Energy 214 (2021) 118874

Contents lists available at ScienceDirect

Energy

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

A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting

Mohamed Massaoudi [a, b, *], Shady S. Refaat [a], Ines Chihi [c], Mohamed Trabelsi [d], Fakhreddine S. Oueslati [b], Haitham Abu-Rub [a]

[a] Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
[b] Unité de Recherche de Physique des Semi-Conducteurs et Capteurs, Carthage University, Tunis, Tunisia
[c] Laboratory of Energy Applications and Renewable Energy Efficiency (LAPER), El Manar University, Tunisia
[d] Department of Electronic and Communications Engineering, Kuwait College of Science and Technology, Kuwait

ARTICLE INFO

Article history:
Received 4 April 2020
Received in revised form
16 September 2020
Accepted 17 September 2020
Available online 24 September 2020

Keywords:
Light gradient boosting machine (LGBM)
Multi-layer perceptron (MLP)
Short-term load forecasting (STLF)
Staking approach
Extreme gradient boosting machine (XGB)
Hyperparameter optimization

ABSTRACT

This paper proposes an effective computing framework for Short-Term Load Forecasting (STLF). The proposed technique copes with the stochastic variations of the load demand using a stacked generalization approach. This approach combines three models, namely, Light Gradient Boosting Machine (LGBM), eXtreme Gradient Boosting machine (XGB), and Multi-Layer Perceptron (MLP). The inner mechanism of Stacked XGB-LGBM-MLP model consists of generating a meta-data from XGB and LGBM models to compute the final predictions using MLP network. The performance of the proposed Stacked XGB-LGBM-MLP model is validated using two datasets from different locations: Malaysia and New England. The main contributions of this paper are: 1) A novel stacking ensemble-based algorithm is proposed; 2) An effective STLF technique is introduced; 3) A critical multi-study analysis for hyperparameter optimization with five techniques is comprehensively performed; 4) A performance comparative study using two datasets and reference models is conducted. Several case studies have been carried out to prove the performance superiority of the proposed model compared to both existing benchmark techniques and hybrid models.

1. Introduction

With the recent waves of digitization and the fast pervasiveness of information and communication technologies, many initiatives have placed an added emphasis on the development and modernization of the traditional centric power grids [1]. The balance between electricity generation and load demand must be optimally maintained to avoid fatal disturbances on the grid due to overloads [2]. To achieve that aim, electric Load Forecasting (LF) offers the necessary tools for stakeholders and energy suppliers to increase their profitability from Renewable Energy Sources (RES) and meet the ever-growing electricity demand. The high complexity of the utility grid operations paved the way for LF to monitor, control, and manage the electric system operations with high efficiency [3].

Recently, LF reached an overall state of maturity that guarantees its safe applicability and profitability in smart grids and traditional utility grids with satisfactory results. LF reveals a cost-effective, efficient, and reliable technique within the energy management framework [4]. Electrical LF assists the scheduling of load response and maintains the fast and economic dispatch to its optimal. Furthermore, it provides a reliable indicator for managing the complex pricing strategies in liberalized and deregulated energy markets with higher financial benefits [5]. Efficient energy management systems strongly require intelligent algorithms to effectively support the electric operations strategy, decrease the electricity bills, and enhance the energy trading and planning [5]. LF is conducted using a variety of features' inputs including social, economic, and weather conditions [3]. Based on the application type, the LF is divided into two categories: 1) the time horizon 2) scope of the variables employed [3].

For the time horizon, most of the LF employed techniques can be divided into four major classes; long-term LF valid for years,

* Corresponding author. Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar.
E-mail addresses: mohamed.massaoudi@qatar.tamu.edu, mohamedsadeg_123@hotmail.com (M. Massaoudi).

https://doi.org/10.1016/j.energy.2020.118874
0360-5442/

Nomenclature

Functions

$\Omega(t)$	Regular function
$L(t)$	Loss function
ϕ_H	Activation function for hidden layers
ϕ_O	Activation function for output layers
$log_e(y)$	Natural logarithm of y
K	Total fold number
C	Extra parameter
w_{ij}^H	Hidden layer weights
w_{jp}^O	Output layer weights

Variables

K	Width of boosting models
LVD	Local Voting Decision
y_i	Actual values (MW)
$\widehat{y}_i$	forecasted values (MW)
S_i	Input feature for testing
N	Samples number
T	Sampling time(hour)
F	Features number

Abbreviations

Adam	Adaptive Moment Estimation optimizer
BO	Bayesian Optimization
CNN	Convolutional Neural Network
CV	Cross-Validation
DFO	Derivative-Free Optimization
DT	Decision Tree
ES	Evolution Strategy
GS	Grid Search
KNN	K-nearest Neighbors
LF	Load Forecasting
LGBM	Light Gradient Boosting Machine
LSTM	Long-Short Term Memory
LVD	Local Voting Decision
MA	Mooving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSLE	Friction Index
OOF	Out-Of-Fold method
PSO	Particle Swarm Optimization
R^2	Coefficient of determination
RF	Random Forest
RMSE	Rooted Mean Squared Error
SA	Simulated Annealing
STLF	Short-Term Load forecasting
XGB	Extreme Gradient Boosting

medium-term LF valid from months to years, Short-Term Load Forecasting (STLF) used from minutes up to one week, and very short-term LF valid from seconds to minutes [6]. The STLF ensures high asset commitment and flexibility of the grid to enhance the serviceability of electricity operations over several time scales for day-to-day operations [7]. In Ref. [3], it was concluded that the most useful LF horizon is STLF.

STLF models are classified into three categories: soft computing techniques, conventional forecasting techniques, and modified traditional techniques [8,9]. Traditional techniques involve regression methods, Iterative reweighted Least Squares, and Exponential Smoothing [10]. However, there are three major drawbacks associated with these methods such as overfitting problems with a massive amount of data, difficulties in feature engineering processing, and relatively less accuracy compared to advanced techniques. Modified techniques include adaptative demand forecasting, stochastic times series, auto-regressive, and moving average-based models, support vector machine-based techniques [3]. Soft computing techniques essentially consist of genetic algorithms, fuzzy logic, neural networks, and knowledge-based expert systems [11]. However, there are some major drawbacks associated with these methods including loss of model interpretability, higher execution time and computational burden, and limited generalization capabilities. Moreover, Hyperparameter Optimization (HO) for soft techniques is a very computationally expensive and time-consuming task [12].

Various algorithms were proposed to solve the optimization problems and enhance the ML model performance. In Refs. [13], the authors proposed an Exchange Market-Genetic Algorithm (EMGA) technique to solve optimization problems with less iterations and better-quality results. The proposed technique combines the merits of the genetic algorithm, and exchange market algorithm [14]. The execution time of the EMGA algorithm took 2.82 min for 641 iterations in solving twelve benchmark functions. Despite the fast execution time and low error rate, the simulation results show that EMGA exhibits a high time iteration ratio. Authors in Ref. [15] used a Simulated Annealing (SA) algorithm for HO of Deep Neural Network (DNN). The proposed SA-DNN achieves accurate results in terms of RMSE. However, the search space of the SA-DNN HO is very limited to avoid computational burden (it only includes the neuron numbers). This leads to low accuracy improvement compared to DNN. In Ref. [16], the authors used the spearmint Bayesian Optimization (BO) method for the HO of recurrent neural networks. The proposed technique is hyper-effective for both short/long-term forecasting. The authors in Ref. [17] used a Derivative-Free Optimization (DFO) technique with deep learning models. The proposed technique uses an efficient feature selection via ensemble structures to predict a variety of RES. However, the comparative analysis of DFO with other HO benchmark techniques is missing.

Ensemble methods have been widely deployed for forecasting applications due to their ease of implementation. In Refs. [18], the authors employed the extreme Gradient Boosting technique (XGB) to predict the load based on similar days using cluster analysis. The presented results confirmed the superiority of the ensemble methods in terms of high accuracy and generalization capabilities compared to the deep learning techniques such as Long Short-Term Memory (LSTM). Nevertheless, the results are unsatisfactory with a relatively poor Mean Square Error (MSE). In Ref. [19], a combination between Convolutional Neural Network (CNN) and Light Gradient Boosting Machine (LGBM) was proposed. The feature extraction process was carried out using CNN model from five wind turbines. The reliability of the proposed technique was verified according to the low registered error metrics values. However, this technique shows sensitivity to time-order character size and requires high computing resources.

In this paper, a Stacked Generalization approach between XGB, LGBM, and Multi-Layer Perceptron (MLP) models named Stacked XGB-LGBM-MLP, is firstly explored for STLF. To the best of the authors' knowledge, no prior work has addressed this architecture for STLF. The Stacked XGB-LGBM-MLP model is characterized by high

accuracy, excellent performance, and ease of implementation. Moreover, the simulation results with an open data portal demonstrated that the Stacked XGB-LGBM-MLP model manages to outperform 11 benchmarks for STLF application. The main contributions of this paper are given as follows:

- A novel Stacked XGB-LGBM-MLP model is proposed to improve the overall regression performance. Despite the potential learning ability and rigorous mathematical theory of XGB and LGBM models, they can only use tree models with the same category, and it is difficult to fundamentally overcome the inherent defects of the tree models. Using MLP model, the metadata enhances its training performance to generate a better-quality result.
- A novel STLF technique is explored and developed in this study. Most of the existing techniques use ensemble models or neural networks for STLF. However, mixing both ensembles and neural networks in one single framework has not received enough attention in the previous studies.
- A comparative analysis of five HO algorithms was comprehensively presented for STLF. Previous studies focused mainly on using various HO techniques to enhance the performance of the ML models. However, selecting the most appropriate HO technique received little attention in the field of ML.
- An assessment of the proposed technique is conducted using two real datasets. The sensibility of the proposed technique to the size and nature of the data has been given significant importance in this research study.
- A comparative study with the recent benchmark techniques is performed. A large comparative study with 11 benchmarks has been conducted to demonstrate the high performance of the proposed stacked XGB-LGBM-MLP model.

Therefore, the paper is organized as follows: Section 2 presents the preliminaries of the proposed technique. In Section 3, two case studies have been conducted. Several existing Machine Learning (ML) models are compared to the proposed technique. Furthermore, a comparative study with the recent benchmark technique regarding the same dataset has been discussed. Finally, section 3 draws conclusions to end this paper.

2. Preliminaries on ML models and stacked XGB-LGBM-MLP method

In this section, three ML models are introduced according to their distinguished architecture, namely, LGBM, XGB and MLP Network. Moreover, the proposed technique is comprehensively investigated.

2.1. Light Gradient Boosting model

LGBM is a boosting ensemble model that transforms coupled weak learners into a potential model [19]. In 2017, Microsoft provides this algorithm on open-source [20]. LGBM enhances the capabilities of Gradient Boosted Decision Trees (GBDT) models in terms of running time acceleration and mitigation of memory consumption while conserving a high accuracy [21]. With a massive volume of data, the traditional GBDT-based models' accuracy decreases, and the forecasting speed significantly declines. LGBM model employs a histogram-based algorithm to mitigate the effects of high dimensional data, accelerates the computational time, and prevents the forecasting system from overfitting. This boosting technique consists of transforming the continuous floating-point eigenvalues into l integers and builds a histogram shape with depth restrictions and k width. LGBM differs from XGB as it adopts a pre-sorted based Decision Trees (DT) technique. Furthermore, parallel learning using a parallel voting DT is adopted during the training process of LGBM. This allows parallel learning for the model. The initial samples are distributed to multiple trees to choose the top-k samples using Local Voting Decision (LVD). The global voting decision collects the top-k LVD attributes to compute the top-2k attributes for k iterations process. In the optimization process, LGBM employs the Leaf-wise method to find suitable leaves. The objective function of LGBM is given by Refs. [22]:

$$Obj(t) = L(t) + \Omega(t) + c \tag{1}$$

where, $\Omega(t)$ and $L(t)$ denote the regular and loss functions respectively, while c and t denote the extra parameter and the sampling time respectively. The extra parameter c prevents overfitting and optimizes the depth of the tree. The regular function reflects the complexity of the model. The parameter $L(t)$ makes the difference between LGBM and the rest of GBDT in terms of computational work acceleration and model feasibility. The loss function represents the fitness of the model from the comparison of real value y_i, and predicted output $\widehat{y}_i$ for N samples defined as [22]:

$$L(t) = \sum_{n=1}^{n} (y_i(t) - (\widehat{y})_i(t))^2 \tag{2}$$

The regression trees are coupled in series to transfer residual information conducted from the previous learners in the chain. The final output result $\widehat{y}_i$ is generated from the accumulation of the residual trees.

2.2. Extreme Gradient Boosting model

The XGB model is one of the commonly used algorithms in forecasting problems [23]. The XGB incorporates an ensemble of DT to build a strong regressor. This large-scale ML method is apt to automatically apply multi-threaded parallelism for accelerating the execution time [23]. Contrariwise to GBDT models, XGB employs the second-order Taylor expansion of the loss function. Moreover, the regularization terms, namely, tree depth and leaf nodes' weights, are part of the objective function of XGB. Thus, the iteration process is decreased and the performance of building trees is enhanced. A level-wise decision tree growth technique is implemented to lessen the model complexity. XGB shares the objective function as LGBM. The loss function of XGB model is given as [23]:

$$L(t) \approx \sum_{n=1}^{n} \left(L\left(y_i, \widehat{y}_{i-1}\right) + g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i) \right) \tag{3}$$

where g_i and h_i functions are given by (4) and (5) respectively [23]:

$$g_i = f'(t) = \frac{\partial L(y_i, \widehat{y}^{t-1})}{\partial \widehat{y}^{t-1}} \tag{4}$$

$$h_i = f''(x) = \frac{\partial^2 L(y_i, \widehat{y}^{t-1})}{\partial \widehat{y}^{(t-1)}} \tag{5}$$

2.3. Multi-Layer Perceptron model

Neural network architectures include many types of algorithms with different designs to suits well-defined applications [24,25]. The difference between these algorithms is mostly related to the information processing techniques adopted. The MLP model is

frequently used for prediction systems [25]. This data-driven technique analyses the feature patterns to produce a meaningful description of the next time steps. The MLP architecture consists of three layers, namely, input layer, hidden layer, and output layer. The learning process comes from the error backpropagation using the gradient descent research approach. In Ref. [26], the symbolic function of the predicted result $\widehat{y}$ is given as follows:

$$\widehat{y} = \phi_o\left\{\sum_{i=1}^{F} w_{jp}^{o}\left[\phi_H\left(\sum_{i=0}^{n} w_{ij}^{H} x_i\right)\right]\right\} \tag{6}$$

where w_{ij}^{H} and w_{jp}^{o} represent the hidden and output layer weights respectively. ϕ_H and ϕ_o denote the activation function for the hidden layers and output layers respectively. x_i denotes the input of F features at a given sample time t. The error function is minimized using the gradient-based function optimization algorithm.

Despite the numerous merits of this architecture including the good accuracy with a higher number of hidden layers and neurons, the unidirectional learning mechanism is inefficient with high dimensional space which may lead to model overfitting. Furthermore, the number of hyperparameters is relatively high including the learning rate, the batch size, the activation function, etc. The optimization of these parameters requires high levels of expertise and computing performance. Moreover, the training process causes a high computational burden to be processed with a slow convergence of weights.

2.4. Stacked XGB-LGBM-MLP method

In this paper, the proposed model is an assembling combination of XGB, LGBM and MLP networks to build a powerful meta-learner. Stacked Generalization (Stacking) is defined as a high-level nonlinear method of models' association [27]. This combination strategy applies non-linear weightings for low-level predictors to boost the forecasting system accuracy. The numerical simulations verify that in many cases, the Stacking technique achieves better results rather than any other base learners [28,29]. The Stacked scheme has two-layered structures (level 0, and level 1) as illustrated in Fig. 1.

According to Fig. 1, the architecture mechanism consists of generating temporal predictions by a set of learners in the first level, called base layer [27]. At level-0, the generalizing biases are collectively predicted. In level-1, named Meta-layer, these predictions are fed to a Meta-learner to calculate the prediction outputs using a Cross-Validation (CV) approach. This methodology aims to filter the output results from the first level of generalization. The training of the forecasting system proceeds as follows: The authors assume that N samples in the training set (S_i, y_i) with $1 \leq i \leq N$ is the sample time. The training set is randomly split into r folds with almost equal size to constitute $(S_i, y_i)_k$ where $k' = (1, \ldots, r)$ is the fold number. The $(S_i, y_i)_k$ satisfy the following condition:

$$\begin{cases} (S,y)_k \cup \overline{(S,y)_k} = (S,y) \\ (S,y)_k \cap \overline{(S,y)_k} = \varnothing \end{cases} \tag{7}$$

Level-0 takes the second part N/r of the dataset $(S_i, y_i)_k$. The weak learners $(L_1, \ldots, L_N)$ predict the Yi using S/S i feature vectors. At this level, the test part Si is computed using weak learners. The output results with the actual dataset y_i shape the meta-level dataset MSi with a new feature vector. The MSi dataset is fed to a meta-learner to form a meta-level vector from the base-level regressors. The Stacking concept adopted in level-0 consists of two training models with 5 folds as shown in Fig. 2.

According to Fig. 2, the level-0 learners consists of XGB and LGBM while the meta-learner is an MLP model. The concept of the proposed approach is hierarchically coupling ensemble methods and Neural Networks symbolized by MLP model to form a multi-modal forecasting system. The intuition of selecting XGB and LGBM as base models is conducted due to their high performance in other forecasting applications. We believe that the stronger forecasting potential of each heterogeneous base model, the higher performance of the overall stacking ensemble. The objective here is to build a perfectly tailored forecasting system to cope with nonlinear system variations, specifically STLF problem. A binary combination of level-0 models takes into consideration the input parameters, specifically, the temperature and the DateTime to be introduced to the XGB and LGBM models. These two techniques generate predictions for each test set using an Out-Of-Fold (OOF) method. In

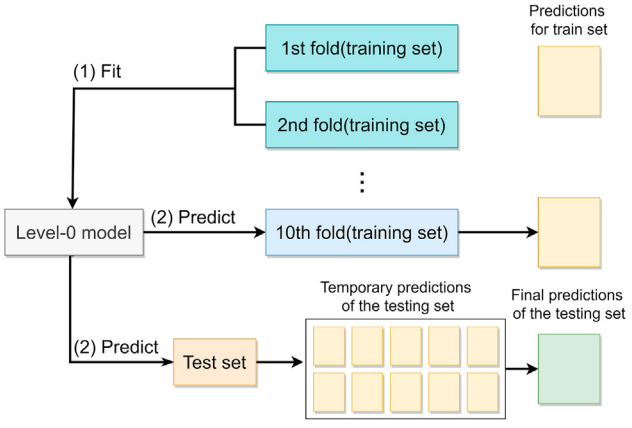


Fig. 2. Schematic representation of the level-0 of stacked architecture framework built from base learners: XGB and LGBM.

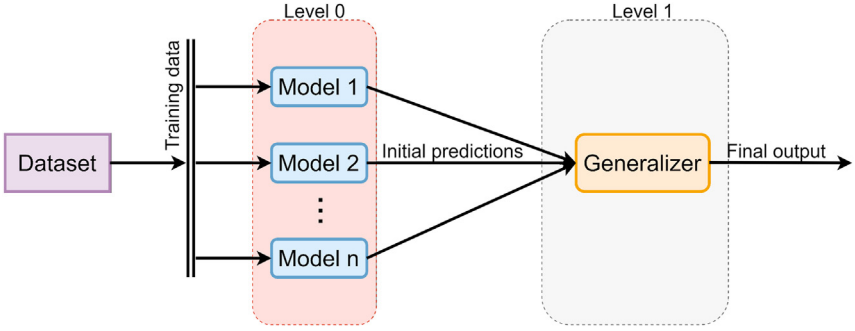


Fig. 1. Graphical representation of the stacking generalization approach.

other words, the training part uses a sub-fold for a CV-such as fashion training which is different from the validation part to avoid the overfitting problem or miscellaneous learnings. A mean value of all folds is calculated for the temporary test sets. The forecasting inputs are considered as parameters for the next level. At level-1, the MLP model fits the model with temporary predictions in the training set. Once MLP is trained, the forecasting system considers only the output results generated at this stage. It is worth saying that the training system takes more time to fit all the sub-folds with different meta-learners and base learners. Furthermore, the higher-level of predictors does not necessarily ensure better results as well as the total stacking concept. The detailed model synoptic is illustrated in Fig. 3.

Regarding Fig. 3, four stages are processed, specifically, Feature engineering, object determination, Forecasting, and evaluation stages. In the engineering stage, the information is processed with data cleaning, feature extraction, and feature selection. The dataset, specifically, temperature, month, day, hour and load feature vectors are integrated into the object determination stage. This stage contains the problem formulation, the data split into training and testing. Furthermore, the ML models' hyperparameters are optimized to select the most suitable values according to the single learners and the meta-learner individually. In the forecasting stage, XGB and LGBM are trained in the level-0 of the Stacking approach. Then, the meta-data outputs are fed for the MLP meta-learner model. The meta-learner analyses the attributes inputs to compute the final output. Fig. 4 clearly presents a flowchart of the whole procedure.

The evaluation stage is conducted to analyze the performance of the forecasting system using point forecasting assessment, K-fold cross-validation, and visualization graphs.

3. Case studies and performance assessment

Based on a publicly available datasets, a real case studies were conducted to assess the proposed approach and illustrate the prediction performance of the hybrid model. Furthermore, a comparative analysis with the recent benchmarks is performed. Moreover, the high accuracy of the proposed method with the latest hybrid STLF technique for the same dataset has been verified.

3.1. Data analysis and processing

The high performance of a forecasting system is essentially related to two factors: the quality of the input data and the forecasting engine. Therefore, data analysis and feature engineering are crucial for enhancing the data quality of the underlying system. To verify the efficiency of the proposed approach, two real datasets are employed. For the first case study, the data are taken on an open-access base from a power supply industry in the city of Johor,

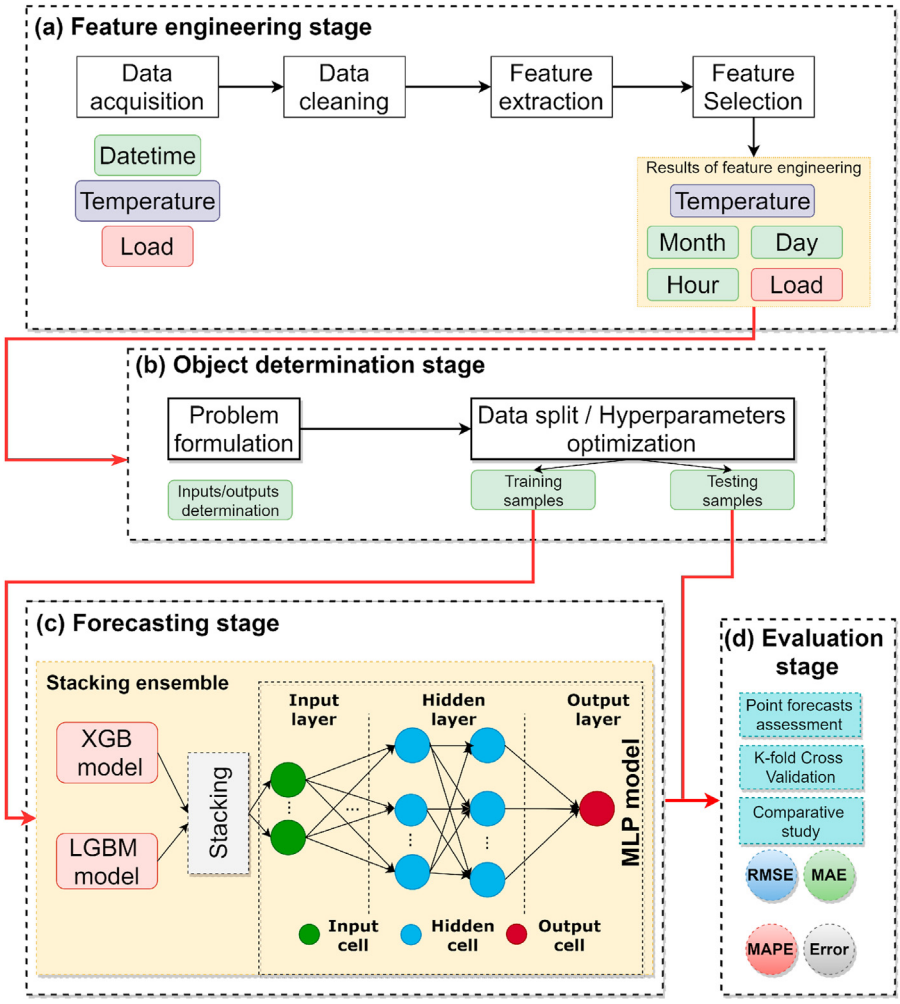


Fig. 3. Block diagram of the proposed stacked ensemble forecasting framework.

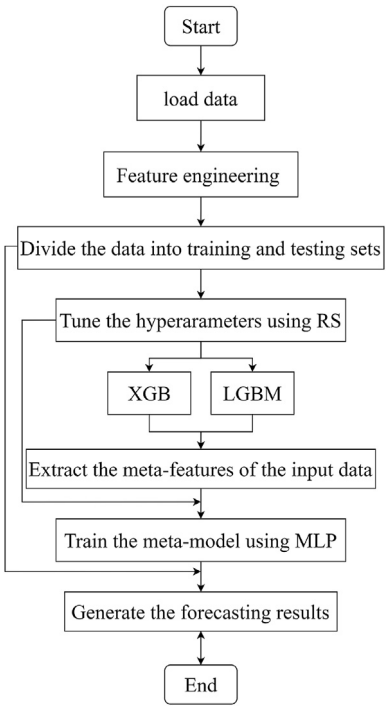


Fig. 4. Flowchart of the proposed Stacked XGB-LGBM-MLP method.

Malaysia [30]. The collected information contains high-resolution data with 17519 samples and a sampling frequency of 1 h. The database covers a range of time from 01/01/2009 to 01/01/2011. The measured hourly temperature (°C) and load demand (MW) as shown in Fig. 5.

According to Fig. 5a, it can be observed that the load demand presents a high variation associated with some outliers (samples = [14748−14749]). The load pattern consumption stochastically changes over time with high instabilities and nonlinear behavior. The load data is employed to verify the feasibility of the proposed architecture using the temperature as input presented in Fig. 5b. The long-term evolution of the temperature is assumed as the key indicator of the load demand variation.

For the second case study, the used dataset comes from ISO New England control area and its eight-wholesale load (ISO-NE) records [31]. The records comprise Boston, Bridgeport, Burlington, Concord, Portland, Providence, Windsor Locks, and Worcester data. The data reports from ISO-NE are employed for planning and monitoring purposes. It is worth noting that the system load is the sum of metered generation, and metered net interchange in addition to the demand from pumped storage units. The data acquired includes the hourly temperature and load taken from 2003 to 2014. The total vector rows used is 103775. The two datasets were sliced into training data and testing data at a ratio of 80% and 20% respectively. The testing set for the first and second case studies contain 1755 and 103774 vector rows respectively. The goal of using these datasets with different sizes tackles testing the forecasting performance on a small and large sets.

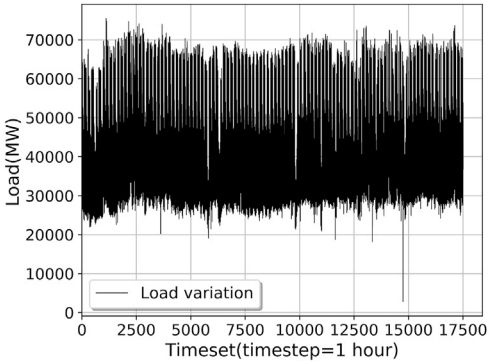
For the data pre-processing, the dataset was cleaned from outliers and abnormal values. Moreover, the DateTime inputs were given as categorical. Therefore, the DateTime were converted to numerical values and split to separated features inputs (hour, day, month, year) to enhance the continuous forecasting of time series labels and simplify the forecasting process. Then, the data is separated into two folds for training and testing. The final input features include the yearly, monthly, daily and hourly DateTime inputs associated with the temperature while the outputs were the load forecasts.

3.2. Hyperparameter optimization

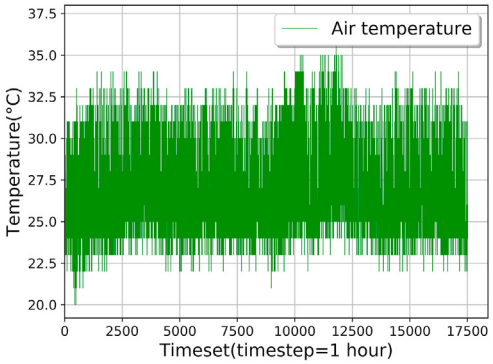
In this paper, several HO techniques have been investigated in order to select the most suitable technique for STLF. Traditionally, the hyperparameter values were selected based on trial and error method. This is conducted by interpreting the model performance and progressively refining the parameters. However, manual tuning is often time-consuming yielding to unsatisfactory results without deep expertise. To solve these problems, Grid Search method (GS) is used as an automatic HO technique to track all the possible values in the search space. The GS method is only used when the dimension of the search space is relatively low. Furthermore, GS is computationally extensive since all cases in the search space are investigated and simulated. Alternatively, the use of meta-heuristics for HO can significantly reduce the computational effort and the fair amount of time associated with GS. In this paper, five methods were assessed and compered for HO. Specifically, PSO, SA, ES, RS, and BO [32]. A brief description for these methods is given in the following subsections.

3.2.1. PSO

PSO is a population meta-heuristic method used for optimization problems. This method was inspired by the behavior of



(a) Electric load(MW) between 2009-2011



(b) Hourly temperature(°C) between 2009-2011.

Fig. 5. The time series data between 2009 and 2011 (17519 samples).

migratory birds for energy and time optimization during their movement [33]. The proposed PSO based methodology uses a set of particles, called initial population, moving at the same time in a defined search space. The distance between the particles, and the best position decreases after each iteration. The mechanism of PSO is conducted by updating the position x, and the velocity υ until reaching the optimum values. The symbolic equations of x and υ are given respectively as [32]:

$$x_i^{t+1} = x_i t + v_i^{t+1} \tag{8}$$

$$v_i^{t+1} = w v_i^{t+1} + c_1 r_1 \left(P_{besti} - x_i^t\right) + c_2 r_2 \left(G_{best} - x_i^t\right) \tag{9}$$

$$G_{best} = max\{f(G_0), f(G_1), \ldots, f(G_m)\} \tag{10}$$

where G_{best} is the best position for the entire population, and Pi is the best position found so far. c_1 and c_2 are constants greater than 0. w and f are the inertia weighted factor and the objective function respectively. The merits of PSO method include fast convergence, simple implementation, and parallel processing which leads to higher computational efficiency [33]. Furthermore, the good performance under multipeak function optimization makes PSO a good candidate for HO problems.

3.2.2. BO

BO method also named as Sequential Model-Based Optimization method (SMBO) is classified as a derivative-free technique. The BO method consists of estimating the objective function using the Gaussian process model [33]. Firstly, two random sets of hyperparameters are assessed. A prior knowledge of the optimization problem is sequentially built using a probabilistic model. Then, the scalar objective function $f : X \rightarrow \mathbb{R}$ for a subset $X \subseteq \mathbb{R}^d$ is optimized by Ref. [34]:

$$x^* = \underset{x \in X}{arg\,max} f(x) \tag{11}$$

where x^* is the global optimum of $f(x)$ constrained domain including real, integer or categorical feature values. The advantages of BO include its fast convergence, high performance, scalability and suitability for HO problems, especially when the features are non-parametric. However, the disadvantages of the BO-based HO could be classified in two classes: the training time and the fine tuning of BO parameters. Since BO is sequential method, its parallelization in order to decrease the computational time is difficult. Furthermore, the kernel function of BO is hard to be tuned. A recent research work addresses these problems such as standardizing the BO parameters [33].

3.2.3. RS

RS is an automatic hyperparameter selection technique that can be considered as a speed-up of GS method. The mechanism of RS goes through iteratively moving to suitable positions in the research space [32]. RS is advantageous in terms of parallelization and the ease of implementation especially in case of sophisticated models. This is due to the absence of gradient optimization where all iterations are independent. Thus, the application of RS includes discontinuous functions. However, the limitation of RS lies in missing optimum values due to the random iterations' procedure. This problem can be solved by giving a more spread values for sampling the RS method.

3.2.4. ES

ES is a heuristic search technique inspired from the natural biological evolution [35]. It consists of a mutation and combination of the best individuals of a certain population of solutions following the Darwinian evolution [35]. According to the fitness values, natural selection empowers the best candidate solutions for mutation and reproduction. Therefore, these solutions dictate the distribution of future generations. Reciprocally, the weak candidates are removed from the population [35]. The ES presents a robust algorithm and highly parallelizable. Hence, the disadvantages of ES include the risk of being trapped in local minimum [35].

3.2.5. SA

SA technique is an efficient meta-heuristic technique for HO [32]. This technique mimics the physical process of molten material with a uniformly re-partitioned temperature. The SA mechanism is governed by randomly generating a possible solution. Then, the SA method generates a successive modifications to this solution. The probability p of moving to a new SA solution is given by Ref. [32]:

$$p = exp\left(-\frac{\Delta f_{norm}}{T}\right) \tag{12}$$

where Δf_{norm} represents the difference between the current individual and the candidate individual, T presents the temperature. The Δf_{norm} function is calculated as follows [32]:

$$\Delta f_{norm} = \frac{f(y) - f_{min}(y)}{f_{max}(y) + f_{min}(y)} \tag{13}$$

The limitations of SA are mainly resumed in fine-tuning its parameters, specifically, epsilon value, number of neighbor solutions, starting temperature, and annealing rate. Furthermore, the inner mechanism of SA limits the possibility of being deployed on parallel computing with faster potential.

3.3. Comparative results

The selection of the most suitable HO techniques is often critical for achieving satisfactory performance with minimum time. This is due to the large number of HO techniques in the literature. To the best of the authors' knowledge, very few research studies have considered this issue. Therefore, five HO techniques are tested in order to select the best HO method for the proposed technique. A search space for every component of the proposed Stacked XGB-LGBM-MLP has been tuned individually. The testing set was performed using a Lenovo Intel Â®i7 Â®Nvidia Geforce GTX 1650@ 2.30 GHz. For the modeling, we used Python programming language and Hyperactive library [32]. The search space contains 9 selected parameters, specifically, the number of estimators, the learning rate, maximum depth of both XGB and LGBM, the maximum of iterations, and the hidden layer size for MLP model. The search space is very large due the number of combinations from assigned hyperparameter values in order to reach to the close optimums. Table 1 presents the simulation results of each model.

According to Table 1, it can be observed that the fine-tuning of the hyperparameters is computationally demanding for all the tested models but with different duration. The fastest algorithms for HO are ES and SA which scientifically reduced the computational cost compared to the other techniques with a consuming time of 82 min and 133 min respectively. On the other side, the most effective technique according to the R^2 is RS method. This is explained by the fact that expanding the running time leads to higher chances to obtain results closer to optimum. Hypothetically, the registered $R^2 = 98\%$. It is worth mentioning that, Despite the performance superiority of RS, this method is computationally expensive compared to the rest of tested HO methods. Taking into

Table 1
Hyperparameter settings and optimization results for the proposed Stacked XGB-LGBM-MLP model.

ML Method	Score Function	Default value	Search space	SA	PSO	BO	RS	ES
XGB	Number of estimators	100	[100, 500, 750, 1000, 1500, 2000, 2500, 3000]	1500	1000	100	2000	2000
	Learning rate	0.1	[10^{-3}, 10^{-2}, 0.1, 0.3, 0.2, 0.5, 0.8, 1]	0.1	0.3	0.2	0.1	0.3
	Max depth	3	−2~12/step = 1	4	2	7	12	3
LGBM	Number of estimators	100	[100, 500, 750, 1000, 1500, 2000, 2500, 3000]	750	2000	1500	500	2500
	Learning rate	0.1	[10^{-3}, 10^{-2}, 0.1, 0.2, 0.3, 0.5,0.8, 1.0]	0.5	0.2	0.3	0.3	0.001
	Max depth	−1	−2~12/step = 1	2	−1	4	11	3
MLP	Max iterations	200	1000~3000/step = 500	2500	2000	2000	1000	1500
	Hidden layer sizes	100	[100, 200, 300, 400, 500]	100	200	500	200	300
			R^2(%)	90	91	88	98	89
			Time(min)	133	204	212	259	82
			Time/iteration(min)	4.42	6.79	7.07	8.62	2.71

account the HO method accorded with higher R^2 value for the rest of hyperparameter tuning problems in this paper, The RS is adopted since it generates the best results.

3.4. Evaluation criteria

With the increasing number of candidate ML techniques devoted to fit well-defined prediction problems, the selection of the most suitable technique that copes with uncertainty and seasonality of weather parameters is a challenging problem. This is explained by the fact that some ML methods are designed to fit specific structured data, often losing sight of the significance with low generalization capabilities for other types of applications or data. Model selection process is characterized by the heavy computational burden and the confusion in the determination of the adequate criteria by which the performance of the model is considered satisfactory. The most relevant criteria for model selection include score metrics, training speed, ease of implementation, and training burden. ML techniques inevitably must be verified in terms of accuracy, soft computing, and computational time for execution. In this investigation, the evaluation criteria are conducted using three methods for the sake of integrity and reliability of assessment procedure, specifically, graphical visualizations, point forecast assessment, and K-fold cross-validation. The score metrics used in the point assessment methods include Rooted Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Logarithmic Error (MSLE), and Coefficient of determination(R^2). The error metrics calculation was computed using Scikit-learn with their parametric equations as follows [36]:

$$MAE = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \widehat{y}_i| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} y_i - \widehat{y}_i)} \tag{15}$$

$$MAPE = \frac{100(\%)}{n}\sum_{i=0}^{n-1}\left|\frac{\widehat{y}_i - y_i}{y_i}\right| \tag{16}$$

$$R^2 = 1 - \frac{\sum_{i=0}^{n-1}(\widehat{y}_i - y_i)^2}{\sum_{i=0}^{n-1}(\overline{y}_i - y_i)^2}, \overline{y} = \sum_{i=0}^{n-1} y_i \tag{17}$$

$$MSLE = \frac{1}{n}\sum_{i=0}^{n-1}\left(log_e(y_i+1) - log_e(\widehat{y}_i+1)^2\right) \tag{18}$$

$$MdAE = \underset{j=(1,\ldots,n)}{Md}\left(\left|y_i - \widehat{y}_i\right|\right) \tag{19}$$

where $\widehat{y}_i$ and y_i present the i^{th} forecasted values and the actual values respectively. Here n denotes the total number of samples.

3.5. Numerical evaluation

3.5.1. 1st case study

The performance assessment of the proposed model for different time scales ahead is carried out in this section. The proposed algorithm was implemented using Python programming language. Several Python packages were included in the model design including Scikit-learn, LightLBM, and XGBoost [36,37]. Furthermore, the stacked generalization was build using Vecstack package [38]. The simulation results were generated from a Lenovo Intel Â®i7 9th Generation Â®Nvidia Geforce GTX 1650@ 2.30 GHz@16 GB RAM. The HO was conducted using RS tool for the sake of enhancing the forecasting accuracy by the comprehensive selection of the most suitable parameters [39]. To assess the proposed method effectiveness, the model is trained and compared to individual models, specifically, XGB, MLP, and LGBM [40]. Initially, these techniques were employed as benchmarks to verify the effectiveness of the proposed approach. Obviously, the forecasting performance of the proposed ensemble method depends largely on the prediction performance of the base models. The simulation procedure was repeated 10 times for providing higher reliability to the forecasting system. To make the comparison more intuitive, a single step ten-fold CV(10-CV) was conducted to follow the variations of the models' performance. The results of 10-fold CV is computed and shown in Table 2 for the ingredients of the proposed method. Furthermore, Table 3 resumes the 10-fold CV results for the Stacked XGB-LGBM-MLP model.

According to Table 2, the analysis of the prediction performance of each base model demonstrates that LGBM generates the best forecasting results in terms of the score performance measures. The average R^2 for LGBM model is R^2 = 89,14% compared to R^2 = 87,89% and R^2 = 86,52% for MLP and XGB models respectfully. The raported results include a mean RMSE = 1006,87 MW, 953,22 MW and 1061,95 MW for MLP, LGBM, and XGB respectively. Consequently, the stacked XGB-LGBM-MLP succeeds to preform best of all single models with a mean R^2 = 94,31% as reported in Table 3. The high forecasting accuracy achieved by the stacking ensemble is verified from a lower mean RMSE = 691,40 MW. The stacked XGB-LGBM-

Table 2
Error metrics of MLP, LGBM, XGB models for the 1st case study.

Model	MLP			LGBM			XGB		
Fold number	MAE (MW)	RMSE (MW)	R^2(%)	MAE (MW)	RMSE (MW)	R^2(%)	MAE (MW)	RMSE (MW)	R^2(%)
0	771,15	1003,47	88,05	714,62	943,51	89,44	802,91	1059,83	86,67
1	766,97	1017,14	87,16	724,45	973,56	88,24	814,18	1090,52	85,24
2	736,62	991,25	88,44	705,14	949,31	89,40	790,72	1057,65	86,84
3	758,88	1013,63	87,68	726,33	970,30	88,71	798,82	1060,68	86,51
4	744,16	988,11	88,36	705,20	939,96	89,47	792,17	1056,48	86,69
5	759,27	1007,75	87,77	710,56	949,88	89,14	791,84	1056,93	86,55
6	762,86	1000,71	87,91	709,99	942,76	89,27	793,99	1054,44	86,58
7	780,16	1028,25	87,44	716,76	959,15	89,07	799,85	1068,04	86,45
8	790,56	1029,11	87,74	727,56	962,48	89,27	807,24	1068,88	86,77
9	744,61	989,23	88,30	711,90	941,29	89,41	788,60	1046,00	86,92
Mean	761,52	1006,87	87,89	715,25	953,22	89,14	798,03	1061,95	86,52
SD	15,92	14,35	0,39	7,91	11,72	0,37	7,78	11,37	0,45
Computational Time (second)		3238.38			3664.78			3643.24	

Table 3
Error metrics of Stacked XGB-LGBM-MLP model for the 1st case study.

Stacked XGB-LGBM-MLP model			
Fold number	MAE(MW)	RMSE(MW)	R^2(%)
0	489,74	707,29	94,05
1	477,22	683,82	94,45
2	480,60	687,10	94,24
3	487,03	707,94	94,17
4	481,47	690,45	94,35
5	483,04	718,25	93,88
6	488,27	696,69	94,33
7	475,68	675,73	94,47
8	475,64	680,11	94,84
9	471,58	666,61	94,84
Mean	481,03	691,40	94,31
SD	5,75	15,29	0,25
Computational Time (second)		2033.08	

MLP model can make full use of the advantages of various models with different training mechanism and realize the complementarity between the advantages of different models. The relative stability of the proposed technique with a minimum Standard Derivation SD = 0,25% compared to its ingredients reflects the effectiveness of the model against the stochastic variations. For a real testing environment on a 24-h ahead, Table 4 resumes the error metrics calculation.

Regarding Table LABEL:missed, the RMSE value of the proposed approach is the lowest by a value of RMSE = 1509.74 MW compared to an RMSE = 5641.67 MW, RMSE = 2143.23 MW and RMSE = 5135.50 MW for the individual XGB, LGBM, and MLP models respectively. The MAE value of the proposed approach is also the lowest equal 1070.67 MW compared to 5220.20 MW and 1764.14 MW for XGB and LGBM respectfully. Furthermore, the Stacked XGB-LGBM MLP model outperforms the rest of the models in terms of MAE, R^2, MAPE, and MSLE scores. A daily STLF is simulated and shown in Fig. 6.

Globally, it can be noticed that the predicted values follow the

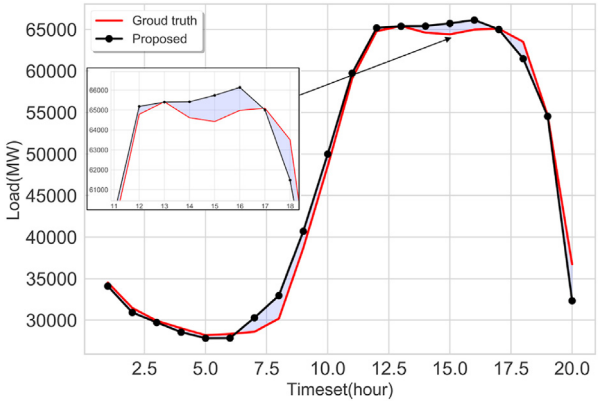


Fig. 6. Actual and predicted demand for 31/12/09.

Table 4
Score errors for the stacking ensemble and its components for 24-h ahead.

Model	RMSE (MW)	MAE (MW)	R^2 (%)	MdAE (MW)	MAPE (%)	MSLE (10^{-2})
XGB	5641.67	5220.20	0.87	4455.34	12.77	1,8
LGBM	2143.23	1765.14	0.98	1404.88	3.85	0,23
MLP	5135.50	4623.32	0.89	4344.50	10.36	1,47
Stacked XGB-LGBM-MLP	1509.74	1070.67	0.99	597.64	2.69	0,17

real values as shown in Fig. 6. The largest error (4652.57 MW) is shown in hour 16 while the lowest error is equal to 297.71 MW. The increase of the error is due to the high value at the load peak. The relationship between real load and forecast points is investigated with the marginal distribution displayed in Fig. 7.

Regarding Fig. 7, the majority of the predicted values during the testing phase are very close to the zero references, which leads to conclude that the system does not have an unbiased probability. The graphical mapping of the proposed model as shown in Fig. 8.

According to Fig. 8, the proposed model is the closest to the ground truth compared to the rest of the models. It is remarkably observed that the stacked generalization is by far ameliorating the system performance compared to single methods. The simulation results have been compared to 11 techniques proposed with the same testing conditions and settings and reported in Ref. [41]. These techniques include hybrid Convolutional Neural Networks-Fuzzy Time Series (FTS-CNN), Seasonal Auto-regressive Integrated Moving Average (SARIMA), Probabilistic Weighted Fuzzy Time Series (PWFTS), Weighted Fuzzy Time Series (WFTS), Integrated Weighted Fuzzy Time Series (IWFTS) and LSTM Neural Networks [41]. The LSTM architecture includes two layers with 256 and 64 units respectively, sequence input layer, dropout layer with a forgetting rate of 0.4, a softmax function, a batch size of 72, a fully connection layer and regression layer, a maximum of iterations of 1000 with an early stopping function. The reference models were adopted from Ref. [41] where it has been reported in that the lagged hours significantly affects the forecasting performances. Therefore, the tested LSTM 1,2,3 were assigned to three configurations according to 24, 48, 72 former lags fed as feature inputs. The best performance of LSTM is attributed to lagged hour 168. Furthermore, KNN, and RF are added to the list of reference models. The configurations of reference models are given in Table 5. An illustration of the proposed technique compared to the state-of-the-art models is displayed in Fig. 9 Moreover, Fig. 10 shows a graphical histogram shape of MAPE and RMSE values of the Stacked XGB-LGBM-MLP compared to the benchmarks' models.

According to Fig. 9, it is remarkably observed that the stacked approach achieves the best results compared to the rest of the methods in terms of MAPE values. The Stacked XGB-LGBM-MLP model succeeds to eliminates the poor prediction aspects in single models. As illustrated in Fig. 10, the proposed method is giving the lowest MAPE value which confirms its superiority for STLF. Table 6 presents the numerical RMSE and MAPE values of all of the

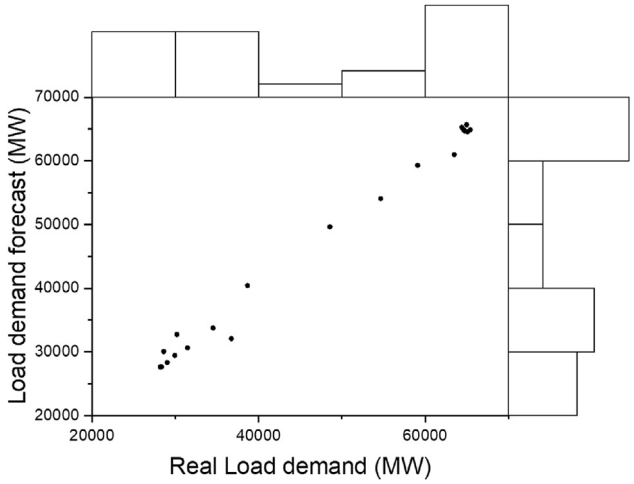


Fig. 7. Scatter graph of the joint distribution of the real load demand and its forecast.

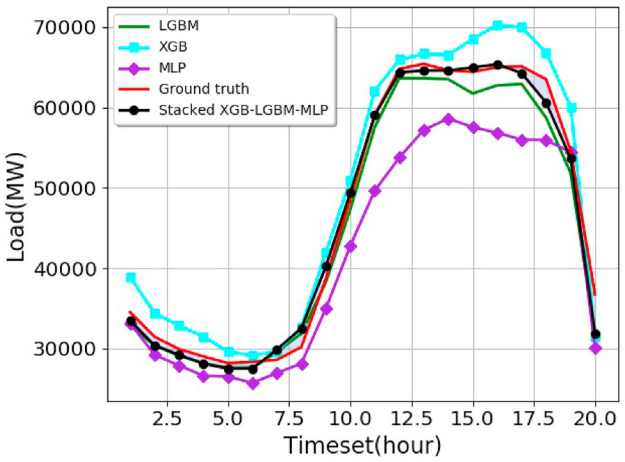


Fig. 8. Actual and predicted load demand with XGB, LGBM, MLP, and stacked model.

cited methods.

According to Table .6, The superiority of the proposed method is expressed by a decrease of an RMSE of 192,96 MW compared to FTS-CNN. The stacked approach presents a better accuracy performance since the meta-learner and base learners analyses data patterns from different feature spaces and structures. The extension of the forecasting period was investigated to explore the influence of the prediction horizon on the prediction accuracy. Fig. 11 illustrates the prediction results for 48 h.

Regarding Fig. 11, it can be concluded that the horizon extension for more than 24 h decreases the accuracy of the forecasting system, especially after the first 30 h of prediction. Therefore, the proposed architecture only fits the short-term forecasting horizon for one day. Table 7, resumes the quantification of the score metrics for two days of forecasts.

According to Table 7, the 48-h ahead forecasting is conducted with an RMSE = 3033.57 MW compared to an RMSE = 1509.74 MW for one day-ahead forecasting. The MAE is 1984.33 MW compared to 1070.67. The RMSE value approximately decreases to half, which presents a serious problem for longer forecasting dependencies. However, the proposed model still follows the real load demand with the same behavior. In general, the effect of longer forecasting dependencies is a serious dilemma that limits the capabilities of ML models associated to the scalability of forecasting systems.

3.6. 2nd case study

In the second case study, ISO-NE has been used in order to assess the effectiveness of the proposed model on a larger set. The goal of this section is to quantify the sensitivity and robustness of the model with a large data for 11 years. The aforementioned adopted methodology is kept the same for the model construction. To make the comparison more intuitive, a single step ten-fold CV was conducted to follow the variations of the models' performance. Here, the results were performed using google Collaboratory (Collab) in order to alleviate the computational effort from using the large ISO-NE dataset [31]. Collab was used as a cloud service with a GPU-centric application. The output results of the proposed models individually assessed are given in Table 8. Additionally, the proposed XGB-LGBM-MLP model results for 10 fold CV are resumed in Table 9 and the resulted mean R^2 score is plotted shown in Fig. 12.

According to Table 8, The LGBM generates the best performances compared to XGB and MLP models with a mean $R^2 = 94{,}13\%$. Hence, a significant enhancement has been demonstrated for the proposed

Table 5
Hyperparameters settings for reference models.

Base models	Hyperparameter settings
XGB	The number of estimators is 2000; The learning rate is 0.001; The tree complexity is 2; The gamma value is 0.5; The max depth is −1
XGB	The number of estimators is 2000; The learning rate is 0.001; The tree complexity is 2; The gamma value is 0.5; The max depth is −1
LGBM	The number of estimators is 2000; The learning rate is 0.3; The max depth is 11; The tree complexity is 3; Number of leaves 130
MLP	The maximum iterations are 1000; The hidden layer sizes are 500,100,50, and 30; the solver is Adam
FTS-CNN	The image size is 32; the batch size is 100; the number of epochs is 20; the learning rate is 0.001; The convolutional layers' number are 2; the fully connected layers are 5; the dropout layer is 40%; the activation function is Rectified Linear Units (ReLU) the max pooling is 2*2
SARIMA	The structure is (1,0,1)(1,1,2)
PWFTS	Default parameters from PyFts
WFTS	Default parameters from PyFts
IWFTS	Default parameters from PyFts
LSTM	The number of layers is 2; The activation function is ReLU; The data used contains 24, 48, 72, and 168 lagged hours for LSTM, LSTM1, LSTM2, and LSTM3
KNN	The algorithm is KDTree; the nearest neighbor number is 7; the leaf size is 90; the distance function is Minkowski distance
RF	The maximum depth is 50; the minimum samples split is 10; The number of estimators = is 140

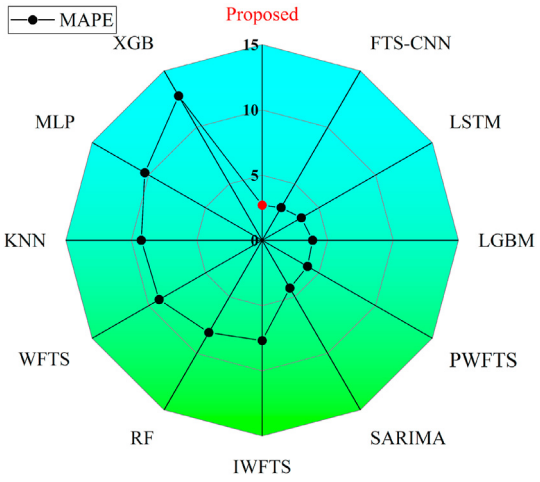


Fig. 9. Comparison of MAPE performance for ML models.

technique that achieves a mean $R^2 = 97{,}73\%$. The forecasting results can be always kept above $R^2 = 97{,}37\%$. Fig. 12 presents the high stability of the proposed XGB-LGBM-MLP model with a minimum R^2 equal to 95%. The mean RMSE and MAE for Stacked XGB-LGBM-MLP are 481,03 MW and 691,40 MW respectively. Fig. 13 presents a Box-and-Whisker plot of RMSE values for STLF.

Table 6
Comparison of related work with hybrid and deep learning models.

Model	RMSE (MW)	MAPE (%)
Stacked XGB-LGBM-MLP	1509.74	2.69
FTS-CNN	1702.70	2.89
SARIMA	2501.25	4.23
PWFTS	2162.57	4.00
WFTS	4419.11	9.09
IWFTS	4663.17	7.69
LSTM	2037.49	3.45
LSTM model 1	2044.68	4.21
LSTM model 2	2483.71	4.23
LSTM model 3	2279.23	3.88
RF	5308.89	8.16
KNN	5851.56	9.24

According to Fig. 13, the proposed technique achieved a significant improvement compared to single models with a lower RMSE value. The simulation results shows that the prediction potential could be enhanced by a augmenting the size of the data. Nevertheless, the Stacked XGB-LGBM-MLP also has good robustness under the condition of the small dataset. It can be seen that the stacking ensemble learning model combines the merits of every single models to overcome the limitations of low-precision prediction of single models. Therefore, it can be deduced that the proposed stacking technique is perfectly tailored for STLF.

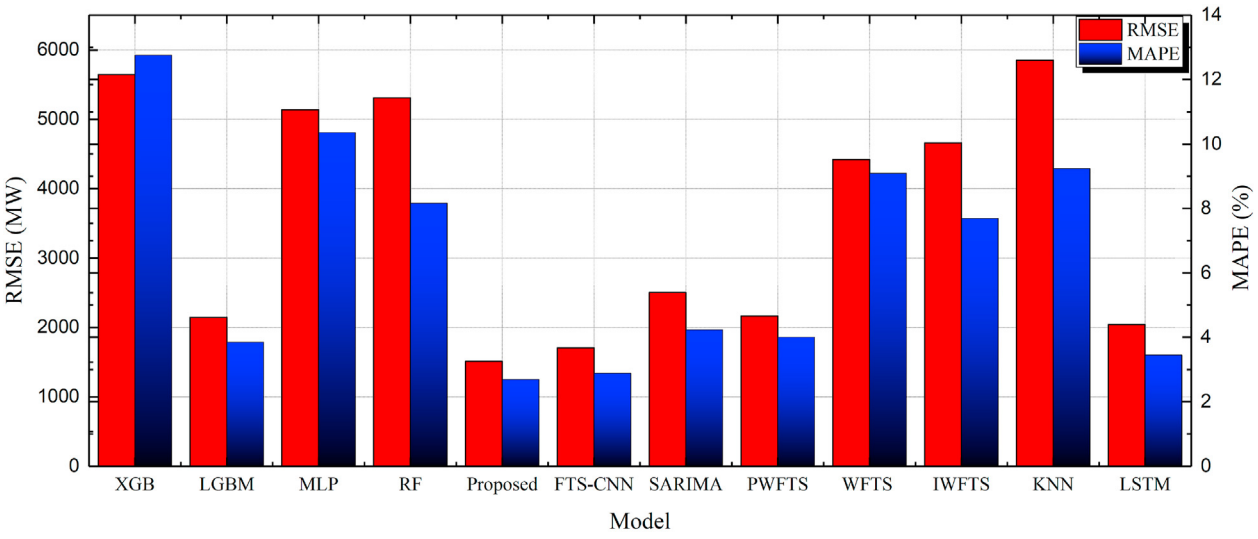


Fig. 10. RMSE and MAPE values for the reference models and the proposed technique.

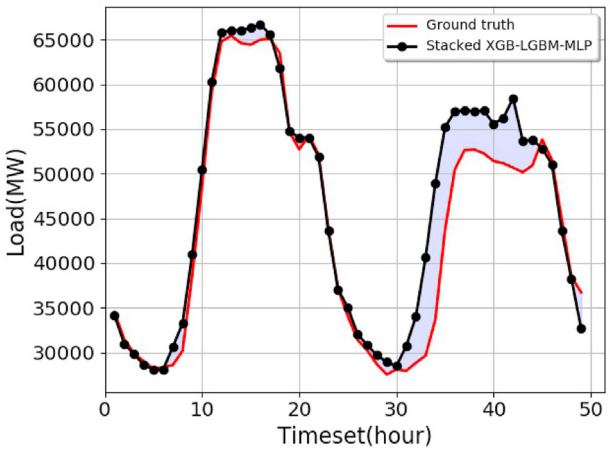


Fig. 11. Actual and predicted demand for 31/12/09 and 01/01/10.

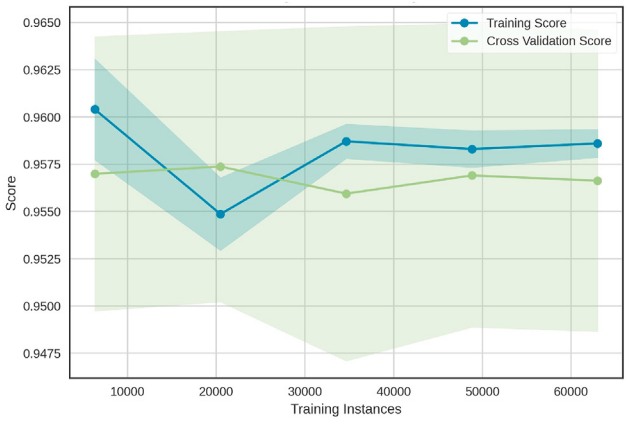


Fig. 12. 10-fold CV graph for Stacked XGB-LGBM-MLP.

Table 7
The impact of forecasting horizon of the Stacked XGB-LGBM-MLP model performance.

Horizon	RMSE (MW)	MAE (MW)	R^2	MdAE (MW)	MAPE (%)
24 h	1509,74	1070,67	0.99	597,64	2,69
48 h	3033.57	1984.33	0.94	1253.52	4.96

4. Conclusions and future work

This paper proposes a novel computing framework based on stacked generalization method for STLF. In order to improve the accuracy of single techniques, the proposed technique combines three efficient methods, namely, Extreme Gradient Boosting (XGB), Light Extreme Gradient Boosting, and Multi-Layer Perceptron (MLP) models. The components of the proposed model are selected based on individual performance, training time, and ease of implementation trade-off. The presented experiments strongly confirm the effectiveness of the Stacked XGB-LGBM-MLP model

Table 8
Error metrics of MLP, LGBM, XGB models for the ISO-NE dataset.

Model	XGB			LGBM			MLP		
Fold number	MAE (MW)	RMSE (MW)	R^2(%)	MAE (MW)	RMSE (MW)	R^2(%)	MAE (MW)	RMSE (MW)	R^2(%)
0	680,86	923,13	89,67	499,24	699,55	94,07	983,12	1223,75	81,85
1	679,10	931,23	99,78	499,49	711,96	94,03	950,65	1196,69	83,12
2	668,59	912,48	90,10	483,88	682,19	94,47	946,90	1185,69	83,28
3	664,57	921,06	89,92	501,88	722,47	93,80	992,68	1248,93	81,47
4	681,95	943,98	89,36	503,28	726,16	93,71	839,45	1116,10	85,13
5	657,03	900,76	90,49	498,37	701,10	94,24	927,56	1168,89	83,99
6	650,13	884,33	90,67	487,10	685,76	94,39	982,65	1226,55	82,04
7	673,56	914,08	89,64	495,53	693,51	94,04	999,57	1234,09	81,12
8	666,06	905,12	90,49	491,19	684,10	94,57	1017,51	1274,83	81,14
9	661,34	90,02		490,69	699,92	94,03	979,11	1223,37	81,75
Mean	668,32	90,01	0,90	495,06	700,67	94,13	961,92	1209,89	82,49
SD	10,07	15,96	0,40	6,20	14,64	0,27	48,13	42,68	1,27

Table 9
Error metrics of Stacked XGB-LGBM-MLP for the ISO-NE dataset.

	Stacked XGB-LGBM-MLP model		
Fold number	MAE(MW)	RMSE(MW)	R^2(%)
0	312,39	439,58	97,66
1	310,47	443,55	97,68
2	301,60	421,25	97,89
3	306,20	470,11	97,37
4	307,31	458,15	97,49
5	311,68	434,53	97,87
6	300,63	422,21	97,87
7	301,56	421,81	97,79
8	301,18	414,26	98,01
9	298,03	430,46	97,74
Mean	305,01	435,59	97,73
SD	5,00	16,84	0,18

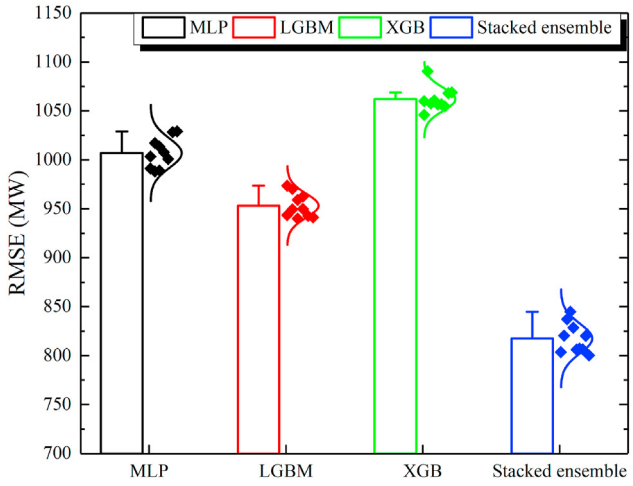


Fig. 13. Box-and-Whisker plot of RMSE errors for STLF models.

using two real datasets. According to multiple case studies, it can be concluded that the large data enhances the learning potential of the proposed technique. The main conclusions of this paper are summarized as follows:

1) The proposed technique is perfectly tailored for STLF achieving a low Rooted Mean squared Error RMSE = 1509.74 MW for a 24-h ahead.
2) The proposed technique outperforms 11 recent benchmarks from a fair assessment based on a forecasting horizon of a 24-h ahead. The Stacked XGB-LGBM-MLP can successfully reduce the error loss to a 192,96 MW compared to the best FTS-CNN model.
3) The comparative study between HO techniques for the load data demonstrate that meta-heuristics significantly speed up the search process especially for evolution strategy and simulated annealing. Nevertheless, Random Search (RS) and Bayesian optimization methods outperforms the rest of HO techniques in terms of R^2. This is explained by the fact that RS method enriches the search space to test further cases. Thus, RS method can reach to the optimum solution but with longer time-consuming and heavy computational burden.

However, the proposed technique is sensible to two elements: the forecasting horizon and the size of the data. The performance of the Stacked XGB-LGBM-MLP model decreases for 48-h ahead forecasting. The future work will include testing deep learning architectures in the meta-learning stage and supplementing the data representation by adding other alternative features such as customers behavior to further enhance the model accuracy.

Credit author statement

Mohamed Massaoudi: Conceptualization, Methodology, Software, original draft preparation. **Shady S. Refaat:** Validation, writing—review & editing, project administration, funding acquisition. **Ines Chihi:** Validation, formal analysis, Data curation, Writing – review & editing. **Mohamed Trabelsi:** Formal analysis, investigation, writing—review & editing, resources. **Fakhreddine S. Oueslati:** Supervision, project administration. **Haitham Abu-rub:** Proofreading & editing, review & editing.

Declaration of competing interest

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

Acknowledgements

This publication was made possible by NPRP grant [NPRP10-0101-170082] from the Qatar National Research Fund (a member of Qatar Foundation), the co-funding by IBERDROLA QSTP LLC and sponsorship by Texas A&M Energy Institute Fellowship. The statements made herein are solely the responsibility of the authors.

References

[1] Faheem M, Shah S, Butt R, Raza B, Anwar M, Achraf M, Ngadi M, Gungor V. Smart grid communication and information technologies in the perspective of Industry 4.0: opportunities and challenges. Comput Sci Rev 2018;30:1–30.
[2] Esteves GR, Bastos BQ, Cyrino FL, Calili RF, Souza RC. Long term electricity forecast: a systematic review. 2015. https://doi.org/10.1016/j.procs.2015.07.041.
[3] Hernandez L, Baladron C, Aguiar JM, Carro B, Sanchez-Esguevillas AJ, Lloret J, Massana J. A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. IEEE Commun Surv Tutorials 2014;16(3):1460–95. https://doi.org/10.1109/SURV.2014.032014.00094. ISSN 1553877X.
[4] Yildiz B, Bilbao JI, Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev 2017;73(March 2016):1104–22. https://doi.org/10.1016/j.rser.2017.02.023. ISSN 18790690.
[5] van der Meer DW, Widén J, Munkhammar J. Review on probabilistic forecasting of photovoltaic power production and electricity consumption. Renew Sustain Energy Rev 2018;81(May 2017):1484–512. https://doi.org/10.1016/j.rser.2017.05.212. ISSN 18790690.
[6] Chen K, Chen K, Wang Q, He Z, Hu J, He J. Short-term load forecasting with deep residual networks. IEEE Trans Smart Grid 2019;10(4):3943–52. https://doi.org/10.1109/TSG.2018.2844307. ISSN 19493053.
[7] Quan H, Srinivasan D, Khosravi A. Short-term load and wind power forecasting using neural network-based prediction intervals. IEEE Trans Neural Netw Learn Syst 2014;25(2):303–15. https://doi.org/10.1109/TNNLS.2013.2276053. ISSN 2162237X.
[8] Wanqing S, Chen X, Cattani C, Zio E. Multifractional Brownian motion and quantum-behaved partial Swarm optimization for bearing degradation forecasting. Energy; 2020. https://doi.org/10.1155/2020/8543131. ISSN 10990526.
[9] N. Zhang, Z. Li, X. Zou, S. M. Quiring, Comparison of three short-term load forecast models in Southern California, Energy 189, ISSN 03605442, doi: 10.1016/j.energy.2019.116358.
[10] Sigauke C, Chikobvu D. Peak electricity demand forecasting using time series regression models: an application to South African data. J Stat Manag Syst 2016;19(4):567–86. https://doi.org/10.1080/09720510.2015.1086146. ISSN 0972-0510, https://www.tandfonline.com/doi/full/10.1080/09720510.2015.1086146.
[11] Y. Yang, W. Hong, S. Li, Deep ensemble learning based probabilistic load forecasting in smart grids, Energy 189, ISSN 03605442, doi:10.1016/j.energy.2019.116324.
[12] Zhou S-M, Gan JQ. Low-level interpretability and high-level interpretability: a unifiedview of data-driven interpretable fuzzy system modelling. Fuzzy Set Syst 2008;159:3091–131.
[13] Jafari A, Khalili T, Babaei E, Bidram A. A hybrid optimization technique using exchange market and genetic algorithms. IEEE Access 2020;8:2417–27. https://doi.org/10.1109/ACCESS.2019.2962153. ISSN 21693536.
[14] Khalili T, Jafari A, Abapour M, Mohammadi-Ivatloo B. Optimal battery technology selection and incentive-based demand response program utilization for reliability improvement of an insular microgrid. ISSN 03605442 Energy 2019;169:92–104. https://doi.org/10.1016/j.energy.2018.12.024.
[15] Tsai CW, Hsia CH, Yang SJ, Liu SJ, Fang ZY. Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. ISSN 15684946 Appl Soft Comput J 2020;88:106068. https://doi.org/10.1016/j.asoc.2020.106068.
[16] M. Pirhooshyaran, L. V.Snyder, Forecasting, hindcasting and feature selection of ocean waves via recurrent and sequence-to-sequence networks, Ocean Eng 207 (107424).
[17] Pirhooshyaran M, Scheinberg K, Snyder LV. Feature engineering and forecasting via derivative-free optimization and ensemble of sequence-to-sequence networks with applications in renewable energy. ISSN 03605442 Energy 2020;196:117136. https://doi.org/10.1016/j.energy.2020.117136.
[18] Liao X, Cao N, Li M, Kang X. Research on short-term load forecasting using XGBoost based on similar days. In: Proceedings - 2019 international conference on intelligent transportation, big data and smart city, vol. 2019. ICITBS; 2019. p. 675–8. https://doi.org/10.1109/ICITBS.2019.00167.
[19] Ju Y, Sun G, Chen Q, Zhang M, Zhu H, Rehman MU. A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. IEEE Access 2019a;7(c):28309–18. https://doi.org/

10.1109/ACCESS.2019.2901920. ISSN 21693536.
[20] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y. Lightgbm: a highly efficient gradient boosting decision tree. In: Advances in neural information processing systems; 2017. p. 3146–54.
[21] Ju Y, Sun G, Chen Q, Zhang M, Zhu H, Rehman MU. A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. IEEE Access 2019b;7:28309–18. https://doi.org/10.1109/ACCESS.2019.2901920. ISSN 21693536.
[22] Meng Q, Ke G, Wang T, Chen W, Ye Q, Ma ZM, Liu TY. A communication-efficient parallel algorithm for decision tree. Adv Neural Inf Process Syst 2016:1279–87. ISSN 10495258.
[23] Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining; 2016. p. 785–94.
[24] Raza MQ, Khosravi A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. ISSN 18790690 Renew Sustain Energy Rev 2015;50:1352–72. https://doi.org/10.1016/j.rser.2015.04.065.
[25] Hong T, Fan S. Probabilistic electric load forecasting: a tutorial review. 2016. https://doi.org/10.1016/j.ijforecast.2015.11.011.
[26] He T, Dong Z, Meng K, Wang H, Oh Y. Accelerating multi-layer perceptron based short term demand forecasting using graphics processing units. In: 2009 transmission & distribution conference & exposition: Asia and Pacific. IEEE; 2009. p. 1–4.
[27] Wolpert DH. Stacked generalization. Neural Network 1992;5(2):241–59. https://doi.org/10.1016/S0893-6080(05)80023-1. ISSN 08936080.
[28] Ma Z, Dai Q. Selected an stacking ELMs for time series prediction. Neural Process Lett 2016;44(3):831–56. https://doi.org/10.1007/s11063-016-9499-9. ISSN 1573773X.
[29] Luo X, Sun J, Wang L, Wang W, Zhao W, Wu J, Wang JH, Zhang Z. Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy. IEEE Trans Ind Inf 2018;14(11):4963–71. https://doi.org/10.1109/TII.2018.2854549. ISSN 15513203.
[30] Efendi R, Ismail Z, Deris MM. A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand. 2015. https://doi.org/10.1016/j.asoc.2014.11.043.
[31] Carneiro T, Medeiros RV, Nóbrega DA, Nepomuceno T, Bian G-b, Albuquerque VHCDE. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access; 2018. p. 1–9. https://doi.org/10.1109/ACCESS.2018.2874767.
[32] Simon Blanke, Hyperactive. A hyperparameter optimization and meta-learning toolbox for machine-/deep-learning models. 2019. since, https://github.com/SimonBlanke.
[33] Khalid R, Javaid N. A survey on hyperparameters optimization algorithms of forecasting models in smart grid. ISSN 22106707 Sustain Cities Soc 2020;61(June):102275. https://doi.org/10.1016/j.scs.2020.102275.
[34] Nguyen V. Bayesian optimization for accelerating hyper-parameter tuning. In: Proceedings - IEEE 2nd international conference on artificial intelligence and knowledge engineering, vol. 2019. AIKE; 2019. p. 302–5. https://doi.org/10.1109/AIKE.2019.00060.
[35] Choi K, Jang D-h, Kang S-i, Lee J-h, Chung T-k, Kim H-s. Evolution strategy for antenna design. IEEE Trans Magn 2016;52(3):3–6.
[36] Varoquaux G, Buitinck L, Louppe G, Grisel O, Pedregosa F, Mueller A. Scikit-learn. GetMobile: Mobile Comput Commun 2015;19(1):29–33. https://doi.org/10.1145/2786984.2786995. ISSN 23750529.
[37] Ali Moez. Home - PyCaret. 2020. https://pycaret.org/.
[38] Ivanov I. vecstack: Python package for stacking (machine learning technique) [????], https://github.com/vecxoz/vecstack.
[39] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in Python fabian. J Mach Learn Res 2011;12:128–54. https://doi.org/10.4018/978-1-5225-9902-9.ch008.
[40] Seabold S, Perktold J. Statsmodels: econometric and statistical modeling with Python. In: Proceedings of the 9th python in science conference (Scipy), vol. 57; 2010. http://statsmodels.sourceforge.net/.
[41] Sadaei HJ, de Lima e Silva PC, Guimarães FG, Lee MH. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. 2019. https://doi.org/10.1016/j.energy.2019.03.081.