IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 July 2024, accepted 18 August 2024, date of publication 26 August 2024, date of current version 6 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3449875

RESEARCH ARTICLE

A Novel Ensemble Learning-Based Method for Online Total Transfer Capability Assessment of New Power System With Increased Stochastics

HUIMIN GAO[1,2], (Member, IEEE), YI LU[2], (Student Member, IEEE), RUISHENG DIAO[3], (Senior Member, IEEE), JING ZHANG[4], (Member, IEEE), ZHANNING LIU[3], (Student Member, IEEE), AND AO ZHAO[3]

[1]College of Information Engineering, Hangzhou Dianzi University, Hangzhou 311305, China
[2]College of Automation, Hangzhou Dianzi University, Hangzhou 310018, China
[3]The ZJU-UIUC Institute, Zhejiang University, Haining, Zhejiang 314400, China
[4]State Grid Zhejiang Electric Power Company, Hangzhou, Zhejiang 310000, China

Corresponding author: Ruisheng Diao (RuishengDiao@intl.zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U22B6007, and in part by Hangzhou Dianzi University Information Engineering College Science Research Project under Grant KYP0223003.

ABSTRACT With the ever-increasing penetration level of renewable energy and emerging electricity demand, it is very important to ensure the safe and reliable operation of the modern transmission system with strong stochastics and dynamics. Exceeding transmission limits will lead to line overloading, equipment damage, voltage instability, and even cascading failures. Therefore, it is of great significance to accurately evaluate the total transfer capability (TTC) in real-time. This paper proposes a novel ensemble model-based method to learn the nonlinear mapping between online operating conditions and TTCs of key transmission corridors. Massive representative operational samples are generated using a high-fidelity power system security analysis engine involving a variety of operating conditions, and thermal stability, static voltage security, and transient stability are considered. After feature extraction and label labeling of these samples, the dataset was split into training and test sets. The samples were used to predict TTC using three types of models: Light Gradient Boosting Machine (LGBM), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). The hyperparameters of these models were optimized through random search and 50% discounted cross-validation methods to enhance the accuracy of TTC estimation. Finally, the optimized models were combined using the stacking method. In this approach, LGBM, XGBoost, and RF served as base models, while LGBM functioned as the meta-model, leveraging the strengths of each model. The proposed method was validated on a 500-node network model with real-world operational characteristics, demonstrating improved TTC evaluation accuracy and higher efficiency.

INDEX TERMS Ensemble learning, total transfer capability, parameter optimization, power grid operation.

I. INTRODUCTION

In large-scale interconnected power grids, to guarantee the secure operation of the power system and optimize the utilization of transmission capacity, real-time evaluation of available transfer capability (ATC) and TTC becomes critical [1], [2]. The evaluation methods are divided into deterministic, probabilistic, optimization, and machine learning methods, where deterministic methods include continuous power flow, optimal power flow, repeated power flow, linear programming, and so on [3], [4], [5], [6].

In [3] and [4], ATC evaluation is based on the optimal power flow model. In [5], an improved repetitive AC power flow method (MRACPF) is proposed. In [6], the value of TTC is estimated by assessing the system's current state and evaluating the system's static or thermal stability index. However, the deterministic method can only be used for a fixed grid operating condition, which cannot reflect the system's dynamic characteristics over time. The uncertainties of load, power generation, and other system

The associate editor coordinating the review of this manuscript and approving it for publication was Nagesh Prabhu.

parameters are considered in probabilistic methods and optimization algorithms. In [7] and [8], an interval-based ATC evaluation model is proposed, where the objective function is divided into subproblems for the lower and upper boundaries. The assumptions about the upper and lower bounds of the interval determine the accuracy of this method. If these assumptions are not accurate, the evaluation results may deviate from the actual situation, thus limiting the practical application of the model. With respect to probabilistic methods, a data-driven method using the sparse polynomial chaos expansion is proposed in [9] to evaluate the TTC. Although the proposed sparse scheme can improve computational efficiency, calculating polynomial chaos expansion is still complicated. In [10], a fast differential equation power flow method minimizes the frequency of solving linear equations. Simultaneously, dynamic ATC is evaluated using the Newton-Raphson-Seydel method. In [11], a probabilistic method based on multi-objective clustering technology is proposed. Probability model analysis is done by subdivding a random set into smaller clusters. For the optimization algorithm in [12], a hybrid chaotic selfish herd optimizer method is proposed to evaluate ATC.

In [13], the square sum relaxation method is proposed for the solution of TTC. In [14], TTC computation of population-based bioinspired optimization techniques, but this algorithm does not always find the optimal solution. In [15], a hybrid method, which does not consider the fluctuation of wind power and load, is proposed by combining the improved Gray Wolf optimization algorithm and the primal-dual interior point method. Evaluating probabilistic methods and optimization algorithms requires complex simulations and calculations, making achieving real-time applications difficult. Many factors must be considered in the TTC calculation because the system's operating condition changes frequently. Stability analysis involves many constraints that should be considered, such as thermal stability, static stability, dynamic stability, transient stability, and other factors. In practice, it can be very complicated to consider all constraints. These constraints make the calculation large and the calculation process complex. The stability indices considered in the calculation of ATC or TTC in the above literature and those considered in the calculation of TTC in this paper are shown in TABLE 1.

As can be observed in TABLE 1, the transient stability of the system is rarely considered in most of the literature when estimating ATC or TTC. In [10], a transient energy function is proposed to calculate the ATC of N-1 contingency. However, the calculation of this method is too simplified, resulting in insufficient accuracy. This paper calculates the TTC under transient stability using the DSA Tools software. The condition of the power system changes frequently, but the optimization and probability-based methods need to run from scratch without the ability to accumulate prior knowledge. In contrast, machine learning-based algorithms have the ability to learn from data and historical experiences and gradually improve model performance. The prediction process of machine learning-based models after training is relatively simple and fast. It can dynamically adapt to system changes and maintain prediction accuracy by constantly updating the model.

TABLE 1. Comparison table of indicators.

Literature serial number	Thermal arrest	Static stability	Dynamic stability	Transient stability
[12]	✓	✗	✗	✗
[2]、[3]、[4]、[5]、[6]、[15]	✓	✓	✗	✗
[7]、[8]、[9]、[11]、[13]、 [14]	✓	✓	✓	✗
[10] and this paper	✓	✓	✓	✓

In power systems, machine learning can typically be applied to power forecasting, load forecasting, predicting outages, predicting grid failures, and so on [16], [17], [18], [19]. Machine learning methods for prediction include random forest decision trees [20] and gradient-boosting decision trees (XGBoost, LGBM) [21]. Random forests usually have a stronger anti-overfitting ability due to integrating multiple independent trees, but the prediction accuracy is slightly lower. Gradient lifting is achieved by gradually optimizing the model. Although the accuracy is high, it is easy to overfit. Therefore, a stacked ensemble learning method is proposed based on a single machine learning model. Stacked ensemble learning combines the advantages of multiple models and makes up for the shortcomings of a single model [22], [23], [24]. In [22], the GBRT, SVR, BPNN, and RF models are proposed, and the predicted results of each model are integrated according to the optimized weights. In [23], a combination of three basic learners, MLP, CNN, and LSTM is proposed to construct wind power prediction models. In [24], an integrated neural network framework is proposed to predict wind and solar power generation. The framework is composed of LSTM, SVM, BP, and ELM. However, the disadvantage of neural networks is that they are easy to over-fit and have a high demand for computing resources. In general, stacked ensemble learning can train multiple models in parallel and reduce the influence of outliers and noise by voting or weighted average. In addition, combined with hyper-parameter optimization, the performance and stability of the model can be further improved, and the optimal combination of parameters can be found in order to achieve the best prediction effect.

Inspired by the above efforts, this paper proposes a novel method for real-time TTC evaluation by means of ensemble learning methods. By optimizing the selection of sub-models and hyperparameters, an accurate and fast TTC evaluation can be achieved online. The following is a summary of the main contributions of this paper: The sample data of this paper is generated by DSAtools software, which covers different operation modes and multiple operation indexes of power grid. By integrating LGBM, XGBoost and RF methods for TTC prediction, and optimizing sub-models and hyperparameters, accurate and fast online TTC evaluation is achieved. A summary of the main contributions of this paper can be found below:

1) During sample data generation, the data is generated by adjusting the load level of the entire power system to mimic realistic operating conditions. The constraints of thermal limits, voltage limits, transient stability, and voltage stability are fully considered.

2) To evaluate the TTC, the stacked ensemble learning method is proposed with XGBoost, LGBM, and RF models as the base models and LGBM as the meta-model. By optimizing the selection of sub-models and hyperparameters, accurate and fast TTC online prediction can be achieved.

i) In the XGBoost model, not only the first derivative is used, but the second derivative is also used to optimize the loss function so as to fit the loss function more accurately. LGBM can better capture complex relationships when dealing with large data sets and high-dimensional features. There are many parameters in XGBoost and LGBM, and the parameter adjustment is complex, but the prediction accuracy of both methods is high. Random forest integrates multiple independent trees, which usually have the stronger anti-overfitting ability, but the prediction accuracy is slightly lower.

ii) Random search is an effective method for solving XGBoost and LGBM parameter optimization problems. In the parameter optimization of the stacking model, a random search strategy can be used, combined with five-fold cross-validation, to search for the best parameter combination.

The remainder of this paper is organized in the following manner: Section II introduces the proposed implementation process of the stacking model. Section III introduces the generation of sample data for TTC assessment. Section IV introduces the principle of XGBoost, LGBM, RF model, and the process of model hyperparameter optimization. In Section V, case studies are performed on the 500-node power grid model to demonstrate the effectiveness of the proposed method. Finally, conclusions are drawn in Section VI.

II. THE TRAINING OF STACKING MO•DELS FOR TTC ASSESSMENT

A. MAIN FLOWCHART

The main flow of the proposed method can be seen in Figure 1. Before constructing the Stacking model, we first use the power system software DSATools to establish a power grid model, simulate it to generate sample data, and preprocess the sample data. Secondly, three models of XGBoost model, LGBM model, and RF model are trained, and random search is used to optimize the hyperparameters to obtain the best combination of hyperparameters. Finally, the optimized stacking model is used for online applications.

The stacked ensemble learning algorithm minimizes the error by finding the optimal model combination weights and structure so that the combination of the base model and the meta-model can improve the overall prediction performance [25], [26], [27]. The objective function of the stacking ensemble learning algorithm to solve the regression prediction problem can be expressed as minimizing regression loss functions such as mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R-squared). Therefore, the objective function of the stacking ensemble learning algorithm is shown as below:

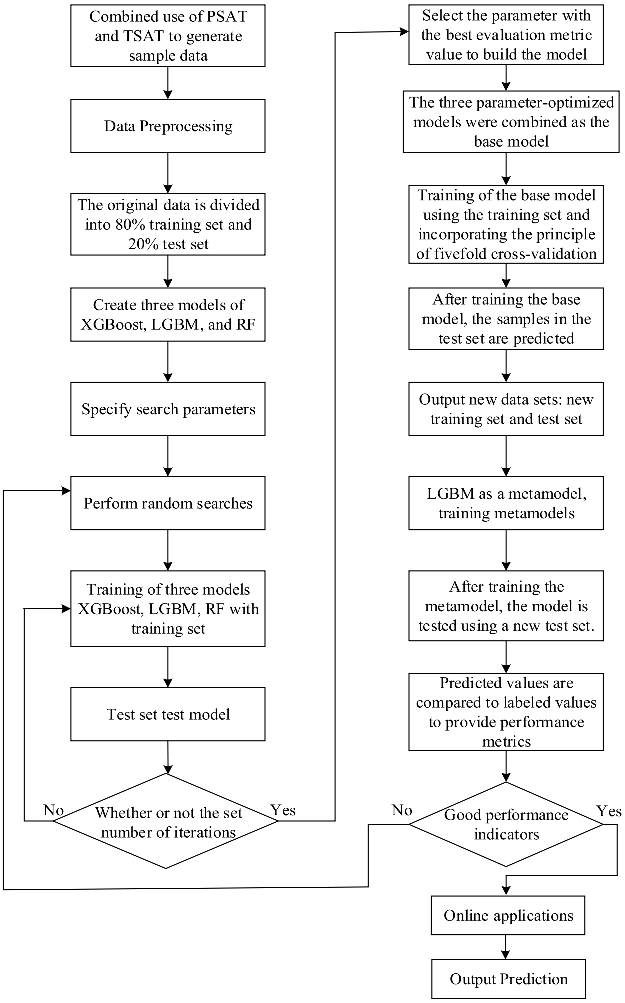

Combined use of PSAT and TSAT to generate sample data
Data Preprocessing
The original data is divided into 80% training set and 20% test set
Create three models of XGBoost, LGBM, and RF
Specify search parameters
Perform random searches
Training of three models XGBoost, LGBM, RF with training set
Test set test model
No
Whether or not the set number of iterations
Yes
Select the parameter with the best evaluation metric value to build the model
The three parameter-optimized models were combined as the base model
Training of the base model using the training set and incorporating the principle of fivefold cross-validation
After training the base model, the samples in the test set are predicted
Output new data sets: new training set and test set
LGBM as a metamodel, training metamodels
After training the metamodel, the model is tested using a new test set.
Predicted values are compared to labeled values to provide performance metrics
No
Good performance indicators
Yes
Online applications
Output Prediction

FIGURE 1. Method flowchart.

$$\min L(y, F(x)) \tag{1}$$

where L is the loss function, y is the real label of the sample, and $F(x)$ is the prediction result of the meta-model based on the combination of the prediction results of the base model. The input of the stacking model is the generated sample data, and the stacking model output is TTC results.

In this paper, the TTC prediction method based on the stacking algorithm takes the LGBM model, XGBoost model, and RF model as the base model. The TTC prediction results of the base model are integrated again and trained with the five-fold cross-validation method.

Then, the LGBM model is used as a meta-model to output the final results of the prediction. The final prediction results and the label values of the original test set are used to evaluate the model's performance. The stacking algorithm combines the advantages of considering different models and improves the accuracy and stability of TTC prediction. The framework of the stacking algorithm is shown in Figure 2.

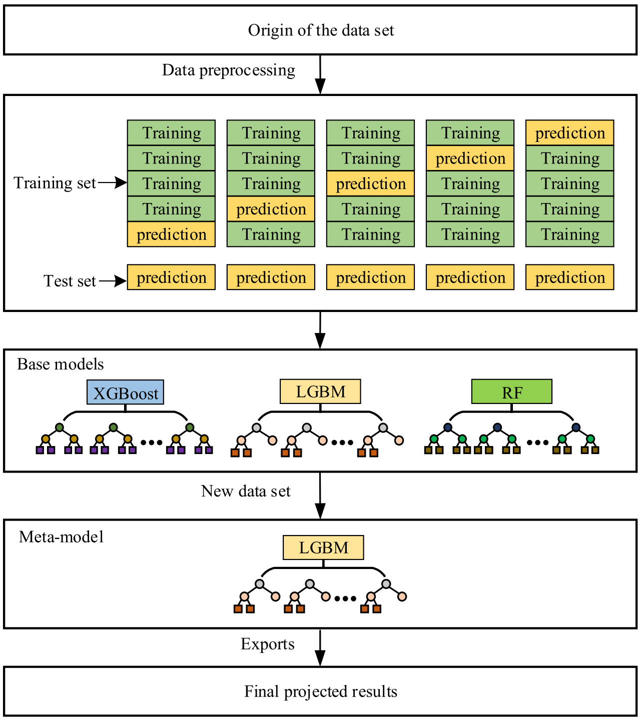


FIGURE 2. Stacking algorithm.

B. KEY STEPS

a) Step 1: The power grid model is established and simulated in the power system software DSA Tools to generate sample data.

b) Step 2: Preprocess the sample data generated in Step 1.

c) Step 3: Three models of XGBoost, LGBM and RF are constructed.

d) Step 4: Hyperparameter optimization of the three models constructed in Step 3 is carried out by random search, and the optimal hyperparameter combination is obtained.

e) Step 5: The stacking model is constructed and trained by using the model that completes the hyper-parameter optimization in Step 4, and then use the trained stacking model to predict TTC.

f) Step 6: Online application of TTC prediction based on the stacking algorithm.

III. GENERATION OF SAMPLE DATA

DSATools is a comprehensive analysis software package for power system dynamic security developed by Powertech Labs. It provides a wide range of functional modules to meet the requirements of the power industry for power grid planning, operation, and control. The main functions include Powerflow and Short-circuit Analysis Tool (PSAT), and Transient Security Assessment Tool (TSAT). PSAT is a tool for calculating power flow solutions such as voltage, power, and current in the power system. TSAT is utilized to assess the transient security and stability of the system and determine the maximum transient stability power transmission capacity for the specified section. The total transmission capacity search process comprehensively considers the four indicators of system transient stability, minimum damping ratio index, voltage stability, and frequency stability.

The power system model and power flow analysis are performed using PSAT, generate the convergent power flow data file and obtain the feature set. In TSAT, the location of the transmission section is determined, the expected fault set is set, and the power adjustment strategy is specified to generate the files necessary for TSAT simulation. The convergence power flow data file and other necessary files calculated by PSAT are used as the input files of the simulation software TSAT, and the total transmission capacity is obtained by relevant scanning and evaluation, and the TTC which meets the requirements of safe transmission in TSAT is retained as the label data. The TTC calculation process of a single sample is shown in Figure 3.

IV. ENSEMBLE MODELS OF XGBoost, LGBM, AND RF

This paper uses the stacking method, taking advantage of the three models: XGBoost, LGBM, and RF. Firstly, these three models are constructed and combined as base models for training. Then, the LGBM model is retrained as a meta-model to obtain the final TTC results.

A. BUILDING THE XGBoost MODEL

Model input: given the training set sample $\{(x_i, y_i)\}_{i=1}^{n}$, where x_i is the i sample feature, y_i is the i sample label, and n is the total number of samples [28].

Internal structure: It is assumed that t-round iteration is performed to establish a decision tree. Each iteration is the generation of a new decision tree. The number of iterates or the number of generated decision trees begins at zero and increases incrementally until the specified maximum number of iterates has been achieved. The value range of the decision tree is 0 to the maximum number of iterations minus one.

Model output: the prediction result $\widehat{y_i}^{(t)}$ of the model to the sample after the t-round iteration, that is, the TTC value.

In the training process, the first decision tree is established, and the features and labels of the training samples are used for fitting. Then, the residual is calculated and used to train the next decision tree. The principle of t-iteration training is as follows:

For the 0th iteration, the initial prediction value $\widehat{y_i}^{(0)}$ is 0:

$$\widehat{y_i}^{(0)} = 0 \tag{2}$$

$$\widehat{y_i}^{(1)} = f_1(x_i) = \widehat{y_i}^{(0)} + f_1(x_i) \tag{3}$$

$$\widehat{y_i}^{(2)} = f_1(x_i) + f_2(x_i) = \widehat{y_i}^{(1)} + f_2(x_i) \tag{4}$$

$$\widehat{y_i}^{(t)} = \sum\nolimits_{k=1}^{t} f_k(x_i) = \widehat{y_i}^{(t-1)} + f_t(x_i) \tag{5}$$

where x_i is the i-th sample feature, y_i is the i-th sample label, $\widehat{y_i}^{(t)}$ is the prediction result of the model after the t-th iteration, $f_k(x_i)$ is the prediction value of the k-th decision tree to the

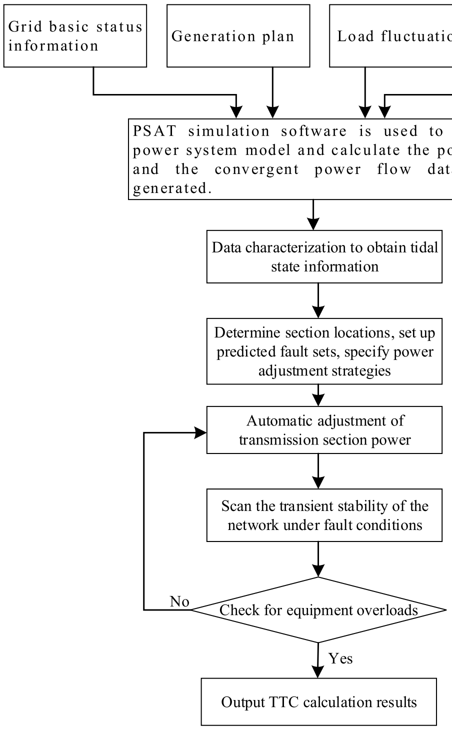

Grid basic status information
Generation plan
Load fluctuation
Fault factor
PSAT simulation software is used to build the power system model and calculate the power flow, and the convergent power flow data file is generated.
Data characterization to obtain tidal state information
Determine section locations, set up predicted fault sets, specify power adjustment strategies
Automatic adjustment of transmission section power
Scan the transient stability of the network under fault conditions
Check for equipment overloads
No
Yes
Output TTC calculation results

FIGURE 3. TTC calculation process.

sample, $\widehat{y_i}^{(t-1)}$ is the sample prediction value of the previous $t-1$ iteration, and $f_t(x_i)$ is the new prediction function of the t-th iteration.

The central idea of the XGboost algorithm is to fit the residuals, so that the residuals continue to decrease. The calculation of the residual E_t of the t-tree decision tree is shown as follows:

$$E_t = y_i - \widehat{y_i}^{(t-1)} \tag{6}$$

The XGBoost algorithm needs to calculate residuals during each iteration. The residual error can be calculated in Eq.(6), and the sample is re-weighted according to the calculation results. Then, the next decision tree is trained again using the corrected training set. Finally, the predicted results of each decision tree model are summed and added. The predicted result of the overall model is shown below:

$$F(x) = \sum\nolimits_{k=1}^{T} \alpha_k f_k(x) \tag{7}$$

where k represents the k^{th} decision tree; T represents the number of rounds of iteration, that is, the number of decision trees; $F(x)$ is the prediction result of the overall model; α_k is the weight of the decision tree $f_k(x)$; and $f_k(x)$ represents the prediction result of the k decision tree.

In summary, the XGBoost algorithm is an improved version based on the boosting tree algorithm. Compared with the traditional boosting tree algorithm, XGBoost introduces regularization terms and second derivative information and uses Taylor expansion to approximate the loss function. By optimizing the loss function, XGBoost can fit data and adjust model parameters more effectively.The principle of the XGBoost algorithm is shown in Figure 4.

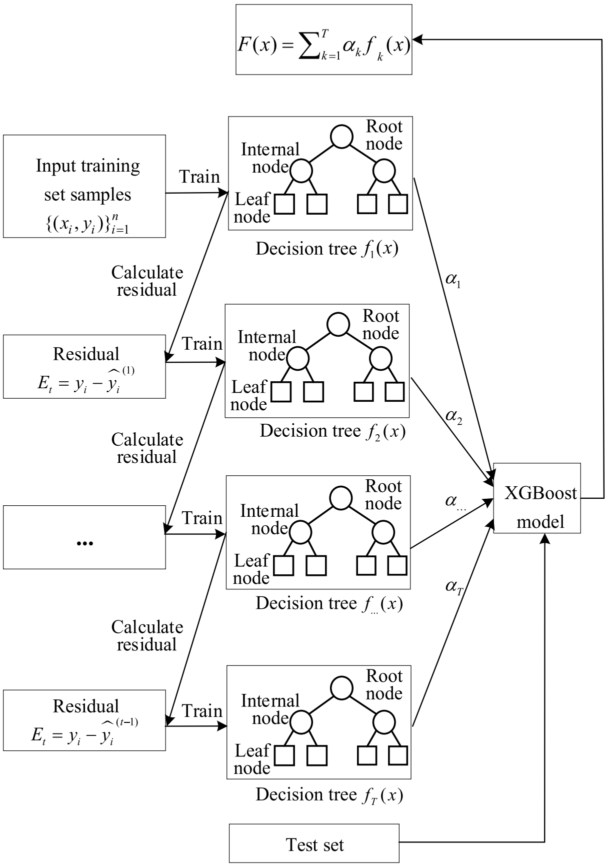

$F(x)=\sum_{k=1}^{T}\alpha_k f_k(x)$
Input training set samples $\{(x_i, y_i)\}_{i=1}^{n}$
Train
Internal node
Root node
Leaf node
Decision tree $f_1(x)$
Calculate residual
Residual $E_t = y_i - \widehat{y}_i^{(1)}$
Decision tree $f_2(x)$
...
Decision tree $f_{...}(x)$
Residual $E_t = y_i - \widehat{y}_i^{(t-1)}$
Decision tree $f_T(x)$
α_1
α_2
$\alpha_{...}$
α_T
XGBoost model
Test set

FIGURE 4. The principle of XGBoost algorithm.

B. BUILDING THE LGBM MODEL

LGBM and XGBoost are two commonly used gradient-boosting tree algorithms that are different from traditional decision trees in the splitting strategy, leaf node growth mode, and data processing method [29].

Firstly, LGBM adopts a decision tree growing by leaves, while XGBoost adopts a decision tree growing by layers. LGBM selects the optimal leaf node for splitting each time to minimize the loss. XGBoost splits nodes in a hierarchical order to ensure that each layer has the same number of leaf nodes. Secondly, the growth pattern of leaf nodes is also different. LGBM uses a depth-constrained leaf-growing strategy as trees grow, which can reduce the risk of overfitting. XGBoost uses a pre-sorting algorithm to select split points to improve training efficiency. In addition, data processing methods are also different. LGBM internally uses a histogram-based discretization method to process continuous features, transforming continuous values into discrete histogram blocks. XGBoost uses the gradient histogram method to process features to make better use of the gradient information of the feature value.

C. BUILDING THE RF MODEL

RF is based on the combination of Bagging (self-sampling method) and decision tree algorithm. RF is an improved

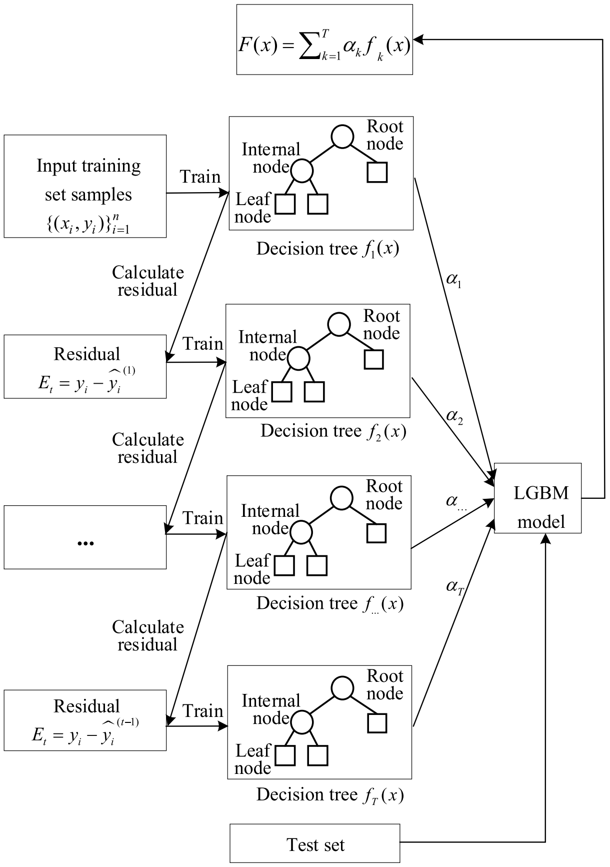

$F(x)=\sum_{k=1}^{T}\alpha_k f_k(x)$
Input training set samples $\{(x_i, y_i)\}_{i=1}^{n}$
Train
Internal node
Root node
Leaf node
Decision tree $f_1(x)$
Calculate residual
Residual $E_t = y_i - \hat{y}_i^{(1)}$
Decision tree $f_2(x)$
...
Decision tree $f_{...}(x)$
Residual $E_t = y_i - \hat{y}_i^{(t-1)}$
Decision tree $f_T(x)$
α_1
α_2
$\alpha_{...}$
α_T
LGBM model
Test set

FIGURE 5. The principle of the LGBM algorithm.

version of the Bagging algorithm. Firstly, RF replaces the base learner in Bagging with a decision tree. At the same time, RF introduces more random elements, which are random not only in sample extraction but also in feature extraction. This randomness can have the effect of a reduction in the correlation between features and an increase in the generalization ability of the model [30].

Given a training set sample $s = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the i^{th} sample feature and y_i is the i^{th} sample label. In the feature set $f = \{f_1, f_2, \ldots, f_M\}$, each sample corresponds to a set of features.

The output of RF is the TTC prediction results, and the RF training process is as follows:

1) Define hyperparameters: the number of decision trees T, the maximum depth of the decision tree max _ depth.

2) Random sampling: N samples are randomly selected as sub-training sets from the original training set with put-back, and the process is repeated T times to form T sub-training sets.

3) Randomly selected features: For each decision tree, P features are randomly selected from the sub-training set corresponding to the decision tree to construct a feature subset of the node. This P is usually a fixed value that is much smaller than the total characteristic number M.

4) Construction of decision tree:

Calculation of the MSE of each set/node is shown as follows:

$$mean = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{8}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - mean)^2 \tag{9}$$

where *mean* represents the mean value in the set, N represents the total number of samples in the set, y_i represents the target value of each sample in the set.

5) According to the above steps, T decision trees are constructed, and all the generated decision trees are accumulated together to obtain the final RF model, which can be used to predict the labels of new samples. The final predicted score is calculated by averaging the predicted scores of all the trees. The final prediction results of the RF are presented as follows:

$$H(x) = \frac{1}{T}\sum_{j=1}^{T} h_j(x) \tag{10}$$

where $h_j(x)$ is the prediction result of each decision tree, and j is the j^{th} decision tree. The principle of the RF algorithm is shown in Figure 6.

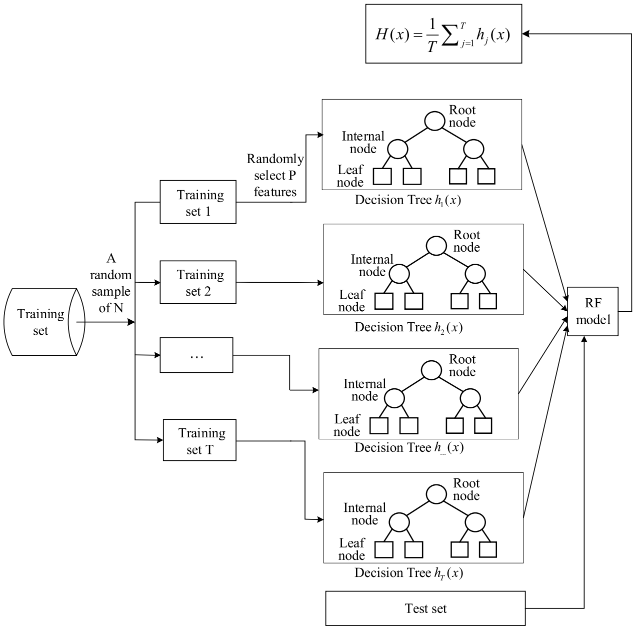


FIGURE 6. Principle of random forest algorithm.

D. HYPERPARAMETER OPTIMIZATION

Hyperparameters refer to some configuration options in the machine learning model. These options are manually set and cannot be automatically optimized by functions within the model. Random search is an effective method in the case of large hyperparameter space and limited computing resources. Since random search does not depend on gradient information or other additional computational overhead, it can quickly perform parameter searches with relatively few computing resources. By randomly selecting parameter combinations in the parameter space for evaluation, random search can quickly explore different parameter combinations and find relatively good solutions. Taking XGBoost as an example, the following is the process of random search:

1) Setting the search space: determining the number of samples (or iterations) for the random search and the random selection range of parameters for each sample. For the XGBoost model, the focus is on a small number of parameters that have a more significant impact on the results: the maximum depth $\overline{D}$ defines the maximum number of layers the decision tree can grow, limiting the complexity of a single decision tree, and plays a role in the objective function of the regularization. α_k controls the contribution of each decision tree to the final model predictions. α_k is multiplied by the prediction of each of the decision tree and added to the cumulative prediction of all previous decision trees to obtain the prediction of the final model; $\widehat{W}$ is the sum of the weights of the smallest child nodes required during the node splitting process. $\widehat{W}$ defines the minimum sum of sample weights allowed on a leaf node. If the sum of the sample weights on a leaf node is less than $\widehat{W}$, the split is not continued, and the node is treated as a leaf node; conversely, if the sum of the sample weights is greater than or equal to $\widehat{W}$, the split is continued.

2) Specify search parameters: Specify how to iterate and fold the search. The number of iterations defines the number of times a hyperparameter combination is randomly selected in the hyperparameter space. The number of folds determines how many segments the cross-validation splits the data into, allowing each hyperparameter combination to be trained and evaluated across these segments.

3) The objective function is defined with MAE, RMSE, and R-squared, which are used as performance evaluation metrics for the model.

4) Data preparation: The training set in the original data is used for hyperparameter optimization.

5) Random Search: Use the specified XGBoost model, search space, and search parameters to create a random search object for a random search. In each iteration, a set of parameter values is randomly selected, and the model is trained on the training set. Then, the model's performance is assessed on the test set.

6) Record results and parameter combinations: Record the results of each evaluation and the corresponding parameter combinations.

7) Iterate Steps 5) and 6) until the specified number of iterations is achieved.

8) The parameter combination yielding the highest evaluation index value is chosen as the final model parameter.

V. CASE STUDIES

Case studies are performed on a 500-node grid model to showcase the effectiveness of the proposed method. The model training process is as follows:

1) Using the PSAT and TSAT in the power system software DSATools, a power grid model with 500 nodes is established and simulated to generate sample data. Firstly, The power system simulation software PSAT is utilized to create a 500-node power grid model and conduct power flow analysis, resulting in a converged power flow data file. The specific process is to adjust the load level of the entire power network, solve power flow, and save the adjusted condition in files. Operating data can be extracted from the files, encompassing load active and reactive power, generator active and reactive power, bus voltage, and line active and reactive power. In TSAT software, determining the section location, defining the expected fault scenarios, and specifying the power adjustment strategy are essential for modeling and simulating the dynamic responses of the 500-node power grid under various operating conditions.

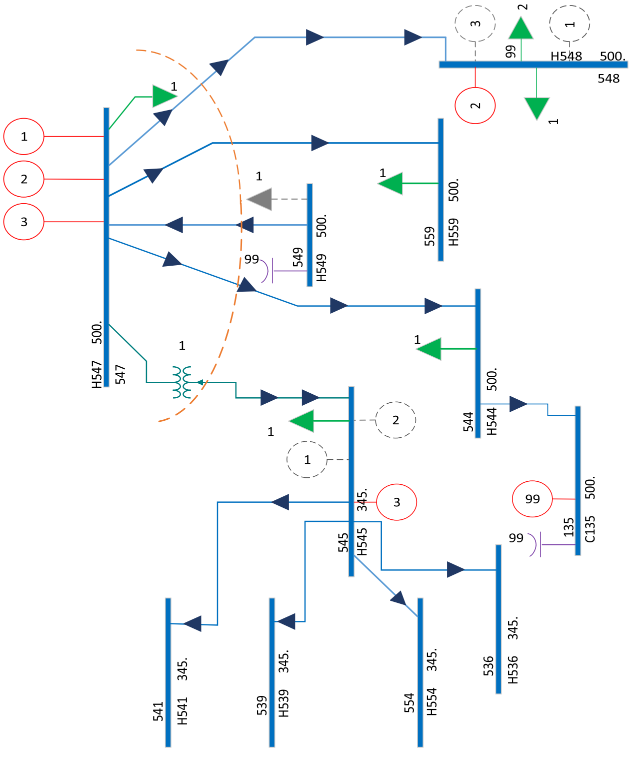


FIGURE 7. The one-line diagram of the transmission corridor.

The transmission corridor of this example consists of a series of lines, which are connected to bus 547 and other nodes, respectively. These include lines from 547 to 548, lines from 547 to 559, lines from 547 to 549, lines from 544 to 547, and lines from 545 to 547. The section setting is shown in Figure 7.

Through these lines, the power on bus 547 can be transmitted to the corresponding node and supplied to the right area, and then the TTC value is obtained by automatically searching the safety limit. The original data includes feature set and label data. The feature set includes bus voltage amplitude V_m and bus voltage phase angle V, generator active power P_{gen}, generator reactive power Q_{gen}, load active power P_{load}, load reactive power Q_{load}, line active power P_{line}, line reactive power Q_{line}, The label data is TTC.

According to the distribution of TTC values shown in Figure 8, it can be observed that the number of samples with TTC values of about 800MW and 1000MW is small. The value regions of sample features and label values are displayed in TABLE 2.

TABLE 2. Feature and label value range table.

Feature set	Maximum value	Minimum value
V_m	1.07123	0.80451
V	42.82791	-69.68275
P_{load} (MW)	19.71428	-9.31097
Q_{load} (Mvar)	4.79252	-5.03693
P_{gen} (MW)	8.24449	-0.02149
Q_{gen} (Mvar)	1.65418	-0.85974
P_{line} (MW)	38.31682	-1.74286
Q_{line} (Mvar)	6.5	-3.78682
TTC (MW)	1045.96	682.65

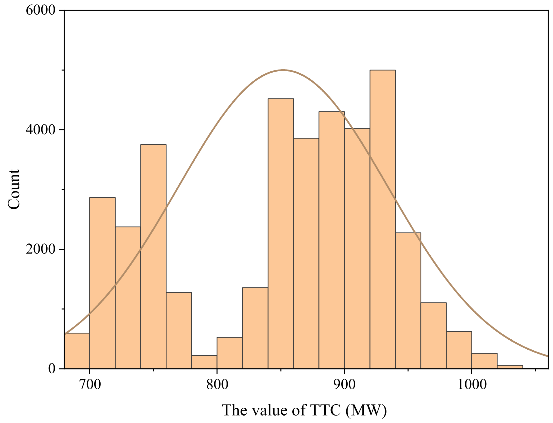


FIGURE 8. TTC sample distribution diagram.

Preprocessing the sample data.

The original data set includes feature sets and label data. The feature set of each sample has a total of 3432 features. Among them, bus voltage amplitude V_m has 563 data, and bus voltage phase angle V has 563 data. the sum of generator active power P_{gen} and generator reactive power Q_{gen} has 280 data, the sum of load active power P_{load} and load reactive power Q_{load} has 884 data, the sum of line active power P_{line} and line reactive power Q_{line} has 1142 data. The label data is a total of 39,000 sample data for TTC values.

2) The optimal parameter combination is obtained by random search using the following steps.

Define the parameter space: the continuous range of the maximum depth $\overline{D}$ search is an integer between 1 and 20; the continuous range of the learning rate search is a float number between 0.01 and 0.2; and the continuous range of $\widehat{W}$ search is an integer between 1 and 20.

Specify the search parameters: specify the number of iterations is 10, and the fold of cross-validation is 5.

Define the objective function: The search parameter is specified to define the objective function. The mean square error is used as the evaluation index of the model performance.

Data preparation: The training set in the original data is used for hyperparameter optimization, and there are 31200 samples in the training set. Then, 20% of the training set samples are reserved for evaluating the model's performance.

Random search: Use the specified XGBoost model, search space, and search parameters to create a random search object, and the sample is passed to the random search object for a random search.

Record results and parameter combinations: Record each evaluation result on the validation set and the corresponding parameter combination.

Repeat the random search process 10 times according to the specified number of iterations. Based on the recorded parameter combinations from each iteration, the parameter set yielding the highest evaluation index value is selected as the final model parameter. The XGBoost, LGBM, and RF models undergo iterative steps as described. Changes in their configurations before and after parameter adjustments are detailed in Table 3:

TABLE 3. Comparison of different algorithms before and after parameter adjustment.

Model name	Hyperparameters before tuning	Hyperparameters after tuning
XGBoost	$\overline{D}$ =5	$\overline{D}$ =7
	α_k =0.01	α_k =0.1
	$\widehat{W}$ =1	$\widehat{W}$ =3
LGBM	T=1000	T=3000
	$\overline{D}$ =10	$\overline{D}$ =7
	α_k =0.01	α_k =0.1
RF	T=10	T=14
	$\overline{D}$ =10	$\overline{D}$ =None

3) The construction of the stacking model and the training and prediction process

Firstly, the original data set has 39,000 samples (each sample includes 3,432 features and 1 label value), which are partitioned into training and test sets. There are 31,200 samples in the training set and 7,800 samples in the test set. Then, the training set is segmented into five parts; each fold contains 6,240 samples. Taking the XGBoost model as an example, the hyperparameters of the model are set, the XGBoost model object is created, and the hyperparameters are passed to the model. The XGBoost model is trained using four folds of the training set. The XGBoost model will learn from the data and hyperparameters, fit the samples in the training set, and generate a trained model. The trained XGBoost model is evaluated using a previously reserved validation set and test set. Features from these sets are fed into the model to obtain predictions. Specifically, the validation set yields 6,240 predictions, and the test set yields 7,800 predictions after five cycles. The validation set predictions are concatenated row-wise to form a dataset of 31,200 samples. The test set predictions are averaged across the five cycles to produce 7,800 averaged predictions.

Replace the XGBoost model with the LGBM and RF models and perform the same steps. Finally, concatenate the validation set predictions from the three models column-wise, resulting in a dataset with 31,200 rows and 3 columns. Similarly, concatenate the test set predictions from the three models column-wise to obtain a dataset with 7,800 rows and 3 columns. Combining the sample data of 31,200 rows and 3 columns with the features of the original training set samples results in a new dataset of 31,200 rows and 3,435 columns, which serves as the input for the new training set. The labels from the original training set are used as the target variable. Similarly, merging the 7,800 rows and 3 columns of sample data with the features of the original test set produces a new dataset of 7,800 rows and 3,435 columns for the new test set, using the labels from the original test set as the target variable.

LGBM is utilized as a meta-model, and the new training set is used for its training. Once the meta-model training is completed, the new test set will be employed to make the final predictions.

4) Real-time application of TTC prediction using the stacking algorithm

5) The comparison results of prediction errors before and after optimizing each model's parameters are shown in Table 4.

TABLE 4. (a) Prediction error of test set before parameter adjustment. (b) Prediction error of test set after parameter adjustment.

(a)

Performance index	RF	LGBM	XGBoost	stacking
RMSE(MW)	9.072	6.716	7.005	6.472
MAE(MW)	5.650	3.319	3.704	2.768
R-squared	0.988	0.993	0.993	0.994

(b)

Performance index	RF	LGBM	XGBoost	stacking
RMSE(MW)	8.016	6.124	6.485	6.107
MAE(MW)	4.357	2.072	2.178	1.921
R-squared	0.991	0.994	0.993	0.995

From the comparison before and after parameter adjustment, the prediction error after random search hyperparameter optimization is smaller than that before optimization. This indicates that the optimized model achieves a better fit to the training data and yields more accurate predictions on the test set. From the perspective of prediction error, the prediction results obtained by using this method are better than those obtained by using XGBoost model, LGBM model, and RF model alone. This shows that the stacking model plays a good role in integrating the prediction results of multiple basic models, such as the LGBM model.

The prediction results of RF, LGBM, XGBoost, and stacking models in the test set before and after hyperparameter optimization are shown in Figure 9 and Figure 10. The curve in the figure is the prediction error comparison result of 200 randomly selected sample data.

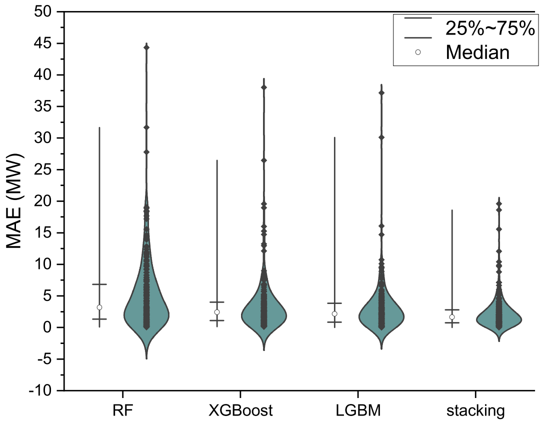


FIGURE 9. Comparison of prediction errors of each model before parameter adjustment.

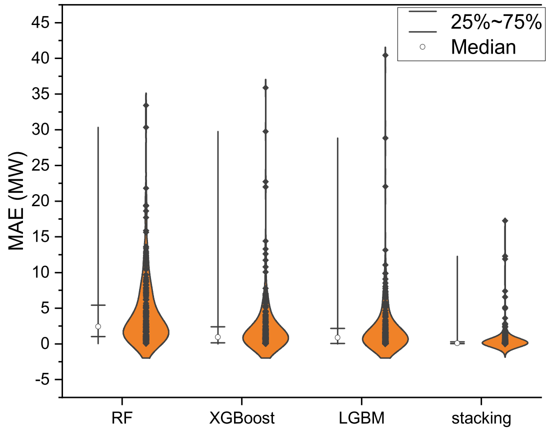


FIGURE 10. Comparison of prediction errors after parameter adjustment of each model.

Ten samples were extracted from the test set for detailed analysis. The data distribution of these sample features is shown in Figure 11. The label values TTC of these samples are listed in TABLE 5. After hyper-parameter optimization, the prediction errors of these 10 samples by different methods are shown in Figure 12. By analyzing the data in the table and the curve in the figure, it is evident that the sample errors at the limit values of 787.42 MW and 1020.22 MW are

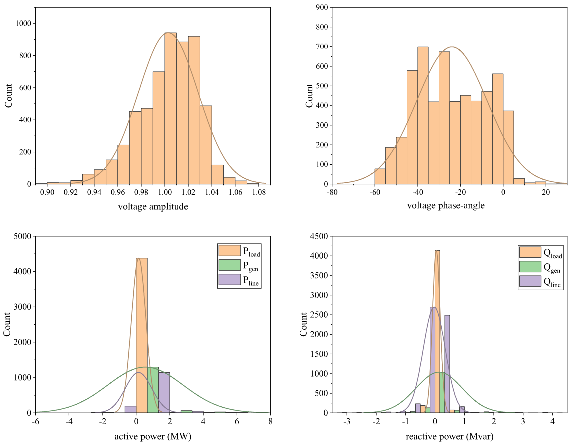


FIGURE 11. Sample feature distribution map.

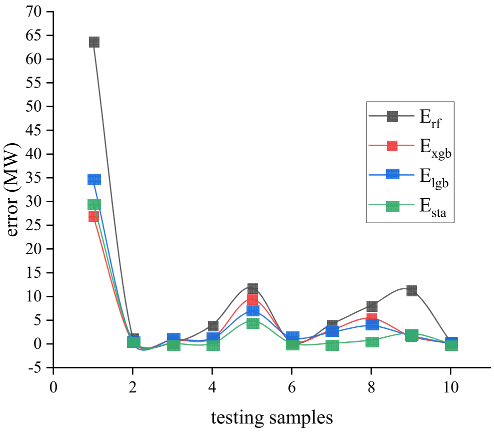


FIGURE 12. Comparison of prediction errors of different methods.

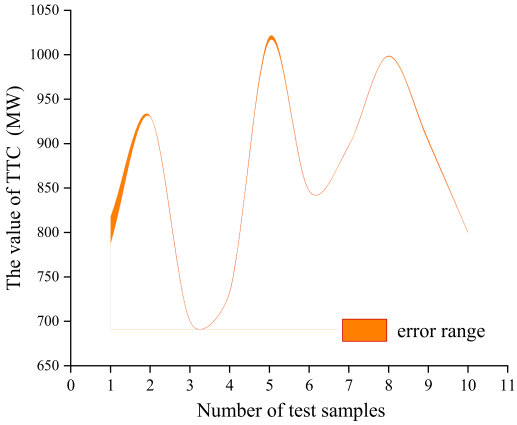


FIGURE 13. The error comparison diagram of TTC predicted value and real value.

TABLE 5. Sample label value.

Testing samples	TTC(MW)
Test1	787.42
Test2	930.25
Test3	700.11
Test4	732.43
Test5	1020.22
Test6	846.93
Test7	897.27
Test8	998.83
Test9	904.1
Test10	800.29

significant. Overall, the stacking algorithm provides superior prediction results compared to other algorithms, effectively capturing the actual data trend. Figure 13 compares the predicted values with the real values for the stacking method optimized by hyperparameters, with the orange shaded area representing the error range. The figure shows that the errors within the orange region are minimal, indicating a very small deviation between the predicted and actual values. This demonstrates that the optimized stacking method has high prediction accuracy and reliability.

VI. CONCLUSION

This paper proposes a method for real-time TTC evaluation using a stacked ensemble learning model. The following conclusions can be made:

(1) In the process of sample generation, the constraints of thermal limits, voltage security, transient stability, and voltage stability are comprehensively considered. This method ensures that the generated sample data can accurately reflect the operating characteristics of the actual power system under different load conditions, which provides a reliable data basis for the subsequent TTC prediction.

(2) By integrating LGBM, XGBoost, and RF models for TTC prediction, this paper fully combines the advantages of each model. Specifically, the efficiency and accuracy of LGBM, the ability of XGBoost to deal with nonlinear relationships, and the anti-overfitting characteristics of RF have been fully utilized, thus significantly improving the overall performance of TTC prediction.

(3) This paper combines a random search strategy and a five-fold cross-validation method to search for the best parameter combination of XGBoost, LGBM, and RF models. It improves the accuracy of TTC evaluation and achieves higher efficiency.

REFERENCES

[1] L. Hou, W. Li, K. Zhou, and Q. Jiang, "Integrating flexible demand response toward available transfer capability enhancement," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113370, doi: 10.1016/j.apenergy.2019.113370.

[2] S. Kesherwani, A. Mohapatra, and S. C. Srivastava, "An efficient holomorphic embedded based approach for available transfer capability evaluation," *Int. J. Electr. Power Energy Syst.*, vol. 122, Nov. 2020, Art. no. 106164, doi: 10.1016/j.ijepes.2020.106164.

[3] J.-H. Liu and C.-C. Chu, "Iterative distributed algorithms for real-time available transfer capability assessment of multiarea power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2569–2578, Sep. 2015, doi: 10.1109/TSG.2015.2388777.

[4] A. M. Alshamrani, M. A. El-Meligy, M. A. F. Sharaf, W. A. Mohammed Saif, and E. M. Awwad, "Transmission expansion planning considering a high share of wind power to maximize available transfer capability," *IEEE Access*, vol. 11, pp. 23136–23145, 2023, doi: 10.1109/ACCESS.2023.3253201.

[5] R. Chauhan, R. Naresh, M. Kenedy, and A. Aharwar, "A streamlined and enhanced iterative method for analysing power system available transfer capability and security," *Electr. Power Syst. Res.*, vol. 223, Oct. 2023, Art. no. 109528, doi: 10.1016/j.epsr.2023.109528.

[6] A. M. Glazunova and E. S. Aksaeva, "Estimation of total transfer capability in intersystem tie lines of electric power systems," *IFAC-PapersOnLine*, vol. 51, no. 32, pp. 331–336, 2018, doi: 10.1016/j.ifacol.2018.11.405.

[7] T. Jiang, X. Li, X. Kou, R. Zhang, G. Tian, and F. Li, "Available transfer capability evaluation in electricity-dominated integrated hybrid energy systems with uncertain wind power: An interval optimization solution," *Appl. Energy*, vol. 314, May 2022, Art. no. 119001, doi: 10.1016/j.apenergy.2022.119001.

[8] Y. Huang, T. Ding, P. Wang, W. Jia, C. Ju, X. He, H. Zhang, Z. Zhang, and Y. Sun, "Linearized AC power flow model based interval total transfer capability evaluation with uncertain renewable energy integration," *Int. J. Electr. Power Energy Syst.*, vol. 154, Dec. 2023, Art. no. 109440, doi: 10.1016/j.ijepes.2023.109440.

[9] X. Wang, X. Wang, H. Sheng, and X. Lin, "A data-driven sparse polynomial chaos expansion method to assess probabilistic total transfer capability for power systems with renewables," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2573–2583, May 2021, doi: 10.1109/TPWRS.2020.3034520.

[10] M. Eidiani, "A new hybrid method to assess available transfer capability in AC–DC networks using the wind power plant interconnection," *IEEE Syst. J.*, vol. 17, no. 1, pp. 1375–1382, Mar. 2023, doi: 10.1109/JSYST.2022.3181099.

[11] S. Madadi, B. Mohammadi-Ivatloo, and S. Tohidi, "Probabilistic available transfer capability evaluation considering dynamic line rating based on a sequential game-theoretic approach," *IEEE Syst. J.*, vol. 16, no. 1, pp. 891–901, Mar. 2022, doi: 10.1109/JSYST.2021.3060286.

[12] K. Majumdar, P. K. Roy, and S. Banerjee, "Calculation of available transfer capability using hybrid chaotic selfish herd optimizer and 24 hours RES-thermal scheduling," *Chin. J. Electr. Eng.*, vol. 9, no. 4, pp. 54–72, Dec. 2023, doi: 10.23919/cjee.2023.000032.

[13] Y. Sun, T. Ding, M. Qu, F. Wang, and M. Shahidehpour, "Interval total transfer capability for mesh HVDC systems based on sum of squares and multi-dimensional holomorphic embedding method," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4157–4167, Nov. 2022, doi: 10.1109/TPWRS.2022.3145065.

[14] W. Peres, J. A. Passos Filho, F. C. R. Coelho, R. P. B. Poubel, and J. N. N. Costa, "Dynamic transmission capability calculation using bioinspired optimization," *Int. J. Electr. Power Energy Syst.*, vol. 133, Dec. 2021, Art. no. 107227, doi: 10.1016/j.ijepes.2021.107227.

[15] H. W. Reyad, M. Elfar, and E. E. El-Araby, "Probabilistic assessment of available transfer capability incorporating load and wind power uncertainties," *IEEE Access*, vol. 11, pp. 39048–39065, 2023, doi: 10.1109/ACCESS.2023.3268544.

[16] M. F. Tahir, M. Z. Yousaf, A. Tzes, M. S. El Moursi, and T. H. M. El-Fouly, "Enhanced solar photovoltaic power prediction using diverse machine learning algorithms with hyperparameter optimization," *Renew. Sustain. Energy Rev.*, vol. 200, Aug. 2024, Art. no. 114581, doi: 10.1016/j.rser.2024.114581.

[17] R. Eskandarpour and A. Khodaei, "Machine learning based power grid outage prediction in response to extreme events," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3315–3316, Jul. 2017, doi: 10.1109/TPWRS.2016.2631895.

[18] M. Madhukumar, A. Sebastian, X. Liang, M. Jamil, and M. N. S. K. Shabbir, "Regression model-based short-term load forecasting for university campus load," *IEEE Access*, vol. 10, pp. 8891–8905, 2022, doi: 10.1109/ACCESS.2022.3144206.

[19] A. S. Alhanaf, M. Farsadi, and H. H. Balik, "Fault detection and classification in ring power system with DG penetration using hybrid CNN-LSTM," *IEEE Access*, vol. 12, pp. 59953–59975, 2024, doi: 10.1109/ACCESS.2024.3394166.

[20] J. Hao, C. Zhu, and X. Guo, "Wind power short-term forecasting model based on the hierarchical output power and Poisson re-sampling random forest algorithm," *IEEE Access*, vol. 9, pp. 6478–6487, 2021, doi: 10.1109/ACCESS.2020.3048382.

[21] X. Yao, X. Fu, and C. Zong, "Short-term load forecasting method based on feature preference strategy and LightGBM-XGboost," *IEEE Access*, vol. 10, pp. 75257–75268, 2022, doi: 10.1109/ACCESS.2022.3192011.

[22] S. Zhang, M. Liu, M. Liu, Z. Lei, G. Zeng, and Z. Chen, "Day-ahead wind power prediction using an ensemble model considering multiple indicators combined with error correction," *Appl. Soft Comput.*, vol. 148, Nov. 2023, Art. no. 110873, doi: 10.1016/j.asoc.2023.110873.

[23] H. Wang, Z. Tan, Y. Liang, F. Li, Z. Zhang, and L. Ju, "A novel multi-layer stacking ensemble wind power prediction model under tensorflow deep learning framework considering feature enhancement and data hierarchy processing," *Energy*, vol. 286, Jan. 2024, Art. no. 129409, doi: 10.1016/j.energy.2023.129409.

[24] Z. Wu and B. Wang, "An ensemble neural network based on variational mode decomposition and an improved sparrow search algorithm for wind and solar power forecasting," *IEEE Access*, vol. 9, pp. 166709–166719, 2021, doi: 10.1109/ACCESS.2021.3136387.

[25] G. Qiu, J. Liu, Y. Liu, T. Liu, and G. Mu, "Ensemble learning for power systems TTC prediction with wind farms," *IEEE Access*, vol. 7, pp. 16572–16583, 2019, doi: 10.1109/ACCESS.2019.2896198.

[26] S. Wen, C. Zhang, H. Lan, Y. Xu, Y. Tang, and Y. Huang, "A hybrid ensemble model for interval prediction of solar power output in ship onboard power systems," *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 14–24, Jan. 2021, doi: 10.1109/TSTE.2019.2963270.

[27] L. Von Krannichfeldt, Y. Wang, and G. Hug, "Online ensemble learning for load forecasting," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 545–548, Jan. 2021, doi: 10.1109/TPWRS.2020.3036230.

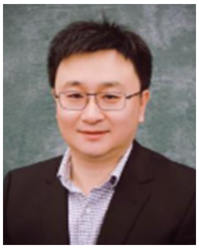
[28] T. Zhang, X. Zhang, O. Rubasinghe, Y. Liu, Y. Chow, H. Iu, and T. Fernando, "Long-term energy and peak power demand forecasting based on sequential-XGBoost," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 3088–3104, Mar. 2024, doi: 10.1109/TPWRS.2023.3289400.

[29] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks," *IEEE Access*, vol. 9, pp. 83654–83665, 2021, doi: 10.1109/ACCESS.2021.3087175.

[30] Y. Xuan, W. Si, J. Zhu, Z. Sun, J. Zhao, M. Xu, and S. Xu, "Multi-model fusion short-term load forecasting based on random forest feature selection and hybrid neural network," *IEEE Access*, vol. 9, pp. 69002–69009, 2021, doi: 10.1109/ACCESS.2021.3051337.

HUIMIN GAO (Member, IEEE) was born in Jiangxi, China, in May 1978. She received the B.S. and M.S. degrees in electrical engineering from Zhengzhou University, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Zhejiang University, China, in 2006. She is now an Professor with the College of Information Engineering, Hangzhou Dianzi University. Her main field of interests include power system stability and control and planning.

YI LU (Student Member, IEEE) received the bachelor's degree in electrical engineering from Zhejiang University of Water Resources and Electric Power. She is currently pursuing the master's degree in control engineering with the College of Automation, Hangzhou Dianzi University. Her research interests include total transfer capability assessment and its operation mode adjustment.



RUISHENG DIAO (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Arizona State University, in 2009. He is currently working with The ZJU-UIUC Institute, as the Director of renewable power system simulation and intelligent control, and a tenured Associate Professor. His research interests include high-fidelity simulation techniques and AI-based methods for grid planning and operation. He serves as an Associate Editor for IEEE TRANSACTIONS ON POWER SYSTEMS, IEEE ACCESS, and *IET Generation, Transmission, and Distribution*. He is a recipient of the 2018 Research and Development 100 Awards and multiple IEEE PES Best Paper Awards.

JING ZHANG (Member, IEEE) was born in Henan, China, in November 1980. He received the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2009. He is currently working as a Senior Engineer with State Grid Zhejiang Electric Power Company. His research interests include power system stability, HVDC and FACTS, and signal processing.

ZHANNING LIU (Student Member, IEEE) received the bachelor's degree in electrical engineering from Zhejiang University, where he is currently pursuing the master's degree in electrical engineering. His research interests include modeling of power converters and frequency control.

AO ZHAO received the bachelor's degree in electrical engineering from Zhejiang University, where he is currently pursuing the master's degree in electrical engineering. His research interests include modeling of renewable generation and intelligent parameter identification.