50% [4]. Meanwhile, more sophisticated forecasting methods are highly needed to meet the technical requirements of actual PV plants.

Recently, PVPF has become an attractive research area for scientists and engineers. Meteorological Data-Driven Approach (MDDA) is one of the commonly used approaches in PVPF. This approach deals with the weather parameters having a direct impact on the PV power production. For a typical forecasting system, the database collection has a major impact on the forecasting quality [5], [6]. Moreover, the forecasting horizon significantly impacts the quality of the prediction [7], [8]. According to the cut-off time horizon, the data-driven methods are categorized into short, medium, and long-term predictions [2]. The Short-Term PVPF (STPF) ranges from minutes to hours, the medium horizon from hours to a few weeks, while the long-term prediction is implemented up to years ahead. Artificial intelligence methods were frequently adopted to cope with the stochastic interaction of PV systems with their external environment. A recent selection of these methods is listed in Table 1 [5], [9]–[20].

In [5], a Weighted Gaussian Process Regression (WGPR) approach has been proposed to alleviate the negative impact of outliers on the PVPF accuracy. In the latter approach, the samples with higher outlier have a lower weight. From the experimental results, the proposed method shows a slight improvement in terms of RMSE compared to standard Gaussian process regression (GPR). However, the joint distribution of input features needs to be calculated, which may lead to poor performance for multidimensional input data. A deep Residual Network (ResNet) and Dense convolutional Network (DenseNet) have been proposed in [20]. In these architectures, shortcut connections were utilized to skip one or more layers while preventing the learning degradation problem. It has been found that ResNet achieves higher performance than DenseNet, with a coverage error rate of less than 1%. Nonetheless, the major limitation of these deep networks is the long training time and the heavy computational requirements due to the high complexity with the increased network depth. In [18], a Support Vector Machines (SVM)-based Ant Colony Optimization (ACO) approach has been proposed for PVPF. Although the latter approach achieved excellent performance with a coefficient of determination (R^2) of 0.997, the scalability to large-scale problems has not been addressed. Besides, the comparison of the optimization method with other up-to-date metaheuristic methods is missed. Some scholars incorporate the persistence model, Auto-encoder, and Long Short Term Memory (LSTM) into the forecasting process of PVPF [15].

Nonetheless, the training time of LSTM is much longer than that of other algorithms. Reducing the training time under the premise of ensuring high accuracy is still a challenge worth studying. Paper [12] has adopted a solar-time-based analog ensemble for regional PVPF. In this specific design, six forecasting engines were fully utilized to generate the PV installed capacity output, leading to an impressive performance with a Mean Absolute Error (MAE) of 54.82MWh. From the employed forecasting engines, LSTM and Convolutional Neural Network (CNN) were incorporated, which could entail excessive parameters and

TABLE 1. Literature review of PV power forecasting methods.

| Method | Ref. | Benchmarks | Inputs | data set source |
|---------------------------|------|---|--|---|
| WGPR | [5] | NN ¹ , SVM, GPR | Solar radiation, time, temperature, wind direction, wind speed, precipitation, gust speed, humidity | PV station, Nanyang Technological University |
| ResNet, DenseNet | [20] | SVM, ES ² , MLP ³ , CNN, QR ⁴ | Historical PV power, Numerical weather prediction, diffuse and global horizontal radiation | Desert Knowledge Australia Solar Center (DKASC) |
| SVM | [18] | ACO-SVM | global and diffuse horizontal radiation, temperature, wind direction, wind direction, relative humidity, sampling time | Desert Knowledge Australia Solar Center (DKASC) |
| AE-LSTM-PM | [15] | PM ⁵ , FNN ⁶ , DNN ⁷ , LSTM, AE-LSTM | Irradiance, solar Zenith angle, Temperature, Minute index, cloud cover, day index | SOA university Bhubaneswar, Odisha, India |
| Ensemble method | [12] | SVM, Blending | Irradiance, temperature, solar hour, earth declination angle | PV plants from Southeastern Massachusetts region and ISO-New England |
| FSEL | [11] | LSTM, ARIMA ⁸ , VAR ⁹ | installation price, customer price, temperature, average sunshine, population density, treasury yield | local PV installation in Jeollanam-do and Gwangju |
| LSTM | [19] | LSTM, BPNN ¹⁰ , SVM, PM ¹¹ | Historical PV power | Desert Knowledge Australia Solar Center (DKASC) |
| Copula function-LSTM | [16] | LSTM, SVM | Temperature, wind speed, wind direction, cloud cover ratio, radiation, surface pressure, precipitation, humidity | PV power station data from China and the US. |
| CNN | [10] | SVM, RF ¹ , DT ¹² , MLP, LSTM | temperature, solar radiation, historical PV power | PV station in Taiwan |
| EMD-SCA-ELM ¹³ | [9] | SCA-ELM, EMD-ELM, ELM | Solar radiation, temperature | Electrical department block of SOA university Bhu-baneswar, Odisha, India |
| MARS | [13] | SVM, ANN, ARIMA | temperature, dew point, humidity, irradiance, wind direction, wind speed, | Data from Beijing, China |
| Deep ConvNets | [14] | ResNet, PM ⁵ | solar radiation, sky images | Folsom, CA, USA -NREL Golden, CO Data Set |
| LSH | [17] | BPNN ¹⁰ | atmospheric irradiance, direct irradiance, diffusion of energy, average wind speed, temperature, and humidity | PV power station in Ashland |

¹ Neural Networks ² Exponential Smoothing ³ Multi-layer Perceptron ⁴ Quantile Regression ⁵ Persistence Model ⁶ Feed Neural Network
⁷ Deep Neural Network ⁸ Auto-Regressive Integrated Moving Average ⁹ Variational Auto-Regressive ¹⁰ Back-propagation Neural Network
¹¹ Random Forest ¹² Decision Trees ¹³ Empirical Mode Decomposition-Sine Cosine Algorithm-ELM

high calculation costs. Unfortunately, the practical application of this design may require big data solutions to counter the additional computational complexity and meet the technical requirements. Besides, the hyperparameter optimization of these algorithms is overlooked. In [11], a novel long-term PVPF assembled by fusing a hybrid feature selection approach and stacking ensemble model is explored. However, the proposed method ignores the impact of historical trends on the future PV power output. Authors in literature [19] exploit the LSTM model based on time correlation modification for PVPF. The LSTM model exhibits the best performance among benchmarks in solving non-linear and time-varying problems.

Nevertheless, the comparison with the emerging powerful Deep Learning (DL) techniques such as Deep Belief Network (DBN) is not considered, making the competitiveness of the depicted approach with top-level DL models questionable. In a similar vein, paper [16] adopted LSTM architecture associated with copula function-based feature extraction to handle mid-to-long term PVPF. The proposed model efficiently extracts the relevant weather features with a lower

Mean Absolute Percentage Error (MAPE) of 5.95%, owing to the strong ability of LSTM in characterizing the dependence relationship of TS data. It was pointed out that meteorological feature extraction is highly recommended for capturing long-term dependencies. However, the model inputs of the used data set are limited to five, which poses a question of the usefulness of feature extraction in such a reduced dimension. The authors in [10] applied a CNN model for day-ahead PV power forecasting. However, the proposed model lacks enough capability to model the complex temporal characteristics of load series regarding the high error measurement leading to an unsatisfactory Root Mean Square Error (RMSE) of 163.15W.

The authors in [21] implemented an Extreme Learning Machine (ELM) for PVPF. The ELM architecture provides an effective selection of random nodes to determine the output weights to protect the system from slow gradient-based learning. To validate the competitiveness and feasibility of the proposed forecasting scheme, SVM and ANN were chosen as the reference models. From the error measures, it has been found that ELM provides more precise forecasting

results at the hourly forecasting horizon, with a MAPE = 3.56%, compared to SVM and ANN with MAPE = 4.56% and 5.41%, respectively. Besides, the training time of ELM was the lowest with 0.34 seconds compared to ANN (0.74 seconds) and SVR (2.91 seconds). However, the ELM model still has room to improve in terms of higher accuracy and efficiency since this model faces a complex issue in computation with a large number of hidden nodes. A recent work [22] deploys a new model named Expanded ELM (EELM) for PV power forecasting. The proposed EELM contributes to the original ELM by an automatic selection of the hidden layer number and random input weights. Although the proposed model outperforms ELM and Functional Link Neural Network (FNLNN), the higher extrapolation capabilities of EELM have been only demonstrated for a forecasting horizon of less than 1 hour. Authors in [23] proposed Kernel Extreme Learning Machine (KELM) model to predict daily global solar radiation. The supremacy of the proposed method is verified using different optimal penalty parameters and kernel width. However, the proposed KELM does not have a high generalization capability, and its performance is proved in limited problems. The authors in [24] implemented a stacked ELM (SELM) model for Time Series (TS) prediction. Despite the superior performance of the S-ELM, it has been conducted that S-ELM faces a heavy computational burden compared to the traditional ELM. To the best of the authors' knowledge, Mondrian Forest (MF) has never been implemented for PV and solar forecasting purposes yet [25].

In summary, based on the above-mentioned research works, the effectiveness of the PVPF can be further enhanced via performing some feasible scenarios. The main contributions of this paper rely on the following aspects:

- An effective ensemble learning-based stacked generalization approach and an intelligent optimizer are firstly proposed. The Enhanced DBN (EDBN) model creates effectively high-level abstractions by leveraging meta-data solutions.
- A potential application of the proposed model is appraised for PV power forecasting and assessed through a real-data set. The proposed method demonstrates a high extrapolation capability for STPF application. The assessment of the proposed model has been conducted using score metrics and a comparative study with multiple benchmark models.
- Tree-structured of Parzen Estimators (TPE) algorithm is employed for Hyperparameters (HP) tuning of machine learning and DL models. It is quite challenging to decide the initial HP values. The TPE optimizer has provided an efficient automatic HP selection based on simulation results.

The remaining parts of this paper are organized as follows. In section II, a comprehensive overview of the adopted methodologies and the proposed method is presented. Section III assesses the proposed method for daily

PV power generation using three real data sets. Finally, section IV concludes this study.

II. METHODOLOGIES AND PROPOSED APPROACH

This section briefly describes five ML models employed in this research work, including MF, ELM, KNN, DBN, and Extremely Randomized Trees (ET). The cited methods have been optimized based on TPE. Furthermore, the proposed approach is comprehensively described.

A. MONDRIAN FOREST

Recently, the MF model has been introduced as an enhanced version of Random Forest (RF). For the conventional RF, assuming $(\hat{f}_n)(n \geq 1)$ as a randomized estimate, $x \in [0, 1]^d$ denotes the query point, $\{\Pi^{(m)}, m \in \{1, \dots, M\}\}$ denotes the random partitions of $[0, 1]^d$, and $\hat{f}_n(x, \Pi^{(m)})$ presents the prediction output. The RF prediction is computed by [26]:

$$\hat{f}_n(x, \Pi_M) = \frac{1}{M} \sum_{m=1}^M \hat{f}_n(x, \Pi^{(m)}) \tag{1}$$

In order to avoid the inconsistency and complexity of RF, a Mondrian Process (MP) has been applied for MF in a scaled time domain. A family distribution $\{MF_t, t \in [0, \infty)\}$ makes hierarchical replications $\{MF_s, s \in [0, \infty)\}$ with an accuracy enhancement for each $s > t$. MF_s are shaped recursively according to an improved probability distribution and the hierarchical bayesian prior to the leaf parameters. For each distribution, the nodes are updated for each timestamp following the conditional mondrian algorithm. For the sake of conciseness, MF is explained from a mathematical perspective. From (1), the random partitions are accorded with timesteps λ from the MP distribution $MP(\lambda|[0, 1]^d)$. The MF parametric equation is defined as [26]:

$$\hat{f}_{\lambda,n,M}(x, \Pi_{\lambda,M}) = \frac{1}{M} \sum_{m=1}^M \hat{f}_{\lambda,n}^m(x, \Pi_{\lambda}^m) \tag{2}$$

Fig. 1 presents the global structure of the MF algorithm in which incremental learning is getting proceeded over time.

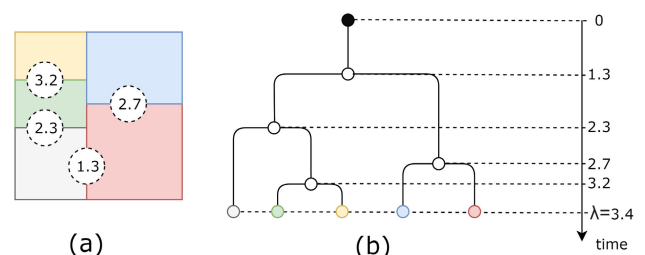


FIGURE 1. (a) MF partition, with (b) corresponding tree structure [26].

Regarding Fig. 1, each node is split for a specific λ timestep. MR has the propriety of no limitation for continuous learning. This MF ensures a falling back to the prior mean and variance for samples far away from the train sets.

B. EXTREME LEARNING MACHINE

The ELM is a learning model for the generalized single-hidden Layer feedforward neural Networks [27]. The ELM acquires a fast learning and high cost-effectiveness of computational complexity compared to the back-propagation algorithm and the Levenberg–Marquardt algorithm. Unlike slow gradient-based algorithms for neural networks, this algorithm’s hidden weights and bias parameters are randomly selected, and the output weights are analytically computed. The ELM learning process targets minimizing the training error in tandem with the smallest norm of output weights as following [27]:

$$\text{Minimize} : \|\beta\|_u^{\sigma_1} + \lambda \|H\beta - T_v\|_{\sigma_2} \quad (3)$$

where $\sigma_1 > 0, \sigma_2 > 0, u, v = 0, \frac{1}{2}, 1, 2, \dots, +\infty$. T denotes the training data-target matrix. H is the Hidden layer output matrix written as [28]:

$$H = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{pmatrix} = \begin{pmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{pmatrix} \quad (4)$$

The ELM mechanism consists of choosing the hidden perceptron and calculating the output weights of Single-hidden Layer Feed Forward Neural Networks (SLFNs). The ELM model with the activation function $g_i(\cdot)$ for the i^{th} hidden node and N hidden nodes is presented as [27]:

$$\sum_{i=1}^N \beta_i g_i(x) = \sum_{i=1}^N \beta_i g_i(w_i x_i + c_i), \quad N \in \mathbb{N} \quad (5)$$

where w_i is the weight between the hidden nodes and the input nodes, β_i denotes the output weights. c_i is the bias. This algorithm aims to get a faster training process with a minimum norm of output weights. Eq. 5 can be simplified as [27]:

$$H\beta = T \quad (6)$$

Since the hidden node’s parameters are tuned randomly, the output weight vector is calculated by a simple multiplication of the training data-target matrix, and the Moore-Penrose generalized inverse of H denoted as H^\dagger presented by [27]:

$$\beta_i = H^\dagger T = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{pmatrix}^\dagger \cdot \begin{pmatrix} t_1^T \\ \vdots \\ t_n^T \end{pmatrix} \quad (7)$$

The orthogonal projection method is successfully employed for the calculation of inverse matrix: $H^\dagger = (H^T H)^{-1} H^T$, if $H^T H$ is non-singular; or $H^\dagger = H^T (H H^T)^{-1}$, if $H H^T$ is non-singular. Following the ridge regression theory, it is recommended to add a positive value ($1/\lambda$) to the diagonal matrix $H H^T$. Thus, the corresponding output function $f(x)$ is given as follows [29]:

$$f(x) = h(x)\beta = h(x) \left(\frac{1}{\lambda} + H H^T \right)^{-1} T H^T \quad (8)$$

C. K-NEAREST NEIGHBORS

The traditional KNN algorithm is an instance-based learning supervised learning method to find out k training samples closest to the target object based on the available input patterns. The traditional KNN implementation is characterized by high simplicity and generalization potential for regression and classification problems. The Euclidean distance $D(X, Y)$ is often implemented through the standard KNN to measure the distance between two points $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ illustrated as [30]:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2 + \dots + (x_n - y_n)^2} \quad (9)$$

where n denotes the number of features. From the Euclidean distance calculation, only saving the input representations is required, and the prediction is achieved locally based on some near patterns. Despite being a mature algorithm with wide use, the traditional KNN has several shortcomings, including high sensitivity to the local structure of the data, large memory requirements, poor performance with unbalanced data, and ineffectiveness with large data sets. Therefore, several research papers have been focused on tackling these problems, including the Weighted KNN, Mixed KNN, and fuzzy KNN [31]–[33].

D. EXTREMELY RANDOMIZED TREES

The ET is an ensemble of untrimmed Decision Trees (DT) that promotes tree diversity based on classical top-down procedure [34]. The ET employs the whole training examples to build each tree in the tree ensemble [34]. The ET model separates the learning nodes by selecting action-points totally randomized for growing these trees. This process makes the ET model more computationally efficient compared to other tree-based ensemble methods [34]. This process reduces the variance of the model and prevents training overfitting [34]. Empirically, given a training data set, $X = \{x_1, x_2, \dots, x_N\}$, where the sample $x_i = f_1, f_2, \dots, f_D$ denotes a D -dimensional vector with f_j as the inputs and $j \in \{1, 2, \dots, D\}$, ET generates M individual DT. Here, S_p presents the subset of training data set X at child node p . At each node p , the ET model selects the best split based on S_p and a randomized inputs’ subgroup. Gini impurity is employed as a score function to select the best split rule. In each child node, the iterations are repeated until obtaining a minimum number of samples required to split, or when all the samples in subset S_p have an identical label. Each leaf node is represented by the label of the samples in subset S_p .

E. DEEP BELIEF NETWORK

The DBN is a variant of the Boltzmann Machine (BM) model formed by h^0 and L computational layers h^i ; ($i = 1, 2, \dots, L$) [35]. Every layer h^i is a Restricted Boltzmann Machine (RBM) RBM^i . The RBM algorithm contains a visible layer with visual neurons designed $v \in \{0, 1\}^{s_v}$ hidden layer containing hidden neural units named as $h \in \{0, 1\}^{s_h}$

with g_v and g_h present the number of visible and hidden units, respectively [35]. As depicted in Fig.2(a), the standard BM has a limited potential in digging out discriminant data of the input data layer by layer.

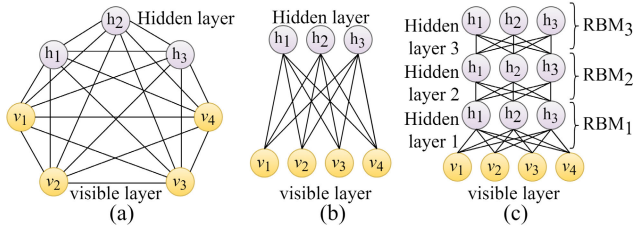


FIGURE 2. Simplified structure of (a) BM (b) RBM (c) DBN, respectively.

Thus, the RBM is employed as an advanced and more straightforward form of BM as there is no linkage between nodes in the same layer, while the nodes between the visible layer and the hidden layer are fully connected, as shown in Fig.2(b). Cascading multiple RBMs serve as the nuclear components of DBN, as illustrated in Fig.2(c).

The RBM belongs to energy-based models, and the energy function is defined by the visible layer $v = (v_i)_n$ and the hidden layer $h = (h_j)_m$ computed as [36]:

$$E(v, h) = - \sum_{i=1}^{g_v} \sum_{j=1}^{g_h} W_{ij} \frac{v_i}{\sigma_i} h_j - \sum_{i=1}^{g_v} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{g_h} b_j h_j \quad (10)$$

with W_{ij} and σ_i present the weight between visible and hidden units and the Gaussian standard deviation of the visible units, respectively. a_i and b_j present the bias terms. Gaussian unit-based RBM is simplified for practical use as [35]:

$$E(v, h) = - \sum_{i=1}^{g_v} \frac{(a_i - v_i)^2}{2\delta_i^2} + \sum_{j=1}^{g_h} \frac{(b_j - h_j)^2}{2\gamma_j^2} - \sum_{i=1}^{g_v} \sum_{j=1}^{g_h} \frac{v_i}{\delta_i} \frac{h_j}{\gamma_j} w_{ij} \quad (11)$$

where δ_i and γ_j and are the standard deviations of the Gaussian noise of the visible unit i and the hidden unit j , respectively. The joint distribution of the visible and hidden units is defined by [37]:

$$P(v, h) = \frac{1}{\aleph} \exp(-E(v, h)) \quad (12)$$

\aleph denotes a normalizing constant as the sum of $E(v, h)$ over all pairs of visible and hidden vectors. \aleph is calculated as [35]:

$$\aleph = \sum_{i=1}^{g_v} \sum_{j=1}^{g_h} \exp(-E(v, h)) \quad (13)$$

The marginal distributions of the visible and the hidden layer are defined as [36]:

$$P(v) = \frac{1}{\aleph} \sum_{i=1}^{g_v} \exp(-E(v, h)) \quad (14)$$

$$P(h) = \frac{1}{\aleph} \sum_{j=1}^{g_h} \exp(-E(v, h)) \quad (15)$$

At each RBM^i with $i = \{1, 2, \dots, L\}$, the conditional probability density function of the j -th neurons in the visible layer h^{i-1} and hidden layer h^i can be expressed as [37]:

$$P(h_j^i = 1 | h^{i-1}) = \sigma \left(\sum_k W_{kj}^i h_k^{i-1} + b_j^i \right) \quad (16)$$

$$P(h_j^{i-1} = 1 | h^i) = \sigma \left(\sum_k W_{jk}^i h_k^i + b_j^{i-1} \right) \quad (17)$$

With σ denotes the sigmoid function formulated as [38]:

$$\sigma(u) = \frac{1}{1 + e^{-u}} \quad (18)$$

The RBM training requires learning model parameters that boost the log-likelihood of the probabilistic distribution of the train visible set, following [37]:

$$\ln(P(D)) = \sum_{t=1}^T \ln P(v^t) = \sum_{t=1}^T \left[\ln \sum_{j=1}^{g_h} e^{-E(v^t, h)} - \ln \sum_{j=1}^{g_h} \sum_{i=1}^{g_v} e^{-E(v, h)} \right] \quad (19)$$

where T denotes the sample number for the train set D , in which $D = \{v^1, v^2, \dots, v^T\}_{t=\{1,2,\dots,T\}}$. Hence, v^t presents the t th training sample. For a representative sample v^t , Differentiating a log-likelihood of $P(v^t)$ with regard to Θ as follows [37]:

$$\frac{\partial \ln P(v^t)}{\partial \Theta} = - \sum_{j=1}^{g_h} P(h_j | v^t) \frac{\partial E(v^t, h)}{\partial \Theta} + \sum_{j=1}^{g_h} \sum_{i=1}^{g_v} P(v, h) \frac{\partial E(v, h)}{\partial \Theta} \quad (20)$$

where Θ is the network parameter set with $\Theta = \{w_{ij}, a_i, b_j\}$. $P(h_j | v^t)$ and $P(v, h)$ denote the expectations of the gradient function under distribution specified by the data and the model, respectively. Due to the high computational burden for solving Eq. 20, the Contrast Divergence (CD) algorithm is employed to measure the joint distribution of v and h [37].

F. TREE-STRUCTURED OF PARZEN ESTIMATORS

The HP optimization is a tedious task for ML methods [39]. To remedy this task, the TPE algorithm is presented as a sequential model-based optimization (SMBO) approach that effectively handles categorical (such as date or weather type)

and conditional parameters (such as learning algorithm and learning rate) [40]. The TPE algorithm transforms the configuration space into a non-parametric Parzen-window density estimation with a function evaluation $f(\theta)$ [41]. The configuration space can be modeled by uniform distribution, discrete uniform distribution, or logarithmic uniform distribution [42]. Hence, this variety of configurations contributes to the flexibility of the TPE compared to the standard BO [42]. For the iterative process, the TPE approximates $f(\theta)$ in a configuration space that is probabilistically supervised and limited according to the observation history [41]. Expected Improvement (EI) criterion is employed to locate the best HP θ^* from the search space [41]. The algorithm defines the probability distribution $p(\theta|y)$ by splitting the configuration space into good and bad samples as [41]:

$$p(\theta|y) = \begin{cases} Pr_{good}(\theta) & \text{if } y < y^* \\ Pr_{bad}(\theta) & \text{if } y \geq y^* \end{cases} \quad (21)$$

where $Pr_{good}(\theta)$ and $Pr_{bad}(\theta)$ are parzen estimators used to estimate the density formed by using the observations θ_i such that $f(\theta_i)$ are less than and greater than y^* , respectively. $y < y^*$ indicates that the value of the objective function is less than the threshold, and $y > y^*$ denotes that the value of the objective function is higher than the threshold. The optimal HP value θ is formulated as [41]:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{Pr_{bad}(\theta)}{Pr_{good}(\theta)} \quad (22)$$

To better describe the optimization procedure of TPE, Fig. 3 presents the flow chart of the iterative steps.

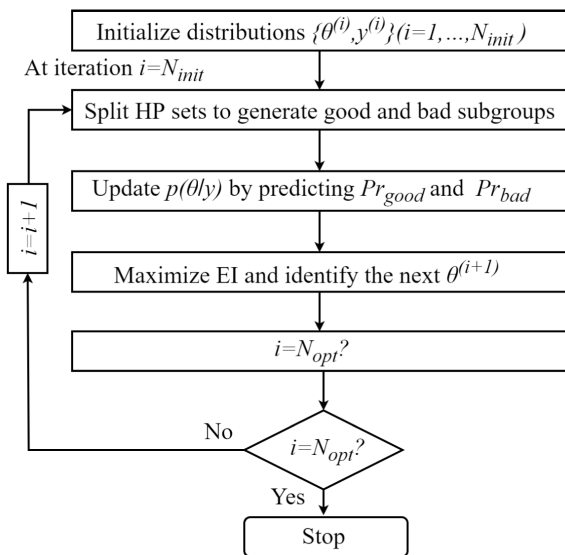


FIGURE 3. Flow chart of the TPE method.

G. PROPOSED APPROACH

To avoid the insufficiency of standalone prediction models, stacked generalization, so-called stacking, is proposed as a heterogeneous integration strategy that comprises multiple

base learners [43]. Thus, the stacking approach efficiently achieves an excellent non-linear fitting ability based on the diversity principle of base learners [44]. In this two-tier framework, the models in the first layer (or level-0), named the base estimator, employs the same target function to obtain the best hypothesis. Then, the second layer (or level-1) named the meta-estimator, manages the balance among the hypothesis obtained from level-0, and makes the final decision. The proposed model is a combination of MF, ELM, ET, KNN, and DBN networks to strengthen the learning effect of standalone models. The stacked scheme runs the individual base models (first-level learners) to train the meta-learner (second-level learner). The complete model architecture is shown in Fig. 4.

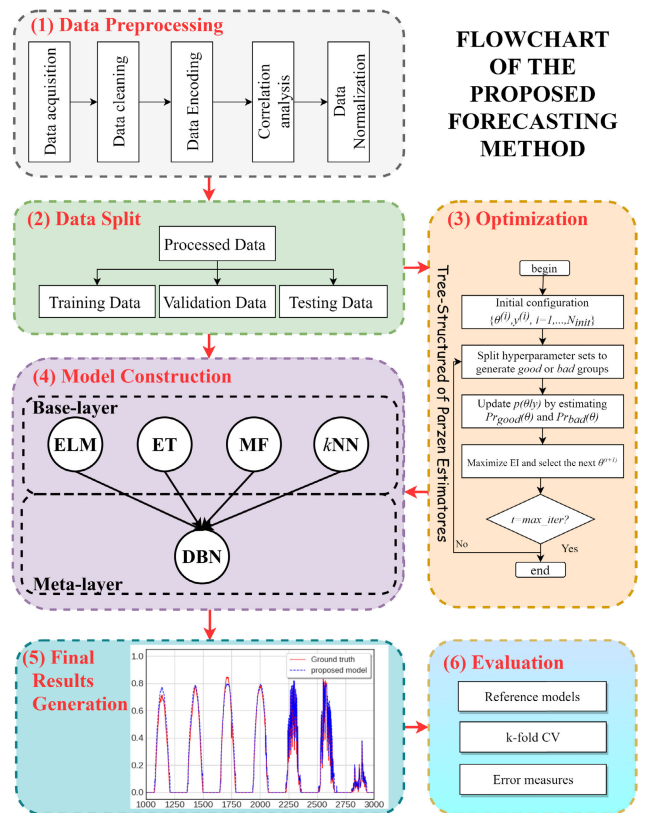


FIGURE 4. Block diagram of the EDBN method.

As seen in Fig. 4, six modules are consecutively computed, namely, data preprocessing, data split, optimization, model construction, results generation, and assessment modules. In the data preprocessing module, the data collected from the data set is cleaned from inconsistent measurements, missing data points, and outliers using Scikit-Learn python library to avoid false interpretations afterwards. Next, data encoding, correlation analysis, and data normalization take place to prepare the time-series data to be fed into the stacking fusion model. The normalized feature representations are integrated into the data split module, where the data is separated into training, validation, and testing. In addition, the predictors' HP were tuned using TPE optimizer in the third module to boost the overall model accuracy. In the model

construction module, MF, ELM, ET, and KNN are trained by applying fivefold cross-validation in the first level of the stacked generalization strategy to promote predictor diversity. A series of four meta-features of base learners were merged to construct new inputs. Then, the new inputs are fed to the DBN meta-learner model. The meta-learner exploits the new features to make the final forecasting. The prediction quality is assessed in the evaluation module. The evaluation module assesses the model’s applicability in real-world PV plants to justify its practical utility compared to benchmarks. The flow chart of the adopted framework is demonstrated in Fig. 5.

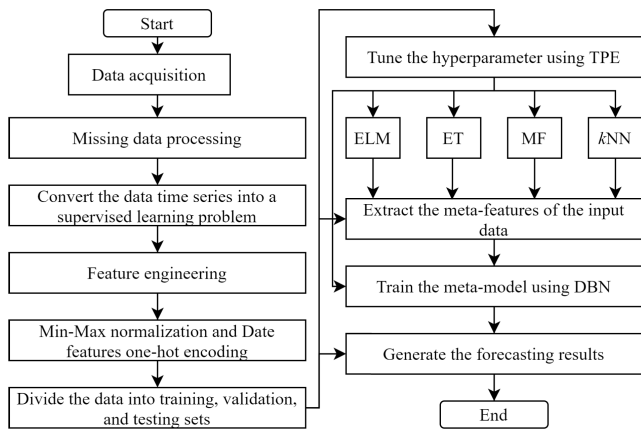


FIGURE 5. The PV power forecasting framework based on the proposed stacking ensemble.

III. EXPERIMENTAL RESULTS

The evaluation of the recommended model is performed based on numerical test results, using real PV plant data traces. This section investigates the data description and preprocessing stage. Next, the evaluation measures are presented. Finally, the simulation results are thoroughly discussed. The learning environment setup is run on a Google Colaboratory, a free cloud service supported by Google with Graphics Processing Unit (GPU) enabled. The experimental simulations tests were carried out on the computational environment described in Table 2.

TABLE 2. Experimental environment.

| Working environment | Proprieties |
|----------------------|--|
| Operating system | Windows 10 |
| Processor | Intel Core i7 9 th Generation |
| Graphics card | NVIDIA GeForce GTX 1650 |
| Programming language | Python 3.7 |
| Learning environment | Google Colaboratory |
| DL framework | Keras 2.4.3 |
| Platform | Tensorflow 2.3.0 |
| Supplementary tools | Scikit-Learn 0.23.2 |

A. DATA DESCRIPTION AND PREPROCESSING

The evaluation process is conducted using a real-world data set to prove the predictive performance of the proposed model. Desert Knowledge Alice Springs Center (DKASC)

in Central Australia data was collected and employed for the validation process [45]. The DKASC flagship facilities of Alice Springs contain 38 sites. The DKASC has a semi-arid climate BWh according to the Köppen climate taxonomy exploiting 9% of Northern Territory. This renewable energy plant is located in a town in the northern territory and is considered as one of the Australia’s top producers of solar energy [45]. Meteorology (global horizontal irradiance (W/m^2), diffuse horizontal irradiance (W/m^2), relative humidity (%), wind direction (\hat{A}°)), sampling time (min), temperature ($^\circ C$)), and historical power data of PV arrays (kW) from March 1, 2016, to December 1, 2019, in sampling intervals of five minutes has been employed for the numerical study. The meteorological factors and the one-year lagged PV power were associated as feature inputs to predict the PV power. The training, validation, and testing sets occupy 60%, 10%, and 30% of the total collected data, and the five-minutes ahead PV generation are the target.

For data cleaning purposes, removing invalid observation samples from the database such as Not A Number (NaN) or unreal values is highly required to not skew model training and gives a reliable sign of the data collected. The low timestep (5 minutes) improves the clearness of the data and the easiness of dirty sample detection. To ensure the consistency of features’ dimensions, the data samples have been linearly scaled into [0, 1] following the Min-Max normalization approach [46]. This feature scaling approach boosts the model convergence as [46]:

$$x_n = \frac{x_r - x_{min}}{x_{max} - x_{min}} \quad (23)$$

where x_n presents the normalized value. x_r , x_{max} , and x_{min} are the datum, maximum, minimum, and scaled datum, respectively.

B. EVALUATION CRITERIA

To perform a fair assessment of the investigated prediction models, the deterministic forecasting performance of EDBN is assessed by RMSE, MAE, MAPE, and R^2 error metrics to their common applicability in TS forecasting problems and wide acceptability within the PV forecasting research community [47]. Mathematically, the score metrics are formulated as [46]:

$$MAE = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \quad (24)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2} \quad (25)$$

$$MAPE = \frac{100(\%)}{n} \sum_{i=0}^{n-1} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (26)$$

$$R^2 = 100\% \left(1 - \frac{\sum_{i=0}^{n-1} (\hat{y}_i - y_i)^2}{\sum_{i=0}^{n-1} (\bar{y} - y_i)^2} \right), \quad \bar{y} = \frac{1}{n} \sum_{i=0}^{n-1} y_i \quad (27)$$

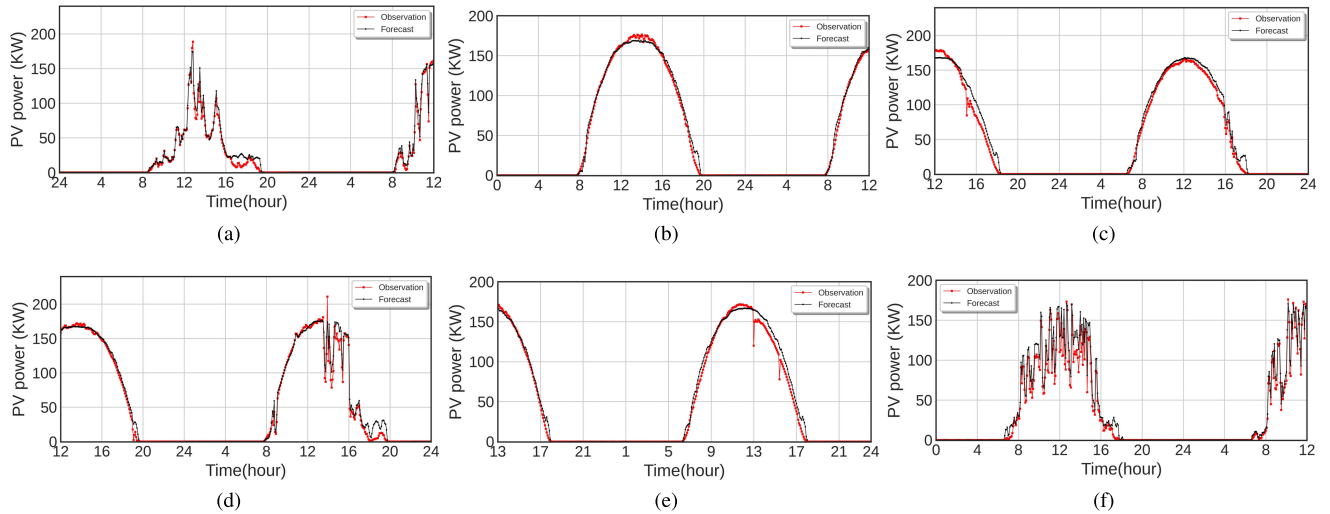


FIGURE 6. PV power of each month using EDBN model for 5-min daily forecasting on: (a) March 20, Rainy. (b) May 20, sunny. (c) June 20, sunny. (d) July 20, sunny to cloudy. (e) September 20, Sunny. (f) November 20, foggy to cloudy.

where n denotes the total number of samples. \hat{y}_i and y_i denote the i^{th} forecast and the actual value, respectively.

C. RESULTS AND DISCUSSION

In the beginning, predictive analysis on the improvement of the proposed model in the PVPF due to the inclusion of the stacking ensemble model was evaluated and compared to its original counterpart models. These models include ELM, KNN, ET, DBN, MF, and EDBN. In this stage, four types of fluctuations were considered for the assessment procedure, namely, the sunny, partially cloudy, foggy, and rainy weather types. Then, the evaluation procedure is extended to compare the proposed model with the popular single DL models. These models include Multi-Layer Perceptron (MLP), Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (BiLSTM), and LSTM. Finally, the proposed model is compared to a list of hybrid methodologies presented in the recent literature [47]–[54]. Towards a reliable evaluation of the proposed model for 5-min ahead daily forecasting, score errors are simultaneously computed with identical conditions for the proposed framework. The simulated models pass by an HP optimization using TPE method. For the TPE implementation, the Γ_{TPE} is used to classify the explored positions into good and bad, while the RPP presents the probability for the TPE to jump to a random position in an iteration step. The TPE optimal configuration adopted in this study lies in thirty iterations, Γ_{TPE} of 0.5, and a Random Position Probability (RPP) of 0.03. To enable rigorous assessment of the ensemble model with respect to individual models, the EDBN model adopts the optimal HP settings of its base and meta-learners to attribute the same level of tuning for all the simulated models. Subsequently, some popular neural network benchmarks, such as MLP, LSTM, BiLSTM, and GRU, are compared. In these benchmarking DL methods, the number of units, the hidden layer size, the activation function, and the optimizer function need to be optimized. It is worth noting that the adopted TPE model generates the optimal

TABLE 3. Configurations of HP settings for benchmarks.

| Base models | HP settings |
|-------------|--|
| ELM | The number of hidden units is 60; The activation function is Sigmoid; The Radial basis function width=1. |
| KNN | The leaf size is 400; The nearest neighbor number is 6. |
| ET | The maximum depth is 22; The number of estimators is 80. |
| MF | The maximum depth is 20; The minimum samples split is 12; The number of estimators is 80. |
| DBN | The number of RBMs is 1500; The batch size is 30; The learning rate is 0.1. |
| LSTM | The number of LSTM cells for the first layer is 64; The number of LSTM cells for the second layer is 36; The optimizer function is Adam. |
| BiLSTM | The number of Bidirectional LSTM cells for the first layer is 64; The number of LSTM cells for the second layer is 36; The optimizer function is Adam. |
| GRU | The number of Bidirectional GRU cells for the first layer is 64; The number of GRU cells for the second layer is 36; The optimizer function is Adam. |
| MLP | The hidden layer size is 64; The activation function is Relu, The learning rate unit is 0.1. |

HP for the reference models used in this study. The specific HP settings of the developed model and benchmarks are found in Table 3.

The proposed model is tested for the different seasons of the year to evaluate the prediction performance of the obtained architecture over the possible meteorological conditions. The forecast results of the EDBN model are vividly visualized in Fig. 6. It is worth noting that the forecasting outputs are converted to the original range using the Max-Min denormalization approach.

As it can be seen from the simulation results presented in Fig. 6, the proposed PVPF method is well-performing. From the first look at the forecast curves, the prediction curves

show excellent agreement with the measured PV power taken into account the number of forecasted points close to the real values. The high accuracy confirms that the proposed model is perfectly designed for PVPF. This assessment procedure investigates the impact of the rapid changes of meteorological factors on the forecasts' quality. To have a more precise judgment, the quantification of the error values for the EDBN model and the counterpart models (ELM, ET, KNN, and DBN models) are summarized in Table 4, where boldface error measures represent the optimal values in each situation.

TABLE 4. Performance comparison of the simulated models for daily PVPF.

| Condition | Model | RMSE (kW) ±SD | MAE (kW)±SD |
|------------------|-------|-------------------|-------------------|
| Sunny | MF | 37.66 | 32.26 |
| | ELM | 45.43 | 41.32 |
| | ET | 17.07 | 14.22 |
| | KNN | 14.09 | 11.01 |
| | DBN | 10.79 | 8.08 |
| | EDBN | 4.76 | 3.67 |
| Partially cloudy | MF | 24.97 | 21.82 |
| | ELM | 15.22 | 13.76 |
| | ET | 16.15 | 12.62 |
| | KNN | 19.54 | 11.53 |
| | DBN | 8.41 | 6.54 |
| | EDBN | 5.15 | 3.38 |
| Cloudy/foggy | MF | 32.49 | 26.05 |
| | ELM | 24.26 | 20.38 |
| | ET | 17.06 | 13.92 |
| | KNN | 19.26 | 13.50 |
| | DBN | 8.06 | 6.43 |
| | EDBN | 4.59 | 3.21 |
| Rainy | MF | 14.37 | 13.74 |
| | ELM | 16.81 | 15.38 |
| | ET | 7.51 | 6.54 |
| | KNN | 11.62 | 5.33 |
| | DBN | 7.04 | 8.95 |
| | EDBN | 1.03 | 0.53 |
| Overall | MF | 27.37 ±8.76 | 23.47 ±6.73 |
| | ELM | 25.43 ±12.04 | 22.71 ±11.02 |
| | ET | 14.45 ±4.02 | 11.82 ±3.11 |
| | KNN | 16.13 ±3.39 | 10.34 ±3.04 |
| | DBN | 8.57 ±1.37 | 7.50 ±1.06 |
| | EDBN | 3.88 ±1.66 | 2.70 ±1.26 |

Note: SD stands for standard deviation.

According to Table 4, the EDBN model has an excellent performance for an overall $MAE = 2.70kW \pm 1.26$, demonstrating the high effectiveness of the improvement strategy. Regarding error metrics, ET, KNN, and DBN exhibit an $RMSE = 8.57kW$, $RMSE = 16.13kW$, $RMSE = 14.45kW$ while the achieved for EDBN is $RMSE = 3.88kW$. The EDBN can always obtain better testing accuracy than standalone models. The proposed framework succeeds in performing the best accuracy. The high performance of the proposed method is explained by combining the benefits of the model components. To further demonstrate the performance of the proposed model, the results are compared with those of MLP, LSTM, BiLSTM, and GRU. To visually display the predictive ability of the proposed model and other DL benchmarks, Fig. 7 further plots the prediction outputs of

the simulated models for typical days in four seasons of the year 2018.

According to Fig. 7, the EDBN model provides high effectiveness in tracking the daily PV power and outperforms all the benchmark models for different seasons of the year. During sunny and partially cloudy days in summer and spring, the EDBN and DBN perform best than other models. It can be noticed that EDBN provides more stable results with fewer fluctuations compared to DBN. On foggy and rainy days, the kNN, DBN, ET, and DBN generate satisfying forecasts despite the high volatility of meteorological conditions. It is worth mentioning that MF and ELM models provide the worst performance in all weather types, possessing high errors compared to DBN, and ET models. Although the predictive performance of the EDBN model is clearly demonstrated compared to comparison models, it could be remarked that the proposed method slightly outperforms ET and KNN models. To establish the robustness of the proposed framework compared to reference models, the scatter plot of the simulated models is illustrated in Fig. 8.

As seen from Fig. 8(a), the forecasted PV power pattern coincides mostly with the real PV power values. It is clear that the EDBN outputs rise linearly with the actual PV measurements. It should be noted that the presented methodology shows a strong correlation between the forecasted and real PV power, demonstrating the best performance compared to counterparts models. More importantly, compared to the original DBN in Fig. 8(c), the predicted points of the DBN are very dispersed, reflecting a weaker correlation with the actual outputs. It should be mentioned that the more the dispersion is low, the more the accuracy is high, and vice versa. From Fig. 8(d) and Fig. 8(g), it can be derived that the weak learners of the stacking approach, namely ELM and KNN, are badly correlated to the PV outputs. The linear relationship deduced from Fig. 8(e) between the ET, and actual values show a good correlation compared to ELM and KNN models. In Figs. 8(b), 8(f), 8(h), 8(i) presenting the MLP, GRU, LSTM, and BiLSTM outputs, respectively, it can be remarked that these DL models achieve a satisfactory performance for a low PV generation lower than 150kW. In other words, It can be found that the forecasting accuracy of these models is very close to the EDBN model. But, the forecast accuracy of the counterpart models degrades for PV power outputs higher than 150kW.

Interestingly, it is remarked that the correlation among the PV generation measurements decreases as the produced PV energy is increased. Fig. 9 shows the error distribution denoted as the difference between the predicted and actual PV generation output. Empirically, the error $error(i) = y(i) - \hat{y}(i)$ is the test error for the i^{th} sample, $y(i)$ is the target value and $\hat{y}(i)$ is the output for time step i . The error distributions plot reveals a direct interpretation of forecast errors for all weather conditions. Regarding Fig. 9(a), the error distributions of the proposed model are mostly concentrated on the zero-axis, which reveals that the EDBN can produce more accurate results. Compared to the LSTM model shown in Fig. 9(h),

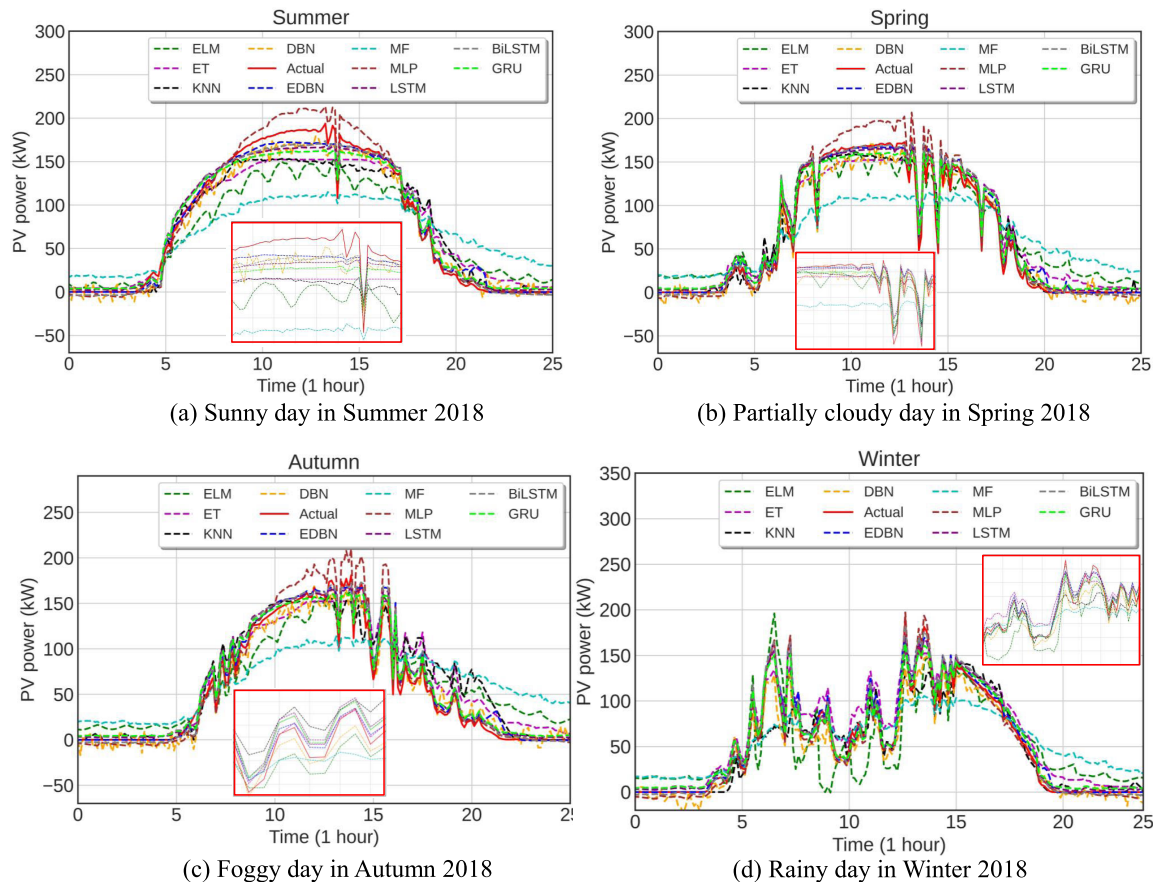


FIGURE 7. Performance comparison of considered models for daily PVPF in (a) Sunny day in Summer 2018 (b) Partially cloudy day in Spring 2018, (c) Foggy day in Autumn 2018 (d) Rainy day in Winter 2018. For visualization purposes, the red box presents a zoomed plot for better visibility of the forecast results.

the error values range from -20kW to 30kW , while the LSTM errors range from -40kW to 30kW . Compared with the prediction outputs of DL models, it can be conducted that DL models can provide better performance than machine learning prediction models. Fig. 10 shows the coefficient of determination for the competing models using the 5-minutes daily generated PV outputs in different seasons of the year. Ideally, an R^2 ratio closer to 1 is better.

As seen from Fig. 10, the box plots illustrate the noticeable difference between the prediction engines' performance. It can be noticed from Fig. 10 that the EDBN model predicts the target accurately compared to ET and KNN models despite the non-symmetry behavior of PV power over the seasons of the year. The size of the box in the plot shows that the coefficients of determination are mostly concentrated at a high range. The EDBN model has the smallest box for different weather types, reflecting the model robustness compared to other models. To better demonstrate the feasibility of the proposed model, Fig. 11 presents the spider chart of MAPE and R^2 of the proposed algorithm and simulated benchmarks.

As presented in 11(a), the MAPE error of the proposed EDBN is 2.30% which is lower than the reference models. Furthermore, the R^2 value of the EDBN algorithm yields

a 97%, which is higher than the other benchmarks. It is worth mentioning the DL models possess a slight lower error values than the proposed model, which demonstrates the high competitiveness of the proposed model. In this work, daily comparisons based on scale-independent forecast error metric results are conducted in this study with the recently proposed models. Comparing with other methods presented in literature using other PV databases, the proposed model performance is analysed aside with CNN-LSTM (CLSTM) [48], LSTM-based Attention Mechanism (LSTM-AM) [49], PVPF Network (PV-Net) [50], Multi-Channel CNN (MC-CNN) [51], Radial Basis Function Neural Network (RBFNN) [52], Wavelet Packet Decomposition-LSTM (WPD-LSTM) [53], LSTM [47], GRU [47], Improved Moth-Flame Optimization algorithm-SVM (IMFO-SV) [54], and Particle Swarm Optimization-SVM (PSO-SVM) [54]. The models' results for PVPF are listed in Table 5.

As per Table 5, the EDBN model yields accurate forecasting results, exhibiting a MAPE error of 2.30%. The proposed model produces better results than high-performing benchmarks such as WPD-LSTM, IMFO-SVM, and PSO-SVM with 2.40%, 3.92%, and 2.85%, respectively. Although hybrid models generally yield better performance, the CLSTM, LSTM-AM, and MC-CNN generate low MAPE

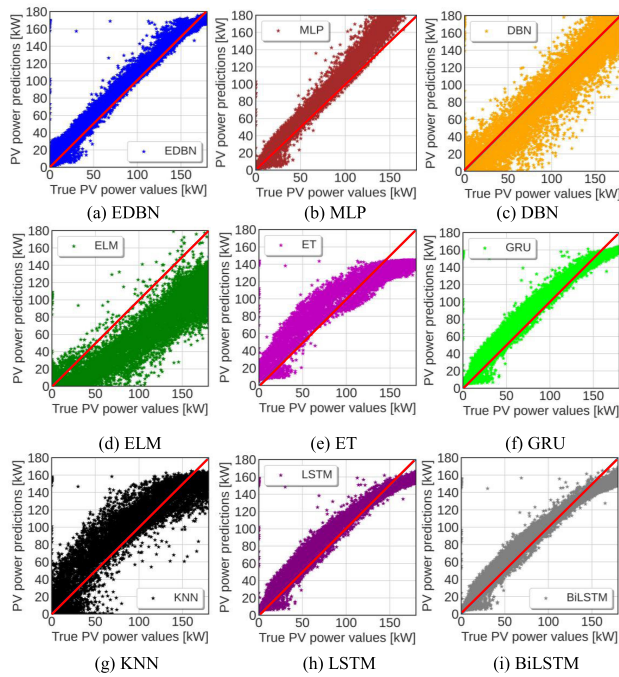


FIGURE 8. Regression plot of predicted power and measurements using nine models: (a) EDBN, (b) MLP, (c) DBN, (d) ELM, (e) ET, (f) GRU, (g) KNN, (h) LSTM, and (i) BiLSTM.

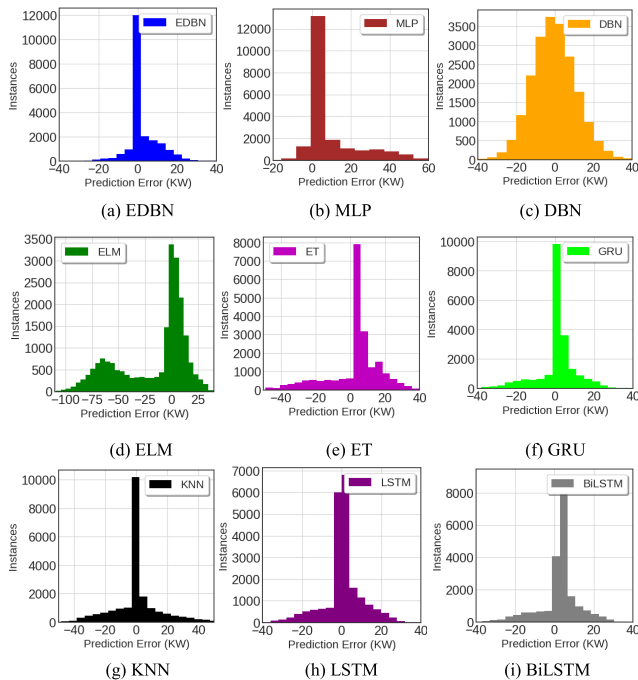


FIGURE 9. Error distribution plot of predicted power and measurements using nine models: (a) EDBN, (b) MLP, (c) DBN, (d) ELM, (e) ET, (f) GRU, (g) KNN, (h) LSTM, and (i) BiLSTM.

errors of 7.53%, 7.10%, and 8.63%, respectively. The RBFNN, LSTM, and GRU yield an MPAE of 3.71%, 3.61%, and 3.42%, respectively. It is worth mentioning that the EDBN model produced slightly better results than WPD-LSTM and PSO-SVM models (the best algorithms

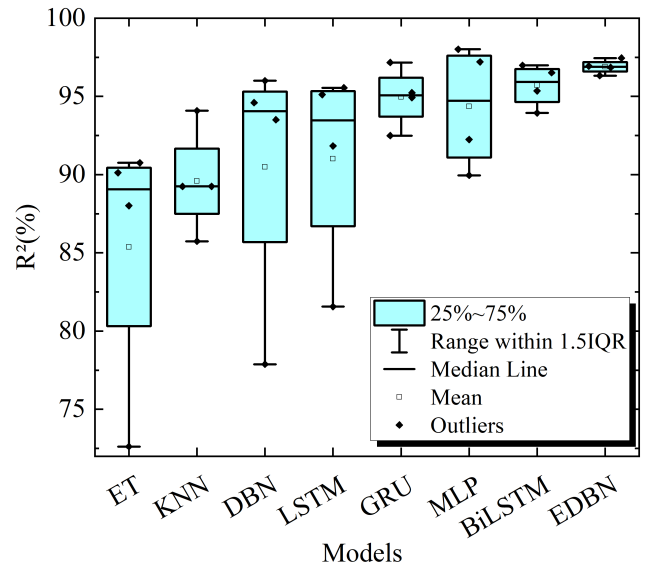


FIGURE 10. Box plot exemplifying the R^2 veracity of the proposed model over different seasons of the year.

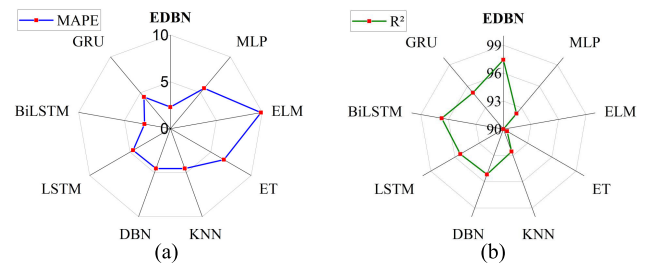


FIGURE 11. Comparison results obtained from the simulated models with (a) MAPE and (b) R^2 .

TABLE 5. Model comparison of daily PVPF results.

| Model | MAPE (%) |
|---------------|----------|
| MC-CNN [51] | 8.63 |
| CLSTM [48] | 7.53 |
| LSTM-AM [49] | 7.10 |
| PV-Net [50] | 6.61 |
| IMFO-SVM [54] | 3.92 |
| RBFNN [52] | 3.71 |
| LSTM [47] | 3.61 |
| GRU [47] | 3.42 |
| PSO-SVM [54] | 2.85 |
| WPD-LSTM [53] | 2.40 |
| EDBN | 2.30 |

after the proposed one) with 2.40% and 2.85%. To assess the significance of the differences among the results listed in Table 5, Fig. 12 presents the MAPE results of the obtained EDBN and recent works.

Fig. 12 reveals that the proposed model is highly competitive compared to the results reported in the literature. This is

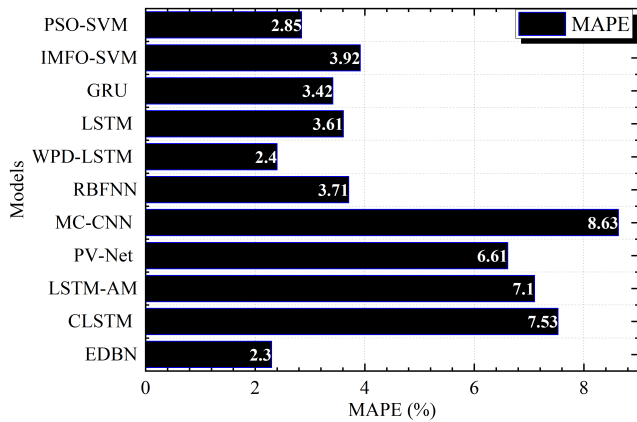


FIGURE 12. Performance (MAPE) for the PVPF methods.

TABLE 6. Testing time for PVPF models.

| Proposed model versus baseline methods | | | | | |
|--|------|------|------|--------|------|
| Model | MF | ELM | ET | KNN | DBN |
| Time (s) | 3.4 | 0.8 | 0.77 | 3.06 | 1.13 |
| Proposed model versus benchmarks | | | | | |
| Model | MLP | GRU | LSTM | BiLSTM | EDBN |
| Time (s) | 0.42 | 1.67 | 1.76 | 2.86 | 1.67 |

deduced since the overall MAPE obtained from the adopted methodology is comparatively lower than those obtained in [47]–[54]. Therefore, the obtained EDBN model is found perfectly tailored for daily PVPF. The time complexity of the ML algorithm is an essential ingredient taken into consideration when devising novel models for real-world implementation. A proposed model requires having a running time less than the response time needed for the underlying problem. In this work, the computational complexity of the testing model is resumed in Table 6.

From Table 6, it is observable that the proposed technique generates the testing prediction output with an acceptable computational testing time-cost (1.67 seconds). The PV forecasts generation in a few seconds reflects the fast convergence of the stacking ensemble and its adaptability in practical applications. Empirically, the MLP, ET, ELM, and DBN present the fastest algorithms, followed by the EDBN and GRU models with 0.42, 0.77, 0.8, and 1.13 seconds, respectively. It is conducted that the relative complexity and the superior accuracy of the EDBN model did compromise its low testing time for generating the final outputs. It must be pointed out future work of this study will give a particular focus on the emerging materialistic optimization algorithms that can further boost the model performance, such as grey wolf optimizer and whale optimization algorithms [38]. In summary, the proposed EDBN model is able to model the periodical data in complex non-linear relationships and irregular dependencies through the stacking mechanism in a timely manner, demonstrating its high applicability in real-world PV plants.

IV. CONCLUSION

PV power forecasting (PVPF) is essential and fundamental in decision-making processes for smart grid systems. This paper proposes a novel PVPF framework named Enhanced Deep Belief Network (EDBN). The proposed EDBN model incorporates Extreme Learning Machine-Mondrian Forest-K-Nearest Neighbors, Extremely Randomized Trees, and deep belief network to cope with the heteroskedasticity of PV power. This model employs the correlation patterns between the meteorological data and the meta-learning ensemble to provide a precise daily estimation of the PV power. Tree-structured of parzen estimators algorithm is employed to optimize the model performance through an effective hyperparameter tuning. As demonstrated in the case study, the error measurements of the developed system are the lowest among all involved methods, with an RMSE and R^2 of 3.88kW and 97%, respectively. The mean absolute error has been reduced from 7.5kW to 2.70kW when compared with the original DBN model. In sum, the simulation results demonstrate that the EDBN provides excellent performance for non-linear PV power output forecasting and high generalization potential compared to single reference models.

In this study, the most suitable base learners were selected based on the repetitive experiments, which can present a cumbersome task for complex time series forecasting problems. An automatic knowledge generation method will be considered to define the best model stack for various prediction problems in future works. Furthermore, a missing-data tolerant strategy is highly required to guarantee the proper operation of the proposed method in real-world environments, which will be the topic of a future study.

REFERENCES

- [1] R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif, "A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization," *Renew. Sustain. Energy Rev.*, vol. 124, May 2020, Art. no. 109792.
- [2] M. Massaoudi, S. S. Refaat, H. Abu-Rub, I. Chihi, and F. S. Wesleti, "A hybrid Bayesian ridge regression-CWT-catboost model for PV power forecasting," in *Proc. IEEE Kansas Power Energy Conf. (KPEC)*, Jul. 2020, pp. 1–5.
- [3] M. Massaoudi, I. Chihi, H. Abu-Rub, S. S. Refaat, and F. S. Oueslati, "Convergence of photovoltaic power forecasting and deep learning: State-of-art review," *IEEE Access*, vol. 9, pp. 136593–136615, 2021.
- [4] G. Li, S. Xie, B. Wang, J. Xin, Y. Li, and S. Du, "Photovoltaic power forecasting with a hybrid deep learning approach," *IEEE Access*, vol. 8, pp. 175871–175880, 2020.
- [5] H. Sheng, J. Xiao, Y. Cheng, Q. Ni, and S. Wang, "Short-term solar power forecasting based on weighted Gaussian process regression," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 300–308, Jan. 2018.
- [6] R. Corizzo, M. Ceci, H. Fanaee-T, and J. Gama, "Multi-aspect renewable energy forecasting," *Inf. Sci.*, vol. 546, pp. 701–722, Feb. 2021.
- [7] Y. Zhao, L. Ye, P. Pinson, Y. Tang, and P. Lu, "Correlation-constrained and sparsity-controlled vector autoregressive model for spatio-temporal wind power forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5029–5040, Sep. 2018.
- [8] J. W. Messner and P. Pinson, "Online adaptive lasso estimation in vector autoregressive models for high dimensional wind power forecasting," *Int. J. Forecasting*, vol. 35, no. 4, pp. 1485–1498, Oct. 2019.
- [9] M. K. Behera and N. Nayak, "A comparative study on short-term PV power forecasting using decomposition based optimized extreme learning machine algorithm," *Eng. Sci. Technol., Int. J.*, vol. 23, no. 1, pp. 156–167, Feb. 2020.

- [10] C.-J. Huang and P.-H. Kuo, "Multiple-input deep convolutional neural network model for short-term photovoltaic power forecasting," *IEEE Access*, vol. 7, pp. 74822–74834, 2019.
- [11] H. Eom, Y. Son, and S. Choi, "Feature-selective ensemble learning-based long-term regional PV generation forecasting," *IEEE Access*, vol. 8, pp. 54620–54630, 2020.
- [12] X. Zhang, Y. Li, S. Lu, H. F. Hamann, B.-M. Hodge, and B. Lehman, "A solar time based analog ensemble method for regional solar power forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 268–279, Jan. 2018.
- [13] X. Zhang, F. Fang, and J. Liu, "Weather-classification-MARS-based photovoltaic power forecasting for energy imbalance market," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8692–8702, Nov. 2019.
- [14] H. Wen, Y. Du, X. Chen, E. Lim, H. Wen, L. Jiang, and W. Xiang, "Deep learning based multistep solar forecasting for PV ramp-rate control using sky images," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1397–1406, Feb. 2021.
- [15] Y. Zhang, C. Qin, A. K. Srivastava, C. Jin, and R. Sharma, "Data-driven day-ahead PV estimation using autoencoder-LSTM and persistence model," *IEEE Trans. Ind. Appl.*, vol. 56, no. 6, pp. 7185–7192, Nov/Dec. 2020.
- [16] S. Han, Y.-H. Qiao, J. Yan, Y.-Q. Liu, L. Li, and Z. Wang, "Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network," *Appl. Energy*, vol. 239, pp. 181–191, Apr. 2019.
- [17] M. Yang and X. Huang, "Ultra-short-term prediction of photovoltaic power based on periodic extraction of PV energy and LSH algorithm," *IEEE Access*, vol. 6, pp. 51200–51205, 2018.
- [18] M. Pan, C. Li, R. Gao, Y. Huang, H. You, T. Gu, and F. Qin, "Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization," *J. Cleaner Prod.*, vol. 277, Dec. 2020, Art. no. 123948.
- [19] F. Wang, Z. Xuan, Z. Zhen, K. Li, T. Wang, and M. Shi, "A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework," *Energy Convers. Manage.*, vol. 212, May 2020, Art. no. 112766.
- [20] H. Zang, L. Cheng, T. Ding, K. W. Cheung, Z. Wei, and G. Sun, "Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning," *Int. J. Electr. Power Energy Syst.*, vol. 118, Jun. 2020, Art. no. 105790.
- [21] M. Hossain, S. Mekhilef, M. Danesh, L. Olatomiwa, and S. Shamshirband, "Application of extreme learning machine for short term output power forecasting of three grid-connected PV systems," *J. Cleaner Prod.*, vol. 167, pp. 395–405, Nov. 2017.
- [22] S. Mishra, L. Tripathy, P. Satapathy, P. K. Dash, and N. Sahani, "An efficient machine learning approach for accurate short term solar power prediction," 2020, *arXiv:2003.12088*.
- [23] S. Shamshirband, K. Mohammadi, H.-L. Chen, G. N. Samy, D. Petkovi, and C. Ma, "Daily global solar radiation prediction from air temperatures using kernel extreme learning machine: A case study for Iran," *J. Atmos. Sol.-Terr. Phys.*, vol. 134, pp. 109–117, Nov. 2015.
- [24] X. Luo, J. Sun, L. Wang, W. Wang, W. Zhao, J. Wu, J.-H. Wang, and Z. Zhang, "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4963–4971, Nov. 2018.
- [25] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3140–3148.
- [26] J. Mourrada, S. Gaïffas, and E. Scornet, "Minimax optimal rates for Mondrian trees and forests," *Ann. Statist.*, vol. 48, no. 4, pp. 2253–2276, Aug. 2020.
- [27] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
- [28] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2011.
- [29] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, Jun. 2011.
- [30] G. Batista and D. F. Silva, "How k-nearest neighbor parameters affect its performance," in *Proc. Argentine Symp. Artif. Intell.*, 2009, pp. 1–12.
- [31] G.-F. Fan, Y.-H. Guo, J.-M. Zheng, and W.-C. Hong, "Application of the weighted K-nearest neighbor algorithm for short-term load forecasting," *Energies*, vol. 12, no. 5, p. 916, Mar. 2019.
- [32] F. Harrou, B. Taghezouit, and Y. Sun, "Improved kNN-based monitoring schemes for detecting faults in PV systems," *IEEE J. Photovolt.*, vol. 9, no. 3, pp. 811–821, Mar. 2019.
- [33] J. Maïllo, S. Garcia, J. Luengo, F. Herrera, and I. Triguero, "Fast and scalable approaches to accelerate the fuzzy k-nearest neighbors classifier for big data," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 5, pp. 874–886, May 2020.
- [34] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.
- [35] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi, and F. S. Oueslati, "Deep learning in smart grid technology: A review of recent advancements and future prospects," *IEEE Access*, vol. 9, pp. 54558–54578, 2021.
- [36] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [37] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Toronto, ON, Canada: Univ. Toronto, 2010, pp. 599–619.
- [38] M. Massaoudi, S. S. Refaat, and H. Abu-Rub, "On the pivotal role of artificial intelligence toward the evolution of smart grids: A review of advanced methodologies and applications," *Smart Grid Enabling Technol.*, vol. 16, pp. 359–420, Aug. 2021.
- [39] M. Massaoudi, S. S. Refaat, H. Abu-Rub, I. Chihi, and F. S. Oueslati, "PLS-CNN-BiLSTM: An end-to-end algorithm-based Savitzky-Golay smoothing and evolution strategy for load forecasting," *Energies*, vol. 13, no. 20, p. 5464, 2020.
- [40] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 1–9.
- [41] F. He, J. Zhou, L. Mo, K. Feng, G. Liu, and Z. He, "Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest," *Appl. Energy*, vol. 262, Mar. 2020, Art. no. 114396.
- [42] H. Dong, D. He, and F. Wang, "SMOTE-XGBoost using tree Parzen estimator optimization for copper flotation method classification," *Powder Technol.*, vol. 375, pp. 174–181, Sep. 2020.
- [43] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, "A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting," *Energy*, vol. 214, Jan. 2021, Art. no. 118874.
- [44] M. H. D. M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105837.
- [45] *DKA Solar Centre*. Accessed: Sep. 23, 2019. [Online]. Available: <http://dkasolarcentre.com>
- [46] M. Massaoudi, I. Chihi, L. Sidhom, M. Trabelsi, S. S. Refaat, H. Abu-Rub, and F. S. Oueslati, "An effective hybrid NARX-LSTM model for point and interval PV power forecasting," *IEEE Access*, vol. 9, pp. 36571–36588, 2021.
- [47] A. A. du Plessis, J. M. Strauss, and A. J. Rix, "Short-term solar power forecasting: Investigating the ability of deep learning models to capture low-level utility-scale photovoltaic system behaviour," *Appl. Energy*, vol. 285, Mar. 2021, Art. no. 116395.
- [48] K. Wang, X. Qi, and H. Liu, "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113315.
- [49] A. Heidari and D. Khovaly, "Short-term energy use prediction of solar-assisted water heating system: Application case of combined attention-based LSTM and time-series decomposition," *Sol. Energy*, vol. 207, pp. 626–639, Sep. 2020.
- [50] M. Abdel-Basset, H. Hawash, R. K. Chakraborty, and M. Ryan, "PV-Net: An innovative deep learning approach for efficient forecasting of short-term photovoltaic energy production," *J. Cleaner Prod.*, vol. 303, Jun. 2021, Art. no. 127037.
- [51] J. Heo, K. Song, S. Han, and D.-E. Lee, "Multi-channel convolutional neural network for integration of meteorological and geographical features in solar power forecasting," *Appl. Energy*, vol. 295, Aug. 2021, Art. no. 117083.
- [52] H. J. Lu and G. W. Chang, "A hybrid approach for day-ahead forecast of PV power generation," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 634–638, 2018.

- [53] P. Li, K. Zhou, X. Lu, and S. Yang, "A hybrid deep learning model for short-term PV power forecasting," *Appl. Energy*, vol. 259, Feb. 2020, Art. no. 114216.
- [54] G.-Q. Lin, L.-L. Li, M.-L. Tseng, H.-M. Liu, D.-D. Yuan, and R. R. Tan, "An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation," *J. Cleaner Prod.*, vol. 253, Apr. 2020, Art. no. 119966.



MOHAMED MASSAOUDI (Member, IEEE) received the bachelor's degree from IPIEN, University of Carthage, in 2015, and the B.E. degree in energy engineering from ENIM, University of Monastir, in 2018. He is currently pursuing the Ph.D. degree with the National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Tunisia.

He has five years of hands-on experience in applying deep learning and machine learning strategies to tackle real-world problems. He is the first author of more than 20 peer-reviewed journals and conference publications and one book chapter during his work at Texas A&M University at Qatar. He has a H-Index of six. His research interests include machine learning and deep learning techniques for power system planning and energy management in smart grids and innovative prediction models.



HAITHAM ABU-RUB (Fellow, IEEE) received the two Ph.D. degree.

He has worked at many universities in many countries, including Poland, Palestine, USA, Germany, and Qatar. Since 2006, he has been with Texas A&M University at Qatar. For five years, he has served as the Chair for the Electrical and Computer Engineering Program, Texas A&M University at Qatar. He is currently serving as the Managing Director for the Smart Grid Center. He has published more than 550 journals and conference papers, five books, and six book chapters. He has supervised many research projects on smart grid, power electronics converters, and renewable energy systems. His main research interests include electric drives, power electronic converters, renewable energy, and smart grid. He was a recipient of many national and international awards and recognitions. He was a recipient of the American Fulbright Scholarship, the German Alexander von Humboldt Fellowship, and many others.



SHADY S. REFAAT (Senior Member, IEEE) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical engineering from Cairo University, Giza, Egypt, in 2002, 2007, and 2013, respectively. He has worked in the industry for more than 12 years as the Engineering Team Leader, a Senior Electrical Engineer, and an Electrical Design Engineer on various electrical engineering projects. He is currently an Associate Research Scientist with the Department of Electrical and

Computer Engineering, Texas A&M University at Qatar. He has published more than 125 journals and conference papers and one book. His principal work area focuses on electrical machines, power systems, smart grid, big data, energy management systems, reliability of power grids and electric machinery, fault detection, and condition monitoring and development of fault-tolerant systems. He has participated and leads several scientific projects over the last eight years. He has successfully realized many potential research projects. He is a member of The Institution of Engineering and Technology (IET) and the Smart Grid Center–Extension in Qatar (SGC-Q).



MOHAMED TRABELSI (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from INSAT, Tunisia, in 2006, and the M.Sc. degree in automated systems and the Ph.D. degree in energy systems from INSA Lyon, France, in 2006 and 2009, respectively. From October 2009 to August 2018, he has been holding different research positions at Qatar University and Texas A&M University at Qatar. Since September 2018, he has been an Associate Professor with

the Kuwait College of Science and Technology. He has published more than 100 journals and conference papers. He is an author of two books and two book chapters. His research interests include systems control with applications arising in the contexts of power electronics, energy conversion, renewable energies integration, and smart grids.



INES CHIHI received the Ph.D. and Habilitation degrees from the National Engineering School of Tunis (ENIT), Tunisia, in 2013 and 2019, respectively. She is currently an Associate Professor of automation and industrial computing with the National Engineering School of Bizerte (ENIB), Tunisia, and a member of the Laboratory of Energy Applications and Renewable Energy Efficiency (LAPER), University Tunis El Manar. Her research interests include intelligent model-

ing, control, and fault detection of complex systems with unpredictable behaviors.



FAKHREDDINE S. OUESLATI received the Civil Engineering, Ph.D., and Habilitation degrees in physics to direct research from the National Engineering School of Tunis (ENIT) and the Faculty of Sciences of Tunis, University of Tunis El Manar, Tunisia, in 1990, 2002, and 2012, respectively. He worked as a Project Engineer at AFH, Tunis, from 1990 to 2002. In 2002, he worked as an Assistant Professor with ISSAT, Higher Institute of Applied Sciences and Technology of Gabes,

INRST, National Institute of Scientific and Technical Research, Borj Cedria, University of Carthage, Tunis, Tunisia. In 2013, he worked as an Assistant Professor with ESTI, Higher School of Technology and Computer, University of Carthage. In 2013 and 2018, he was an Associate Professor at the National Engineering School of Carthage (ENICarthage), University of Carthage, where he has been a Professor, since 2018. He is currently responsible for the axis of the flow and transport phenomena at the Laboratory of Energy and Heat and Mass Transfer (LETTM), Faculty of Sciences of Tunis (FST), Tunisia. He has been a Visiting Professor in several universities and international and prestigious organizations. He is an Invited Professor of many institutes. He has several international publications. His research interests include materials science, energy systems, pollution, and renewable energies, with expertise in multi-component multi-phase convection-diffusion problems. He is a member of several boards of directors or scientists and conference organizing committees. He has chaired many international conferences. He is a regular reviewer for journals and international research projects.

...