Optimised Extreme Gradient Boosting Model for Short Term Electric Load Forecasting: Application to the French Regional Grid

ZHAO Qinghe
Northeast Agricultural University
XIANG Wen
Northeast Agricultural University
HUANG Boyan
Northeast Agricultural University
Jong WANG
Northeast Agricultural University
Junlong FANG (jlfang@neau.edu.cn)
Northeast Agricultural University

Article

Keywords:

Posted Date: May 17th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1621432/v1

Optimised Extreme Gradient Boosting Model for Short Term Electric Load Forecasting: Application to the French Regional Grid

ZHAO Qinghe[1], XIANG Wen[1,2], HUANG Boyan[1],WANG Jong[1], FANG Junlong[1]*
[1]Electrical Engineering and Information College, Northeast Agricultural University
[2]Economic and Technological Research Institute of State Grid Heilongjiang Electric Power Co., LTD,

* Corresponding authors: FANG Junlong (jlfang@neau.edu.cn);

Abstract

Load forecast provides effective and reliable guidance for power construction and grid operation. It is essential for the power utility to forecast the exact in-future coming energy demand. Advanced machine learning methods can support competently for load forecasting, and extreme gradient boosting is an algorithm with great research potential. We take the extreme gradient boosting algorithm as the original model and combine the Tree-structured Parzen Estimator method to design the TPE-XGBoost model for completing the high-performance single-lag power load forecasting task. We resample the power load data of the Île-de-France Region Grid provided by Réseau de Transport d'Électricité in the day, train and optimise the TPE-XGBoost model by samples from 2016 to 2018, and test and evaluate in samples of 2019. The optimal window width of the time series data is determined in this study through Discrete Fourier Transform and Pearson Correlation Coefficient Methods, and five additional date features are introduced to complete feature engineering. By 500 iterations, TPE optimisation ensures nine hyperparameters' values of XGBoost and improves the models obviously. In the dataset of 2019, the TPE-XGBoost model we designed has an excellent performance of MAE=166.020 and MAPE=2.61%. Compared with the original model, the two metrics are respectively improved by 14.23% and 14.14%; compared with the other eight machine learning algorithms, the model performs with the best metrics as well.

Introduction

Load forecasting is a technique used by the energy-providing utility to predict the electrical power needed to meet the demand and supply equilibrium[1]. The technique can provide a reference for the daily operation of regional power grids and the formulation of dispatching plans. According to the results of power load forecasting, dispatchers can reasonably coordinate the distribution of the output of each power plant, maintain a balance between supply and demand, and ensure power grid stability. This determines the start-stop arrangement of the generator set, reduces the redundant generator reserve capacity value, and reduces the power generation cost[2,3]. Time series forecasting with the Machine Learning technique is the application of a model to predict future values through experience and by the use of previously observed values automatically. In recent years, power load forecasting combining machine learning methods, as a special sequence with stable data sources from grid operators or energy utilities, has broad research prospects, and it is also an essential part of the Data-Driven Smart Energy Assessment[4].

Gradient boosting is a state-of-the-art Machine learning algorithm. The Extreme Gradient Boosting is an important applicated popular algorithm developed by Tianqi in 2014. And because of its excellent performance on regression and classification problems, it is recommended as the first choice in many cases, such as industry and the Internet applications, which is even implemented in machine learning platforms. In time series forecasting, the research results using the Extreme Gradient Boosting solution in recent years are also very rich, including metal manufacture process[5], crowd flow prediction[6], lithology prediction[7], weather forecast[8], transportation system[9] and other related research in various fields. Although Extreme Gradient Boosting is one of the most widely used effective models, there are still many challenges in applying it for load forecasting. First, the Extreme Gradient Boosting algorithm relies on many hyperparameters to tune during the model building, and the reasonable hyperparameters directly determine the final prediction effect of the model. For the reason that an optimisation algorithm that can balance both data characteristics and model characteristics is very vital[10,11]. Secondly, when transforming time series into a general supervised regression problem in machine learning, it is complex to construct the data to have both historical memories and ensure the model has sufficient generalization ability after training[12,13]. The Two issues above are the key to combining Extreme Gradient Boosting even for all Machine learning algorithms with load forecasting tasks or time series.

In this study, (1) we completed data exploration for regional power grid consumption demand load data in the Ile-de-France region of France, ensured the best width of the sliding window for the machine learning model by the Discrete Fourier Transform and Pearson Correlation Coefficient methods, and added 5 date features in the dataset for feature engineering work. (2) We designed the TPE-XGBoost algorithm by combining the Tree-structured Parzen Estimator method and the Extreme Gradient Boosting model. By comparing with the original unoptimised model and other 8 machine learning algorithms, our proposed model can effectively improve the prediction performance for power demand load forecasting in the individual testing dataset. (3) We conducted a model evaluation on the TPE-XGBoost model we designed and discussed in detail the feature engineering of the dataset and the modelling effect of the TPE optimisation for the XGBoost model.

Material and Methods

Loading forecasting dataset and data exploring

Île-de-France (literally "Isle of France") is one of the 13 administrative regions in mainland France and the capital circle of Paris. The average temperature is 11°C, and the average precipitation is 600 mm. Île-de-France is the most densely populated region of France. According to the 2019 report, this region provides France with a quarter of jobs in total employment, of which the tertiary sector accounts for near nine-tenths of jobs. Agriculture, forests and natural areas cover nearly 80% of the surface. As well, the region, as the first industrial zone in France, includes electronics and ICT, aviation, biotechnology, finance, mobility, automotive, pharmaceuticals and aerospace.

We analysed the power load in the Île-de-France region with a 30-minute sampling rate with a total of 70128 records over four years, from 2016 to 2019. The data is from the éCO2mix API provided by the RTE (Réseau de Transport d'Électricité). The original data were resampled as the maximum daily power into 1461 records in Figure 1, whose y-axis is the real-time demand load power (unit: MWatt). As shown below, the trend between the years of the series is similar and has an evident periodicity; each cycle is V-shaped with visible seasonality. Due to the characteristics of power load and the region's actual situation, each cycle's trend is stable without an apparent growth or decay.

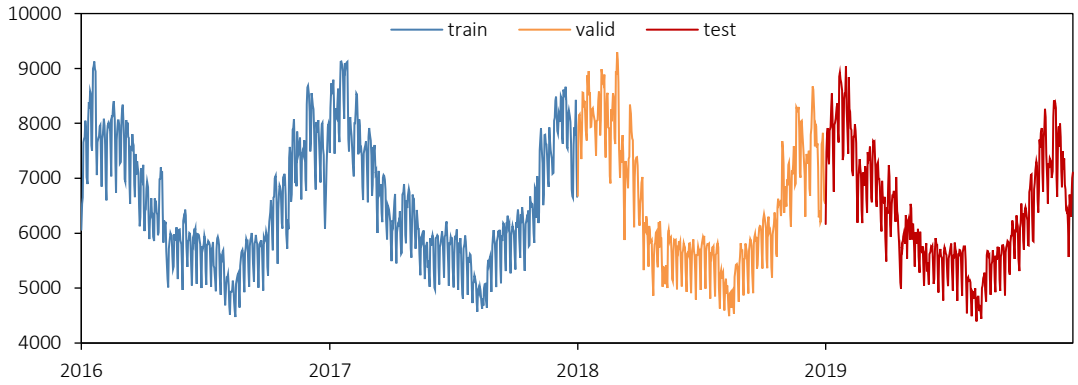


Figure 1 Demand consumption of electrical power (daily, MWatt)

The dataset we collected from éCO2mix is divided into three parts, the blue training dataset (2016, 2017), the orange validation dataset (2018) and the red testing dataset (2019)[14] in Figure 1. The training one builds the main models, and the validation one is analysed for optimisation eval. And testing one will check the models' performance on several different metrics. The data exploring part of the time series will be finished in the validation one in 2018 to avoid data leakage.

In the following Table 1, we use feature engineering to transform time series into a supervised learning dataset for machine learning as the additional date feature[15].

Table 1 The date features added in the dataset

Feature index	Data type	Description
date_feature_1	integer	Day of the week (Monday=0 and Sunday=6)
date_feature_2	integer	Day of the month (from 1 to 28~31)
date_feature_3	integer	the day of the year (from 1 to 365/366)
date_feature_4	integer	the week of the year (from 1 to 53/54)
date_feature_5	integer	the month this year falls on (from 1 to 12)

Finally, loading forecast values at the first N moments will be added to the dataset as a memory feature in the form of a sliding window called momery_feature_1~momery_feature_N. However, the choice of N, the memory length or the time lag is not casual. We will use a method combining Discrete Fourier Transform and Pearson Correlation Coefficient to complete the memory length determination.

The Best width of windows analysis

Data-driven loads forecasting issues of machine learning require the datasets to be produced in the form of sliding windows. Then, the time series issue transforms into a supervised regression in machine learning. And there is a complex effect on the window width or called lags count of the dataset. The longer width of the window, the more abundant the memory information as more features in the sample. However, for the machine learning algorithms based on statistics experience, more features would cause unideal results for practical application by too many irrelevant features. On the other hand, too short a window means fewer features, which might be underfitting for insufficient information.

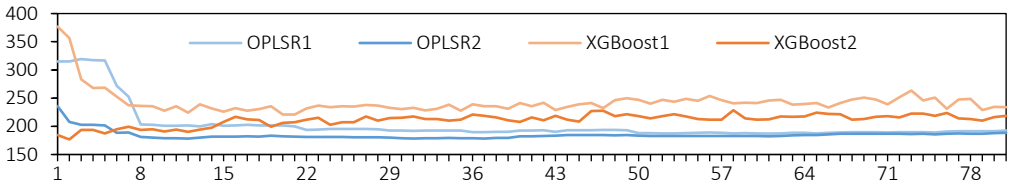


Figure 2 MAE metrics with wider windows of testing dataset

Figure 2 above is the effect of different window widths in the testing dataset of the XGBoost model and Linear Regression (OPLS), whose x-axis is the window's width of data features and the y-axis is the mean absolute error (units: MWatt) in the testing dataset (2019), and the models indexed 1 mean no date features adding. It can be seen that the relationship between the performance and window width is not a simple linear relationship. This figure shows a dramatic decline in MAE with wider windows, it reached a low point, and then the MAE fluctuates within a specific range and worsens when the windows widen.

The Fourier Transform is a practical tool for extracting frequency or periodic components in signal analysis. Generally, the synthetic signal $f(t)$ can be converted to frequency domain component signals $g(freq)$ as below if it satisfies the Dirichlet conditions in the range of $(-\infty, +\infty)$:

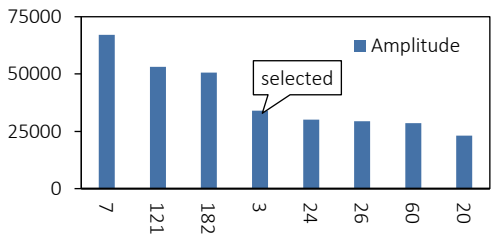
$$g(freq) = \int_{-\infty}^{+\infty} f(t) \cdot e^{-2\pi i \cdot freq \cdot t} d_t$$

the power loads time series in this paper are sampled discretely with limited length, and the Fast Discrete Fourier method proposed by Bluestein[16] is used instead as below:

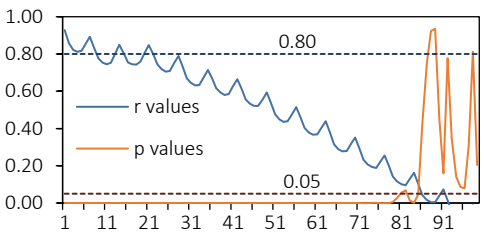
$$g(freq) = \sum_{t=1}^{N} \left[f(t) \cdot e^{-2\pi i \cdot freq \cdot \frac{t}{N}} \right]$$

where $N = 365$ is from the validation dataset in 2018, and the $freq$ series contain the frequency bin centers in cycles per unit of the sample spacing with zero at the start. The second half of $freq$ series is the conjugate of the first half, only the positive is saved. And bring $period = \frac{1}{freq}$ back as below:

$$g(period) = \sum_{t=1}^{N} \left[f(t) \cdot e^{-2\pi i \frac{t}{period \cdot N}} \right]$$



(a)DFT analysis for series' cycle-period



(b) Pearson's Correlation Coefficient

Figure 3 Features engineering of windows width analysis

Remove the inf and the $period = N$ item $g(period)$, and the first eight amplitude of $period - g(period)$ bar plots as shown in Figure 3**Error! Reference source not found.**(a) [17]. Some of the periods are related to the natural time cycle: 121 as a quarter of a year, 182 as a semi-annual. And not all of the meaning is clear, such as 3, 24 and 26, which are difficult to have sufficient explanation.

Further, the Pearson correlation coefficient (PCCs) are used to calculate a more detailed period. The PCCs measure the linear relationship between two datasets as below:

$$PCC(\boldsymbol{x}(t_0), \delta) = \frac{\sum\left(\boldsymbol{x}(t_0 - \delta) - \overline{\boldsymbol{x}(t_0 - \delta)}\right)\left(\boldsymbol{x}(t_0) - \overline{\boldsymbol{x}(t_0)}\right)}{\sqrt{\sum\left(\boldsymbol{x}(t_0) - \overline{\boldsymbol{x}(t_0)}\right)^2 \cdot \sum\left(\boldsymbol{x}(t_0) - \overline{\boldsymbol{x}(t_0)}\right)^2}}$$

where the $\boldsymbol{x}(t_0)$ is the series to predict as y target of dataset and the $\boldsymbol{x}(t_0 - \delta)$ is the δ lag series of t_0. The larger PCC means there are more correlated relationships between two series.

In Figure 3(b), the X-axis is the time interval numbers and the Y-axis is the Pearson's Correlation Coefficient values (blue) and Two-tailed p-value (orange). According to experience in the general statistical sense[18], when the Pearson Coefficient is greater than 0.80 (blue dotted line), it can be considered that the two series have a strong correlation; when the p-value is less than 0.05 (orange dotted line), the hypothesis is established. The orange curve shows that the about first 50 memory features have a positive correlation with the predicted target, so the minimum period should be less than 50. Furthermore, Memory-feature 1~5 have Pearson's correlation coefficient values greater than 0.80; that is, the values within 5 are strongly correlated. And the number of periods in the FFT to satisfy this value requirement is 3, therefore, our model will use 3 as the window width.

We will further compare the three kinds of widths, 7, 14 and 28, as a control to complete the sequential modelling.

Extreme Gradient Boosting Optimised by Tree-structured Parzen Estimator

Gradient Boosting originates from the paper by Friedman in 2011[19]. XGBoost is an open-source software library of extreme gradient boosting developed by CHEN Tianqi[20] that ensembles tree models by a series of strategies and algorithms such as a greedy search strategy based on gradient boosting. As an additive ensemble model, XGBoost considers the gradient of first-order derivative and second-order derivative in the Taylor series for the loss function and constructs in the case of probability approximately correct (PAC). The objective function is as follows:

$$obj(t) = \sum_{i=1}^{n} loss\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_i)$$

where the t means the rounds of ensemble processing and the ω means the regularisation part.

Take a second-order Taylor expansion on the loss function and add the parameters of the tree structure in the regular term, Then the objective function transforms into below:

$$obj(t) = \sum_{i=1}^{n} \left[g_i^t \cdot f_t(x_i) + \frac{1}{2} \cdot h_i^t \cdot f_t(x_i)^2 \right] + \gamma \cdot T + \frac{1}{2} \cdot \lambda \cdot \sum_{j=1}^{T} \omega_j^2$$

where the g and the h is the derivative term of the loss function; the T and ω are the parameters of ensembled decision trees' structure parameters; γ is the minimum loss required for further partitioning on the leaf nodes of single tree; λ is the L2 regularization term.

A greedy strategy to solves the $obj(t)$ for a local optimal solution $\omega = -\frac{G}{\lambda + \mathrm{H}}$ then Bring back:

$$best_{obj(t)} = -\frac{1}{2} \cdot \sum_{j=1}^{T} \left(\frac{G_j^2}{H_j + \lambda} \right) + \gamma \cdot T$$

With the meta, weak learner t generated in each round, $best_{obj(t)}$ is used as the basement strategy for the growth of the decision tree, which controls the generalisation ability for the boosting process.

Most specific detail for XGBoost can refer to the paper, *XGBoost: A scalable tree boosting system*[20].

XGBoost is a powerful ensemble algorithm, and there are numerous hyper-parameters to tune for the best performance in application, however, it is a black-box process widely recognised[21]. We adopt the Tree-structured Parzen Estimator[22] (TPE), one of the sequential model-based optimisation methods (SMBO) based on the Bayesian theorem, to optimise our XGBoost model for time-series forecasting. The TPE pseudocode as below shown in Figure 4:

Algorithm. TPE algorithm (with XGBoost)
1: Initialization $H_0 = \emptyset$ and $\boldsymbol{z} = \boldsymbol{z}_0$
2: **for** $i = 1\ to\ I_{max}$ **do**
3: $\boldsymbol{z} * = argmin(EI_i(k, \boldsymbol{z}_i[H_{i-1}]))$
4: XGB modelling and validation for s_i
5: Update $H_i = H_{i-1} \cup < EI_i(k), s_i >$
6: **end for**
7: **Return** $argmin_{\boldsymbol{z}}\left[s\left(\mathrm{xgb}(\mathbf{z}, \mathbf{X})\right)\right]$

Figure 4 Pseudocode of Tree-structured Parzen Estimator optimising XGBoost

Where the $\boldsymbol{z}$ is the set of hyperparameters of the search space, the s is the metrics score of XGBoost with $\boldsymbol{z}$ in the validation dataset, and the H is the history of validation scores and the selected $\boldsymbol{z}$.

The EI is the core of TPE, which builds a probability model of the objective function and uses it to select the most promising hyperparameters to evaluate in the true objective function:

$$EI = \frac{\int_{-\infty}^{+\infty} \max(\boldsymbol{s} * -\boldsymbol{s}[H_{i-1}], 0)\, p(\boldsymbol{s}[H_{i-1}])ds}{k + \frac{(1-k)g(\boldsymbol{z}[H_{i-1}])}{l(\boldsymbol{z}[H_{i-1}])}}$$

the $l(\boldsymbol{z})$ is the value of the objective function less than the threshold k, and $g(\boldsymbol{z})$ is the objective function greater than the threshold.

Table 2 Hyperparameters to be optimised of XGBoost in paper

Hyperparameter to tuning	Data type	Default value	Searching space
Alpha	Float	0	0.01 to 1.00
Learning rate	Float	0.3	0.01 to 0.20
Max depth of single tree	Integer	6	2 to 5
Minimized child weight	Float	1	0.50 to 0.60
Minimized split loss	Float	0	1e-10 to 1.00
Subsample	Float	1	0.90 to 1.00
Col sample by tree/level/node	Float	1	0.90 to 1.00

We first build the XGBoost model (XGBoost) by default values in Table 2, and then nine hyper-parameters in Table 2 are going to be optimized in the searching space below by the TPE algorithm for TPE-XGBoost models. We limit the tuning iterations to 500 and the target of each iteration will be set of the MAPE in the validation dataset (2018).

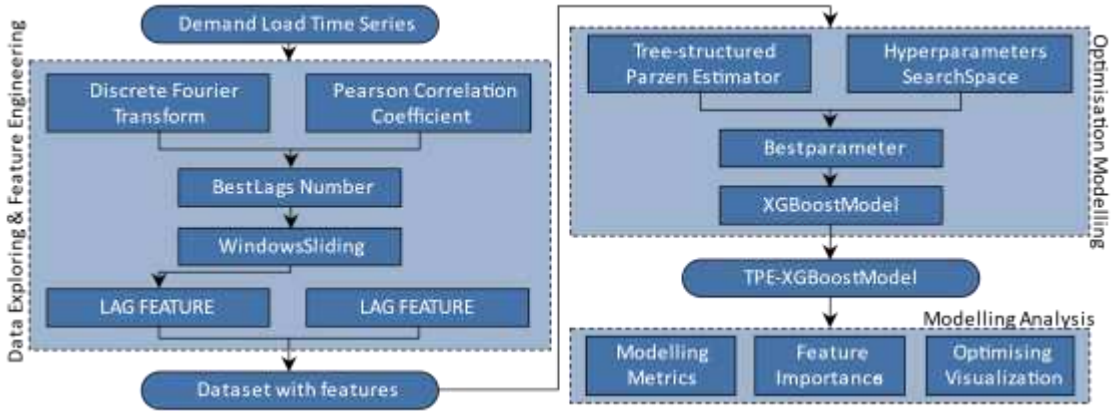


Figure 5 Feature Engineering and Optimised Modelling Process

Results

XGBoost and TPE-XGBoost we recommended have been modelled in four kinds of windows width data: 3d, 7d, 14d and 28d in train dataset (2016,2017). In addition to this, eight other below machine learning algorithms have been conducted as comparative experiments on at the same time, including:

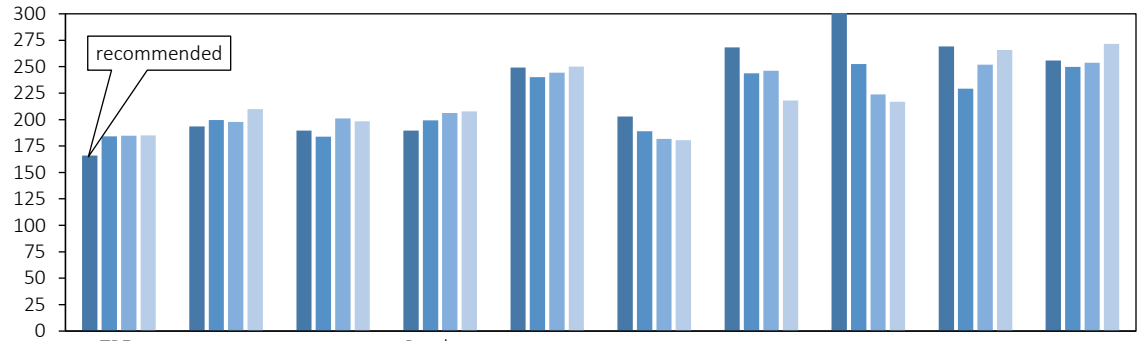
- Three ensemble models: Gradient Boosting Decision Tree models (*pGBDT*) and Adaptive Boost models (Adaboost) based on scikit-learn; Random Forest models (*RandomForest*) based on XGBoost.
- One linear model: built by Ordinary Least Squares Method (*OPLSR*).
- One Support Vector Machine model: based on libsvm algorithm with a Radial basis function kernel (*RBF-SVR*).
- One Neural Network model: Perceptron model (*Perceptron*) with triple hidden layers shaped (256,128,64) built by scikit-learn framework.
- One Neighbours Model: K-Nearest Neighbours model (*KNN*) with Euclidean distance metrics.
- Single Decision Tree model: Tree (*SingleTree*) model built by scikit-learn framework without max depth limit as the contrast of ensemble models.

Prediction models of testing dataset

Figure 6 shows mean absolute error values (MAE values) respectively, where the top model valued at 166.02 is the XGBoost optimised by the TPE algorithm with data wide of 3d.

Obviously, TPE method does improve XGBoost performance. All MAE metrics of four XGBoost models trained with different windows width data are apparent to improve after being optimised by the TPE algorithm. They decrease from 193.57, 199.46, 197.82, 209.93 to 166.02, 184.11, 184.65, 185.09 respectively, whose optimization achieves 14.23%, 7.70%, 6.66%, and 11.83%.

The MAEs of Five ensemble learning methods (TPE-XGBoost, XGBoost, pGBDT, Random Forest and Adaboost) get a slight rise with longer windows. This proves from the side that proper selection of window width is vital for such ensemble learning models to predict time series correctly. However, OPLSR, Perceptron, and RBF-SVR models have the opposite trend after training with more previous features.



	TPE-XGBoost	XGBoost	pGBDT	Random Forest	Adaboost	OPLSR	Perceptron	RBF-SVR	kNN	SingleTree
3d	166.020	193.570	189.551	189.701	249.247	202.800	268.280	303.880	269.076	255.827
7d	184.116	199.463	183.865	199.293	240.083	188.983	243.604	252.430	229.124	249.655
14d	184.650	197.824	201.201	206.144	244.429	181.718	246.142	223.655	251.880	253.611
28d	185.090	209.931	198.254	207.796	250.189	180.468	218.015	216.951	265.896	271.688

Figure 6 Mean Absolute Error values (MAE values)

But as Figure 2 shown, this trend would reach a limit value not as good as TPE-XGBoost models' scores and then it will begin fluctuating in ranges. The best MAE during the 1 to 81 is 178.539 (width = 31), which gaps obviously with our TPE-XGBoost with 166.02. We will discuss in-depth for this phenomenon in the next part of our paper.

Another three metrics (MAPE, R2 and MaxError) from the testing set are listed partly in Table 3. Shorter windows of our method can reach higher on two overall metrics: MAPE=2.61%, R2=0.9471. And the max residual metrics don't result in ideal results ranged 1183 to 1436. However, the 3d-TPE-XGBoost's max residual is still in an acceptable value on par with the best 14d-OPLSR model, scored 895, and the TPE process suppresses it compared to XGBoost not optimised to a certain extent in 7d, 14d and 21d data.

Table 3 Results metrics from testing dataset of models

		TPE XGBoost	XGBoost	Random Forest	OPLSR	Perceptron	RBF-SVR
MAPE	3d	**2.61%**	3.04%	2.99%	3.19%	4.19%	4.80%
	7d	2.88%	3.09%	3.16%	2.98%	3.82%	3.94%
	14d	2.90%	3.06%	3.26%	2.84%	3.77%	3.49%
	28d	2.90%	3.29%	3.28%	2.80%	3.36%	3.38%
R2 Score	3d	**0.9471**	0.9328	0.9355	0.9307	0.8885	0.8430
	7d	0.9389	0.9297	0.9317	0.9388	0.9052	0.8978
	14d	0.9413	0.9319	0.9271	0.9419	0.8983	0.9153
	28d	0.9387	0.9255	0.9261	0.9421	0.9198	0.9208
Maximum Residual	3d	**1183**	1174	1294	975	1506	1448
	7d	1436	1550	1305	979	1190	1125
	14d	1209	1249	1226	895	1450	1216
	28d	1257	1529	1230	919	1126	1082

Figure 7 is the sequence comparison figure of random selected 21-day predicted and real values among four seasons of 2019. The model of 3d we proposed can make excellent predictions of the periodic and frequency trend of the real time series, mutually confirmed by its better MAE, MAPE, and R2 metrics.

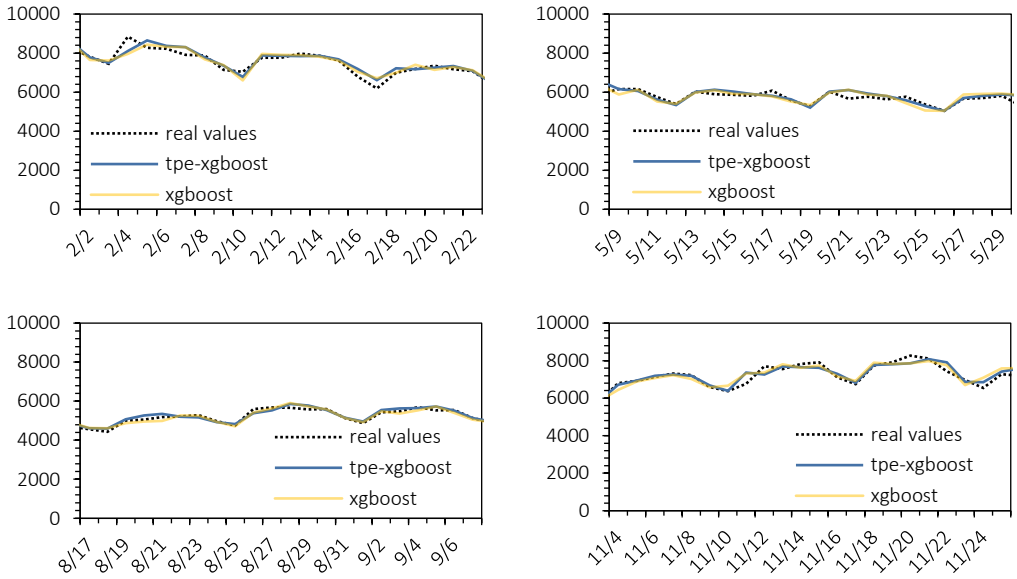


Figure 7 Forecasting and true values of testing dataset in 2019

The TPE-XGBoost model performs well before December. A suppressing fixing by TPE can be observed compared to the unoptimised XGBoost method, especially in January, April and June. The December series forecast is terrible. This month, the negative impact of almost all models is contributed by the max residual of 3d-TPE-XGBoost.

TPE optimisation processing for XGBoost models

The visualization process of nine hyperparameter tuned by the TPE algorithm is shown in **Error! Reference source not found.**. The object value for loop iterations is set to the MAPE (mean absolute percentage error) from the validation dataset in 2018, which is neither included in the training nor the testing dataset. After 500 iterations of learning, the model can gain an acceptable excellent target in the validation set of MAPE= 0.02647. With the iterations increasing, the validation target is gradually distributed to a tight range. The max depth of the trees in XGBoost is selected to 3 in a range from 2 to 5; the learning rate(eta) is around 0.11 from 0.02 to 0.2.

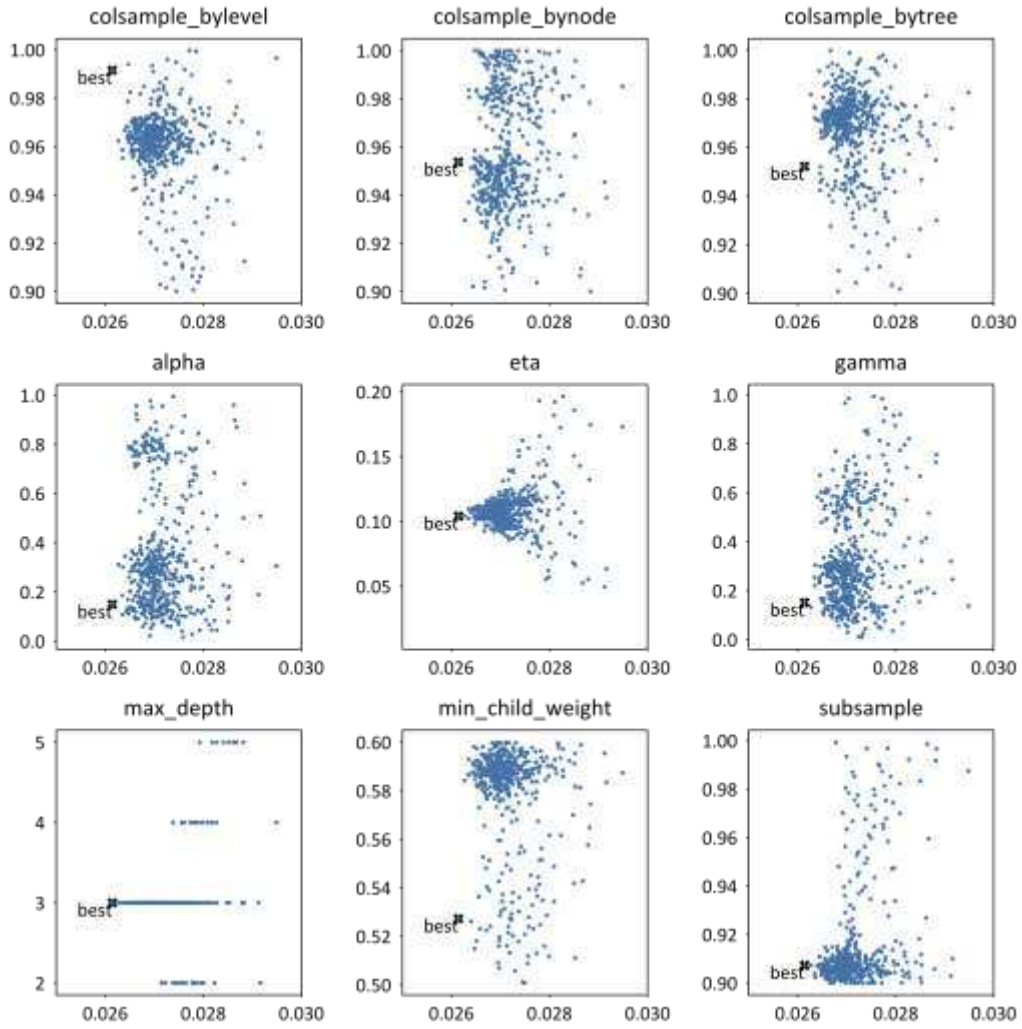


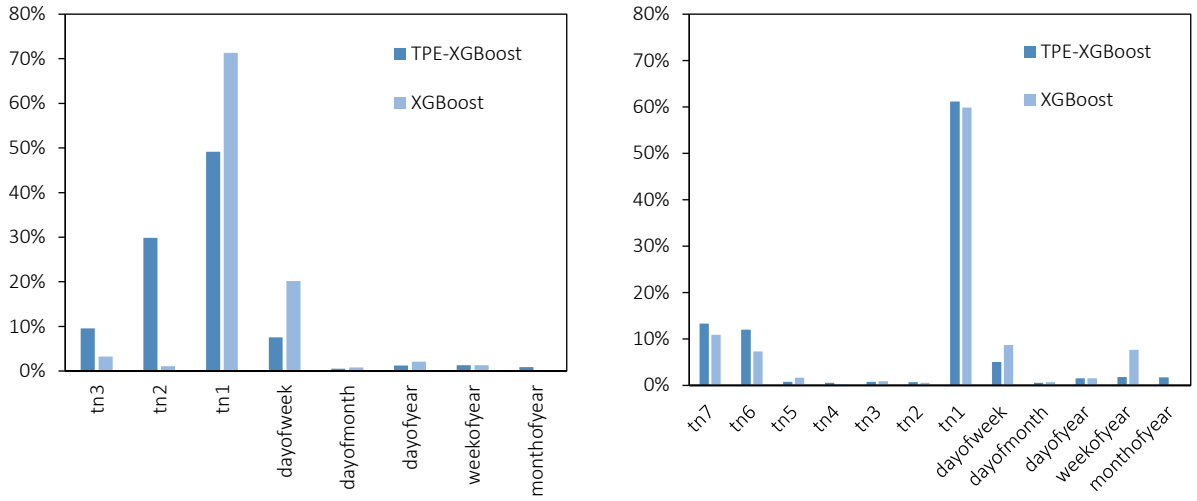
Figure 8 Nine Tuning hyperparameters with MAPE values

The best hyperparameter set appearing in the 499th iteration of 500 rounds is listed in Table 4. More searching rounds would gain a better MAPE of validation set but it needs more time to run.

Table 4 Best hyperparameters of XGBoost optimised by TPE in paper

eta	alpha	gamma	Subsample	Max depth
0.10411162943929	0.14962743583968	0.1513032788308	0.90727951205518	3
Min child weight	Colsample by level	Colsample by node	Colsample by tree	Iteration
0.52728463623425	0.99153864007020	0.95380256761024	0.95222870269938	499/500

As an ensemble algorithm based on the tree model, FI (feature importance rate) stands for the weights in the modelling of features in the dataset. Figure 9 shows the two different width series (3d we preferred and longer 7d) of TPE-XGBoost and the Unoptimized one. The models optimized have higher FI values of the features before tn1 time of 3d one: almost 40% FI values of tn2&tn3 but the unoptimized one's less than 10%. Wider windows models focus more on tn6 and tn7, and the TPE processing rises the rate of FI values of them and suppresses the contribution of the tn2~tn5.



(a) 3d window dataset (b) 7d window dataset

Figure 9 Feature importance percentages of XGBoost models

In summary, these results show that (1) fewer window features are capable to revert the power loads time series we are concerned about. The three-day width of windows analysed from FFT and Pearson correlation have enough information to do better than longer ones. (2) XGBoost as a practical and effective algorithm can achieve the forecasting task with fewer features, however, there is remarkable necessary to add optimising processing by the TPE method. Hyperparameters from TPE will get the most out of the performance of XGBoost with short windows.

Discussion

- The power load data has a clear time continuity. That is the load data will not change abruptly only in the case of extreme events (such as grid crashing, etc). This is the reason why linear regression (OPLSR) and the simplest model perform still well for the wider windows. The XGBoost method, even most machine learning methods, is based on historical data and does not have the concept of temporal continuity. We make sliding windows to provide the memory for them and so transformed time series problems into regression problems, trains and forecasts data will reference through this window feature. As for the TPE method, from the discussion of FI above, it can be seen that it suppresses the modelling weight of the near memory features, and increases the model's attention to farther ones. It is the immediate reason why the TPE method can improve modelling performance by hyperparameters controlling.

  We believe that it is necessary to use the minimum period as the window width, which is a targeted treatment of the continuity characteristics of time series or load forecasting data. The window with the shortest period includes at least the complete memory of the data of interest and does not contain redundant information of multiple periods. Although a wider window will provide richer historical information, the XGBoost algorithm's focus on data continuity will suffer, which is regulated to control the risk of generalization.

We believe that the main impact of adding the date features to the model is to ensure that the algorithm can have the ability to extract other periodicities. Time series data, including our load data, is of course highly cyclical, and the cycles it contains may be related to the real world with clear explanations. Monthly and weekly data are also cyclical, with stronger or weaker correlations between these cycles. Therefore, we have added five date features to the feature engineering. The five date features can help the algorithm to extract information from multiple cycles as much as possible. If there is no data feature, machine learning methods will maybe not perform effective fitting and prediction on periodic time series.

- TPE-XGBoost as mentioned above is a two-step process with modelling-then-validation, that is, given the hyperparameter search space of XGBoost, adjusted and optimized by TPE for finally taking the increase or decrease of a certain target objective value as the goal. In the optimization process, which kind of metrics to use and where the metrics are from are two crucial issues.
  Usually, the optimization target metrics are one of the metrics for evaluating the modelling. Such as MAE or RMSE, two reasonable targets can both describe the error value between predicted data and real data from a certain scale. However, the metrics for optimization are different from evaluating purposes. In our TPE-XGBoost algorithms, setting MAPE as the target value through the data modelling from 2016 to 2017 and predicted of true values of 2018 by 500 iterations, the MAPE can effectively improve MAE\RMSE\R2, etc. Other evaluation metrics can only improve their own performance in an independent test dataset. Other metrics are not unchanged but are less obvious than MAPE.
  In addition, the usual source of objective values is the k-folds Cross-validation of the training set itself, this way can maximize the use of existing sequences for modelling and evaluation when the data is not sufficient. However, this method, as literally stated, needs to use 5 times the calculation of single modelling for repeated fitting, and the obtained k-folds have great differences in the dataset in this paper no matter what metrics are used, resulting in a slow and ineffective optimization process. In fact, for the load forecasting request itself, the data is abundant, and even several years of historical data can be traced back at power grid operators, and there is no problem of insufficient data. In the machine learning method, if such a dataset is used for the validation of the model, it is necessary to ensure that the training data and the validation data should be independent and identically distributed, and our training dataset, validation dataset and test dataset, no doubt, are all in the form of there is actually real-world data, and the data itself is consistent, so our approach of using an independent validation set is correct.
  Therefore, as described in this paper's results and discussion, we propose to adopt MAPE metrics from a separate validation dataset as objective values in the optimization process of TPE-XGBoost.
- This paper does not consider the introduction of external variables, and only studies from the time series itself, but it also achieves reliable forecasting results. This is because the external variables such as temperature, wind speed and social practice commonly used in load forecasting problems can be replaced by the date feature we introduced. These external variables are also periodic and to be predicted, and the function of the date feature is to provide a calibration reference for the memory of the time series from another aspect. Such a calibration reference that has an independent and identical distribution in the dataset is more valuable than the actual wind speed and other data.

Data availability

All methods were carried out in accordance with the relevant guidelines and regulations. Data are available from the éCO2mix of Français Réseau de Transport d'Electricité(French language) website at https://www.rte-france.com/eco2mix , or get the copy of CSV file at https://github.com/gniqeh/TPE-XGB-TS by MIT License.

References

1 Hong, T., Wang, P. & Willis, H. L. in *2011 IEEE Power and Energy Society General Meeting.* 1-6 (IEEE).

2 Hong, T. & Fan, S. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting* **32**, 914-938 (2016).

3 Hong, T. & Wang, P. in *2012 IEEE Power and Energy Society General Meeting.* 1-3 (IEEE).

4 Ahmad, T., Madonski, R., Zhang, D., Huang, C. & Mujeeb, A. Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: key developments, challenges, and future research opportunities in the context of smart grid paradigm. *Renewable and Sustainable Energy Reviews* **160**, 112128 (2022).

5 Coelho, D., Costa, D., Rocha, E. M., Almeida, D. & Santos, J. P. Predictive maintenance on sensorized stamping presses by time series segmentation, anomaly detection, and classification algorithms. *Procedia Computer Science* **200**, 1184-1193, doi:10.1016/j.procs.2022.01.318 (2022).

6 Wu, C., Yin, T., Ge, S. & Yu, K. 103-113 (Springer International Publishing).

7 Gu, Y., Zhang, D., Lin, Y., Ruan, J. & Bao, Z. Data-driven lithology prediction for tight sandstone reservoirs based on new ensemble learning of conventional logs: A demonstration of a Yanchang member, Ordos Basin. *Journal of Petroleum Science and Engineering* **207**, doi:10.1016/j.petrol.2021.109292 (2021).

8 Mai, X., Zhong, H. & Li, L. 1313-1319 (Springer International Publishing).

9 Kalvapalli, S. P. K. & Chelliah, M. 341-348 (Springer Singapore).

10 Putatunda, S. & Rama, K. in *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning.* 6-10.

11 Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S. & Schmidt-Thieme, L. Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118* (2021).

12 Norwawi, N. M. in *Data Science for COVID-19* 547-564 (Elsevier, 2021).

13 Mozaffari, L., Mozaffari, A. & Azad, N. L. Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads. *Engineering Science and Technology, an International Journal* **18**, 150-162, doi:10.1016/j.jestch.2014.11.002 (2015).

14 d'Electricité, R. d. T. (ed https://www.rte-france.com/en/eco2mix) (2022).

15 Massaoudi, M. *et al.* A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. *Energy* **214**, doi:10.1016/j.energy.2020.118874 (2021).

16 Rao, K. R., Kim, D. N. & Hwang, J. J. *Fast Fourier transform: algorithms and applications*. Vol. 32 (Springer, 2010).

17 Puech, T., Boussard, M., D'Amato, A. & Millerand, G. in *International Workshop on Advanced Analysis and Learning on Temporal Data.* 43-54 (Springer).

18 Benesty, J., Chen, J., Huang, Y. & Cohen, I. in *Noise reduction in speech processing* 1-4 (Springer, 2009).

19 Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232 (2001).

20 Chen, T. & Guestrin, C. in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.* 785-794.

21 Turner, R. *et al.* in *NeurIPS 2020 Competition and Demonstration Track.* 3-26 (PMLR).

22 Ozaki, Y., Tanigaki, Y., Watanabe, S. & Onishi, M. in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference.* 533-541.