IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 1, 2020, accepted April 25, 2020, date of publication April 29, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991263

Recurrent Graph Convolutional Network-Based Multi-Task Transient Stability Assessment Framework in Power System

JIYU HUANG[1], LIN GUAN[1], (Member, IEEE), YINSHENG SU[2], HAICHENG YAO[2], MENGXUAN GUO[1], AND ZHI ZHONG[1]

[1]School of Electric Power, South China University of Technology, Guangzhou 510641, China
[2]CSG Power Dispatching and Control Center, Guangzhou 510663, China

Corresponding author: Lin Guan (lguan@scut.edu.cn)

This work was supported by the China Southern Power Grid Research Project under Grant ZDKJXM20180084.

ABSTRACT Reliable online transient stability assessment (TSA) is fundamentally required for power system operation security. Compared with time-costly classical digital simulation methods, data-driven deep learning (DL) methods provide a promising technique to build a TSA model. However, general DL models show poor adaptability to the variation of power system topology. In this paper, we propose a new graph-based framework, which is termed as recurrent graph convolutional network based multi-task TSA (RGCN-MT-TSA). Both the graph convolutional network (GCN) and the long short-term memory (LSTM) unit are aggregated to form the recurrent graph convolutional network (RGCN), where the GCN explicitly integrate the bus (node) states with the topological characteristics while the LSTM subsequently captures the temporal features. We also propose a multi-task learning (MTL) scheme, which provides joint training of stability classification (Task-1) as well as critical generator identification (Task-2) in the framework, and accelerate the process with parallel computing. Test results on IEEE 39 Bus system and IEEE 300 Bus system indicate the superiority of the proposed scheme over existing models, as well as its robustness under various scenarios.

INDEX TERMS Deep graph-based learning, transient stability assessment (TSA), graph convolutional network (GCN), recurrent graph convolutional network (RGCN), multi-task learning (MTL).

NOMENCLATURE

ABBREVIATIONS

ACC	Accuracy
ADM	Assistant decision-making
BN	Batch normalization
CCT	Critical clearing time
CE	Cross-entropy
CNN	Convolutional neural network
DL	Deep learning
DT	Decision tree
EJS	Expand Jaccard similarity
ELM	Extreme learning machine
FA	False alarm
FC	Full connected
GCN	Graph convolutional network
GRU	Gated recurrent unit
J	Jaccard similarity
JACC	Jaccard accuracy
JACCU	Jaccard accuracy of unstable
LN	Layer normalization
MA	Miss alarm
ML	Machine learning
MLP	Multilayer perceptron
MTL	Multi-task learning
PMU	Phasor measurement unit
RGCN	Recurrent graph convolutional network
RGCN-MT-TSA	Recurrent graph convolutional network based multi-task transient stability assessment
SAE	Stacked auto encoder
SNR	Signal to noise ratio
SVM	Support vector machine
TDS	Time domain simulation
TSA	Transient stability assessment

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

SYMBOLS OF RGCN-MT-TSA

$\boldsymbol{G}, \boldsymbol{G_c}, \boldsymbol{G_s}$	The sets of generators, critical generators and significant generators
$\tilde{\boldsymbol{A}}'', \boldsymbol{X'}, \boldsymbol{O'}$	The block matrix of adjacency, input node features and output convolved features
$\tilde{\boldsymbol{c}}$	The vector of the confidence of the categories or labels
$\tilde{\boldsymbol{G}}_{\boldsymbol{c}}, \tilde{\boldsymbol{G}}_{\boldsymbol{s}}$	The predicted sets of critical generators and significant generators
$\boldsymbol{H}^{\boldsymbol{l}}, \boldsymbol{W}^{\boldsymbol{l}}, \boldsymbol{b}^{\boldsymbol{l}}$	The input, layer-specific weight and bias matrics of the l^{th} layer
$\boldsymbol{p}, \boldsymbol{p_0}$	The learnable and trained parameter
$\boldsymbol{w_0}, \boldsymbol{b_0}$	The trained parameter of Task-1 of the sharing hidden layers
$\boldsymbol{w}, \boldsymbol{b}$	The learnable network parameters
$\boldsymbol{x_t}, \boldsymbol{c_t}, \boldsymbol{h_t}$	The inputs, cell states and hidden states vectors of LSTM at the moment t
$\boldsymbol{Y}$	The node admittance matrix
$\boldsymbol{z}$	The predictive vector for the categories or labels
$\mathcal{G}$	Graph
$\mathcal{V}, \mathcal{E}$	The set of nodes and edges
$\boldsymbol{A}, \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{A}}'$	The adjacency matrix without and with self-loop, renormalization trick
$\boldsymbol{c}$	The vector of the categories or labels
$\boldsymbol{D}, \tilde{\boldsymbol{D}}$	The diagonal node degree matrix without and with self-loop
δ, δ_a	The thresholds
η	Transient stability index
σ	The activation function
$\lvert\Delta\delta\rvert_{\max}$	The absolute value of the maximum rotor angle of separation between any two generators during the simulation time
$\lvert\Delta\delta\rvert_i$	The absolute value of separation between generator i and the reference generator during the simulation time
$\lvert y_{i,j}\rvert, \lvert y_{i,j}\rvert_{\max}$	The module of an element of $\boldsymbol{Y}$ and the max of the modules
C, F	The size of input and output features of each bus (node)
L	The number of edges
M	The number of snapshots
N	The number of buses (nodes)
N_G	The number of generators
T	The length of the observation window
$\alpha_i, \alpha_u, \alpha_s$	The balanced factor of the i^{th} sample, unstable samples and stable samples
β_1, β_2	The regularization weights
$\tilde{\beta}_{i,j}$	The correction factor
f_s	The sampling frequency
t_0, t_c	The moment of fault occurrence and fault clearance

OTHER SYMBOLS

$\boldsymbol{s}$	The set
$\delta(\cdot)$	The pulse function
$\varepsilon, \varepsilon'$	The Gaussian white noise and color noise
$N(0, 1)$	The Gaussian distribution
$\mathcal{O}(\cdot)$	The operation complexity

I. INTRODUCTION

For the past decades, there has been an increasing demand of loads and large-scale deployment of low-inertia converter-based renewable generation in power system [1]. These changes bring challenges for the security of power system operation, especially for the system stability under faults [2]. Fast online transient stability assessment (TSA) is a fundamental tool to provide early-warning for instability and instruction for preventive control of the system.

Conventionally, the core of online TSA is the time domain simulation (TDS), which solves the high-dimensional nonlinear differential-algebraic model of power system [3]. Although, parallel computing [4], improved integration methods [5], energy functions based direct methods [6], [7] are developed to accelerate the simulation, huge computational burden brought by emerging converter modules and expanding of system scale are still the first challenge for the online TSA.

Data-driven machine learning (ML) techniques provide a new train of thought, where the TSA model is established offline with batches of training samples and applied to rapidly assess new contingencies online. Deployment of the phasor measurement units (PMUs) provides fast and accurate dynamic information of the system for ML models. Hence, a variety of ML approaches, such as decision tree (DT) [8], support vector machine (SVM) [9], multilayer perceptrons (MLPs) [10] and extreme learning machine (ELM) [11] are applied to the online TSA modeling, and the targets cover various TSA scenarios, e.g. the stability classification, the critical clearing time (CCT) prediction and the critical generator identification etc. However, for these schemes, the input features should be carefully constructed based on the expert experience.

Deep learning (DL) can extract fine-grained features from big raw data with the help of more hidden layers or even sharing of modules. Zhu *et al*. [12] employ a two-stage TSA method with stacked auto encoder (SAE). James J Q and his colleagues exploit long short-term memory (LSTM) [13] and stacked gated recurrent units (GRUs) [14] in the correlation learning of temporal series. Another popular technique is the convolutional neural network (CNN), which is capable of learning spatial representations of data. For the TSA problem, time-series data are arranged into multi-channel metrics (i.e., each element in a matrix refers to a vector of observations) such that CNN can learn the mapping function from the

inputs to the stability labels [15], [16]. GUPTA *et al*. [17] consider a new description of measured generator data as an image with each value in the data matrix represented as color intensity. The visual dissimilarity of images of stable and unstable cases is then distinguished by CNN, which is trained simultaneously for both stability classification and critical generators (i.e generators most affected under the disturbance) identification. In [18], the authors adopt discrete Fourier transform to obtain spectrum from the fault-on generator trajectories and arrange them into 2D images, such that CNN can achieve good performance in refined CCT regressions. Shi *et al*. [19] construct larger images with variables of all buses and verify the effectiveness of CNN on instability mode (e.g., caused by insufficient synchronizing or damping torque) prediction. Aimed at a large scale of contingency screening, Yan *et al*. [20] introduce cascade CNNs in stability probability prediction for early TDS termination without losses of accuracy, based on continuously refreshing themselves with the increase of labeled TDS outputs. Nonetheless, all above DL models are not specialized in exploring of observations with explicit topological graph correlation, where power system is such an interconnected network of generators and loads [21]. A series of studies [22]–[24] establish that there exists a close relationship between topology and transient stability. As a result, changes of power system topology, which is frequently triggered by maintenance or faults, may deteriorate the performance of TSA models based on SAE, CNN or recurrent methods.

Correspondingly, graph convolutional network (GCN) develops an explicit way of integrating topological structure into the convolution algorithm [25]. GCN has been proved extremely useful for graph analysis tasks in a wide variety of application areas, such as knowledge graph learning [26], text classification [27] and recommender system prediction [28]. The basic idea behind GCN is to distill the high-dimensional information about a node's graph neighborhood into a vector representation with dimension reduction. With this in mind, GCNs are also employed in the field of power system recently, to deal with fault location and load shedding [29], [30]. Specially, under the context of TSA, James J Q *et al*. [31] designs a GCN model for recovery of the missing PMU data and indicate lower errors than existing implementation [14].

In this paper, we propose a new recurrent graph convolutional network (RGCN) for spatio-temporal feature integration. RGCN adopts cascading architecture where the improved GCN modules process measurements at nodes considering the power system graph structures firstly, and then the LSTM modules accomplish the temporal fusion. Based on the RGCN, we further design a multi-task TSA framework, named as the RGCN-MT-TSA in the paper. Multi-task learning (MTL) is exploited for joint training of two subtasks, i.e. stability classification (Task-1) and critical generator identification (Task-2). The proposed framework provides early-warning based on the results of both tasks such that they can verify each other spontaneously. Comprehensive tests are carried out on IEEE 39 Bus system and IEEE 300 Bus system to validate the generalization and robustness of the proposed scheme.

Generally, this paper is highlighted with the following contributions:

1) The adjacency matrix of GCN is designed to representatively describe the graph topology of power system and effectively reflect the inherent physical characteristics.
2) A block-diagonal sparse matrix is constructed with each block corresponding to the adjacency matrix of a graph. Such an attempt supports batch-wise process of graph data and fully utilize parallel computing.
3) A cost-sensitive cross-entropy function is designed to deal with category-imbalanced problem in critical generator identification.
4) A soft sharing scheme is proposed to accelerate the multi-task training.

The rest of this paper is organized as follows. Section II introduces the design of RGCN. Section III presents the data preprocessing and the application of the RGCN-MT-TSA framework with offline training tricks. Section IV demonstrates cases study on two different benchmark systems and various scenarios. The conclusion is discussed in Section V.

II. RECURRENT GRAPH CONVOLUTIONAL NETWORK

In this paper, we propose a novel aggregating network structure, named as the recurrent graph convolutional network (RGCN) and shown in Fig. 1. RGCN consists of four cascading modules, where GCN and LSTM are hierarchical modules while the time pooling and classifier are single ones. GCN and LSTM play a critical role in addressing graphical and temporal feature extraction. Then the time pooling module aggregates features from the whole time steps and the classifier provides final discrimination.

Both the GCN and the LSTM adopt hierarchical stacked structure containing also the normalization layers and full connected (FC) layers. We select two types of normalization layers, i.e. the batch normalization (BN) and the layer normalization (LN). Details of these layers and modules are introduced as follows.

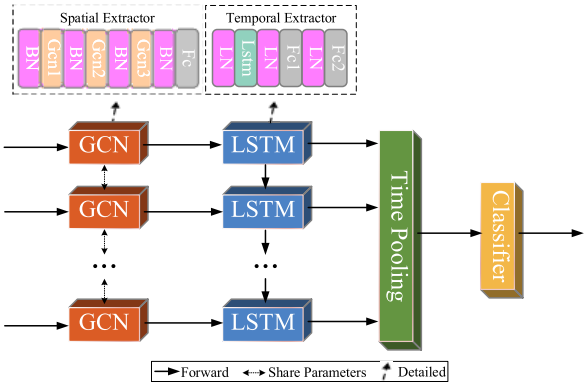

FIGURE 1. Cascade architecture of RGCN.

A. GRAPH CONVOLUTIONAL NETWORK LAYER

CNN performs neighborhood information aggregating on the input data, which is elaborately designed for image-type of signals in Euclidean space. However, fixed filters face difficulty in addressing graph data with irregularity connections of nodes. Instead of convolution with geographical neighborhoods, graph convolutional filters concentrate more on the correlation over graph edges provided by an adjacency matrix. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes an undirected graph where $\mathcal{V} \in \mathbb{R}^N$ denotes the set of nodes and $\mathcal{E} \in \mathbb{R}^L$ denotes the set of edges. Then we have the adjacency matrix $\boldsymbol{A}$ and define a renormalization trick as:

$$\tilde{\boldsymbol{A}}' = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2} \tag{1}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ represents an adjacency matrix with self-connections with its diagonal node degree matrix $\tilde{\boldsymbol{D}}$ calculated by $\tilde{D}_{i,j} = \sum_j \tilde{A}_{i,j}$. The graph-based propagation for a GCN layer that maps a $N \times C$ input feature matrix to a new $N \times F$ output matrix is performed by a nonlinear function [32]:

$$\boldsymbol{H}^{l+1} = \sigma(f(\tilde{\boldsymbol{A}}', \boldsymbol{H}^l)) = \sigma(\tilde{\boldsymbol{A}}'\boldsymbol{H}^l\boldsymbol{W}^l + \boldsymbol{b}^l) \tag{2}$$

where $\boldsymbol{H}^l$ denotes the input matrix of the l^{th} layer, i.e., output of the $(l-1)^{th}$ layer. $\boldsymbol{W}^l \in \mathbb{R}^{C\times F}$, $\boldsymbol{b}^l \in \mathbb{R}^{N\times F}$ are layer-specific weight and bias matrics, respectively. σ is an activation function, where we adopt ReLu(·) in the rest of this paper.

Fig. 2 demonstrates the feed-forward propagation, where each node obtains information from first-order neighbors under (2), i.e., a message passing mechanism [33], and then updates its representation. As the layers stacked, nodes incrementally aggregate more and more "message" from further reaches of the graph. To alleviate the problem of overfitting, all the GCN modules share the same parameters.

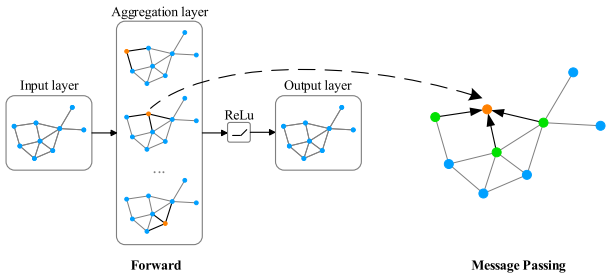

FIGURE 2. Feed-forward propagation with message passing mechanism.

B. LONG SHORT-TERM MEMORY LAYER

As one of the most popular variants of recurrent neural network (RNN), LSTM overcomes the problem of vanishing gradients in deep RNN, which is designed to pass the information of previous time steps to subsequent ones. In Fig. 3, an LSTM cell [34] typically comprises three gates: input, forget and output gates. $\boldsymbol{x}_t$, $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$ represent the vectors of input, cell and hidden states, and σ refers to the sigmoid function $\sigma(x) = 1/(1+e^{-x})$.

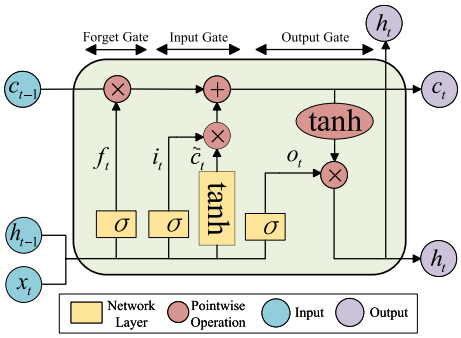

FIGURE 3. An illustration of LSTM cell.

C. FULL CONNECTED LAYER

For an FC layer, the output is a linear transformation of the inputs. Multiple FC layers [10] map the temporal output of LSTM to prediction results at each time step.

D. NORMALIZATION LAYER

In order to reduce the impact of the parameters of all preceding layers to the inputs of current layer in the training process, we adopt the normalization layers, mainly including batch normalization (BN) and layer normalization (LN), to fix the means and variances of each layer's inputs. The effectiveness of BN has been widely proved in conventional CNN [35], and it is introduced between GCN layers similarly in this paper. It is suggested by Kim *et al.* [36] that LN, i.e., a normalization for neurons of the same layer, is preferred to recurrent architecture like LSTM and FC. The normalization promotes the robustness to noises and thus, the model benefits from good performance without fine-tuning parameters of dropout layer [37] or L1 regularization.

E. TIME POOLING MODULE

With sequential inputs of T steps and static outputs, we adopt a global mean pooling approach, called time pooling module to merge per-time step predictions $[z_1, z_2, \ldots, z_T]$ into a single prediction z as:

$$z = \frac{1}{T}\sum_{t=0}^{T} z_t \tag{3}$$

where z denotes a vector of predictive values for the categories or labels.

F. CLASSIFIER

In our application, two different targets, i.e. stability classification and critical generator identification are addressed. These two targets belong to different fields in ML, i.e., binary-category classifier and multi-label classifier respectively.

- **Binary-category classifier**
  The softmax function is utilized and we obtain the confidence $\tilde{c}_i$ for the category i as:

$$\tilde{c}_i = \frac{e^{z_i}}{\sum_i e^{z_i}} (j = 1, 2) \tag{4}$$

where $\tilde{c}_1 + \tilde{c}_2 = 1$. The system is predicted as unstable when $\tilde{c}_1 > 0.5$, which is labeled as $[1, 0]^T$. Otherwise, the system is stable and labeled as $[0, 1]^T$.

- **Multi-label classifier**
  For multi-label classification, each sample is simultaneously associated with a set of labels. We assign a 0/1 binary code for each label to represent False/True. Then the problem can be actually decomposed to multiple related binary-category learning. We adopt the sigmoid function that limits $\boldsymbol{z}$ to $\tilde{\boldsymbol{c}}(\tilde{c}_i \in (0, 1))$, where $\tilde{\boldsymbol{c}}$ denotes a vector of the confidence of all labels. Define a threshold δ, a label i is predicted to be true one when its confidence $\tilde{c}_i \in (0, \delta]$, and the final output is a binary vector.
  Going back to the critical generator identification problem, each label is corresponding to a generator and the set of true labels from output refers to that of the predicted critical generators $\tilde{\boldsymbol{G}}_c$. Furthermore, if there is a set of labels whose confidence belongs to (δ, δ_a) with $\delta < \delta_a < 1$, we say this set refers to the set of predicted significant generators $\tilde{\boldsymbol{G}}_s$.

III. THE RGCN-MT-TSA FRAMEWORK

A. GENERAL INTRODUCTION OF THE FRAMEWORK

In this paper, we propose a multi-task TSA solution based on RGCN. The framework is shown in Fig. 4.

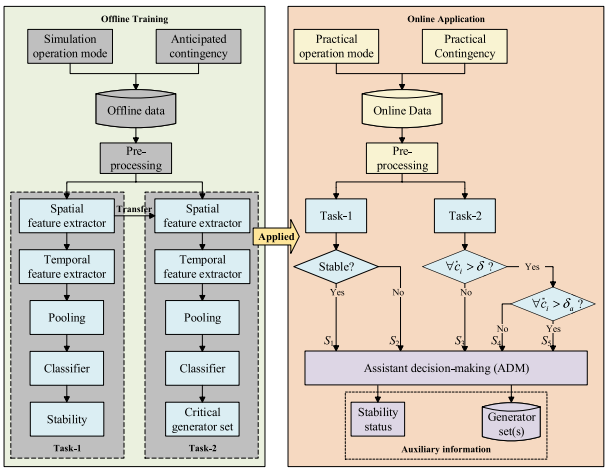

FIGURE 4. The flowchart of the proposed RGCN-MT-TSA framework.

1) MULTI-TASK DESIGN

Our TSA task is composed of two subtasks, i.e. the stability classification (Task-1) and the critical generator identification (Task-2). Conventionally, different tasks may have distinguished parameters or even architectures. However, individual designed blocks the sharing of knowledge in the training process. In fact, for the transient stability problem, the judgment of instability has strong, or even causal, links to the behavior of the critical generators. Hence, we follow the conceptional idea of multi-task learning (MTL) in our design such that the model shares the representation of the related tasks and performs better on the target tasks.

MTL is essentially a multi-objective optimization, i.e., multiple loss functions are simultaneously minimized based on gradient descent. In the context of DL, hard and soft parameter sharing [38], [39] are the most commonly used settings for MTL. The former requires all the tasks to share the same subset of the hidden layers and thus effectively alleviates the chance to overfit. However, the drawback is that we might solve the multi-objective programming directly to obtain the common representation that captures multiple tasks. Another practical way is to merge the weighted loss functions and thus optimize the single-objective problem. The assignment of the weights among the tasks implements a direct effect on the generalization of all the tasks. Here, we adopt the latter setting.

As shown in Fig. 4, task-1 and task-2 have their own models and parameters, but we regularize the distance between the parameters of the two models to encourage their parameters to be similar. Considering that the operation complexity for Task-2 is significantly larger than Task-1 with when there are tens of labels, Task-1 will be trained at first and its spatial feature extractor, GCN modules, are then transferred to Task-2 as an initial setting. A regularization term is merged in the loss function of Task-2 to minimize the distance between its parameters and the trained parameters of Task-1. With the benefit of such a generalization design, we use the implicit experience in Task-1 as guidance for the parameters optimization of Task-2 and the multi-objective problem is simplified to a two-stage single-objective optimization.

2) INPUT VECTOR

We choose three physical variables of each bus (node) to form the input space of GCN. They are the bus voltage magnitude, the bus relative phase and the rotor speeds of generators connected to the power plant bus, i.e., the derivative of rotor angles with respect to time. For the load buses, their values of the third variable (rotor speed) are uniformly set to zero. The observation time window of the model inputs starts from the moment of fault occurrence t_0 and ends at the fault clearance period t_c (including t_{0-} and t_{c+}). Denoting length of the observation window as T and the sampling frequency as f_s, then the number of snapshots of above variables will be $M = Tf_s + 1$. Therefore, for a power system with N buses, we need M RGCNs and each RGCN has an input feature matrix of size $N \times 3$.

In the online TSA, the input data can be obtained from either PMUs or TDS.

3) ADJACENCY MATRIX

For any graph $\boldsymbol{\mathcal{G}} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$ describing the structure of power system, the nodes refer to the buses, while the edges refer to the transmission lines. Typically, the element at (i, j) of the adjacency matrix $\boldsymbol{A}$ is defined as follows:

$$a_{i,j} = \begin{cases} 0 & \mathcal{V}_i, \mathcal{V}_j \in \boldsymbol{\mathcal{V}}, (\mathcal{V}_i, \mathcal{V}_j) \notin \boldsymbol{\mathcal{E}} \\ 1 & \mathcal{V}_i, \mathcal{V}_j \in \boldsymbol{\mathcal{V}}, (\mathcal{V}_i, \mathcal{V}_j) \in \boldsymbol{\mathcal{E}} \end{cases} \tag{5}$$

where $(\mathcal{V}_i, \mathcal{V}_j)$ denotes the edge from i to j.

Note that the power disturbance caused by fault will propagate through transmission lines and the distribution of bus power along the lines is approximately proportional to the admittance of the transmission lines. Thus we take the advantage of the node admittance matrix $\boldsymbol{Y}$ and define the domain-related adjacency matrix of $\boldsymbol{\mathcal{G}}$ in the GCN as follows:

$$a_{i,j} = \begin{cases} \dfrac{|y_{i,j}|}{|y_{i,j}|_{\max}} & y_{i,j} \in \boldsymbol{Y}\ ,\ i \neq j \\ 0 & y_{i,j} \in \boldsymbol{Y}\ ,\ i = j \end{cases} \tag{6}$$

where $|y_{i,j}|$ denotes the module of an element of $\boldsymbol{Y}$ with the maximum $|y_{i,j}|_{\max}$ as the reference value for normalization.

Substitute (23) in (7), we obtain the renormalized matrix with its element as:

$$\begin{cases} \tilde{a}'_{i,j} = \dfrac{1}{\sqrt{\tilde{d}_{i,i}\tilde{d}_{j,j}}}\tilde{a}_{i,j} = \tilde{\beta}^{-1}_{i,j}\tilde{a}_{i,j} \\ \tilde{\beta}_{i,j} = \sqrt{1 + \dfrac{\sum_k |y_{i,k}|}{|y_{i,j}|_{\max}}}\sqrt{1 + \dfrac{\sum_k |y_{j,k}|}{|y_{i,j}|_{\max}}} \\ \tilde{d}_{i,i} = \dfrac{\sum_k |y_{i,k}|}{|y_{i,j}|_{\max}} (i \neq k) \end{cases} \tag{7}$$

where $\tilde{d}_{i,i}$ denotes the degree of the node i with node j as one of its neighbors. $\tilde{a}_{i,j}$, $\tilde{a}'_{i,j}$ are elements of $\boldsymbol{A}'$ and $\tilde{\boldsymbol{A}}'$ respectively, while the correction factor $\tilde{\beta}_{i,j}$ is related to the degrees of the node itself and its neighbor.

In terms of the massage passing, above setting enables the neighbors with larger degrees (i.e., physically the sum of its normalized admittance) being assigned with smaller weighted edges so as to prevent them from occupying a large amount of the neighborhood "message". On one hand, it reduces the difficulty for GCN to focus on critical buses. On the other hand, the neighborhood features might be implicitly utilized as additional information when certain nodes face the data integrity problem with slightly poor measurement or communication quality.

It is worth noting that for the 1^{st} to the $(M-1)^{th}$ RGCNs, the adjacency matrix follows the power system pre-fault topology, while the adjacency matrix for the M^{th} RGCN will reflect the post-fault topology. For instance, if the fault line between bus i and j is tripped, let $\tilde{a}'_{i,j} = 0$.

4) GROUND TRUTH DATA

The ground truth for our method, i.e., the stability status of the system, is decided by the post-fault rotor angle waveforms for a longer period. It can be practically obtained by introducing the transient stability index η:

$$\eta = \frac{180^\circ - |\Delta\delta|_{\max}}{180^\circ + |\Delta\delta|_{\max}} \tag{8}$$

where $|\Delta\delta|_{\max}$ is the absolute value of the maximum rotor angle of separation between any two generators during the simulation time. When $|\Delta\delta|_{\max} > 180^\circ$, i.e., $\eta < 0$, one or more generators lose their synchronization. We defined this as a transient unstable sample, and label the stability status with the vector $\boldsymbol{c} = [1, 0]^T$. Otherwise, it is considered as a stable one and labeled with $\boldsymbol{c} = [0, 1]^T$.

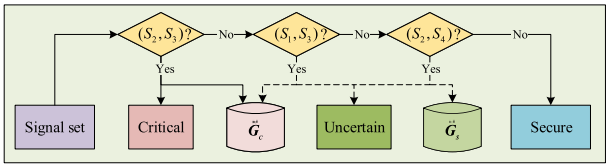

FIGURE 5. The logic diagram of the ADM module.

For an unstable sample, particularly, there exist leading instability generators. Let N_G denote the number of generators, then we define $\boldsymbol{G}$ as the set of generators and $\boldsymbol{c} \in \mathbb{R}^{N_G}$ is a binary vector that denotes the stability status of all the generators. The status of a generator i is expressed as:

$$c(i) = \begin{cases} 0 & |\Delta\delta_i| > 180^\circ \\ 1 & |\Delta\delta_i| \leq 180^\circ \end{cases} \tag{9}$$

where $|\Delta\delta_i|$ is the absolute value of separation between generator i and the reference generator during the simulation time. The set of critical generators $\boldsymbol{G}_c = \{i \in \boldsymbol{G}\,|c(i) = 0\}$ is a subset of $\boldsymbol{G}$ and in particular, $\boldsymbol{G}_c$ is an empty set if and only if the system is transient stable.

5) ASSISTANT DECISION MAKING MODULE

For online application of TSA, there may be conflict in the results of task-1 and task-2. We define an assistant decision-making (ADM) module to provide logically consistent estimation. According to Fig. 5, ADM first analyzes the signal set including two single signals (i.e., S_1 or S_2 from Task-1 and S_3, S_4 or S_5 from Task-2), and simultaneously provides the post-fault stability status as well as the sets of generators based on parallel computing. The detailed logic diagram of ADM is demonstrated in Fig. 5, where "Critical", "Uncertain" and "Secure" refer to the final signals of stability status, while $\tilde{\boldsymbol{G}}_c, \tilde{\boldsymbol{G}}_s$ denote the sets of critical and significant generators.

When ADM receives the set (S_2, S_3), the system enters a state of emergency with the signal "Critical". Meanwhile, $\tilde{\boldsymbol{G}}_c$ is provided for dispatchers to implement critical control. The stability status of the system is uncertain, however, when receiving the set (S_1, S_3) or (S_2, S_4). In this case, ADM outputs $\tilde{\boldsymbol{G}}_c$ as the collection of potentially critical generators and $\tilde{\boldsymbol{G}}_s$ to be monitored with signals (S_1, S_3), or only $\tilde{\boldsymbol{G}}_s$ if $\tilde{\boldsymbol{G}}_c$ remains unknown with (S_2, S_4). TDS is also suggested, if necessary, for further determination. Except for the above cases, the system is definitely secure.

B. CRITICAL TRICKS FOR OFFLINE TRAINING

1) GRAPHICAL PARALLEL COMPUTING

For a GCN module, the adjacency matrix $\tilde{\boldsymbol{A}}'$, as well as the matrix of its degrees are typically sparse matrices. It is easily perceived that the sparseness has a marked increase in a larger graph. According to (2), a node can only communicate with their first-order neighbors in a single propagation, while any two non-connected nodes, i.e., with no available path between

them, cannot gain "message" from each other. Therefore, we consider batches of graphs as subgraphs of one or more large graphs, with the characteristic that any two nodes belonging to two different subgraphs are still separated from each other in the synthetic graph. The parallel computing process for n subgraphs is illustrated as Fig. 6.

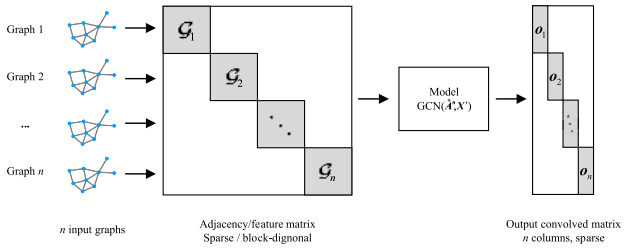

FIGURE 6. Parallel computing process.

When the process is GPU accelerated and the block matrix does not suffer from memory leak, a single sparse operation with complexity $\mathcal{O}(nL)$ in a convolution operation as (2) is converted to n parallel operations with $\mathcal{O}(L_i)$. Hence, The max complexity of the sparse operation drops down to $\mathcal{O}(L_{\max})$. In our model, we assume a sample as a graph-based series of size T, and a block-diagonal matrix for a batch with m samples is

$$\tilde{\boldsymbol{A}}'' = \text{diag}([\tilde{\boldsymbol{A}}'_{1,1}, \tilde{\boldsymbol{A}}'_{1,2}, \ldots, \tilde{\boldsymbol{A}}'_{1,T}, \ldots \tilde{\boldsymbol{A}}'_{2,T}, \ldots, \tilde{\boldsymbol{A}}'_{m,T}]) \quad (10)$$

corresponding to a block matrix of node features $\boldsymbol{X'}$ and an output convolved sparse matrix $\boldsymbol{O'}$.

2) COST-SENSITIVE CROSS-ENTROPY FUNCTION

- **Task-1**
  Cross-entropy (CE) is widely adopted as the cost function for classification tasks. However for the problem with imbalanced samples, the stable (negative) samples attract too much attention and as a result, the unstable (positive) samples suffer a loss of fit and generalization. Here, we adopt the cost-sensitive cross-entropy (CSCE) function with L2 regularization term as:

$$\text{Loss}_1 = -\sum\nolimits_i \alpha_i (\sum\nolimits_j c_{i,j} \log \tilde{c}_{i,j}) + \text{Loss}_{L2} \quad (11)$$

  where α_i is the balanced factor. Normally, α_i of the unstable samples has a bigger value to encourage higher accuracy (ACC) for them. $[c_{i,1}, c_{i,2}]$ is the annotated categories, while $[\tilde{c}_{i,1}, \tilde{c}_{i,2}]$ denotes the softmax function outputs of the i^{th} sample. It follows that

$$\text{Loss}_{L2} = \frac{1}{2}\beta_1(\|\boldsymbol{w}\|^2 + \|\boldsymbol{b}\|^2) \quad (12)$$

  with $\boldsymbol{w}$ and $\boldsymbol{b}$ as learnable network parameters. β_1 is the regularization weight.
- **Task-2**
  Assume $[c_{i,1}, c_{i,2}, \ldots, c_{i,L}]$ to be annotated labels and $[\tilde{c}_{i,1}, \tilde{c}_{i,2}, \ldots, \tilde{c}_{i,L}]$ to be the sigmoid function outputs,

TABLE 1. Confusion matrix for Task-1.

	Actual unstable	Actual stable
Predicted unstable	TN	FN
Predicted stable	FP	TP

we can generalize (28) to L-label learning:

$$\text{Loss}_2 = -\sum\nolimits_i \alpha_i (\sum\nolimits_j c_{i,j} \log \tilde{c}_{i,j} + (1 - c_{i,j}) \times \log(1 - \tilde{c}_{i,j})) + \text{Loss}_{L2} + \text{Loss}'_{L2} \quad (13)$$

Here, an extra regularization term is added to maintain the similarity between both tasks, which is defined as:

$$\text{Loss}'_{L2} = \frac{1}{2}\beta_2(\|\boldsymbol{w} - \boldsymbol{w}_0\|^2 + \|\boldsymbol{b} - \boldsymbol{b}_0\|^2) \quad (14)$$

where $\boldsymbol{w}_0$ and $\boldsymbol{b}_0$ are trained parameters of the sharing hidden layers of Task-1, while β_2 is another regularization weight.
All the loss functions mentioned above are optimized with Adam algorithm [40], which is one of the most commonly-used optimization algorithms for DL.

C. PERFORMANCE METRICS

Taking the difference between the tasks into account, we designed two categories of metrics to measure the performance of the model.

1) CONFUSION MATRIX BASED METRICS

Based on the confusion matrix in Tab. 1, (16) to (19) explain the specific metrics for our model evaluation, including ACC, miss alarm (MA) rate, false alarm (FA) rate and G-mean.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (15)$$

$$\text{MA} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad (16)$$

$$\text{FA} = \frac{\text{FN}}{\text{FN} + \text{TP}} \quad (17)$$

$$\text{G}-\text{mean} = \sqrt{(1 - \text{MA})(1 - \text{FA})} \quad (18)$$

where ACC denotes the proportion of the correctly predicted samples. MA represents the proportion of the correct results in all unstable samples, which reflects the reliability of assessment with a higher risk priority than FA. FA represents the proportion of the correct results of the stable ones, which is used to monitor excessive alarm. Furthermore, G-mean is a comprehensive index for the classification of imbalanced samples.

2) SET SIMILARITY BASED METRICS

Distinguished from Task-1 with scalar based evaluation, Task-2 predicts the set of critical generators. Here, we introduce the Jaccard similarity to evaluate the distance between

sets of integers. Given any two set $s_i, s_j \in \mathbb{N}$, Jaccard similarity is defined as:

$$\mathrm{J}(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|} = \frac{|s_i \cap s_j|}{|s_i| + |s_j| - |s_i \cap s_j|} \tag{19}$$

where $\mathrm{J} \in [0, 1]$ and $\mathrm{J}(s_i, s_j) = 1$. Here, we consider the sample correct only when $\mathrm{J}(\tilde{G}_c, G_c) = 1$. Similar to ACC and MA, we define Jaccard accuracy (JACC) of all samples as well as Jaccard accuracy of unstable (JACCU) samples.

In terms of the parameter similarity between Task-1 and Task-2, we prefer expand Jaccard similarity (EJS) instead of Jaccard similarity that considers the difference in value and direction of an ordered set, e.g. a vector. Given vectors $v_i, v_j \in \mathbb{R}$, EJS is calculated by:

$$\mathrm{EJS}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| + \|v_j\| - v_i \cdot v_j} \tag{20}$$

Here we take account of two parameter sets $p_0 = [w_0, b_0]$ for Task-1 and $p = [w, b]$ for Task-2, and the similarity of the sharing layers can be expressed as:

$$\mathrm{EJS}(p, p_0) = \frac{1}{2}(\mathrm{EJS}(w, w_0) + \mathrm{EJS}(b, b_0)) \tag{21}$$

where $\mathrm{EJS} \in [0, 1]$. The closer the similarity is to 1, the better the two sets of parameters satisfy the similarity constraint.

IV. CASE STUDY

We set up the RGCN-MT-TSA model for two test systems with different sizes: the IEEE 39 Bus system and the IEEE 300 Bus system. The TDS platform PSD-BPA is applied to generate the training and testing samples. The proposed model is implemented in Pytorch [41] developed by Facebook. All the tests are fulfilled on a computer with Intel Core i7-9700 3.0GHz CPU,16GB RAM and GTX 1660Ti 6G GPU.

A. IEEE 39 BUS SYSTEM

1) TEST SYSTEM AND TDS SETTING

The IEEE 39 Bus system has 39 buses, 10 generators, 19 loads and 46 transmission lines. All generators use 6^{th}-order model, with excitation system of IEEE model type I and speed control system of IEEE G1. TDS data are generated according to the principles as follows:

- "N−1" and "N−2" cases are generated with the transmission lines of the basic cases randomly switched off. Those cases with islands are rejected.
- All the loads change within 75% to 120% of basic load level, while the generator outputs are randomly adjusted until the feasible power flow can be obtained.
- Contingency: three-phase to ground faults at the beginning and the end of any transmission line, and cleared after 0.1s with tripping of the faulted line.
- Labels of contingencies are determined according to the TDS lasting 4s.

Finally, 29500 samples are generated with 25168 stable ones and 4332 unstable ones.

TABLE 2. RGCN construction in IEEE 39 Bus system.

Layer	Task-1	Task-2
GCN (input feature size, output feature size, time step)		
gcn1	(39×3, 39×16,11)	
gcn2	(39×16, 39×16,11)	
gcn3	(39×16, 39×8,11)	
fc	(312, 64, 11)	
LSTM (input size, output size,time step)		
lstm	(64, 64, 11)	(64, 128, 11)
fc1	(64, 32, 11)	(128, 84, 11)
fc2	(32, 2, 11)	(84, 10, 11)

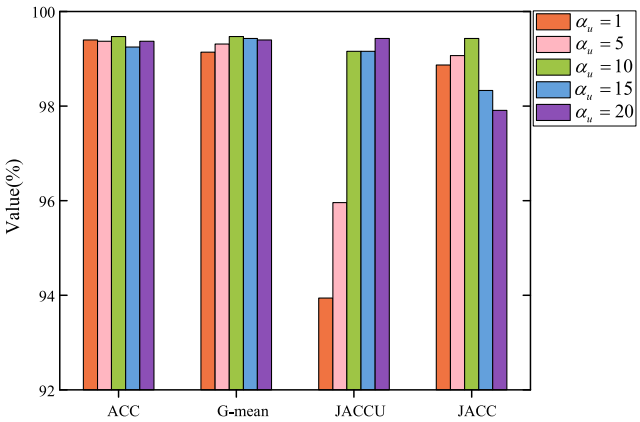

FIGURE 7. Comparison among α_u of different values.

2) MODEL SETTING AND METHODS COMPARISON

The samples from "Base" and "N−1" cases form the training set (60%), while the "N−2" cases are used for validation (20%) and test (20%). This can verify the adaptability and generalization of our method on various topologies. We set the input time window T as 0.1s with reference to the shortest fault duration of the contingencies and the PMU sampling frequency f_s as 100Hz. Hence, the total of input time steps are $M = 11$. Test results are analysed in various aspects with proposed model setting as follows.

- **Design of the RGCN-MT-TSA**
  The learning rate for training is set to 0.001, while the batch size is 256. Regularization weights β_1, β_2 are 5e-4, 0.01 and the balanced factor of the stable α_s is set as 1. Note that the ratio of the stable samples to the unstable samples is about 6:1. We intialize the balanced factor of the unstable α_u as 6. Assume δ equal to 0.5, we compare groups of hyperparameter settings of RGCNs on validation set and definitely select the best model with the detailed setting as Tab. 2.
  To go a step further, we search a more desirable value for α_u according to the metrics comparison in Fig. 7. As the value of α_u shows an growing trend, there are insignificant changes in ACC but distinct rises in JACCU.

TABLE 3. Metrics comparison of methods in IEEE 39 Bus system.

Method	ACC(%)	MA(%)	FA(%)	G-mean(%)
ω-SVM[a]	95.31	3.82	12.40	91.79
SVM	96.52	2.36	3.60	97.02
MLP	96.65	2.53	3.44	97.01
SAE	97.04	2.19	3.05	97.34
SSAE[b]	97.37	2.02	2.70	97.64
CNN	97.32	1.85	2.73	97.69
LSTM	97.27	2.76	2.72	97.21
RSAE	97.50	1.01	2.76	98.16
RCNN	98.33	1.35	1.70	98.48
Proposed	99.47	0.50	0.54	99.47

[a] ω-SVM is SVM that only takes generator speeds ω as input in [42]
[b] Stacked sparse auto encoder (SSAE) is SAE with sparse constraint in [43]

Nonetheless, we can not ignore that JACC suffer a loss when $\alpha_u > 10$. In this case, with high balanced factors, the model over-emphasizes the unstable samples and thus has a worse performance on the stable samples. To balance the trade-off between JACC and JACCU, we ultimately set α_u as 10 such that our model can meet the requirement of both tasks.

- **Single Method Experiment**
  Tab. 3 shows the performance of different ML methods on the test set. Obviously, the generalization of the typical shallow networks, ω-SVM, SVM and MLP, are the weakest with ACC less than 97%, which can be significantly improved by DL methods, e.g., SAE, SSAE, CNN and LSTM. The DL models benefit at least 0.3% increase in ACC and G-mean. Among them, CNN can simultaneously capture the spatio-temporal features and perform the best on the prediction of unstable samples. The MA is reduced by 0.34% and 0.91% compared with SAE and LSTM, respectively. The latter two methods are comparable on ACC and G-mean, while the weight-sharing based LSTM has a lighter storage burden, 1.2MB, only about 1/25 of that of SAE.
- **Ablation Experiment**
  To check the necessity and effectiveness of spatial extraction and GCN modules, we construct another two composite models based on the proposed method by substituting GCN for SAE and CNN as spatial extractors. Then we have a recurrent convolutional neural network (RCNN) and a recurrent stacked auto encoder (RSAE). Multi-dimensional hidden layer activations of the spatial extractors are visualized with t-SNE [42] as Fig. 7, where a circle refers to a sample and the background color intensity denotes the values of confidence. V_0 and V_1 represent the compressed 2D activations.
  Among the models, the spatial extractors of RSAE and RCNN require to be pretrained [12] for 160 iterations. As shown in Tab. 3, the performance of RSAE changes slightly compared with the single methods, while ACC and G-mean of RCNN have a remarkable rise by approximately 1% and 0.8%. From Fig. 8 a and Fig. 8 b, the overlap of samples belonging to different categories decreases if we replace SAE with CNN. These results reflect that CNNs are more efficient modules to discover critical spatial information. Furthermore, in Fig. 8 b there are clusters of samples along the classification boundary, i.e., the white area, while most samples are far from the boundary in Fig. 8 c. We can thus infer that additional representation, graphically learned by GCN, contributes to the compactness of the samples belonging to the same category but enhances the distance between those belonging to different categories on the contrary. It explains clearly why our method has the best performance that ACC and G-mean are more than 99.4% with a gain of 1%, when GCN takes the place of CNN.

TABLE 4. Metrics comparison of composite methods in IEEE 39 Bus system under "N−3" cases.

Method	ACC(%)	MA(%)	FA(%)	G-mean(%)
RSAE	87.87	3.11	17.05	89.64
RCNN	89.47	5.08	13.50	90.61
Proposed	96.31	3.11	4.01	96.44

In order to evaluate on the extent of topology changes that the TSA model can be tolerated, we set up 50 "N−3" operation conditions and randomly generate 2004 new samples to form an extra test set. Three composite models are tested on the set with the results listed in Tab. 4. To face the greater topology changes, the ACC of the non-topological learning models, RSAE and RCNN, decline dramatically to be below 90%. Nevertheless, the proposed method maintains good performance with both the ACC and G-mean over 95%. An explanation may be that with the topology information (adjacency matrix) integrated, the GCN modules are capable of extracting the characteristics similarity between untrained topology variations and the known ones. The results indicate that the topological learning method benefits stronger adaptability and robustness than traditional methods in case of topology changes.
Fig. 9 a depicts the training curves, where our proposed model converges quickly and smoothly. ACC of our method grows up sharply to 95% for less than 200 iterations (nearly 60% of RCNN), without pretraining. As for an iteration, RSAE and RCNN are trained for a relatively short time, while RGCN requires a response time of up to 15810ms without parallel computing. Here the response time refers to the prediction time. The training and test time with our batch-wise method falls down dramatically to 1/176, 1/1486 of the previous method. Furthermore, a single sample is tested for merely 35μs, almost half of that of RCNN, meeting the rapidity of the online application in practice.

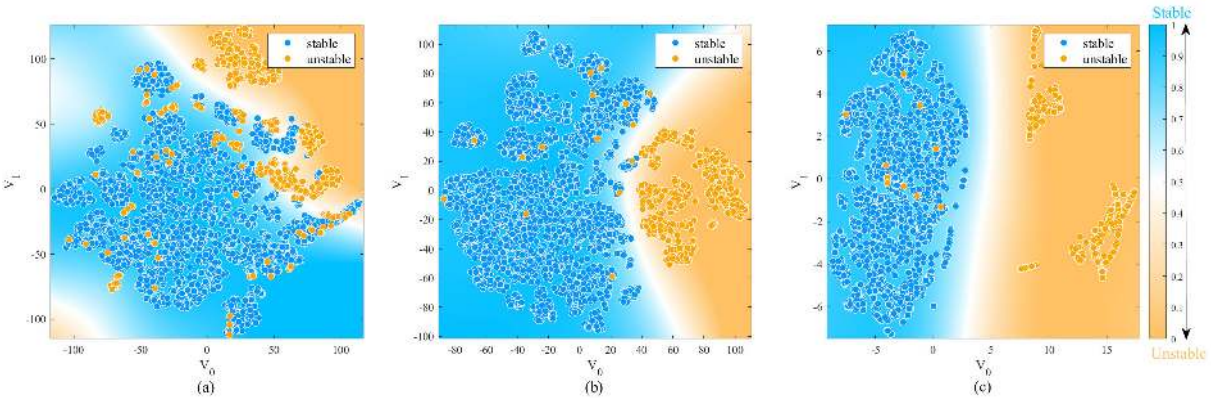

FIGURE 8. Hidden layer activations visualization of the spatial extractors. (a) RSAE. (b) RCNN. (c)RGCN.

3) ROBUSTNESS ANALYSIS

Once the offline model is applied online, we should consider the input damage problem led by loads fluctuation or poor measurement. On one hand, the existing research tends to simplify the distribution of error in the sampling and calculation stage, as ideal Gaussian white noise $\boldsymbol{\varepsilon} \sim N(0, 1)$. However, white noise might be converted to color noise [43] in the low-pass filtering of PMUs. The pulse expression of a low-pass filter is defined as:

$$h(t) = \frac{1}{15} \sum_{i=0}^{15} \delta(t - i) \tag{22}$$

where δ denotes the pulse function. A series of Gaussian color noise $\boldsymbol{\varepsilon'}$ is generated as:

$$\boldsymbol{\varepsilon'} = h * \boldsymbol{\varepsilon} \tag{23}$$

Generally, signal to noise ratio (SNR) is used to calculate distance between noise and signal:

$$\mathrm{SNR} = 20 \lg \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{\varepsilon'}\|} \tag{24}$$

SNR of small values refers to high signal distortion. On the other hand, communication error or signal interference, etc., usually result in data missing or abnormal values in the sampling stage. We simulate this scenario by assuming values of data drop to zero or soar to two times of themselves with an assumed probability.

Extensive performance of the models for both tasks in above multiple scenarios is listed as Tab. 5. Under the ideal scenario, the changes in the loss functions make no significant difference to ACC of Task-1, while we find the improvement of 0.68% in MA with CSCE. JACCU of Task-2 rises by over 5% with the parameter similarity (EJS) of more than 0.99.

All the metrics remain practically unchanged when considering the color noise of big values. Assuming an extreme condition SNR = 20dB where the noise reaches 10% of the original input, MA and FA slightly increase to almost 1%. Due to distinctively more prediction objectives than Task-1, JACCU suffers a drop of around 2%. Nonetheless, our model covers 97% of the unstable samples under large noise interference.

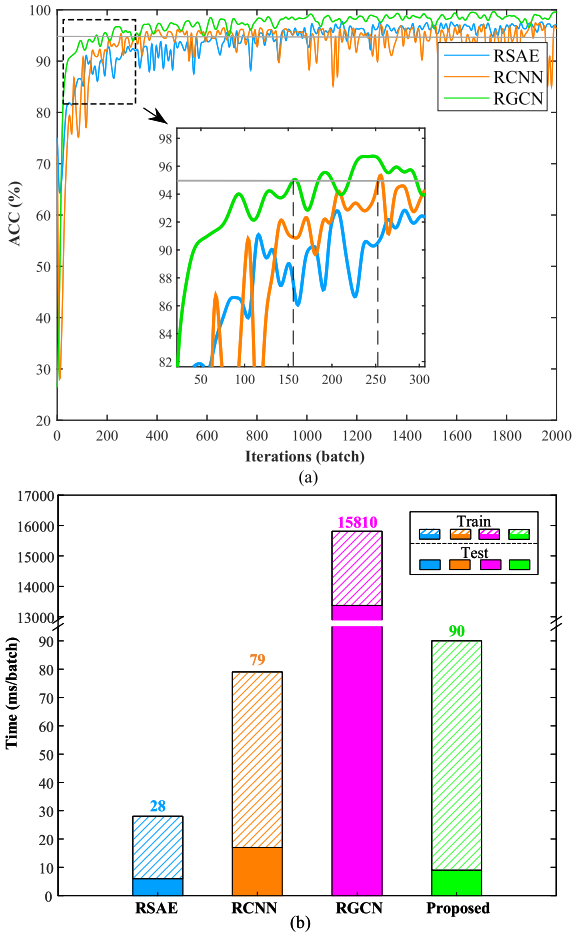

FIGURE 9. Convergence metrics comparison of the composite methods. (a) Training curves. (b) Response time.

Considering wide area abnormal values of 1% to 3%, the proposed method with message passing based GCN modules is only mildly affected by individual abnormal buses. The max loss of metrics in both tasks keeps less than 2% compared with the ideal scenario. On the whole, the proposed method results in desirable performance under the designed scenarios and fulfills the requirements of adaptability and robustness.

TABLE 5. Robustness metrics of Task-1 and Task-2.

Scenario		ACC(%)	MA(%)	FA(%)	G-mean(%)	JACCU(%)	JACC(%)
Ideal	With CSCE	99.47	0.50	0.54	99.47	99.16	99.43
	With CE	99.40	1.18	0.54	99.14	93.94	98.87
Color noise (dB)	40	99.42	0.67	0.57	99.38	99.32	99.18
	30	99.38	0.67	0.61	99.36	99.33	99.18
	20	99.07	1.01	0.92	99.03	96.97	97.93
Abnormal values (%)	1	99.33	0.67	0.67	99.33	98.82	99.40
	2	99.40	1.01	0.55	99.22	98.48	99.37
	3	99.30	0.84	0.68	99.24	97.98	99.33

B. IEEE 300 BUS SYSTEM

1) TEST SYSTEM AND TDS SETTING

We employ our method to a system of larger scale and higher complexity called IEEE 300 Bus system in this subsection. It comprises 300 buses, 69 generators, 203 loads and 411 transmission lines. All generators use 6^{th}-order model, with parameters of the control systems obtained from the practical grid. Similarly, 31062 samples are generated including 26966 stable ones and 4096 unstable ones. Test results are listed in Tab. 6.

2) TEST RESULTS

In contrast with IEEE 39 Bus system, the input scale grows at a geometric progression and as a result, ACC of the shallow networks declines by 8.93% to 13.49%. Deep networks have ACC over 90% and LSTM performs worst among them. It is inferred that a single temporal method generalizes poorly to data with abundant spatial and topological characteristics. In terms of the composite methods, our method is the most reliable one regardless of the system scale. Here ACC and G-mean both maintain about 99%. We then transfer the pretrained GCN modules of Task-1 to more fine-grained Task-2, where the numbers of generators to be predicted are almost 7 times of those in the previous system. JACCU and JACC of the proposed method are 97.32%, 97.98% and meanwhile, the similarity regularization is satisfied with EJS of 0.992.

C. VISUALIZATION VERIFICATION OF RGCN-MT-TSA FRAMEWORK

Assume δ_a equal to 0.9, we apply the proposed framework online based on parallel computing of Task-1 and Task-2. The average time of a batch assessment in IEEE 39 Bus system, as well as IEEE 300 Bus system, is respectively 16ms and 62ms. It follows that ADM generates three signals, e.g., ''Secure'', ''Uncertain'' and ''Critical''. The following typical examples described in Fig. 10 are selected to verify the effectiveness of the designed signals, where Fig. 10 a and

TABLE 6. Metrics comparison of methods in IEEE 300 Bus system.

Method	ACC(%)	MA(%)	FA(%)	G-mean(%)
ω-SVM	84.70	14.43	15.42	85.07
SVM	83.03	5.96	18.40	87.60
MLP	87.72	10.42	12.53	88.52
SAE	95.90	2.53	4.30	96.58
SSAE	96.43	5.21	3.35	95.72
CNN	96.16	1.79	4.11	97.04
LSTM	90.02	6.70	10.41	91.62
RSAE	96.98	2.23	3.12	97.32
RCNN	97.34	2.38	2.70	97.46
Proposed	98.94	0.74	1.11	99.07

Fig. 10 d refers to true stable samples while the others refer to true unstable samples. Hidden activations of generator nodes are similarly compressed as 2D vectors with t-SNE. Here a circle represents a generator and its color intensity is related to the confidence. The set of predicted critical generators is highlighted by a solid oval, and that of predicted significant generators is circled by a dashed one.

From details in Fig.10 a and Fig.10 d, the systems are predicted to be unstable in Task-1, while the whole generators are considered to be stable in Task-2 with all of their confidence over 0.9. Therefore, ADM determines the system status as ''Secure'' and avoids false alarms. In terms of ''Uncertain'' cases in Fig. 10 b and Fig. 10 e, there still exists conflict in the predictions of both tasks, where the model of Task-2 detects the set of critical or significant generators. It is expected to be concerned more about these generators while utilizing TDS to further reduce harmful MA phenomena. When both tasks predict the systems and the generators to be unstable as Fig. 10 c and Fig. 10 f, ADM indicates the state of emergency and prompt critical control can be implemented based on the set of critical generators. Generally due to the visualization, dispatchers might efficiently recognize the set of generators to be controlled based on the color intensity and clusters of

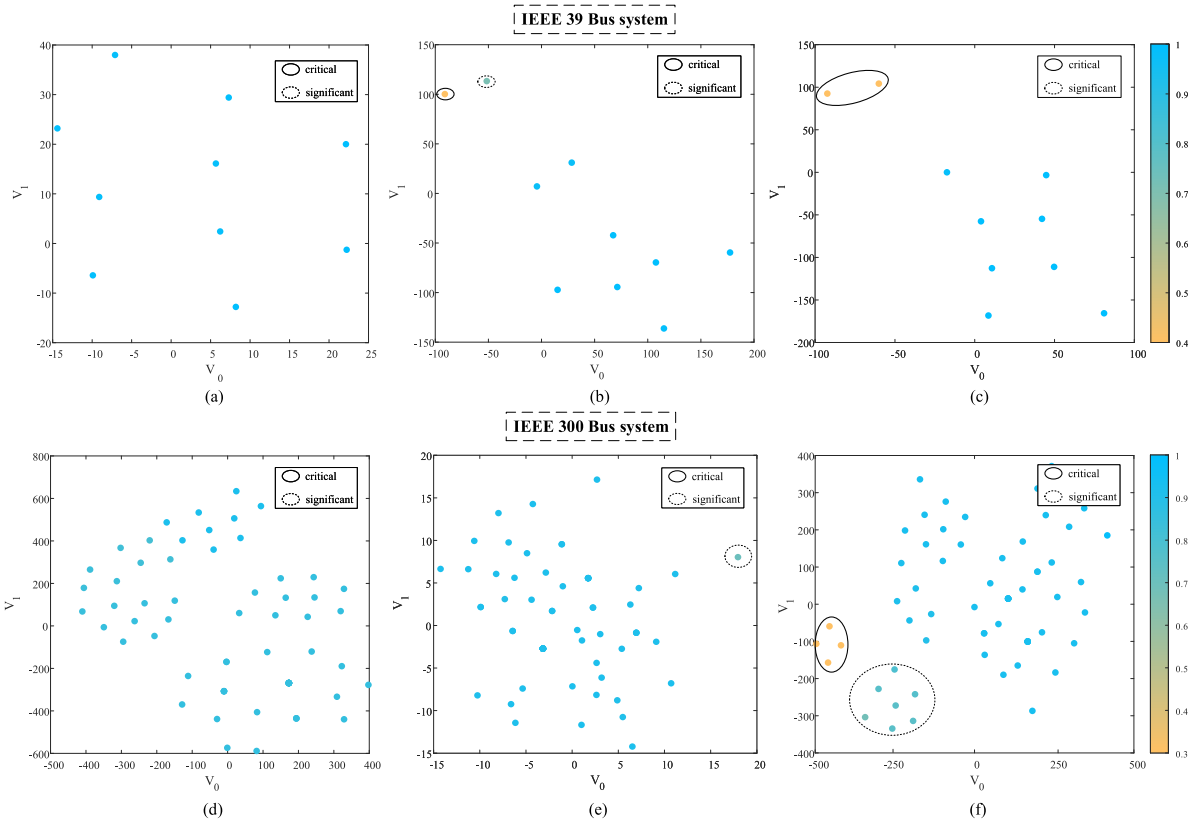

FIGURE 10. Typical visualization results of RGCN-MT-TSA framework. (a)(d) Secure. (b)(e) Uncertain. (c)(f) Critical.

circles. It is convenient to sort the importance of generators and then develop more precise control strategies.

V. CONCLUSION

In this paper, a multi-task transient stability assessment framework is proposed to address early-warning of stability classification and critical generator identification according to PMU data. We design a cascade neural network architecture named RGCN to capture the transient characteristics graphically and temporally, where a state-of-the-art network, GCN, is creatively used to explicitly extract physical topological information of the power system. The offline models of different tasks are trained in a parallel way, with a new cost-sensitive cross-entropy function to handle the imbalanced problem. A similarity regularization item is designed such that the model of Task-1 can be transferred to that of Task-2 and the training difficulty is alleviated. To evaluate the effectiveness and robustness of the proposed method, a series of case studies as well as comparisons with six different single or aggregating models are comprehensively conducted on two benchmark systems of different scales. Test results indicate the desirable performance and reliability of the proposed method. Furthermore, our framework provides comprehensive signals and 2D visualization of the generators, which helps to improve the false alarm rate as well as implement more accurate and timely control.

In future work, we will pay attention to periodic model update in our framework when facing more complex changes in the topology. This adaptive framework is expected to expand on a large practical system with thousands of buses.

REFERENCES

[1] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. A. Canizares, N. D. Hatziargyriou, D. J. Hill, A. M. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal, "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] S. Obuz, M. Ayar, R. D. Trevizan, C. Ruben, and A. S. Bretas, "Renewable and energy storage resources for enhancing transient stability margins: A PDE-based nonlinear control strategy," *Int. J. Elect. Power Energy Syst.*, vol. 116, Mar. 2020, Art. no. 105510.

[3] P. Kundur, N. J. Balu, and M. G. Lauby, *Power System Stability and Control*, vol. 7. New York, NY, USA: McGraw-Hill, 1994.

[4] R. Diao, S. Jin, F. Howell, Z. Huang, L. Wang, D. Wu, and Y. Chen, "On parallelizing single dynamic simulation using HPC techniques and APIs of commercial software," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2225–2233, May 2017.

[5] L. M. Skvortsov, "A fifth order implicit method for the numerical solution of differential-algebraic equations," *Comput. Math. Math. Phys.*, vol. 55, no. 6, pp. 962–968, Jun. 2015.

[6] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 2, pp. 573–584, Mar. 1979.

[7] Y. Xu, Z. Y. Dong, R. Zhang, Y. Xue, and D. J. Hill, "A decomposition-based practical approach to transient stability-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1455–1464, May 2015.

[8] M. He, J. Zhang, and V. Vittal, "Robust online dynamic security assessment using adaptive ensemble decision-tree learning," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4089–4098, Nov. 2013.

[9] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1474–1483, Aug. 2011.

[10] S. A. Siddiqui, K. Verma, K. R. Niazi, and M. Fozdar, "Real-time monitoring of post-fault scenario for determining generator coherency and transient stability through ANN," *IEEE Trans. Ind. Appl.*, vol. 54, no. 1, pp. 685–692, Jan. 2018.

[11] I. B. Sulistiawati, A. Priyadi, O. A. Qudsi, A. Soeprijanto, and N. Yorino, "Critical clearing time prediction within various loads for transient stability assessment by means of the extreme learning machine method," *Int. J. Elect. Power Energy Syst.*, vol. 77, pp. 345–352, May 2016.

[12] Q. Zhu, J. Chen, L. Zhu, D. Shi, X. Bai, X. Duan, and Y. Liu, "A deep end-to-end model for transient stability assessment with PMU data," *IEEE Access*, vol. 6, pp. 65474–65487, 2018.

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

[14] J. J. Q. Yu, D. J. Hill, and A. Y. S. Lam, "Delay aware transient stability assessment with synchrophasor recovery and prediction framework," *Neurocomputing*, vol. 322, pp. 187–194, Dec. 2018.

[15] 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, pp. 379–394, May 2019.

[16] 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.

[17] A. Gupta, G. Gurrala, and P. S. Sastry, "An online power system stability monitoring system using convolutional neural networks," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 864–872, Mar. 2019.

[18] L. Zhu, D. J. Hill, and C. Lu, "Hierarchical deep learning machine for power system online transient stability prediction," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2399–2411, May 2020.

[19] Z. Shi, W. Yao, L. Zeng, J. Wen, J. Fang, X. Ai, and J. Wen, "Convolutional neural network-based power system transient stability assessment and instability mode prediction," *Appl. Energy*, vol. 263, Apr. 2020, Art. no. 114586.

[20] 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, Jul. 2019.

[21] T. Ishizaki, A. Chakrabortty, and J.-I. Imura, "Graph-theoretic analysis of power systems," *Proc. IEEE*, vol. 106, no. 5, pp. 931–952, May 2018.

[22] F. Ebrahimzadeh, M. Adeen, and F. Milano, "On the impact of topology on power system transient and frequency stability," in *Proc. IEEE Int. Conf. Environ. Electr. Eng., IEEE Ind. Commercial Power Syst. Eur. (EEEIC/I&CPS Europe)*, Jun. 2019, pp. 1–5.

[23] Y. Song, D. J. Hill, and T. Liu, "Characterization of cutsets in networks with application to transient stability analysis of power systems," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1261–1274, Sep. 2018.

[24] T. Weckesser, H. Jóhannsson, M. Glavic, and J. Østergaard, "An improved on-line contingency screening for power system transient stability assessment," *Electr. Power Compon. Syst.*, vol. 45, no. 8, pp. 852–863, May 2017.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[26] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang, "A vectorized relational graph convolutional network for multi-relational network alignment," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4135–4141.

[27] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 7370–7377.

[28] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[29] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, "Fault location in power distribution systems via deep graph convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 119–131, Jan. 2020.

[30] C. Kim, K. Kim, P. Balaprakash, and M. Anitescu, "Graph convolutional neural networks for optimal load shedding under line contingency," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2019, pp. 1–5.

[31] J. J. Q. Yu, D. J. Hill, V. O. K. Li, and Y. Hou, "Synchrophasor recovery and prediction: A graph-based deep learning approach," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7348–7359, Oct. 2019.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[33] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1263–1272.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[36] T. Kim, I. Song, and Y. Bengio, "Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition," in *Proc. Interspeech*, Aug. 2017, pp. 2411–2415.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *Proc. Int. Conf. Mach. Learn.*, 1993, pp. 41–48.

[39] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 845–850.

[40] D. P. Kingma and J. Lei Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[41] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[42] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[43] C. K. Papadopoulos and C. L. Nikias, "Parameter estimation of exponentially damped sinusoids using higher order statistics," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 8, pp. 1424–1436, Aug. 1990.

JIYU HUANG received the B.S. degree in electrical engineering from the South China University of Technology (SCUT), Guangzhou, China, in 2019, where he is currently pursuing the Ph.D. degree.

His research interests include deep learning in power system security and transient stability assessment.

LIN GUAN (Member, IEEE) received the B.S. and Ph.D. degrees in electric power engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1990 and 1995, respectively.

She is currently a Professor with the Electric Power College, South China University of Technology, Guangzhou, China. From 2014 to 2015, she was a Visiting Scholar with Stanford University. She is the author of more than 120 articles and a Principal Investigator of more than 50 projects. Her research interests include application of artificial intelligence technology in electrical engineering, power system security and control, and power system planning and reliability.

YINSHENG SU received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiaotong University, Shanghai, China, in 1999 and 2002, respectively. He is currently a Senior Specialist with China Southern Grid Co. Ltd. His research interest includes electric power system operation and control.

HAICHENG YAO received the B.S. degree in water conservancy and hydropower engineering from the Huazhong University of Science, Wuhan, China, in 2004, and the M.S. degree from the State Grid Electric Research Institute, Shanghai, China, in 2007. He is currently a Senior Engineer with China Southern Grid Co. Ltd. His research interest includes electric power system operation and control.

MENGXUAN GUO was born in Hunan, China, in 1997. She received the B.S. degree in electrical engineering from the South China University of Technology (SCUT), Guangzhou, China, in 2019. Her research interests include deep learning in power system security and small-signal stability analysis.

ZHI ZHONG received the B.S. degree in electrical engineering from Southeast University, Nanjing, China, in 2019. He is currently pursuing the M.S. degree in electrical engineering with the South China University of Technology (SCUT), Guangzhou, China. His research interests include machine learning, transient stability assessment of power system, and the applications of big data in smart grids.