

Article

An Automated and Interpretable Machine Learning Scheme for Power System Transient Stability Assessment

Fang Liu ¹, Xiaodi Wang ¹, Ting Li ¹, Mingzeng Huang ², Tao Hu ^{2,*}, Yunfeng Wen ² and Yunche Su ¹¹ State Grid Sichuan Economic Research Institute, Chengdu 610041, China² Engineering Research Center of Power Transmission and Transformation Technology of Ministry of Education, Hunan University, Changsha 410082, China

* Correspondence: htao@hnu.edu.cn

Abstract: Many repeated manual feature adjustments and much heuristic parameter tuning are required during the debugging of machine learning (ML)-based transient stability assessment (TSA) of power systems. Furthermore, the results produced by ML-based TSA are often not explainable. This paper handles both the automation and interpretability issues of ML-based TSA. An automated machine learning (AutoML) scheme is proposed which consists of auto-feature selection, CatBoost, Bayesian optimization, and performance evaluation. CatBoost, as a new ensemble ML method, is implemented to achieve fast, scalable, and high performance for online TSA. To enable faster deployment and reduce the heavy dependence on human expertise, auto-feature selection and Bayesian optimization, respectively, are introduced to automatically determine the best input features and optimal hyperparameters. Furthermore, to help operators understand the prediction of stable/unstable TSA, an interpretability analysis based on the Shapley additive explanation (SHAP), is embedded into both offline and online phases of the AutoML framework. Test results on IEEE 39-bus system, IEEE 118-bus system, and a practical large-scale power system, demonstrate that the proposed approach achieves more accurate and certain appropriate trust solutions while saving a substantial amount of time in comparison to other methods.

Keywords: transient stability; automated machine learning; interpretability; Bayesian optimization; SHAP; CatBoost; PMU



Citation: Liu, F.; Wang, X.; Li, T.; Huang, M.; Hu, T.; Wen, Y.; Su, Y. An Automated and Interpretable Machine Learning Scheme for Power System Transient Stability Assessment. *Energies* **2023**, *16*, 1956. <https://doi.org/10.3390/en16041956>

Academic Editor: J. C. Hernandez

Received: 27 December 2022

Revised: 4 February 2023

Accepted: 10 February 2023

Published: 16 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the ever-increasing penetration of uncertain renewable generation and bulk HVDC infeed into power systems, a considerable number of coal-fired synchronous generation units are being displaced [1]. The dynamic characteristics of modern hybrid AC/DC power systems become increasingly complicated, which have imposed unprecedented challenges on transient stability assessment (TSA) in power system planning and operation. A reduction in system inertia and insufficient dynamic grid support are the main factors that could negatively affect the transient response of power systems [2]. The transient stability of power systems is deteriorated by the integration of large amounts of uncertain and intermittent renewable generation [3].

Currently, commonly used TSA tools rely on analytical model-based methods, such as time-domain simulation (TDS) and direct methods [4]. TDS describes the transient response of a power system with a set of high-dimensional nonlinear differential-algebraic equations, and uses numerical integration methods to solve them. TDS is often considered the most accurate method available for TSA. However, the intensively computational burden for massive operation scenarios and credible contingencies make it intractable for online applications. From the energy point of view, direct methods evaluate transient stability via transient energy function (TEF) [5], extended equal area criterion (EEAC) [6,7], trajectory analysis [8,9], etc. However, due to the model simplifications required, direct methods

have major drawbacks of poor adaptability and conservative results when employed on practical large-scale power systems.

Nowadays, with widespread deployed phasor measurement units (PMU) in power systems, wide area measurement system (WAMS) and synchronous system measurement data are available. Because the traditional TSA methods cannot meet the needs of real-time TSA, it is urgent to develop a new data-driven scheme to support online TSA, i.e., making fast and accurate evaluation decisions based on the real-time power system operating status accessed from PMU measurements.

Owing to the rapid advancement of artificial intelligence (AI) techniques, it has been identified that machine learning (ML) can be used to implement online TSA. Over the past few years, the use of ML-based methods for TSA has been extensively explored, such as long short-term memory (LSTM) [10], extreme learning machine (ELM) [11], convolutional neural network (CNN) [12,13], decision tree (DT) [14], and hybrid ensemble learning [15]. A support vector machine (SVM)-based imbalanced correction method is proposed to address the problem that unstable samples rarely appear in practical systems giving rise to imbalanced samples [16]. To assess the transient stability of power systems by improving classification accuracy, an artificial intelligence method based on sparse dictionary learning is applied to transient stability assessment [17]. Because of the problem that differentiating between short-term voltage instability and transient rotor angle instability is not an easy task, a method based on graph attention networks (GATs) is proposed to solve it [18]. In [19], effective transient stability prediction in a data-driven manner is achieved by a transient stability assessment method which is based on high expressibility and low-depth quantum circuit.

Compared with conventional methods, ML-based methods do not need to model the power system. Instead, they use predetermined transient stability datasets to derive the relationship between system dynamic response and the corresponding stability conditions. New unlearned cases can be assessed with a minimum of computational effort with the relationship established [10]. Therefore, ML-based methods are advantageous for real-time fast TSA, lower data requirement, and the enhanced capability of knowledge discovery.

Despite ML-based TSA approaches having achieved certain developments, it is still difficult to directly apply them in operational planning and dispatch of practical power systems. The main reasons are two-fold:

- (i) To realize successful application, human experts are heavily involved in the four steps of ML, i.e., feature engineering, model selection, hyperparameter optimization, and performance evaluation, especially hyperparameter optimization [20]. Existing work on ML-based TSA assumes that the predefined input features and parameters of the trained model are always optimal for application. However, this assumption may not be true for power systems due to many practical issues, such as model updating, topology change, and variation of operating conditions. Since human experts are usually limited, to improve the accuracy and adaptability, many repeated manual feature adjustments and much heuristic parameter tuning are required during the debugging of ML-based TSA. This is highly tedious, inefficient, and time-consuming [21].
- (ii) On the other hand, the state-of-the-art ML-based TSA approaches have relatively poor transparency, because their models establish the mapping capability through black-box structures, such as deep learning and ensemble learning algorithms [12,13,15]. It is difficult for power system operators to interpret the behaviors of complex ML models and understand how particular decisions are made by these models. Hence, the uninterpretable results produced by ML-based TSA are often not actionable. This lack of interpretability has thus limited the use of ML-based TSA approaches in power industries.

In order to reduce the human effort needed for applying ML, this paper proposes an automated machine learning (AutoML)-based TSA approach. AutoML is naturally the intersection of automation and machine learning, aiming to achieve accurate decisions in a data-driven, automated, and objective manner [22]. In AutoML, feature engineering, model

learning, hyperparameter optimization, and performance evaluation can all be realized by computer programs, which makes ML much more accessible for online TSA. Furthermore, to make the AutoML-based method produce interpretable TSA predictions, we introduce Shapley additive explanation (SHAP) [23] into AutoML to explain why a certain prediction is made for a given case. Therefore, the proposed approach can be both automated and explainable for online use.

This work handles both the automation and interpretability issues of ML-based TSA. The main contributions of this paper are summarized as follows:

- (i) An AutoML scheme is designed for online TSA, which consists of auto-feature selection, CatBoost, Bayesian optimization, and performance evaluation. To ensure the best performance of the CatBoost-based ensemble learning, auto-feature selection and Bayesian optimization are introduced to automatically determine the best input features and optimal hyperparameters, respectively. By leveraging the adopted techniques, performance improvement is achieved while saving a substantial amount of time for manual feature adjustments and heuristic parameter tuning.
- (ii) To address the interpretability issue, this paper introduces SHAP to interpret the outputs of the proposed AutoML-based TSA. The impact of each input feature on AutoML's prediction is represented using SHAP values. The distribution of the SHAP values about the feature with a significant impact on the obtained prediction can provide additional insights into the decision interpretation.

The proposed interpretable AutoML-based TSA is tested on a provincial power system in China, in addition to two IEEE systems. Simulation results compared to other methods demonstrate the effectiveness and superiority of the proposed approach.

2. Automated Machine Learning

To enable faster deployment of ML-based TSA and reduce the heavy dependence on human experts, an AutoML scheme is proposed to achieve real-time evaluation. The proposed AutoML (shown in Figure 1) consists of auto-feature selection, CatBoost, Bayesian optimization, and performance evaluation. Firstly, by combining the feature importance and the corresponding threshold, input features with a high impact on the output are automatically selected from raw input characteristics. Then, to mine the mapping relationship between the input characteristics and the postfault transient stability statuses, CatBoost is introduced as a TSA classifier, which avoids the modeling of complex power systems. Meanwhile, Bayesian optimization is applied to determine the hyperparameters of the CatBoost-based learning model automatically. Finally, evaluation indices are built to evaluate the effectiveness of the AutoML model.

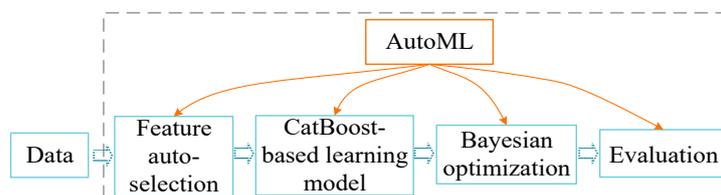


Figure 1. The scheme of AutoML in TSA.

2.1. Auto-Feature Selection

The dimensionality of the original input dataset increases dramatically with the system scale expansion, which could result in the dimension explosion issue. On the other hand, the use of input features that are effective can enable the ML model to learn faster and perform better in classification. Thus, we utilize an auto-feature selection technique to strengthen the mapping relationship between input features and transient stability statuses. To estimate the importance of each input feature, the split information during the offline training of CatBoost is fully made use of via calculation (1) [24]. To automatically filter

useless features, a threshold value p is defined. Only features with impact factor f_{IF} larger than p are selected as input data [25].

$$f_{IF} = \sum \left(v_1 - \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2} \right)^2 \cdot c_1 + \left(v_2 - \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2} \right)^2 \cdot c_2 \quad (1)$$

where c_1 and c_2 represent the total weight of objects in the left and right leaves, respectively; v_1 and v_2 represent the formula value in the left and right leaves, respectively.

2.2. CatBoost

CatBoost is a new ensemble ML method based on gradient boosting over decision trees [26]. With advantages of improved accuracy, categorical features support, multi-GPU support for training, and reduced overfitting, CatBoost has shown fast, scalable, and high performance for classification, regression, and other machine learning tasks [27].

Given a dataset $D = \{(x_i, y_i)\}_{i=1}^m$, where x_i is the input vector of sample i , y_i denotes the TSA result of credible contingency i , m denotes the number of samples. A series of decision tree models are built up based on D , written as [26]:

$$h(x) = \sum_{j=1}^J b_j I_{\{x \in R_j\}} \quad (2)$$

where $I(\cdot)$ is an indicator function; R_j is disjoint regions corresponding to leaves of the tree; J is the number of disjoint regions; b_j is the predicted value of R_j .

In the gradient boosting procedure, a sequence of approximations (F^t) is built iteratively in a greedy manner [26]:

$$F^t = F^{t-1} + \alpha h^t \quad (3)$$

where h^t is a tree in the t -th iteration process; α is the step size.

The loss function of CatBoost is given by [26]:

$$h^t = \operatorname{argmin}_{h \in H} \mathbb{E}(-\tau^t(x, y) - h(x))^2 \quad (4)$$

where τ is the gradient value; The objective of a learning task is to train a series of functions F^t to minimize the loss function. After T iterations, the final model can be obtained:

$$F(x) = \sum_{t=1}^T h^t \quad (5)$$

To overcome the residual shift caused by the target leakage issue, ordered boosting [26] is implemented in CatBoost. The principle of the ordered boosting has been described in [28].

2.3. Bayesian Optimization

The most fundamental objective in AutoML is to set hyperparameters to optimize the overall performance automatically. This paper introduces Bayesian optimization to realize automatic hyperparameter tuning of the CatBoost-based learning model.

Bayesian optimization is an emerging optimization framework for the global optimization of expensive black-box functions [29]. Acquisition function and probabilistic surrogate model are its two key components [30].

Bayesian optimization is used to obtain the global maximum value X^* of function $f(\lambda)$ in candidate hyperparameter set χ , and the corresponding optimal combination of CatBoost's hyperparameters is determined subsequently [31]:

$$\lambda^* = \operatorname{arg}_{\lambda \in \chi} \max f(\lambda) \quad (6)$$

Gaussian process (GP) is employed as the prior function to model the target function due to its closed-form computability and well-calibrated uncertainty estimates [30]. In GP, the combination of finite samples can be represented by [32]:

$$f(\lambda) \sim GP(m(\lambda), k(\lambda, \lambda')) \tag{7}$$

$$k(\lambda, \lambda') = \exp\left(-\frac{1}{2\theta} \|\lambda - \lambda'\|^2\right) \tag{8}$$

where $m(\lambda)$ is the mean function of λ ; $k(\lambda, \lambda')$ is the covariance function of λ ; θ is the length-scale parameter.

Having observed data $\{\lambda_{1:t}, f_{1:t}\}$, the joint distribution of observed data $f_{1:t}$ and prediction $f_{t+1} = f(\lambda_{t+1})$ agrees with the multivariate Gaussian distribution as shown in Equation (9):

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & \mathbf{k} \\ \mathbf{k}^T & k(\lambda_{t+1}, \lambda_{t+1}) \end{bmatrix}\right) \tag{9}$$

where $\mathbf{k} = [k(\lambda_{t+1}, \lambda_1) \ k(\lambda_{t+1}, \lambda_2) \ \dots \ k(\lambda_{t+1}, \lambda_t)]$; K is a covariance matrix; $N(\cdot)$ denotes the joint Gaussian distribution.

The predictive distribution $P(\cdot)$ at a new sampling point λ_{t+1} can be expressed by:

$$P(f_{t+1}|\lambda_{t+1}, \lambda_{1:t}, f_{1:t}) = N\left(\mu_t(\lambda_{t+1}), \sigma_t^2(\lambda_{t+1})\right) \tag{10}$$

where $\mu_t(\lambda_{t+1})$ and $\sigma_t^2(\lambda_{t+1})$ denote the predictive mean and variance, respectively.

The acquisition function of Bayesian optimization is the probability of improvement (POI):

$$PI(\lambda) = \phi\left(\frac{u(\lambda) - f(\lambda^+) - \varepsilon}{\delta(\lambda)}\right) \tag{11}$$

where $\phi(\cdot)$ is the normal cumulative distribution function; ε is a trade-off parameter; function $\delta(\lambda)$ is the standard deviation. By maximizing the acquisition function, the most prospective sampling points for the next search are computed.

2.4. Evaluation Indices

To reasonably evaluate the effectiveness of the proposed AutoML, *Accuracy* and *Recall* are used as evaluation indices. *Accuracy* and *Recall* represent the ability of AutoML to accurately identify all and unstable samples, respectively [33].

$$Accuracy = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{10}} \tag{12}$$

$$Recall = \frac{f_{00}}{f_{00} + f_{01}} \tag{13}$$

where f_{11} and f_{00} are the number of stable instances and unstable instances correctly classified, respectively; f_{10} and f_{01} are the number of stable instances and unstable instances falsely classified, respectively.

AUC, an important metric for measuring the classification performance under imbalanced conditions [34], is also used for the performance evaluation of the AutoML:

$$AUC = \frac{1}{n^+n^-} \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} I\left(f(x_i^+) > f(x_j^-)\right) \tag{14}$$

where n^+ and n^- are the number of instances predicted as stable and unstable, respectively; x_i^+ and x_j^- are, respectively, the instance predicted as stable and unstable; $f(x)$ is the AutoML classifier; $I(\cdot)$ is the indicator function.

3. Proposed AutoML-Based TSA with Interpretability Analysis

This paper establishes a nonlinear relationship between the system dynamic response and the transient stability using the proposed AutoML as the assessment model. The offline training stage and the online application stage form the entire framework of the proposed method. To help operators understand why a stable/unstable status prediction is made, interpretability analysis is embedded into both the offline and online stages of the AutoML framework, which helps to engender appropriate trust in the AutoML-based TSA predictions. Figure 2 provides an illustration of the flowchart of the proposed interpretable AutoML-based TSA method.

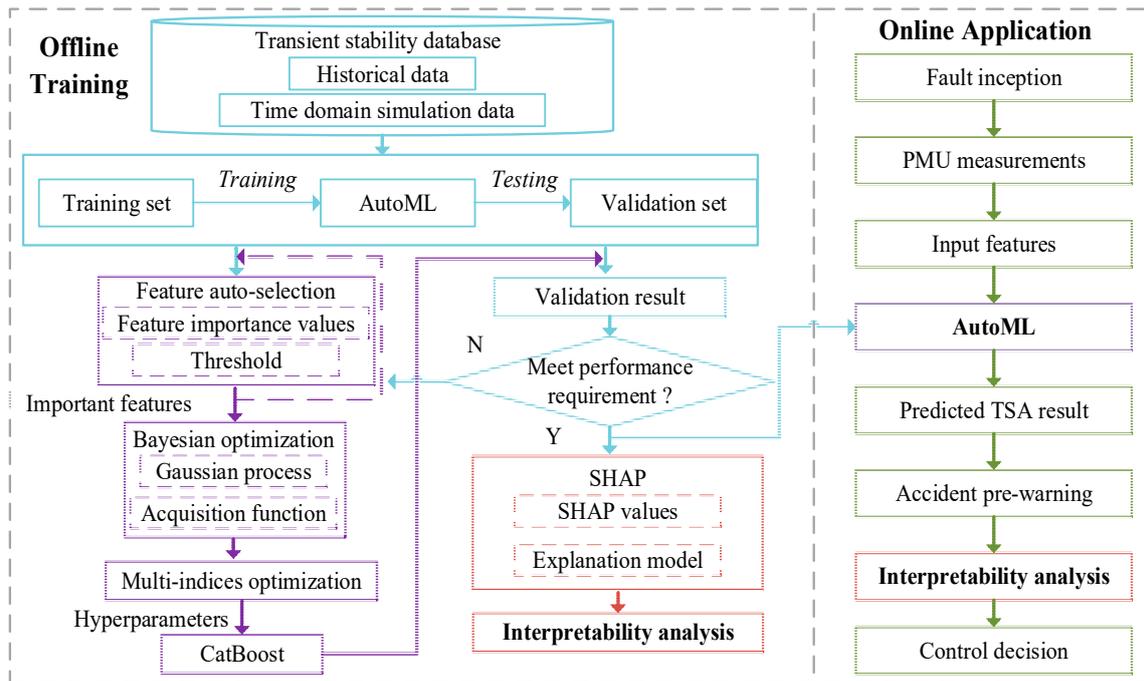


Figure 2. Flowchart of the proposed interpretable AutoML-based TSA method.

3.1. Offline Training

The following are the main steps of the offline training stage:

- (1) Historical PMU data and fault recordings are used as the dataset for offline training. When historical data is insufficient, simulation can be conducted to obtain a certain amount of dataset. The postfault system stability can be determined by the transient stability index (TSI) [16]:

$$\eta = (360^\circ - |\Delta\delta|_{\max}) / (360^\circ + |\Delta\delta|_{\max}) \quad (15)$$

where $|\Delta\delta|_{\max}$ denotes the absolute value of the maximum angle deviation of any two generators. The system is considered stable only when $\eta > 0$. Combining historical and offline simulation data, a sample dataset with diversity can be obtained.

- (2) The comprehensive dataset is divided into a training set and a validation set to test the performance of the AutoML-based TSA.
- (3) During the offline training process of CatBoost, the individual impact factor f_{IF} of each original input feature (i.e., active and reactive power of transmission lines, phase angle, voltage amplitude [11]) is calculated by Equation (1). Then, critical features are automatically selected as input features based on impact factor f_{IF} and threshold value p .
- (4) Bayesian optimization is used to automatically search the optimal hyperparameters of the CatBoost learning model. Because TSA is a non-equilibrium classification problem,

i.e., unstable samples rarely appear in practice, the AutoML's capability to recognize transient instability may be greatly affected if only one single indicator (i.e., accuracy) is used as the objective. To obtain better performance of the CatBoost learning model, we introduce multiple indices (i.e., *Accuracy*, *Recall*, *AUC*) into the objective function:

$$\text{Maximize } w_1 \cdot \text{Accuracy} + w_2 \cdot \text{Recall} + w_3 \cdot \text{AUC} \quad (16)$$

$$p_{\text{imin}} \leq p_i \leq p_{\text{imax}} \quad (17)$$

where w_j is the weight of the j -th evaluation index; p_1 , p_2 , p_3 , and p_4 represent the parameters of CatBoost, i.e., learning rate, l2 leaf regularization (*l2_leaf_reg*), the maximum depth of trees (*max_depth*), and the number of estimators (*n_estimators*). As the unstable status judged to be stable will cause cascading failure or even widespread blackout, the capability to accurately identify unstable samples (*Recall*) needs more attention in the hyperparameter optimization. Hence, the weight of *Recall* in the objective function (16) should be set larger than those of *Accuracy* and *AUC*. Through the GP and POI mechanism, Bayesian optimization iteratively validates hyperparameter combinations to maximize the objective function with multiple indices.

3.2. Online Application

When the AutoML-based TSA is well trained, it can be implemented online to predict system transient stability. Once a fault is detected, the critical features acquired from real-time PMU measurements are fed into the learned AutoML to perform TSA. If the system is insecure, operators will receive an early warning, and then they can quickly take emergency control actions to maintain the stability of the system. To improve the adaptability of AutoML, evaluated instances with accurate TSA assessment can be fed back to the offline stage to enrich the training database.

3.3. Interpretability Analysis

The increasing tension between the accuracy and interpretability of ML-based predictions has motivated the development of approaches that help users understand the predictions. SHAP [18] is a recently proposed novel unified framework for interpreting model predictions. Leveraging SHAP, the proposed AutoML's prediction can be explained as a sum of SHAP values corresponding to each feature. As shown in Figure 2, the original AutoML-based TSA model is approximated using an explanation model g , which is defined as a linear function of binary variables [23]:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z_i' \quad (18)$$

where z_i' represents a feature being observed ($z_i' = 1$) or unobserved ($z_i' = 0$); M is the number of input features; SHAP values ϕ_i (feature attribution values), explaining a prediction f , can be computed by [23]:

$$\phi_i = \sum_{s \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} \Delta_i(S) \quad (19)$$

$$\Delta_i(S) = f_x(S \cup \{i\}) - f_x(S) \quad (20)$$

As shown in Equation (19), SHAP values are evaluated by combining the conditional expectation $E[f(x) | x_S]$ with the classic Shapley values, where $f_x(x) = E[f(x) | x_S]$ corresponds to the desired outcome of the function conditioned on a subset S ; S is a subset of N with non-zero indices in z' ; N is the set of all input features; $\Delta_i(S)$ denotes the contribution of feature i to S .

Specifically, the attribution values of the explanation model g match the AutoML-based TSA model for a specific input, which can be represented as [23]:

$$f(x) = \phi_0 + \sum_{i=1}^M \phi_i \quad (21)$$

where $\phi_0(f,x) = E[f(x)]$ is the desired value of the AutoML-based TSA model over the transient stability dataset.

Aided by the explanation model g and SHAP values, the impacts of each input feature on the transient stability prediction are evaluated. Then, the AutoML-based TSA prediction can be explained based on the summation of SHAP values. Therefore, features with significant impact on the stability prediction can be easily identified by comparing their SHAP values. The process of the interpretability analysis of the AutoML-based TSA is illustrated in Algorithm 1.

Algorithm 1. The process of interpretability analysis

Input: $f(x)$ —Trained AutoML model; X_{train} —Trained data; X_e —A stable or unstable instance that needs to be explained.

Output: SHAP values for each input system features

```

1: Train an explainer through  $(f(x), X_{train})$ 
2: for each  $i \in$  input features( $X_{train}$ ) do
3:   Use EXPVALUE method to estimate  $E[f(x) | x_S]$ 
4:   procedure EXPVALUE( $x, S, tree$ )
5:     procedure G( $j, w$ )
6:       if  $v_j \neq$  internal then
7:         return  $w \cdot v_j$ 
8:       else
9:         if  $d_j \in S$  then
10:          return  $G(a_j, w)$  if  $x_{d_j} \leq t_j$  else  $G(b_j, w)$ 
11:        else
12:          return  $G(a_j, wr_{a_j}/r_j) + G(b_j, wr_{b_j}/r_j)$ 
13:        return G(1,1)
14:      end procedure
15:    end procedure
14:    Calculating attribute values  $\phi_i$  by (19)
15:    Establishing an explanation model  $g$  by (18)
16:    return  $g$ 
17: Interpretability for  $X_e$ 
18: for each  $i \in$  input system features( $X_e$ ) do
19:   SHAP values [ $i$ ]  $\leftarrow$   $explainer.shapvalues(X_e, i)$ 
20: return SHAP values [ $i$ ]
21: A certain predicted status of the power system can be interpreted based on the sum of SHAP values [ $i$ ].

```

4. Case Study

To demonstrate the effectiveness of the proposed approach, two IEEE test systems (IEEE 39-bus and 118-bus systems) and a large-scale provincial power system (PPG) in China were utilized for simulation. The IEEE 39-bus system has 39 buses, 10 generators, 19 loads, and 46 branches. The IEEE 118-bus system has 118 buses, 54 thermal units, 91 loads, and 186 branches. The practical grid has 1154 buses, 208 generators, and 451 loads in 2020. Furthermore, ten HVDC links are fed out to deliver electric power to other provincial systems. PSD-BPA software is used for data generation. The proposed interpretable AutoML-based TSA was deployed on the Python 3.7 platform. The SHAP values of AutoML models were calculated and visualized with the help of a python-based SHAP library [35]. All testing was performed on a personal computer with Intel Core i5-10400F 2.9 GHz CPU, 16.0 GB RAM.

4.1. Database Generation

TDS was implemented on the PSD-BPA software to generate input dataset. As for the IEEE 39-bus and IEEE 118-bus systems, ten load levels (75, 80, 85, 90, 95, 100, 105, 110, 115, 120%) were considered. All three-phase to ground faults at any bus were considered in the contingency set, and the faults were assumed occurring on each transmission line at four locations (at 0, 25, 50, and 70% of the length); Each fault was set to start at 1.0 s and being cleared at two cases: 1.1 s and 1.2 s. The duration of each simulation was 20 s. For the practical grid, four operation modes were considered, all three-phase to ground faults and HVDC blockings were analyzed. The numbers of stable and unstable instances in each dataset of the three test systems are listed in Table 1. For each test system, the dataset was split into a training sample set and a testing sample set in a ratio of 8:2.

Table 1. Database of the three test systems.

System	No. of Instances	Unstable Instances	Stable Instances	Raw Features	Selected Features
39-bus	3668	973	2695	170	36
118-bus	6781	1394	5387	586	37
PPG	9495	2705	7920	828	45

4.2. The Selection of Important Features

Necessary features were automatically selected by the AutoML using Equation (1), in which the threshold p is set at 0.7. The fifth and sixth columns of Table 1 show the dimensions of raw features and selected important features. One can observe that for the IEEE 39-bus system, the dimensions of raw features were 170, but most of them were useless, which posed great computational burden. By leveraging the auto-feature selection mechanism, only 36-dimensional features were retained. Similarly, dimensions of selected important input features for the IEEE 118-bus system and the practical grid were 37 (6.3% of the raw features) and 45 (5.4% of the raw features), respectively. Consequently, the dimension explosion issue when applying AutoML to large-scale systems could be avoided.

4.3. Bayesian Optimization

The effectiveness of utilizing Bayesian optimization (BO) in the AutoML to find optimal hyperparameters was compared with the other two traditional methods, which were grid search (GS) and randomized search (RS). All tests were carried out using the same training and testing dataset. The weights of *Accuracy*, *Recall*, and *AUC* in the objective function of BO were set at 0.25, 0.5 and 0.25, respectively.

The hyperparameter tuning process of RS and BO on the IEEE 39-bus system is shown in Figure 3. To visualize the process of the sequential selection of hyperparameters, the impact of the combination of any two hyperparameters (learning rate, $l2_leaf_reg$, and $n_estimators$) on the evaluation indices (*Accuracy*, *Recall*, *AUC*) was analyzed. The background of Figure 3a–d are colored based on the values of *Accuracy*, *Recall*, and *AUC*, respectively. It can be seen that hyperparameters determined by RS were irregularly distributed in Figure 3a–d, which results in the optimal hyperparameters being easily skipped. BO determines hyperparameters through GP and POI, hence it is more persuasive. Compared with RS, there were more hyperparameters determined by BO located in the optimal space. According to Figure 3, the appropriate search space of learning rate, $l2_leaf_reg$, and $n_estimators$ could be set at [0.01, 0.3], [1, 10], and [40, 120], respectively.

The CPU time and evaluation indices produced by BO, GS, and RS optimized CatBoost are compared and shown in Table 2. For the IEEE 39-bus system, the Bayesian optimized CatBoost had an Accuracy of 98.77%, Recall of 98.44%, and AUC of 98.67%, which was 0.41, 1.57 and 1.04% higher than the RS optimized CatBoost, respectively. GS traverses all parameter combinations to find the best hyperparameters, which makes it easy to encounter the combination explosion issue. Although evaluation indices obtained by GS are closer to

those obtained by BO, it is extremely time-consuming to run GS (the total time required is 25 h). The CPU time (93.1 s) of performing BO was only 0.1% of the GS. By mining the rules on transient data, BO can efficiently find optimal hyperparameters, which saves substantial amounts of time and avoids the inefficiency of heuristic parameter tuning.

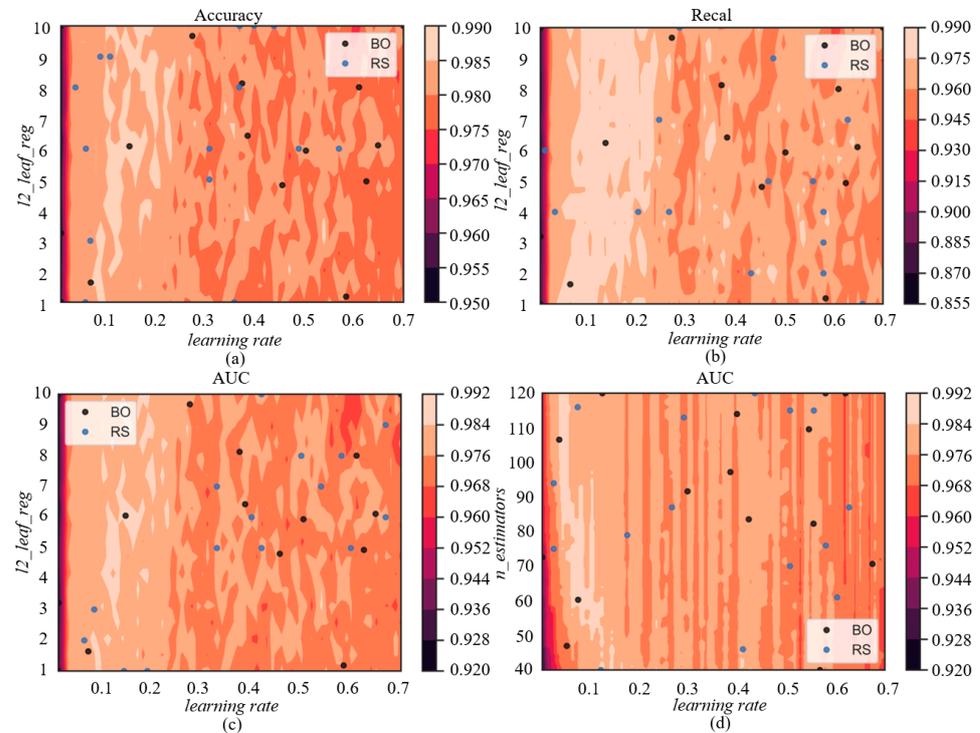


Figure 3. (a) Distribution of Accuracy impacted by learning rate and l2_leaf_reg; (b) distribution of Recall impacted by learning rate and l2_leaf_reg; (c) distribution of AUC impacted by learning rate and l2_leaf_reg; (d) distribution of AUC impacted by learning rate and n_estimators.

Table 2. Performance comparison of different parameter tuning methods.

Method	System	Tuning Time	Accuracy/%	Recall/%	AUC/%
BO	39-bus	93.1 s	98.77	98.44	98.67
GS	39-bus	25 h	98.36	97.39	97.86
RS	39-bus	214.7 s	98.36	96.87	97.63
BO	118-bus	102.3 s	97.19	96.80	97.08
GS	118-bus	25 h	97.13	96.09	96.74
RS	118-bus	220.3 s	96.46	96.08	96.32
BO	PPG	114.1 s	97.85	96.22	97.23
GS	PPG	26 h	97.51	95.41	96.84
RS	PPG	256.8 s	97.18	94.71	96.39

Tests concerning the effectiveness of BO were also performed on the IEEE 118-bus system and the practical grid, and the results are provided in Figure 4 and Table 2 (rows 4–9). As shown in Table 2, the CPU time of performing BO was significantly lower than that of GS and BS, while the CatBoost can achieve the best performance based on hyperparameters found by BO. Figure 4 shows that the highest range of the AUC index achieved by BO optimized CatBoost was [96%, 97.5%] for both the 118-bus system and the practical grid. The space of this range produced by BO (drawn with light color) is quite large in Figure 4a,b, revealing that BO-optimized CatBoost can accurately classify TSA statuses when applied to large-scale power systems.

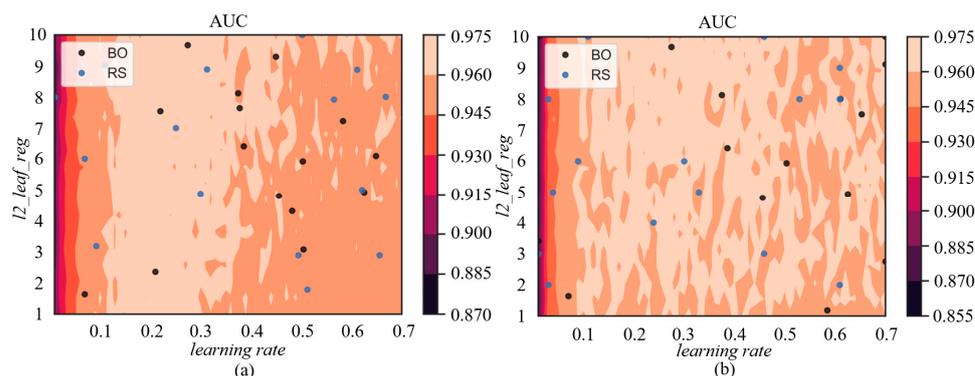


Figure 4. (a) Distribution of AUC impacted by learning rate and $l2_leaf_reg$ on the 118-bus system; (b) distribution of AUC impacted by learning rate and $l2_leaf_reg$ on the practical grid.

4.4. Comparison of the Classifiers' Performance

The proposed AutoML-based TSA has been compared with some other ML methods, such as LSTM, XGBoost, RF, and DT. Table 3 shows the evaluation indices obtained by the five methods in all test systems. For all three test systems, the proposed AutoML-based TSA had the highest Accuracy, Recall, and AUC in comparison with the other four ML-based TSA methods. For example, for the practical power system (Yunan power grid), AutoML achieved an Accuracy of 97.77%, Recall of 96.22%, and AUC of 97.23%. Note that, for the three systems, the Recall of the proposed AutoML-based TSA were all over 96%, demonstrating that AutoML can accurately identify the unstable operating status immediately following a fault. Because of the simple structure, the DT-based TSA method learns restrictedly and had the lowest level of Accuracy, Recall, and AUC for the three systems. XGBoost and LSTM-based TSA approaches have better performance than DT, but evaluation indices obtained by them are all lower than the proposed AutoML-based TSA. As the scale of the system expanded, the Recall index produced by LSTM, XGBoost, RF, and DT-based TSA decreased significantly. Taking the practical grid as an example, the Recall index produced by LSTM and XGBoost were around 93%, while RF and DT both had a Recall lower than 90%. Hence, the ability of these methods to identify unstable samples was not satisfactory when applied to large-scale systems. The proposed AutoML-based TSA still maintained good performance. The averages of the three evaluation indices produced by AutoML on the two large systems exceeded 97%.

Table 3. Results obtained by different methods on the three power systems.

Method	System	Accuracy/%	Recall/%	AUC/%
AutoML	39-bus	98.77	98.44	98.67
LSTM	39-bus	97.57	95.92	97.04
XGBoost	39-bus	97.68	95.83	98.09
RF	39-bus	96.19	94.55	95.71
DT	39-bus	95.43	91.51	94.39
AutoML	118-bus	97.20	96.80	97.05
LSTM	118-bus	95.58	93.23	94.71
XGBoost	118-bus	95.51	95.30	95.45
RF	118-bus	94.85	94.67	94.78
DT	118-bus	93.37	90.75	92.40
AutoML	PPG	97.85	96.22	97.23
LSTM	PPG	96.57	92.94	95.42
XGBoost	PPG	97.37	93.47	96.13
RF	PPG	95.54	89.06	93.47
DT	PPG	94.23	89.94	92.87

Figure 5 shows the 2D visualization of the predicted results obtained by AutoML, XGBoost, and RF-based TSA methods on the IEEE 39-bus system. The background of

Figure 5 is colored based on the probabilistic prediction of these methods. In the evaluation results of CatBoost-based AutoML, the pink background of isolated unstable samples is more obvious, indicating that compared with XGBoost and RF, the proposed approach can better distinguish isolated unstable samples, which is helpful to avoid misjudging the unstable operation state of the system after a disturbance accident.

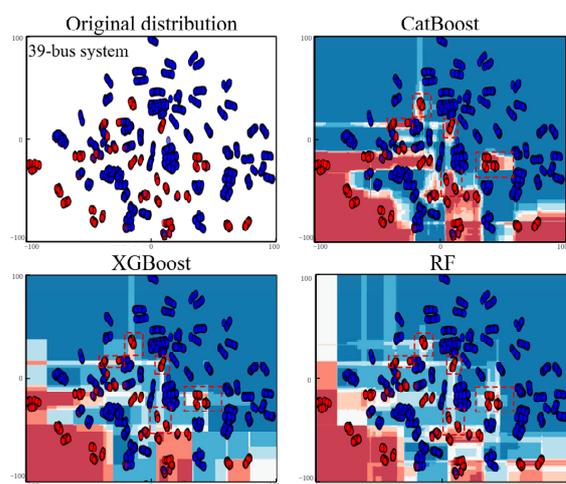


Figure 5. Visualization of prediction results. Red square in the figure represents the isolated unstable samples.

In real-time, the training time and testing time of ML-based TSA tools are of great concern. Table 4 compares the CPU time of AutoML and TDS on the three test systems. The proposed AutoML-based TSA is computationally efficient for both offline training and online testing. Furthermore, the effect of system scale on the model training is not significant, because redundant features are eliminated by using the AutoML. As for TDS, the excessive computational time required prevents its online use. Specifically, the CPU time of TDS is 17.7 h on the practical grid. However, the proposed AutoML-based TSA only needs 2.27 s, which is only 0.003% of that required by TDS. Thus, AutoML is more suitable for online TSA for large-scale power systems, which can help to save more time for subsequent control actions.

Table 4. Solution time on the three power systems.

Systems	Training Time of AutoML	Testing Time of AutoML	Time of TDS
39-bus system	1.18 s	0.02 s	0.69 h
118-bus system	2.16 s	0.02 s	1.13 h
PPG	2.27 s	0.01 s	17.71 h

4.5. Interpretability Analysis

SHAP was embedded into AutoML to interpret the behaviors of the AutoML-based TSA method and understand how a particular decision (the system will be transient stable/unstable) for a given case is made.

For clarity, transient stability prediction following a three-phase short-circuit fault occurring at line 2-30 in the IEEE-39 bus system was taken as an example for interpretability analysis. The fault lasted 0.1 s, and the system became unstable after clearing the fault. Figure 6 explains why the system is classified as instability by the AutoML-based TSA method based on SHAP values. The red-colored features in Figure 6 pushed the risk of transient instability lower, while the blue-colored features pushed the risk higher. Because the influence of the blue-colored features on the output prediction of the contingency was higher, AutoML predicted that the system would lose transient stability after the fault. As shown in Figure 6, input features θ_3 , θ_2 , P_{2-30} , and Q_{2-25} had larger SHAP values, hence

they had much more impact on the obtained prediction. These features were all directed to bus 2, which was consistent with the fault.

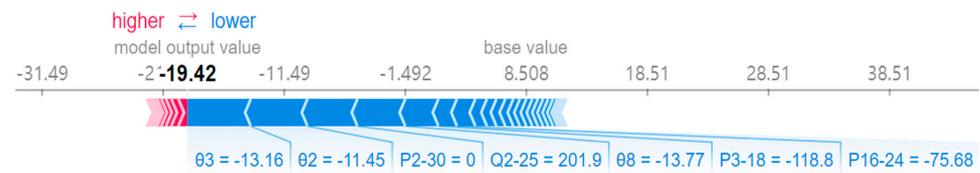


Figure 6. Interpretability of a prediction produced by AutoML on the IEEE-39.

To further interpret the effect of these features on the prediction, the SHAP dependence plots were used. Figure 7a–d show how SHAP values varied as a function of features θ_3 , θ_2 , P_{2-30} , and Q_{2-25} , respectively. The color corresponds to the values of the feature from low (blue) to high (red). The lower the SHAP values were, the higher risk of transient instability. Coloring each dot by θ_3 and θ_2 revealed that transient instability was more alarming when their values were low. When the value of P_{2-30} belonged to the normal range $[-220, -110]$ affected by the active power of generator G32, the risk of transient instability was low. Furthermore, the system will lose stability when the value of Q_{2-30} is higher than 70. Consequently, the values of θ_3 , θ_2 , P_{2-30} , and Q_{2-25} were all in the range where the risk of transient instability was high. Therefore, the unstable result obtained by the AutoML-based TSA was reasonable and credible.

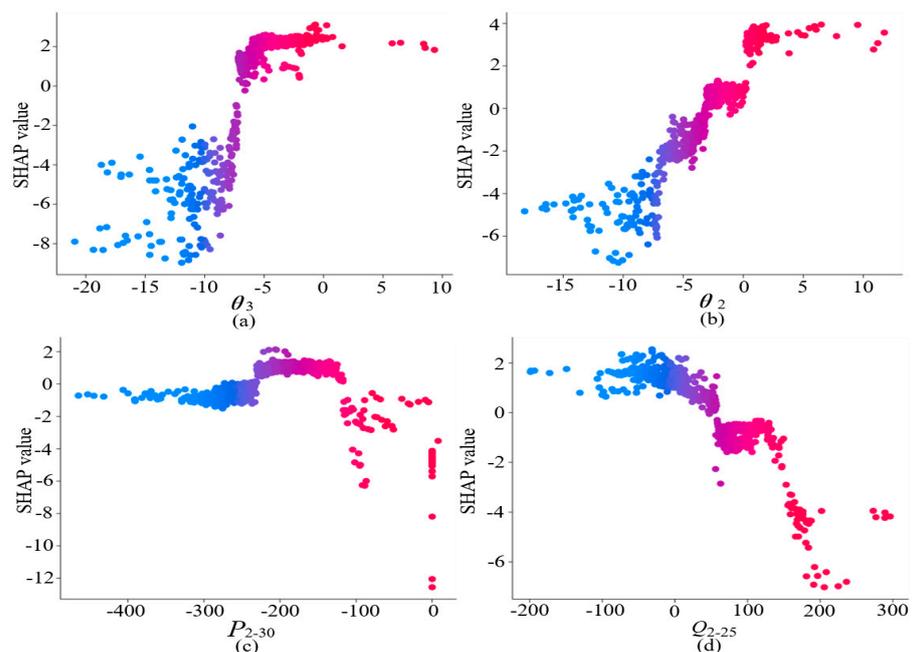


Figure 7. (a) SHAP dependence of θ_3 ; (b) SHAP dependence of θ_2 ; (c) SHAP dependence of P_{2-30} ; (d) SHAP dependence of Q_{2-35} .

Figure 8 explains why the practical grid was considered unstable after a specific fault, i.e., lines 37 and 38 were disconnected following the three-phase short-circuit fault (N-2). Since lines 37 and 38 were disconnected after the fault, the power flow values of P_{37} and P_{38} became 0. Figure 8 reveals that P_{37} and P_{38} significantly pushed the risk of instability higher, which reflected the credibility of the explanation model. Prior to the fault, the values of P_{39} and P_{40} were -278.56 and -282.91 , respectively, whereas, the values after the fault for P_{39} and P_{40} were -774.1 (277% of the raw value) and -802.4 (283% of the raw value), respectively. The values of P_{39} and P_{40} changed significantly after the fault was removed. It can be inferred that this fault caused an enormous impact on the surrounding

lines. Due to the seriousness of the fault, AutoML predicted that the practical grid was unstable after the fault.

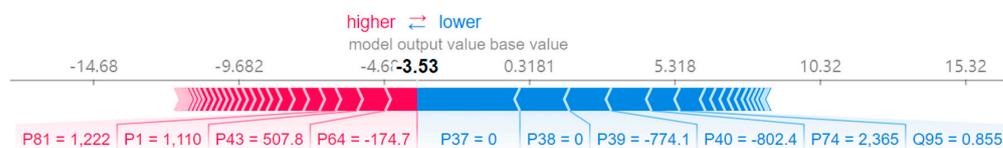


Figure 8. Interpretability of a prediction produced by AutoML on the Yunnan.

Figure 9 shows the distribution of SHAP values about P_{37} and P_{38} . As we can see from Figure 9, the risk of transient instability is high when the value of P_{37} or P_{38} is 0, reflecting the vulnerability of the corresponding lines.

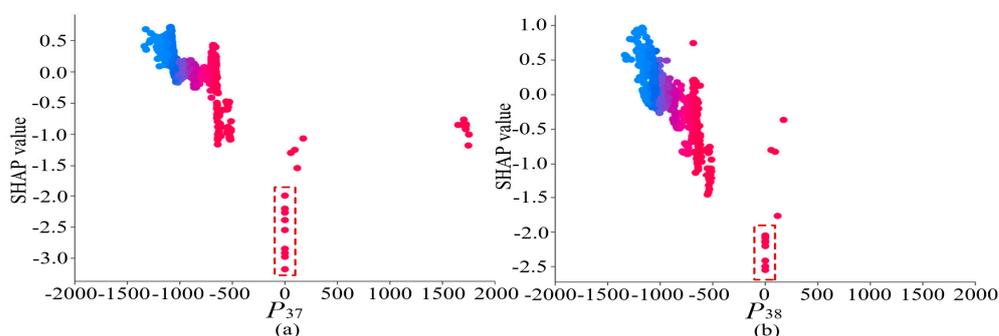


Figure 9. (a) SHAP dependence of P_{37} ; (b) SHAP dependence of P_{38} . The red square means the risk of transient instability is high when the value of P_{37} or P_{38} is 0.

5. Conclusions

An automated ML scheme (AutoML) was proposed to realize power system online TSA. Under this framework, auto-feature selection and Bayesian optimization were respectively introduced to automatically determine the crucial input features and optimal hyperparameters of the CatBoost-based TSA learning model. Multiple evaluation indices were considered to improve the performance of Bayesian optimization. By leveraging these techniques, faster deployment of the ML-based TSA could be achieved, and the dependence on human expertise is significantly reduced.

To help operators understand why a stable/unstable status prediction is made, interpretability analysis was embedded into both offline and online stages of the AutoML framework. By leveraging the SHAP technique, the proposed AutoML's TSA prediction can be explained as a sum of SHAP values corresponding to each input feature. Furthermore, additional insight into the independent effects of important features can be gained by analyzing the distribution of SHAP values.

Test results showed that the evaluation indices obtained by the proposed AutoML-based TSA in all test systems were the best over the other methods, and appropriate trust in the AutoML-based TSA predictions can be achieved. Furthermore, the proposed AutoML-based TSA saves a great deal of solution time and is applicable to large-scale power systems. In future work, an in-depth analysis of the impact of RES penetration on the transient stability of the power system may be conducted.

Author Contributions: Conceptualization, F.L. and T.L.; methodology, X.W.; validation, M.H. and T.H.; formal analysis, X.W.; investigation, M.H.; resources, Y.S.; data curation, F.L.; writing—original draft preparation, M.H. and X.W.; writing—review and editing, Y.W.; supervision, Y.W.; project administration, F.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Program of State Grid Corporation, grant number 521996220005, and the Huxiang Young Talents Science and Technology Innovation Program, grant number 2020RC3015.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is unavailable due to ethical restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wen, Y.; Zhan, J.; Chung, C.Y.; Li, W. Frequency Stability Enhancement of Integrated AC/VSC-MTDC Systems With Massive Infeed of Offshore Wind Generation. *IEEE Trans. Power Syst.* **2018**, *33*, 5135–5146. [CrossRef]
2. Bueno, P.G.; Hernández, J.C.; Ruiz-Rodríguez, F.J. Stability assessment for transmission systems with large utility-scale photovoltaic units. *IET Renew. Power Gener.* **2016**, *10*, 584–597. [CrossRef]
3. Liu, Y.; Wang, J.; Yue, Z. Improved Multi-point estimation method based probabilistic transient stability assessment for power system with wind power. *Int. J. Electr. Power Energy Syst.* **2022**, *142*, 108283. [CrossRef]
4. Kundur, P. *Power System Stability and Control*; McGraw-Hill, Inc.: New York, NY, USA, 1994; ISBN 978-0070-35958-1.
5. Rahimi, F.A.; Lauby, M.G.; Wrubel, J.N.; Lee, K.L. Evaluation of the transient energy function method for on-line dynamic security analysis. *IEEE Trans. Power Syst.* **1993**, *8*, 497–507. [CrossRef]
6. Xue, Y.; Van Cutsem, T.; Ribbens-Pavella, M. Extended equal area criterion justifications, generalizations, applications. *IEEE Trans. Power Syst.* **1989**, *4*, 44–52. [CrossRef]
7. Tang, L.; Sun, W. An Automated Transient Stability Constrained Optimal Power Flow Based on Trajectory Sensitivity Analysis. *IEEE Trans. Power Syst.* **2017**, *32*, 590–599. [CrossRef]
8. Xu, Y.; Dong, Z.Y.; Zhang, R.; Xue, Y.; Hill, D.J. A Decomposition-Based Practical Approach to Transient Stability-Constrained Unit Commitment. *IEEE Trans. Power Syst.* **2015**, *30*, 1455–1464. [CrossRef]
9. Gurusinghe, D.R.; Rajapakse, A.D. Post-Disturbance Transient Stability Status Prediction Using Synchrophasor Measurements. *IEEE Trans. Power Syst.* **2016**, *31*, 3656–3664. [CrossRef]
10. Yu, J.J.Q.; Hill, D.J.; Lam, A.Y.S.; Gu, J.; Li, V.O.K. Intelligent Time-Adaptive Transient Stability Assessment System. *IEEE Trans. Power Syst.* **2018**, *33*, 1049–1058. [CrossRef]
11. Xu, Y.; Dong, Z.; Meng, K.; Zhang, R.; Wong, K. Real-time transient stability assessment model using extreme learning machine. *IET Gener. Transm. Distrib.* **2011**, *5*, 314–322. [CrossRef]
12. Yan, R.; Geng, G.; Jiang, Q.; Li, Y. Fast Transient Stability Batch Assessment Using Cascaded Convolutional Neural Networks. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2–6 August 2020; IEEE: Montreal, QC, Canada, 2020; pp. 2802–2813.
13. Zhu, L.; Hill, D.J.; Lu, C. Hierarchical Deep Learning Machine for Power System Online Transient Stability Prediction. *IEEE Trans. Power Syst.* **2020**, *35*, 2399–2411. [CrossRef]
14. He, M.; Vittal, V.; Zhang, J. Online dynamic security assessment with missing pmu measurements: A data mining approach. *IEEE Trans. Power Syst.* **2013**, *28*, 1969–1977. [CrossRef]
15. Zhang, Y.; Xu, Y.; Dong, Z.Y. Robust Ensemble Data Analytics for Incomplete PMU Measurements-Based Power System Stability Assessment. *IEEE Trans. Power Syst.* **2018**, *33*, 1124–1126. [CrossRef]
16. Wang, H.; Hu, L.; Zhang, Y. SVM based imbalanced correction method for Power Systems Transient stability evaluation. *ISA Trans.* **2022**. [CrossRef] [PubMed]
17. Wang, Q.; Pang, C.; Qian, C. Sparse Dictionary Learning for Transient Stability Assessment. *Front. Energy Res.* **2022**, *10*, 932770. [CrossRef]
18. Zhang, R.; Yao, W.; Shi, Z.; Zeng, L.; Tang, Y.; Wen, J. A graph attention networks-based model to distinguish the transient rotor angle instability and short-term voltage instability in power systems. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107783. [CrossRef]
19. Zhou, Y.; Zhang, P. Noise-Resilient Quantum Machine Learning for Stability Assessment of Power Systems. *IEEE Trans. Power Syst.* **2023**, *38*, 475–487. [CrossRef]
20. He, X.; Zhao, K.; Chu, X. AutoML: A survey of the state-of-the-art. *Knowl.-Based Syst.* **2021**, *212*, 106622. [CrossRef]
21. Li, Z.; Guo, H.; Wang, W.M.; Guan, Y.; Barenji, A.V.; Huang, G.Q.; McFall, K.S.; Chen, X. A Blockchain and AutoML Approach for Open and Automated Customer Service. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3642–3651. [CrossRef]
22. Tuggener, L.; Amirian, M.; Rombach, K.; Lörwald, S.; Varlet, A.; Westermann, C.; Stadelmann, T. Automated Machine Learning in Practice: State of the Art and Recent Results. In Proceedings of the 2019 6th Swiss Conference on Data Science (SDS), Bern, Switzerland, 14 June 2019; pp. 31–36.
23. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
24. Feature Importance. Available online: <https://catboost.ai/en/docs/concepts/fstr> (accessed on 16 March 2022).

25. Anuradha, P.; David, V.K. Feature Selection and Prediction of Heart Diseases Using Gradient Boosting Algorithms. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 711–717.
26. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
27. Punmiya, R.; Choe, S. Energy Theft Detection Using Gradient Boosting Theft Detector With Feature Engineering-Based Preprocessing. *IEEE Trans. Smart Grid* **2019**, *10*, 2326–2329. [[CrossRef](#)]
28. Deng, K.; Zhang, X.; Cheng, Y.; Zheng, Z.; Jiang, F.; Liu, W.; Peng, J. A remaining useful life prediction method with long-short term feature processing for aircraft engines. *Appl. Soft Comput.* **2020**, *93*, 106344. [[CrossRef](#)]
29. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [[CrossRef](#)]
30. van Hoof, J.; Vanschoren, J. Hyperboost: Hyperparameter Optimization by Gradient Boosting Surrogate Models 2021. *arXiv* **2021**, arXiv:2101.02289.
31. Wang, R.; Liu, Y.; Ye, X.; Tang, Q.; Gou, J.; Huang, M.; Wen, Y. Power System Transient Stability Assessment Based on Bayesian Optimized LightGBM. In Proceedings of the 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2), Changsha, China, 8–10 November 2019; IEEE: Changsha, China, 2019; pp. 263–268.
32. Lyu, W.; Xue, P.; Yang, F.; Yan, C.; Hong, Z.; Zeng, X.; Zhou, D. An Efficient Bayesian Optimization Approach for Automated Optimization of Analog Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 1954–1967. [[CrossRef](#)]
33. Li, X.; Liu, C.; Guo, P.; Liu, S.; Ning, J. Deep learning-based transient stability assessment framework for large-scale modern power system. *Int. J. Electr. Power Energy Syst.* **2022**, *139*, 108010. [[CrossRef](#)]
34. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [[CrossRef](#)]
35. Shap PyPI. Available online: <https://pypi.org/project/shap/> (accessed on 22 May 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.