An Interpretable Deep Learning Method for Power System Transient Stability Assessment via Tree Regularization

Chao Ren, *Student Member, IEEE*, Yan Xu, *Senior Member, IEEE*, and Rui Zhang, *Member, IEEE*

***Abstract*—Deep learning (DL) techniques have shown promising performance for designing data-driven power system transient stability assessment (TSA) models. However, due to the deep structure of the DL, the resulting model is always a black-box and hard to explain, which hinders its practical adoption by the industry. This paper proposes an interpretable DL-based TSA model to balance the TSA accuracy and transparency. The proposed method combines the strong nonlinear modelling capability of a deep neural network and the interpretability of a decision tree (DT). Through regularizing DL-based model with the average DT path length in the training process, the proposed interpretable DL-based TSA method can visually explain the TSA decision-making process. Simulation results have shown that the proposed method can deliver highly accurate TSA results and interpretable TSA decision-making rules, which can be used for designing preventive control actions.**

***Index Terms*—Data-driven, deep learning, transient stability assessment, gated recurrent unit, interpretability, transparency, tree regularization.**

I. Introduction

TRANSIENT stability assessment (TSA) aims to evaluate the stability status of a power system under a set of credible contingencies [1]. With the growing integration of intermittent renewable energy sources (such as wind and solar power) and the active demand-side response, the operating condition of the power system becomes much more varying and difficult to predict. In order to timely evaluate the stability level of the power system, the online TSA is a necessity.

As a promising strategy, the machine learning (ML)-based data-driven methods have been proposed for online TSA [2]. The principle is to train a ML model with the selected features from a TSA database. Once well trained, such ML-based TSA model can be directly applied for real-time stability assessment. Compared with traditional model-based methods, the data-driven methods have many merits, including less data requirement, faster assessment speed, strong knowledge discovery ability, and broad extensibility capability [3], etc. Traditional ML algorithms, such as support vector machine, decision tree (DT), and randomized learning have been successfully used in the literature [4], [5].

In recent years, with the continuous advancement of artificial intelligence research, the deep learning (DL) techniques have shown the higher accuracy performance than the conventional ML algorithms. The principle of DL is to use multi-layer neural networks (NN) to progressively extract higher-level features from the input data, where each layer learns to transform its input data into a more abstract and composite representation [6]. Based on its deep structure, more knowledge can be extracted to approximate the distribution of the training data. Compared with the traditional ML algorithms, the DL-based models have the stronger nonlinear modeling capability and tend to be more accurate, especially for more complex data structure.

In the literature, many specific DL techniques have been used for TSA [4], such as deep belief networks (DBNs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs), etc. For DBN-based methods, Ref. [7] uses a local linear interpreter to constraint the DBN but cannot give a detailed explanation for the whole TSA model. CNNs have the better feature extraction ability to speed-up the training process and make the final results accurate. For CNN-based methods, Ref. [8] uses a twin convolutional SVM network to predict the transient stability status, which can mine the internal structure of trajectory features appropriately. In [9], a CNN-based ensemble method is used to train a transient stability predictor, which can be updated rapidly with the informative and representative samples before the operating conditions or topology of the power system change greatly. In [10], a hierarchical and self-adaptive CNN-based method is used to determine the post-disturbance transient stability of the system via integrated decision-making rule for multiple CNNs. In [11], the cascaded CNNs combining with time-domain simulation can improve the computational efficiency for pre-fault transient stability assessment via extracting features from different TDS time intervals. In [12], power system snapshots are represented as the images, hence can be directly

Manuscript received 10 December 2020; revised 25 March 2021, 24 June 2021, and 8 September 2021; accepted 23 October 2021. Date of publication 10 December 2021; date of current version 19 August 2022. This work was supported by the Ministry of Education (MOE), Republic of Singapore, under Grant AcRF TIER 1 2019-T1-001-069 (RG75/19). Paper no. TPWRS-02018-2020. *(Corresponding author: Yan Xu.)*

Chao Ren is with the Interdisciplinary Graduate School, Nanyang Technological University 639798, Singapore (e-mail: renc0003@e.ntu.edu.sg).

Yan Xu is with the School of Electrical and Electronic Engineering, Nanyang Technological University 639798, Singapore (e-mail: eeyanxu@gmail.com).

Rui Zhang is with the University of New South Wales, Sydney NSW 2052, Australia (e-mail: rachel.zhang1@unsw.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2021.3133611.

Digital Object Identifier 10.1109/TPWRS.2021.3133611

applied by CNN to predict the stability status. RNNs can use their internal state to process variable length sequences of inputs, which considers spatial and temporal correlations, such as long short-term memory (LSTM) and gated recurrent units (GRU). In [13], a LSTM-based method is proposed to classify the transient stability status with multiple time-step algebraic variables. In [14], a self-adaptive learning model-based LSTM is applied for online TSA, which can extract both spatial and temporal data dependency from the input features. GRU-based methods in [15] and [16] are applied for real-time transient instability prediction without needing fault information, which are robust to measurement noise and topology changes. Other latest DL technique, such as generative adversarial networks (GAN) [17], is applied for online dynamic security assessment (DSA) with incomplete PMU measurements without considering topologies change.

Although these DL-based methods have achieved the excellent TSA accuracy, the users/system operators cannot completely understand the obtained TSA results because such DL-based models are black-box and not interpretable. Thus, the results may not be convincing and difficult to provide effective decision support knowledge for emergency control design.

On the other hand, DT is an interpretable ML technique [18]. It uses a tree structure to decide the final results via the feature inference. The paths from the root node to the leaf nodes represent the decision-making rules. Compared with DL-based models, DT belongs to a white-box model. With its nodes, thresholds and paths, the decision-making process is transparent and interpretable. E.g., to predict the power system stability status, one can go down each node of the tree structure and understand which features can be used to make the predictions/decisions. In [19], single DT is used to predict the voltage stability status. In order to achieve more accurate decision results, ensemble of DT such as AdaBoost [20], XGBoost [21] and random forest [22], [23] are proposed to predict the transient stability status. In the European iTesla project, DT is used for online DSA of large-scale grids in a massively parallel way [24].

In the past, very few works have been reported on interpretability of data-driven TSA models. In Ref. [25], a hybrid DT-NN approach is proposed, where a DT is firstly trained and then translated into a four-layer feed-forward multilayer perceptron (MLP). But such hybrid DT-NN method comes from Entropy Net [26] and fully depends on the DT model, if DT model is not well-trained at first, the final hybrid DT-NN model will be infeasible. Besides, such method suffers from a heavy computational burden, since it applied BFGS method to optimize the weights whose computational complexity is $O(n^2)$. Ref. [27] has shown a trade-off between accuracy and transparency for data-driven DSA models. The optimal classification tree in [28] optimizes the tree structure to explain the DSA model. Note that the final goal of all above methods aim to optimize DT and obtain the interpretable data-driven DSA models based on the well-trained DT model. However, although the tree-based stability models are interpretable, they may not achieve high stability accuracy in some cases, since these tree-based models are very sensitive to the database and feature inference [29]. Even small noise in training database can result in assessment accuracy decrease.

To acquire both transparent and accurate TSA, this paper proposes an interpretable DL-based TSA method for pre-fault TSA problem. The proposed method is composed of two models, a TSA classification model and an interpretable mimic model with the similar decision results for the TSA classification model. The main contributions and values of this paper are three-fold as follows:

1) The DL-based TSA classification model is based on GRU algorithm, which can improve the TSA speed and achieve accurate TSA results with the less computation burden compared with LSTM. The interpretable mimic model is based on DT which is to provide the TSA decision rules. The proposed method formulates the objective function with the tree regularization, which can encourage the GRU model to become more similar with DT model by multiple iterative training. Thus, DT-based interpretable model can be used for model interpretation. In this way, the proposed method combines the strong nonlinear modeling capability of GRU and the interpretability of DT by tree regularization. Compared with the traditional DT-based interpretable DSA method [25], the proposed interpretable DL-based TSA method is only partially dependent on DT and not sensitive to tree structure.
2) The proposed method can balance the accuracy of GRU-based TSA classification model and the transparency of DT-based interpretable model by controlling the tree regularization term coefficient.
3) For pre-fault TSA, given such interpretable DL-based TSA model, the system operators can obtain the accurate and transparent TSA process. The TSA rules provided by the DT model can support the preventive control design. Meanwhile, one can check such TSA results via human-being expert knowledge, and reason the fairness and systematic deviations in the database according to the actual conditions.

The preliminaries of the used ML algorithms are introduced in Section II. Them, the framework and the training process of the proposed interpretable DL-based method are described in Section III. Finally, simulation results tested on the benchmark testing system are given in Section IV, which have shown the satisfactory TSA accuracy and highest fidelity performance.

II. Preliminaries

The proposed interpretable DL-based TSA method is based on GRU and DT. This section introduces the principle of them.

A. Gated Recurrent Unit (GRU)

GRU, proposed by *Cho* in [30], is a relatively new prediction algorithm, which belongs to RNN. While original RNNs suffer from vanishing and exploding gradient problems, LSTM is firstly proposed to deal with these problems by introducing new gates, such as input and forget gates, which allows for a better control over the gradient flow and enable better preservation of "long-range dependencies". The long-range dependency in

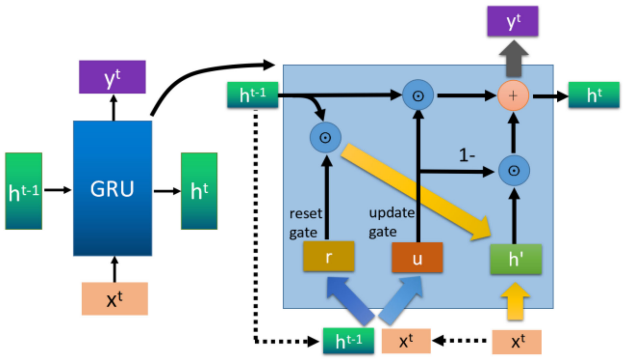


Fig. 1. The structure of a GRU.

RNN is resolved by increasing the number of repeating layer in LSTM. Similar to LSTM, GRU uses a simplified structure and also aims to solve the problems of long-term memory and gradient in back propagation, but GRU only utilizes hidden state to carry information flow without using the memory cell. GRU consist of reset gate and update gate, where the update gate is merged with the forget gate and the input gate of LSTM. In general, GRU is more effective than LSTM because GRU has fewer "gates" and fewer parameters than the LSTM, but it can also achieve the considerable results with the less computation burden than LSTM. Considering the computational efficiency and time cost of the hardware, it will be more inclined to use GRU in many cases. The structure of GRU is shown in Fig. 1, where $\odot$ represents the Hadamard Product.

For the basic GRU, its external inputs are the previous hidden state $\mathbf{h}^{t-1}$ and the current input vector $\mathbf{x}^t$. Such hidden state passed down from the previous node. This hidden state contains information about the previous node. Combining hidden state $\mathbf{h}^{t-1}$ and input vector $\mathbf{x}^t$, GRU will get the output y^t and the hidden state h^t passed to the next node. First, GRU can obtain two gate states by the last transmitted state $\mathbf{h}^{t-1}$ and the input vector $\mathbf{x}^t$ of the current node as follow:

$$\boldsymbol{r}^t = \varphi\left(\mathbf{w}_r \cdot \left[\mathbf{h}^{t-1}, \mathbf{x}^{t-1}\right] + \mathbf{b}_r\right) \tag{1}$$

$$\boldsymbol{u}^t = \varphi\left(\mathbf{w}_u \cdot \left[\mathbf{h}^{t-1}, \mathbf{x}^{t-1}\right] + \mathbf{b}_u\right) \tag{2}$$

where $\boldsymbol{r}$ and $\boldsymbol{u}$ represent the reset gate and update gate respectively; $[\mathbf{h}^{t-1}, \mathbf{x}^{t-1}]$ represents the concatenated vector of previous hidden state $\mathbf{h}^{t-1}$ and input vector $\mathbf{x}^t$; $\mathbf{w}_r$ and $\mathbf{w}_u$ are the weight vector of reset gate and update gate respectively; $\mathbf{b}_r$ and $\mathbf{b}_u$ are the weight vector of reset gate and update gate respectively; $\varphi(\cdot)$ is sigmoid function, which can transform data into values in the range of [0,1] to control the gate signal.

After getting the gate control signal, the reset gate control is used to get the temporary value $(\mathbf{h}^{t-1})' = \mathbf{h}^{t-1} \odot \boldsymbol{r}^t$ after "reset", then the $(\mathbf{h}^{t-1})'$ is spliced with the input vector $\mathbf{x}^t$, then the data is reduced into the range of $[-1,1]$ via the $tanh(\cdot)$ activation function as follow:

$$\widetilde{\mathbf{h}}^t = tanh\left(\mathbf{w}_h \cdot \left[\mathbf{h}^{t-1} \odot \boldsymbol{r}^t, \mathbf{x}^t\right] + \mathbf{b}_h\right) \tag{3}$$

where $\widetilde{\mathbf{h}}^t$ mainly contains the current input vector and can remember the state of the current state.

Finally, in the updating stage, GRU carries out two steps of forget and memory at the same time as follow:

$$\mathbf{h}^t = (1 - z) \odot \mathbf{h}^{t-1} + z \odot \widetilde{\mathbf{h}}^t \tag{4}$$

where the range of gate signal z is [0,1]. The closer the gate signal is to 1, the more data it represents in memory, and the closer it is to 0, the more forgotten it represents. It can be seen that forgetting z and selecting (1-z) are related. For the previous information, GRU selectively forgets via weights z, and then uses the corresponding weights in the $\widetilde{\mathbf{h}}^t$ containing the current input to make up (1-z) in order to maintain a "constant" state.

Given N training instances with corresponding ground truth label $\mathcal{D}^N = \{(\mathbf{x}_n,\ \mathrm{y}_n) |\ \mathbf{x}_n = [\mathbf{x}_n^1, \ldots, \mathbf{x}_n^t],\ \mathrm{y}_n \in \mathbb{R}\}_{n=1,t=1}^{N,T_n}$, GRU can learn the available classifier $\widehat{\mathrm{y}_n}$ as (5), which can map the relationship from $\mathbf{x}$ to y with the model parameters $\boldsymbol{\mathcal{W}}$.

$$f\ (\mathbf{x}_n; \boldsymbol{\mathcal{W}}) = \sigma\left(\boldsymbol{\mathcal{W}} \cdot \mathbf{h}^t\right) \tag{5}$$

The goal of such GRU is to learn the model parameters $\boldsymbol{\mathcal{W}}$ via minimizing the loss function between predicted $f_\theta(\mathbf{x}_n)$ and ground true label y_n as (6).

$$\min_{\boldsymbol{\mathcal{W}}} \sum_{n=1}^{N} \sum_{n=1}^{T_n} \mathcal{L}\left(f\left(\mathbf{x}_n; \boldsymbol{\mathcal{W}}\right), \mathrm{y}_n\right) + \sigma \mathcal{R}\left(\boldsymbol{\mathcal{W}}\right) \tag{6}$$

where $\mathcal{L}(\cdot,\cdot)$ represents the pre-defined loss function; $\mathcal{R}(\boldsymbol{\mathcal{W}})$ represents the regularization term (e.g., L_1 or L_2 norm) with scalar strength $\sigma \in \mathbb{R}^+$. The gradient descent algorithms (e.g., stochastic gradient descent, Adam, etc.) can solve such back-propagation procedure to update the model parameters $\boldsymbol{\mathcal{W}}$. Once training instance $\mathbf{x}$ and label y are available, the GRU model can be obtained.

B. Decision Tree (DT)

DT, as a popular supervised machine learning technique [18], is a tree-structured predictive model for classification or regression on unknown targets given their features. DT is composed of three parts, including the root node, the internal nodes and the leaf nodes. The root node is the start of DT. The internal nodes connect the root node and the leaf nodes, and each internal node represents a feature inference. The leaf nodes are the terminal of the DT and each leaf node holds a class label. The paths from root node to leaf nodes reveal the DT classification rules, and each leaf node has a class label, which represents the outcome of the feature inferences.

During the DT training stage, the feature inference of internal nodes is based on the measurement of the information. The critical features can be selected for feature inference. Generally, *Gini index* is a popular feature selection method for DT, which measures impurity and is utilized to select the features with less impurity [29]. This method checks the reduction of impurities when the selected feature is used. The smaller the impurities, the more useful they are for classification. In the *Gini* algorithm, the features and splitting variables are chosen by the splitting criterion that produces the highest impurity reduction for the next split.

In this paper, DT aims to classify the stability status class labels, i.e., 'secure' and 'insecure' with respect to a contingency.

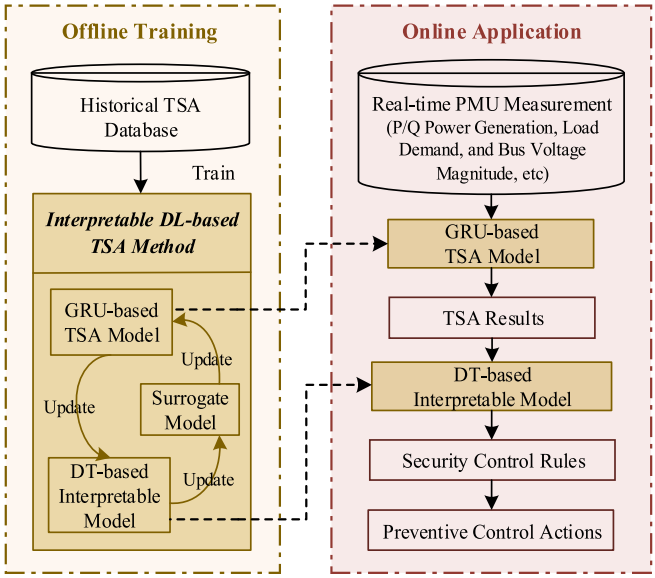


Fig. 2. The framework of proposed interpretable DL-based TSA method.

Once a DT-based TSA model is constructed, the control rules can be obtained by reversely interpreting the TSA decision-making process, e.g., in order to control an insecure operating point to be secure, the value of internal node can be changed to move the operating point from an insecure leaf node to a secure leaf node [31].

III. Proposed Interpretable DL-Based TSA Method

To balance the TSA accuracy and transparency, the proposed method combines the nonlinear modeling capability of GRU and the interpretability of DT, via regularizing GRU with average DT path length. Through such tree regularization [32], the GRU-based model has the similar behavior with a DT, and the decision process of GRU-based model can be explained and used for preventive control action design.

This section firstly describes the whole framework of the proposed method, including offline training stage and online application stage, illustrated in Fig. 2. Then the whole objective function and its solving processes by a surrogate model are presented in detail.

A. General Framework

For pre-fault TSA problem, the training inputs are the operating variables of the system (e.g., P/Q power generation, load demand, and bus voltage magnitude) and outputs are the corresponding class label (secure or insecure).

At the offline training stage, the historical TSA database is utilized to train a GRU-based TSA model and a DT-based interpretable model. Through constantly multiple iterative updating, the GRU-based TSA model can predict the stability status, and the DT-based interpretable model can explain the decision-making process of GRU model. During the online application stage, with the real-time measurements, the TSA results can be rapidly calculated by the GRU-based TSA model.

Algorithm 1: Interpretable DL-based TSA Method.

Input: Given N original training instances with corresponding ground truth label $\mathcal{D}^N = \{(\mathbf{x}_n, \mathrm{y}_n) | \, \mathbf{x}_n, \mathrm{y}_n \in \mathbb{R}\}_{n=1}^N$, the number of training iterations K, learning rate λ, regularization term coefficient σ, γ.

Output: The proposed interpretable DL-based TSA method, including a GRU-based TSA model and a DT-based interpretable model.

Initialize: DT model parameters settings, initial GRU model parameters, surrogate model database $\mathcal{D}_{\mathcal{W}}^{K+1}$.

Begin
Train a GRU-based model $f(\mathbf{x}; \mathcal{W})$ with $\mathcal{D}^N = \{(\mathbf{x}_n, \mathrm{y}_n)\}_{n=1}^N$
For number of training iterations $k = 0, \ldots, K$ **do**
 # Update objective function by GRU and average DT path length
Update GRU-based TSA model parameters by descending gradient algorithm: $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k - \lambda \cdot Adam(\mathcal{W}_k)$.
Calculate predicted label $\widehat{\mathrm{y}_n} = f(\mathbf{x}_n; \mathcal{W}_{k+1}), \forall n \in \{1, .., N\}$.
 # Train a DT-based interpretable model with $\{(\mathbf{x}_n, \widehat{\mathrm{y}_n})\}_{n=1}^N$
$DT \leftarrow Train\_DecisionTree(\{(\mathbf{x}_n, \widehat{\mathrm{y}_n})\}_{n=1}^N)$.
 # Obtain the average DT path length
Calculate the $\mathcal{S}(\mathcal{W})$ via $DTPL(\mathbf{x}_n, f(\mathbf{x}_n; \mathcal{W}), DT)$ as Eq. (7-b).
Collect one instance $\left(\mathcal{W}_{k+1}, \mathcal{S}(\mathcal{W}_{k+1})\right)$ into $\mathcal{D}_{\mathcal{W}}^{K+1}$.
 # Train a surrogate model $\hat{\mathcal{S}}(\mathcal{W}; \boldsymbol{\beta})$
Train a surrogate model $\hat{\mathcal{S}}(\mathcal{W}; \boldsymbol{\beta})$ with current $\mathcal{D}_{\mathcal{W}}^{K+1}$ via MLP as Eq. (8).
Calculate the $\hat{\mathcal{S}}(\mathcal{W}_{k+1})$ via $\hat{\mathcal{S}}(\mathcal{W}; \boldsymbol{\beta})$.
Obtain the objective function of the proposed method via combining the GRU model and surrogate model as Eq. (9).
End for
Collect the surrogate model database $\mathcal{D}_{\mathcal{W}}^{K+1} = \{(\mathcal{W}_k, \mathcal{S}(\mathcal{W}_k))\}_{k=1}^{K+1}$.
Obtain the DT-based interpretable model of the GRU-based model.
Return Interpretable DL-based TSA method.
End

The gradient-based updating can utilize any standard gradient descent algorithm. In this method, *Adam* algorithm is applied here.

In the meantime, based on the DT model, the TSA decision-making process is explained by the DT model and the control rules can also be designed by reversely interpreting the TSA decision-making process, e.g., in order to control an insecure operating point to be secure, the value of an internal node can be changed to move the operating point from an insecure leaf node to a secure leaf node. Detailed preventive control designs based on DT can be found in Ref. [31], [33].

B. Training Process

The proposed interpretable DL-based method regularizes GRU by tree regularization rather than traditional L_1 or L_2 norm, which can construct a mimic DT to approximate the decision-making process of a trained GRU. At the training process, there are three parts of models to be computed, including a GRU-based TSA model trained by the original training instances, a DT-based interpretable model trained by the reconstructed training instances, and a surrogate model trained by the surrogate training instances.

The original training instances are composed of original features and corresponding ground truth labels. With the original training instances, the GRU-based TSA model can be trained and the model parameters $\mathcal{W}$ and predicted labels $\widehat{\mathrm{y}_n}$ can be obtained. Then, the DT-based interpretable model can be trained

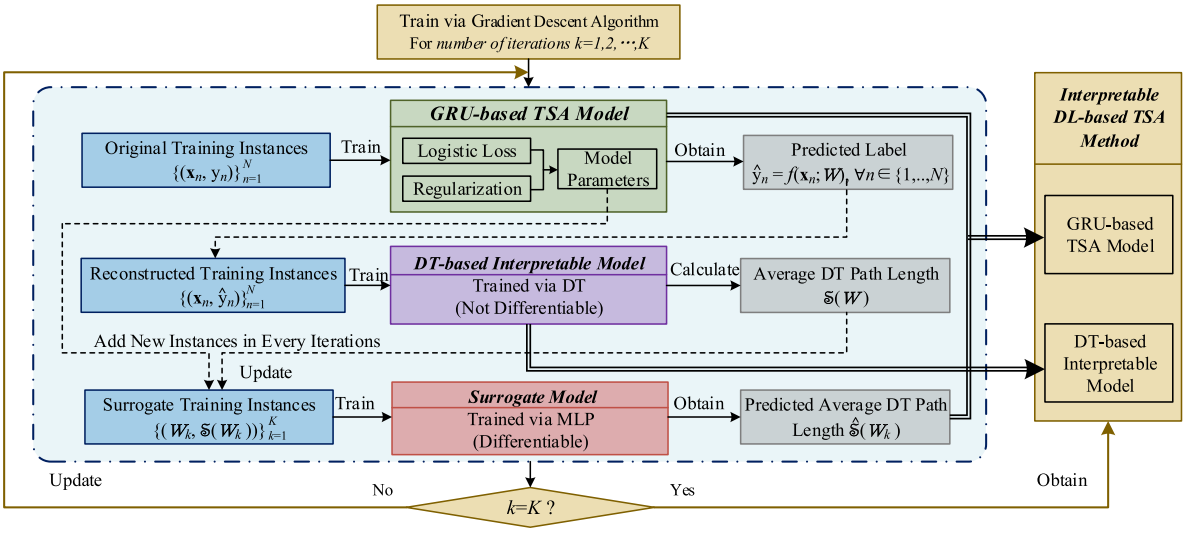


Fig. 3. The training process of the proposed interpretable DL-based TSA method.

with the reconstructed training instances, which are composed of original features and predicted labels $\widehat{\mathrm{y}_n}$ of the GRU-based TSA model. According to the structure of the DT-based interpretable model, the average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ can be calculated, which is used for tree regularization. By using tree regularization term with average DT path length, the GRU-based model is not entirely dependent on the structure of the DT. However, the average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ cannot be directly used for tree regularization, since it is non-differentiable. Therefore, a differentiable surrogate model is utilized to predict the estimated average DT path $\hat{\mathcal{S}}(\boldsymbol{\mathcal{W}})$ by the surrogate training instances, which can map each step of the GRU-based TSA model parameters $\boldsymbol{\mathcal{W}}$ to an average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ of the DT-based interpretable model. The surrogate model can be trained by any effective ML algorithm. In this paper, MLP is applied, which belongs to forward structure artificial neural network.

Finally, after certain iterations, the proposed interpretable DL-based TSA method are completely trained, and a GRU-based TSA model and a DT-based interpretable model are obtained. A complete training process of proposed method is illustrated in Fig. 3 and also summarized in Algorithm 1.

C. Training Objective Function via Tree Regularization

Formally, given N original training instances, the objective function of proposed interpretable DL-based method can firstly be formalized in (7):

$$\min_{\boldsymbol{\mathcal{W}}} \sum_{n=1}^{N} \sum_{n=1}^{T_n} \mathcal{L}\left(f\left(\mathbf{x}_n; \boldsymbol{\mathcal{W}}\right), \mathrm{y}_n\right) + \sigma \mathcal{S}\left(\boldsymbol{\mathcal{W}}\right) \tag{7a}$$

$$where\ \mathcal{S}\left(\boldsymbol{\mathcal{W}}\right) \triangleq \frac{1}{N} \sum_{n=1}^{N} DTPL\left(\mathbf{x}_n, f\left(\mathbf{x}_n; \boldsymbol{\mathcal{W}}\right), DT\right) \tag{7b}$$

where $\mathcal{L}(\cdot,\cdot)$ represents the logistic loss function; $DTPL(\cdot,\cdot)$, called the DT path length, represents the decision path depth of a sample from root node to leaf node, in order to compute the average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ as tree regularization term with scalar strength $\sigma \in \mathbb{R}^+$. In practice, if the system needs an accurate GRU-based model with the detailed complex decision rules for GRU-based model, the proposed interpretable DL-based model trained with small regularization coefficient σ can be selected; if the system requires the GRU-based model with fast computation time and transparency decision rules provided by DT-based model, the proposed interpretable DL-based model trained with large regularization coefficient σ is suitable.

It can be seen that the former and latter term of (7-a) are GRU classification loss and corresponding average DT path length loss, respectively. Therefore, the proposed interpretable DL-based method takes use of the nonlinear capability of GRU and the interpretability of DT, hence has the accurate and interpretable TSA decision-making characteristics.

D. Surrogate Model

Since the average DT path length loss term $\mathcal{S}(\boldsymbol{\mathcal{W}})$ formalized in (7-b) is not differentiable, the objective function (7-a) cannot be directly solved by gradient descent algorithms to update the model parameters $\boldsymbol{\mathcal{W}}$.

The surrogate model trained by MLP aims to acquire the accurate average DT path length in a differentiable way, which is utilized to emulate the tree regularization term $\mathcal{S}(\boldsymbol{\mathcal{W}})$ via predicted $\hat{\mathcal{S}}(\boldsymbol{\mathcal{W}}_k; \beta)$. The surrogate model is trained by the surrogate training instances with K model parameters $\boldsymbol{\mathcal{W}}$ and corresponding average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ value: $\mathcal{D}_{\boldsymbol{\mathcal{W}}}^{K} = \{(\boldsymbol{\mathcal{W}}_k,\ \mathcal{S}(\boldsymbol{\mathcal{W}}_k))\}_{k=1}^{K}$, where $\boldsymbol{\mathcal{W}}_k$ can be obtained from each training step of the target GRU model $\widehat{\mathrm{y}_n} = f(\mathbf{x}_n; \boldsymbol{\mathcal{W}})$. Thus, the surrogate model can be trained by minimizing the sum of squared errors between the average DT path length $\mathcal{S}(\boldsymbol{\mathcal{W}})$ and the predicted $\hat{\mathcal{S}}(\boldsymbol{\mathcal{W}}_k; \beta)$ as (8).

$$\min_{\beta} \sum_{k=1}^{K} \left(\mathcal{S}\left(\boldsymbol{\mathcal{W}}_k\right) - \hat{\mathcal{S}}\left(\boldsymbol{\mathcal{W}}_k; \beta\right)\right)^2 + \gamma \|\beta\|_2^2 \tag{8}$$

TABLE I
CONTINGENCY SET

Contingencies	1	2	3	4	5	6	7	8	9	10
Fault Setting	Fault bus 39, trip 1-39	Fault bus 1, trip 1-39	Fault bus 4, trip 3-4	Fault bus 3, trip 3-4	Fault bus 29, trip 26-29	Fault bus 26, trip 26-29	Fault bus 16, trip 15-16	Fault bus 15, trip 15-16	Fault bus 17, trip 16-17	Fault bus 16, trip 16-17

where β represents the model parameters of the surrogate model, and $\gamma \in \mathbb{R}^+$ represents the regularization coefficient.

Based on the trained surrogate model, the estimated $\hat{\mathcal{S}}(\boldsymbol{\mathcal{W}}_k;\beta)$ can be obtained. Then, the final training objective function of the proposed interpretable DL-based TSA method can be reformulated by replacing the latter term of (7-a) by the estimated $\hat{\mathcal{S}}(\boldsymbol{\mathcal{W}}_k;\beta)$ as (9).

$$\min_{\boldsymbol{\mathcal{W}}} \sum_{n=1}^{N} \sum_{n=1}^{T_n} \mathcal{L}\left(f\left(\mathbf{x}_n;\boldsymbol{\mathcal{W}}\right),\mathrm{y}_n\right) + \sigma\hat{\mathcal{S}}\left(\boldsymbol{\mathcal{W}}_k;\beta\right) \tag{9}$$

In practice, the surrogate model can be trained in parallel with the GRU model $f(\mathbf{x};\boldsymbol{\mathcal{W}})$.

IV. SIMULATION RESULTS

The proposed method is tested on New England 10-machine 39-bus system to validate the performance, which is a standard benchmark power system for stability analysis. The numerical simulation is conducted on a high-performance computer with an Intel Core i7 CPU of 3.3-GHz, 16-GB RAM and GPU with NVIDIA GeForce GTX 1060. The case study in this paper considers the rotor angle stability (i.e., transient stability) criterion. The T-D simulation is implemented in Transient Security Assessment Tool (TSAT). The proposed method is implemented in the Python with Scikit-Learn framework.

A. Database Generation

Simulations are tested on New England 10-machine 39-bus system to evaluate the proposed method. The operating points with their corresponding stability conditions are generated via Monte-Carlo method [34], which is run to sample uncertain power variations under the forecasted load demand level for each bus. The contingencies are the three-phase faults with inter-area corridor trip and cleared 0.25 s after their occurrences. 10 severe N-1 three-phase short-circuit bus faults are simulated using TSAT, as shown in Table I. 320 operating variables (including, 136 bus related features and 184 branch related features) are selected as the primary features, listed in Table II. Then, if the maximum rotor angle separation is beyond 360 degrees, it is labelled as insecure [35]. Finally, 3000 operating instances with their corresponding stability conditions were obtained. Based on the previous studies and experience, 2000 instances were randomly selected for model training, and the remaining 1000 instances are utilized for testing.

B. Testing TSA Accuracy and Fidelity Performance

The proposed DL-based interpretable method shows the better performance in terms of the TSA accuracy and interpretability. In order to verify the TSA accuracy, the average TSA accuracy is applied, which represents the average percentage of the correctly classified testing instances under ten different faults conditions. The proposed method (including the GRU-based TSA model with tree regularization and the DT-based interpretable model) should be compared with GRU-based models with L_1 and L_2 norm regularization term and independent DT-based model. In order to verify the interpretability, this paper uses the fidelity index to measure the consistency from the DT-based TSA models to GRU-based TSA models. Fidelity is denoted as the percentage of testing examples on which the prediction made by a DT-based TSA model agrees with the DL-based TSA model [36]. For the fidelity performance, the proposed GRU-tree regularization TSA model with DT-based interpretable model is compared to GRU-L_1/L_2 norm regularization models with independent DT-based model.

TABLE II
FEATURES FOR NEW ENGLAND 10-MACHINE 39-BUS SYSTEM

Operating Variables		Features Symbol	Number of Features
Bus Features (Total 136 Variables)	generation active power output	$\mathbf{P}_G$	10
	generation reactive power output	$\mathbf{Q}_G$	10
	load bus active power	$\mathbf{P}_L$	19
	load bus reactive power	$\mathbf{Q}_L$	19
	bus voltage magnitude	$\mathbf{V}_M$	39
	bus voltage angle	$\mathbf{V}_A$	39
Branch Features (Total 184 Variables)	transmission line active power flow (from)	$\mathbf{P}_{FROM}$	46
	transmission line reactive power flow (from)	$\mathbf{Q}_{FROM}$	46
	transmission line active power flow (to)	$\mathbf{P}_{TO}$	46
	transmission line reactive power flow (to)	$\mathbf{Q}_{TO}$	46

In order to ensure the fairness of the testing results, we guarantee that the tree structures of the proposed DT-based interpretable model and independent DT model are the same, i.e., the number of the internal nodes and leaf nodes. Note that all the testing results are the average performance value of the ten testing results for the ten contingencies as Table I.

The proposed method includes a GRU-based TSA model and a DT-based interpretable model. Both the GRU-based TSA model and DT-based interpretable model are trained at the offline stage. For the real-time measurements, the trained GRU-based TSA model and DT-based interpretable model can be directly utilized without any other settings. Thus, it is available for real-time applications.

1) Average TSA Accuracy Comparison: The average TSA accuracy of the three different GRU-based TSA models with the different regularization term coefficients σ are listed in Table III. The average TSA accuracy and the tree structure (including maximize number of the internal nodes and leaf nodes) of the

TABLE III
AVERAGE TSA ACCURACY OF GRU-BASED TSA MODELS WITH DIFFERENT REGULARIZATION

DL-based TSA Methods	Regularization Term Coefficient σ			
	0.01	1.0	100	10000
GRU-L_1 Norm	94.46%	92.18%	90.38%	85.24%
GRU-L_2 Norm	94.78%	92.44%	90.44%	86.32%
Proposed GRU-Tree Regularization	94.12%	91.96%	90.12%	84.96%

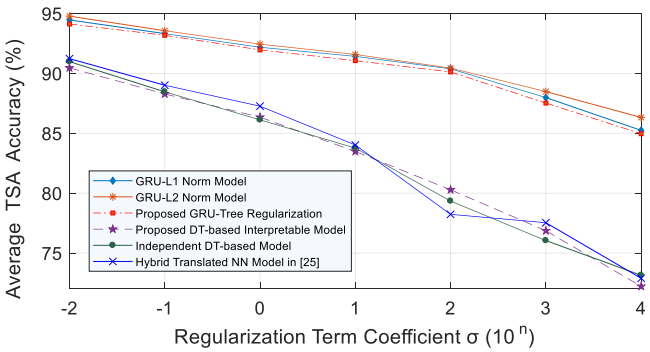


Fig. 4. Average TSA accuracy of different models.

DT-based interpretable model for the GRU-based TSA model with the different regularization term coefficient σ are listed in Table IV. Besides, Fig. 4 shows the average TSA accuracy of the different GRU models and DT models, and each point represents the average TSA accuracy under the different regularization term coefficients.

According to Tables III, IV and Fig. 4, with the increase of the regularization term coefficient, all the data-driven TSA methods show the different degree of decrease in average TSA accuracy, and the DT-based models show the significantly decrease compared with the GRU-based TSA models. Besides, all the GRU-based TSA models (including GRU-L_1/L_2 norm regularization models and the proposed GRU-tree regularization model) always have the far better TSA accuracy performance than the DT-based TSA models (including independent DT model and the proposed DT-based interpretable model), which confirms the stronger accuracy performance of DL technique than DT.

Moreover, it can be seen that the proposed GRU-tree regularization TSA model has the similar average TSA accuracy with GRU-L_1/L_2 norm TSA models, and the corresponding DT-based interpretable model has the similar average TSA accuracy with the independent DT model, which can demonstrate that the TSA accuracy of the proposed DL-based TSA method has not been significantly affected and can make the DL-based TSA model transparent without sacrificing the TSA accuracy. Overall, the above conclusions can be summarized as below:

$$\begin{aligned} &Acc\,(GRU-L_1/L_2\text{Norm Regularization}) \\ &\approx Acc(\text{Proposed GRU} \\ &\text{with Tree Regularization}) \gg Acc\,(\text{Independent DT Model}) \\ &\approx Acc\,(\text{Proposed DT–based Interpretable Model}) \end{aligned} \tag{10}$$

where $Acc\,(\cdot)$ denotes the average TSA accuracy performance. However, for the existing data-driven interpretable TSA models in Ref. [25], [27], these hybrid DT-NN methods need to train the DT at first, then based on the DT structure and selected attributions, the translated NN-based models can be obtained. As shown in Fig. 4, it can be seen that the TSA accuracy performance of the obtained translated NN model is only close to the original DT model as below:

$$Acc\,(Independent\ DT\ Model) \approx Acc(Translated\ NN\ Model) \tag{11}$$

Based on above two inequalities about TSA accuracy of different models, it can be seen that the proposed interpretable DL-based model is far better than the existing hybrid DT-NN interpretable methods, since it can achieve transparency with higher TSA accuracy.

2) Balance Between the Accuracy of the GRU Model and the Transparency of DT Model: According to Table IV and Fig. 5, it can be seen that there exists relationship between the TSA accuracy of GRU-based TSA model and the transparency of the DT-based interpretable model. With the increase of the regularization term coefficient, the structures of DT-based models become simpler and therefore more transparent, since the average number of the internal nodes and leaf nodes of the DT-based TSA models are fewer as Fig. 5(a). Besides, as shown in Fig. 5(b), it can be seen that the more the internal nodes, the higher the TSA accuracy. Thus, in practice, if the operator needs an accurate TSA model with the detailed complex decision rules for GRU-based TSA model, a small regularization coefficient $\boldsymbol{\sigma}$ can be selected, e.g., in the range of smaller than 1.0; if the operator requires the TSA model with fast computation time and transparency decision rules provided by DT-based TSA model, a large regularization coefficient $\boldsymbol{\sigma}$ is suitable for this scenario, e.g., in the range of greater than 100.

Fig. 6(a, b, c) illustrate the acquired DT-based interpretable models of the fault 5 with the different regularization term coefficient (0.1, 10, 1000), respectively. The structure of the acquired DT-based interpretable model includes the internal nodes (square with green in Fig. 6) and the leaf nodes (oval with orange or blue in Fig. 6). Each internal node block consists of the feature judgment conditions, Gini index (represents the impureness of the selected features, the smaller value means be closer to the final stability status decisions; while the leaf node block shows Gini index and decision results. Moreover, in order to clearly distinguish the stability status decisions, two different color are used to represent that the bright color leaf nodes (in orange) can better identity the insecure cases and the dark color leaf nodes (in blue) can better identity the secure cases. For the leaf nodes, if the Gini index value is equal to zero, it means such leaf nodes can accurate distinguish the stability status. When the regularization term coefficient is too large, it can be observed that the leaf nodes with one common internal node cannot distinguish the stability status. Such DT-based interpretable models provide the GRU-based TSA model decision rules, hence, can guide subsequent preventive controls.

3) Fidelity of TSA Predicted Results Comparison: Table V and Fig. 7 show the fidelity performance of the different methods

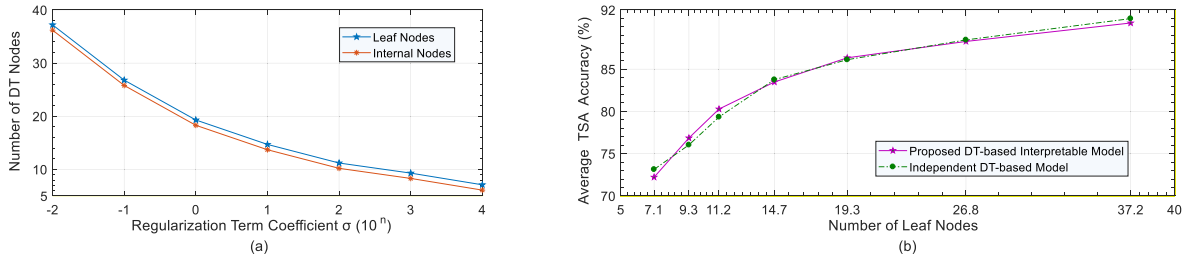
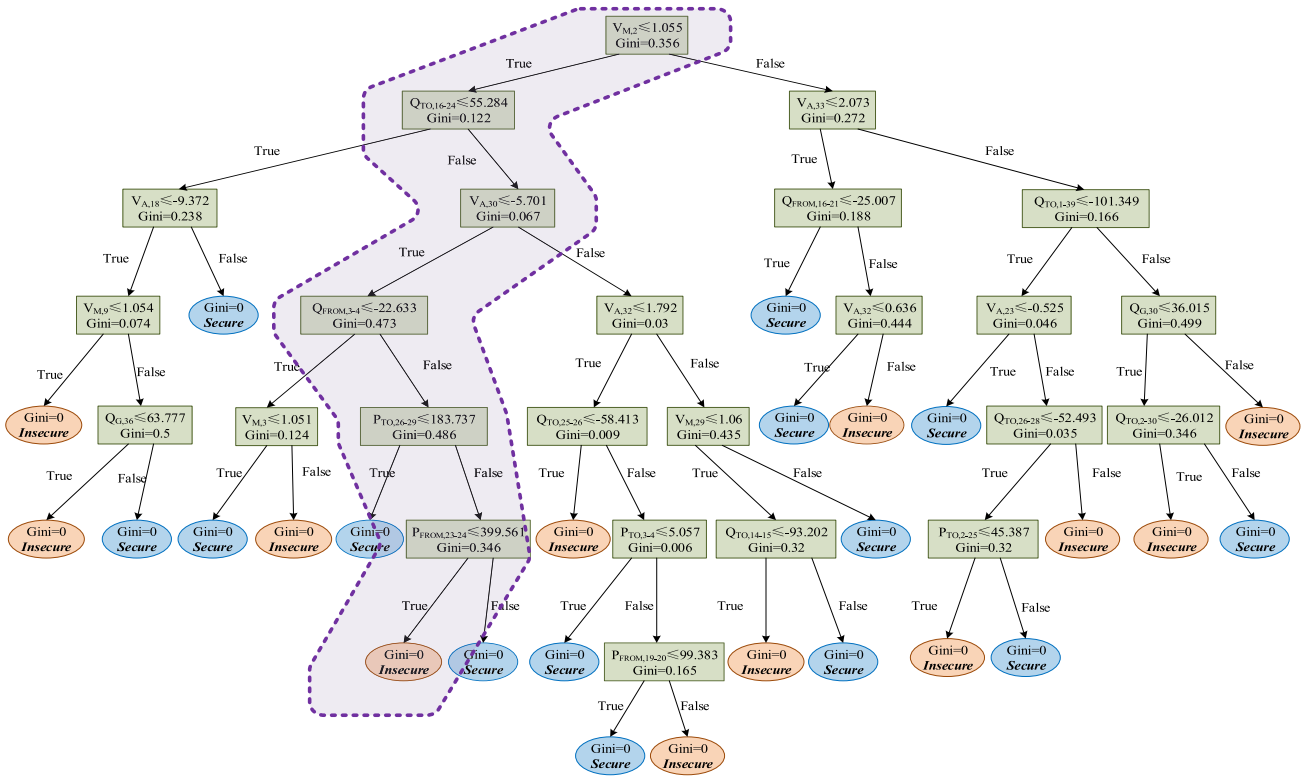
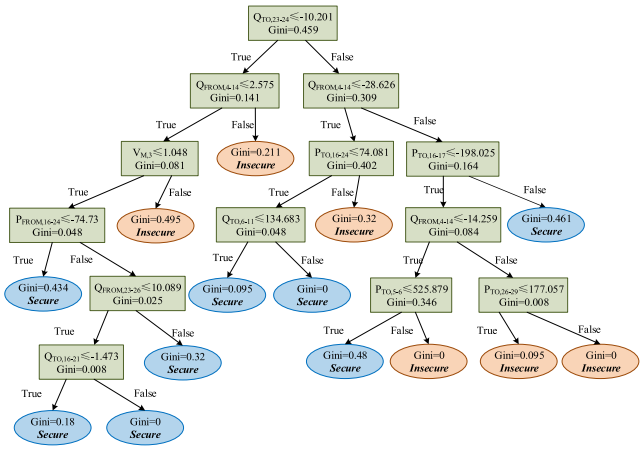


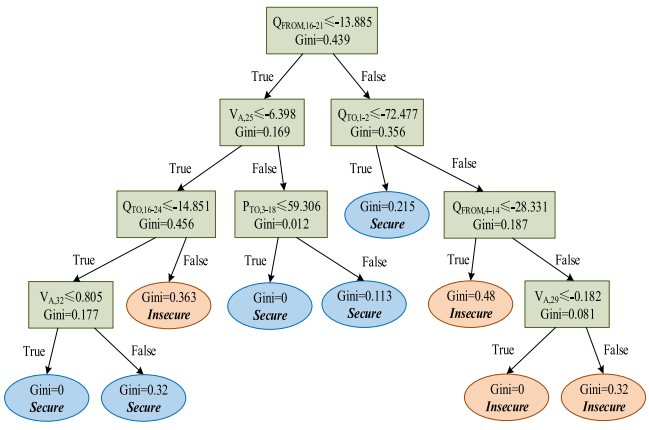
Fig. 5. Testing results of DT-based models. (a) Relationship between regularization term and DT nodes; (b) Relationship between leaf nodes and TSA accuracy.



(a) fault 5 with regularization term coefficient $\sigma = 0.1$



(b) fault 5 with regularization term coefficient $\sigma = 10$



(c) fault 5 with regularization term coefficient $\sigma = 1000$

Fig. 6. DT-based interpretable model of the GRU-based TSA model under the different regularization term coefficient σ for the fault 5. (a) $\sigma = 0.1$; (b) $\sigma = 10$; (c) $\sigma = 1000$.

TABLE IV
AVERAGE NUMBER OF NODES FOR DT-BASED MODELS & AVERAGE TSA ACCURACY OF DT-BASED MODELS AND TRANSLATED NN MODEL BY INDEPENDENT DT

DT-based Model and Translated NN Model		Regularization Term Coefficient σ			
		0.01	1.0	100	10000
Proposed DT-based Interpretable Model	Average TSA Accuracy	90.46%	86.34%	80.28%	72.22%
	Average Number of Internal Nodes	36.2	18.3	10.2	6.1
	Average Number of Leaf Nodes	37.2	19.3	11.2	7.1
Average TSA Accuracy of Independent DT		90.98%	86.12%	79.34%	73.14%
Average TSA Accuracy of Translated NN Model [25]		91.24%	87.26%	79.60%	72.88%

TABLE V
FIDELITY PERFORMANCE OF DIFFERENT METHODS

Fidelity of Different Methods between DL-based TSA Models and DT-based Models	Regularization Term Coefficient σ			
	0.01	1.0	100	10000
GRU-L_1 Norm with Independent DT	87.44%	79.90%	73.78%	64.46%
GRU-L_2 Norm with Independent DT	87.46%	81.16%	74.72%	65.38%
Independent DT with Translated NN Model (Hybrid DT-NN Method [25])	84.02%	79.66%	76.48%	70.34%
Proposed GRU-Tree Regularization Model with DT Interpretable Model	95.12%	92.78%	88.28%	86.12%

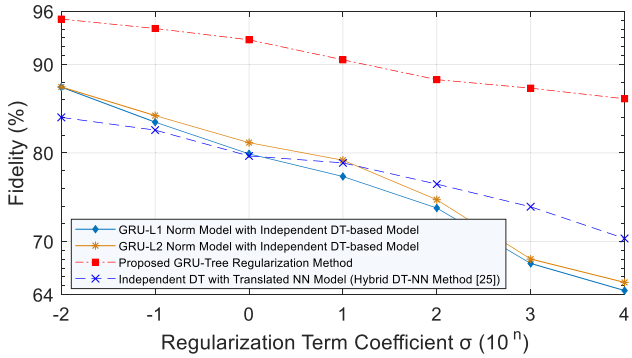


Fig. 7. Fidelity performance of different methods.

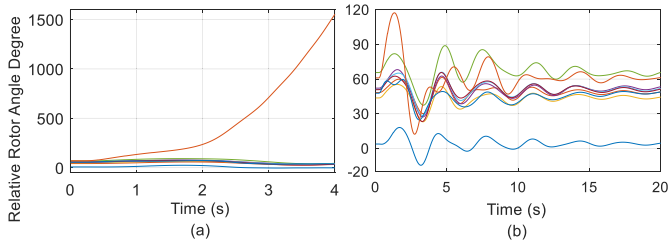


Fig. 8. Relative rotor angle performance under fault 5. (a) the initial operating point without preventive control; (b) the operating point with preventive control based on DT-based interpretable model.

under the different regularization term coefficient. A larger fidelity index value indicates higher consistency between the DT-based and DL-based models. It is clear that the proposed GRU-tree regularization TSA model with its corresponding DT-based interpretable model always has the best fidelity performance around 86.12%-95.12% of testing instances, larger than the GRU-L_1/L_2 norm model with the independent DT-based model around 7.66%-21.66%. Besides, it can be seen that although the existing hybrid DT-NN method [25] can make the independent DT model and translated NN model have the similar TSA accuracy, the fidelity of TSA predicted results is still further lower than the proposed GRU-tree regularization method. Thus, the proposed method can obtain both the accurate and transparent TSA results. Besides, it can be seen that when the regularization term coefficients σ is larger, the proposed method shows the better interpretability performance in terms of the fidelity difference value.

C. DT Rule-Based Preventive Control

In practice, the DT-based rules can be used for preventive control. In order to control an insecure operating point to be secure, the value of an internal node can be changed to move the operating point from an insecure leaf node to a secure leaf node. More detailed preventive control actions based on DT can be found in Ref. [31], [33].

To validate the feasibility and effectiveness of the acquired DT-based rules in this paper, the preventive control actions are tested and the testing results are given in Fig. 8(a) and (b). Taking an example under fault 5, one testing initial operating point without preventive control is classified as the insecure case according to DT rules, as shown in the purple region surrounded by dashed lines in Fig. 6(a). Fig. 8(a) shows the relative rotor angle degree (by reference generator G39) performance of this testing operating point. It can be seen that one generator will be out of stability, and the transient stability index (TSI) [37] of this testing initial operating point without preventive control is -46.39, verifying its insecure case.

Among these DT rules, the partial rule based on two internal nodes closest to the root node is selected and listed in (12).

$$DT\,Rule = \{P_{\mathrm{TO},26-29} \geqslant 183.737 \& P_{\mathrm{FROM},23-24} \leqslant 399.561\} \tag{12}$$

where $P_{\mathrm{TO,26\text{-}29}}183.737$ and $P_{\mathrm{FROM,23\text{-}24}}399.561$ represent that the active power flow to line 26-29 larger than 183.737 and the active power flow from line 23-24 smaller than 183.737, respectively. Such rule explains why such operating point is classified as insecure case compared with other operating points in the vicinity; otherwise, if $P_{\mathrm{TO,26\text{-}29}}183.737$ or

$P_{\mathrm{FROM,23\text{-}24}}$399.561, the testing operating point will be classified as the secure case. According to the generation cost coefficients, the cost of rescheduling generator G38 is smaller than that of generator G36, and the active power flow to line 26-29 is directly linked to generator G38, which is consistent with this case that the generator G38 swings relative to other units of the system. Therefore, for this insecure testing operating point, by preventively reducing the $P_{\mathrm{G},38}$ (generation output of G38) and restricting the $P_{\mathrm{TO,26\text{-}29}}$ within the DT rules, this insecure operating point will be controlled back to be secure, which is shown in Fig. 8(b) and its corresponding TSI is 53.41. More technical details and testing results of this idea can be found in [31] and [33].

V. Conclusion

In this paper, an interpretable DL-based TSA method is proposed which aims to acquire the transparent and accurate TSA results. The proposed method consists of a GRU-based TSA model and a DT-based interpretable model, which takes merits of the nonlinear capability of GRU and the interpretability of DT via tree regularization. The proposed method improves the training process by modifying the objective function with the tree regularization, which can encourage the GRU model to become more similar with DT model by multiple iterative training. The whole training process of proposed method is mutually restricted and iterative, rather than only providing the passive post-hoc explanation for an already trained model.

In practice, the GRU- based TSA model can be directly utilized to predict the stability status for real-time application. The corresponding DT-based interpretable model has the consistency with the GRU-based TSA model, hence, can be used to explain the transparent decision process of the GRU-based TSA model. By this, the transparent and accurate TSA decision-making rules for GRU-based model can be obtained for preventive control actions if necessarily. Besides, the system operators can check such TSA results via human-being expert knowledge, and reason the fairness and systematic deviations according to the actual conditions. Simulation results have shown its highest fidelity performance, achieving up to 95%, without sacrificing the TSA accuracy, which can verify the interpretability ability of the proposed method. Besides, the proposed method can balance the TSA accuracy of GRU-based TSA model and the transparency of the DT-based interpretable model by controlling the regularization term coefficient. For our proposed method, there is no limit to the specific DL algorithms, and any other DL algorithms can be utilized to achieve such purpose, such as, RNN, CNN, etc. To the best of our knowledge, interpretability of DL-based models can be a very promising method to solve other similar data-driven problems in power engineering.

References

[1] P. Kundur *et al.*, "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Trans. Power Syst.,* vol. 19, no. 3, pp. 1387–1401, Aug. 2004.

[2] L. Wehenkel, *Automatic Learning Techniques in Power Systems*. Norwell, MA, USA: Kluwer, 1998.

[3] Z. Y. Dong, Y. Xu, P. Zhang, and K. P. Wong, "Using IS to assess an electric power system's real-time stability," *IEEE Intell. Syst.,* vol. 28, no. 4, pp. 60–66, Dec. 2013.

[4] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," in *Proc. IEEE*, 2020.

[5] Y. Xu, Y. Zhang, Z.Y. Dong, and R. Zhang, "Intelligent systems for stability assessment and control of smart power grids: Security analysis, optimization, and knowledge discovery" Boca Raton, FL, USA: CRC Press, 2020.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] S. Wu *et al.*, "Improved deep belief network and model interpretation method for power system transient stability assessment," *J. Modern Power Syst. Clean Energy*, 2019.

[8] A. Bashiri Mosavi, A. Amiri, and H. Hosseini, "A learning framework for size and type independent transient stability prediction of power system using twin convolutional support vector machine," *IEEE Access*, vol. 6, pp. 69937–69947, Nov. 2018.

[9] Y. Zhou, Q. Guo, H. Sun, Z. Yu, J. Wu, and L. Hao, "A novel data-driven approach for transient stability prediction of power systems considering the operational variability," *Int. J. Elect. Power Energy Syst.*, vol. 107, Dec. 2018, pp. 379–394, 2019.

[10] R. Zhang, J. Wu, Y. Xu, B. Li, and M. Shao, "A hierarchical self-adaptive method for post-disturbance transient stability assessment of power systems using an integrated CNN-based ensemble classifier," *Energies*, vol. 12, no. 17, p. 3217, 2019.

[11] R. Yan, G. Geng, Q. Jiang, and Y. Li, "Fast transient stability batch assessment using cascaded convolutional neural networks," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2802–2813, Jan. 2019.

[12] J.-M. H. Arteaga *et al.*, "Deep learning for power system security assessment," *13th IEEE PowerTech 2019*, pp. 1–6, 2019.

[13] L. Zheng *et al.*, "Real-time transient stability assessment based on deep recurrent neural network," in *Proc. 2017 IEEE Innov. Smart Grid Technol.-Asia: ISGT-Asia*, pp. 1–5.

[14] J. J. Yu, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, "Intelligent time-adaptive transient stability assessment system," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1049–1058, 2018.

[15] A. Gupta, G. Gurrala, and P. S. Sastry, "Instability prediction in power systems using recurrent neural networks," in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, pp. 1795–1801, 2017.

[16] M. Barati, "Faster than real-time prediction of disruptions in power grids using PMU: Gated recurrent unit approach," in *Proc. 2019 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf.*, 2019, pp. 1–5.

[17] C. Ren and Y. Xu, "A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5044–5052, Nov. 2019.

[18] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.

[19] H. Mohammadi and M. Dehghani, "PMU based voltage security assessment of power systems exploiting principal component analysis and decision trees," *Int. J. Elect. Power Energy Syst.*, vol. 64, Jun. 2018, pp. 655–663, 2015.

[20] F. Hang, S. Huang, Y. Chen, and S. Mei, "Power system transient stability assessment based on dimension reduction and cost-sensitive ensemble learning," in *Proc. 2017 IEEE Conf. Energy Internet Energy Syst. Integration (EI2)*.

[21] S. Shen, Q. Liu, X. Tao, and S. Ni, "Application of the XGBOOST on the assessment of transient stability of power system," in *Proc. 2019 Int. Conf. Electronical, Mech. Mater. Eng.*, vol. 181, pp. 6–10.

[22] H. Supreme *et al.*, "Development of new predictors based on the concept of center of power for transient and dynamic instability detection," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3605–3615, Dec. 2016.

[23] C. Liu and C. L. Bak, "An accurate online dynamic security assessment scheme based on random forest," *Energies*, vol. 11, no. 7, 2018.

[24] I. Konstantelos *et al.*, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1417–1426, Sep. 2016.

[25] L. Wehenkeland and V. Akella, "A hybrid decision tree-neural network approach for power system dynamic security assessment," in *Proc. 4th Int. Symp. Expert Syst. Appl. Power Syst.*, Melbourne, Australia, Jan. 1993, pp. 285–291.

[26] I. K. Sethi, "Entropy nets: From decision trees to neural networks," *Proc. IEEE*, vol. 78, no. 10, pp. 1605–1613, Oct. 1990.
[27] I. Kamwa, S. R. Samantaray, and G. Joos, "On the accuracy versus transparency trade-off of data-mining models for fast-response PMU-Based catastrophe predictors," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 152–161, Mar. 2012.
[28] J. L. Cremer, I. Konstantelos, and G. Strbac, "From optimization-based machine learning to interpretable security rules for operation," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3826–3836, Sep. 2019.
[29] J. Han, P. Jian, and Micheline Kamber. *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands; New York: Elsevier, 2011.
[30] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *2014 Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
[31] Y. Zhang, Y. Xu, Z. Y. Dong, R. Zhang, and W. Wei, "Mining transient stability database for rule-based preventive control of power systems," in *Proc. 2016 IEEE Power Energy Soc. Gen. Meeting*, Boston, MA, USA, pp. 1–5.
[32] M. Wu *et al.*, "Beyond sparsity: Tree regularization of deep models for interpretability," in *Proc. 32-th AAAI Conf. Artif. Intell.*, 2018.
[33] Y. Xu, Z.Y. Dong, R. Zhang, and K.P. Wong, "A decision tree-based on-line preventive control strategy for power system transient instability prevention," *Int. J. Syst. Sci.*, vol. 45, no. 2, pp. 176–186, Nov. 2014.
[34] Y. Xu, Z.Y. Dong, K. Meng, R. Zhang, and K.P. Wong, "Real-time transient stability assessment model using extreme learning machine," *IET Gen. Trans. Dist.*, vol. 5, no. 3, pp. 314–322, Mar. 2011.
[35] *Transient Security Assessment Tool User Manual*. Surrey, British Columbia: Powertech Labs, 2013.
[36] M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds. Cambridge, MA, USA: MIT Press, 1996, vol. 8, pp. 24–30.
[37] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1611–1619, Aug. 2010.

Chao Ren (Student Member, IEEE) received the B.E. degree from the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017. He is currently working toward the Ph.D. degree in cross-disciplinary of computer science and electrical engineering from Interdisciplinary Graduate School, Nanyang Technological University, Singapore. His research interests include adversarial machine learning, data-analytics, security assessment, interpretability, and their applications to power engineering. He was the recipient of several programming contest awards, including the Champion of Chinese Software Cup, NeurIPS Competition, International College Student "Internet+" Competition.

Yan Xu (Senior Member, IEEE) received the B.E. and M.E. degrees from the South China University of Technology, Guangzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Newcastle, Callaghan, NSW, Australia, in 2013. He conducted postdoctoral research with the University of Sydney Postdoctoral Fellowship, and then joined Nanyang Technological University (NTU) with Nanyang Assistant Professorship. He is currently an Associate Professor with the School of Electrical and Electronic Engineering and the Cluster Director with Energy Research Institute @ NTU (ERI@N). His research interests include power system stability and control, microgrid, and data-analytics for smart grid applications. Dr. Xu is the Editor of several international journals, including the IEEE TRANSACTIONS ON SMART GRID AND THE IEEE TRANSACTIONS ON POWER SYSTEMS. He is also the Chairman of the IEEE Power & Energy Society Singapore Chapter.

Rui Zhang (Member, IEEE) received the B.E. degree in electrical engineering from The University of Queensland, Brisbane, QLD, Australia, in 2009, and the Ph.D. degree in electrical engineering from the University of Newcastle, Newcastle, NSW, Australia, in 2014. She is currently with the School of Electrical Engineering and Telecommunication, University of New South Wales, Sydney, NSW, Australia. Her research interests include power system operation, control, and stability, data analytics, and machine learning applications in power engineering. She was the recipient of the 2022 Australian Research Council Discovery Early Career Researcher Award (ARC DECRA).