Robust Power System Stability Assessment Against Adversarial Machine Learning-Based Cyberattacks via Online Purification

Tianqiao Zhao, *Member, IEEE*, Meng Yue, *Member, IEEE*, and Jianhui Wang, *Fellow, IEEE*

***Abstract*—The increasing complexity associated with renewable generation brings more challenges to power system stability assessment (SA). Data-driven approaches based on machine learning (ML) techniques for stability assessment have received significant research interest and shown their promising performance. However, ML-based models are recognized to be vulnerable to adversarial disturbances, where a slight perturbation to power system measurements could lead to unacceptable errors. To address this issue, this paper develops a novel lightweight mitigation strategy, i.e., robust online stability assessment (ROSA), to enhance the ML-based assessment model against both white-box and the black-box adversarial disturbances (i.e., purification) in the online implementation. The ROSA involves a supervised learning-based module for the primary stability assessment and a self-supervised learning-based module. The two modules are trained jointly with different objective (loss) functions and implemented in sequence. A suitable purification objective and various time-series data augmentation methods are designed for SA applications to tackle adversarial disturbances adaptively. Case studies are performed, and the comparative results have clearly illustrated the competitive, robust accuracy against various adversarial scenarios and verified the effectiveness of the proposed online purification strategy.**

***Index Terms*—Machine learning, adversarial disturbances, adversarial purification, stability assessment, self-supervised learning.**

I. Introduction

THE development of machine learning (ML) techniques facilitates designing the emerging data-driven approaches to assess power system stability under various situations in real-time [1]. The ML-based approaches have been identified as a promising solution to accelerating the stability assessment (SA) and strengthening the generalization capability of SA models. The fundamental concept of the ML-based SA is to train an ML model offline using proper training data and deploy the trained model online using real-time data measured by specific devices (e.g., PMUs). For instance, transient stability assessment (TSA) can be modeled as a classification problem, where an ML-based TSA utilizes rotor angles as input to predict the system stability [2].

In line with the increasing concerns with the power system stability, a variety of ML-based models have presented detailed problem formulations and notable solutions with satisfactory performance [3], such as the assessment methods based on support vector machine (SVM) [4], [5], decision tree (DT) [6], random forest [7], etc. These ML-based methods have proven to be capable of learning and understanding the characteristics of loads, renewable generation, network data, etc., that are peculiar to the dynamic nature of modern power systems. Recently, deep learning (DL) techniques have been widely used in enhancing the performance of SA models in terms of prediction accuracy, systematicness, and reliability. For instance, convolutional neural networks (CNNs) have been utilized to deal with both stability prediction [8] and short-term voltage stability (STVS) [9] with the enhanced prediction accuracy. In [10], a DL-based model is developed based on deep belief neural networks (DBNNs) with improved practicability. In addition, authors in [11], [12] utilize recurrent neural networks (RNNs) to better capture useful spatial-temporal characteristics in measurements which, in turn, improves the overall SA performance. To handle missing PMU data, authors in [13] develop a dynamic SA model taking advantage of generative adversarial networks (GANs).

In general, the remarkable performance of the existing ML-based SA methods is only preserved when the model inputs are clean and complete. However, a plethora of studies have identified that deep neural networks (DNNs) are vulnerable to cyber-threats [14], implementation errors [15], and mismatches between the training and the deployment environments [16]. This vulnerability raises significant concerns when deploying the ML-based methods to power system applications that would encounter potential adversarial disturbances [17], [18], [19]. The purpose of adversarial strategies is to find the indistinguishable samples that are close to the original data but easily misclassified by the ML-based models [20]. The stakes are high: an incorrect SA gives grid operators erroneous information that

Manuscript received 11 May 2022; revised 28 September 2022 and 8 December 2022; accepted 13 December 2022. Date of publication 2 January 2023; date of current version 20 October 2023. This work was supported by Laboratory Directed Research and Development Project under Grant LDRD 21-032 "Development of a Planning, Operation, and Control Framework for Hybrid Energy Storage and Renewable" at Brookhaven National Laboratory. Paper no. TPWRS-00687-2022. *(Corresponding author: Tianqiao Zhao.)*

Tianqiao Zhao and Meng Yue are with the Brookhaven National Laboratory, Upton, NY 11973-5000 USA (e-mail: tzhao1@bnl.gov; yuemeng@bnl.gov).

Jianhui Wang is with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: jianhui@smu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2022.3233735.

Digital Object Identifier 10.1109/TPWRS.2022.3233735

can lead to incorrect decisions. For example, a false negative in detecting instability can prevent appropriate preemptive actions that otherwise would be taken by the operators, which can potentially result in a unnecessary losses or even blackout threatening properties and lives; on the other hand, a false positive can cause unnecessary load shedding and inconvenience for the affected customers and increases the cost. Authors in [21] have investigated the use of learning-based methods for detecting data anomaly under unknown probability distributions. Rather than detecting adversarial disturbances only, this work aims to purify adversarial samples in testing implementation and further provide an accurate response to stability prediction even under such disturbances. Authors in [22] developed a deep learning-based method for Bayesian state estimation that deals with unobservable distribution systems and bad-data injections. However, it still needs to carefully label PMU data (e.g., samples with/without bad data) that is generally challenging in real applications. Therefore, it is desired to design a method that is independent of data labels without performance reduction.

Recent works have identified a number of strategies to empower ML-based models to defend against these adversaries [14], [23], [24], [25], where adversarial training [14] has been widely adopted to design mitigation strategies. In this regard, authors in [26] analyze the vulnerabilities of the ML-based models and develop a mitigation framework to improve model robustness by presenting the model with clean data and adversarial examples based on adversarial training. Although the mitigation strategy developed in [26] enhances the robustness of the ML-based SA model, this adversarial training-based strategy is computationally costly since the adversarial examples are discovered by computing a sample-wise gradient at every epoch.

To bridge the existing gaps, this paper herein proposes an online defense strategy, Robust Online Stability Assessment (ROSA), which makes the ML-based SA models robust against both white-box and black-box adversarial scenarios in real-time implementation. Self-supervised learning, featuring a better informative and influential data representation, has been used to understand and improve model robustness and save the computation cost via fine-tuning [27]. The proposed ROSA therefore investigates the combination of supervised SA problems and self-supervised learning (e.g., contrastive learning [28]) to enhance the robustness of neural networks (NNs). It contains two independent NN modules, an SA module and a consistent contrastive learning (CCL) module, to predict system stability and purify adversarial samples online, respectively. The two NN modules are trained jointly with different objective (loss) functions. We carefully design a CCL objective for supporting the primary SA assignment such that the proposed strategy is compatible with power system specifications. The trained ROSA is implemented together with an algorithm for online purification under different adversarial scenarios. The main contributions in this paper are summarized as follows:

- A novel ROSA framework is proposed by exploiting the advantages of label-independent self-supervised learning to purify adversarial samples in real-time implementation, thus promising an effective solution to enhancing the performance of SA models.
- An auxiliary objective is designed for the CCL module based on emerging contrastive learning [28], which generates a suitable signal to assist the original SA decisions adaptively. Meanwhile, specific data augmentation methods are introduced to process the time-series PMU measurements, supporting the CCL objective.
- To systematically counteract the adversarial disturbances and relieve computational burdens, an iterative purification algorithm is developed and applied to both clean and adversarial samples, which minimizes the success probability of an adversarial disturbance online and, in turn, enhances the model robustness against the adversarial samples.

The remainder of the paper is organized as follows. Section II introduces the problem formulation. Section III details the proposed ROSA framework, modules, and implementation. Case studies are presented in Section IV to verify and validate the effectiveness of the ROSA framework. Section V concludes the paper.

II. Problem Formulation

A. Machine Learning-Based Assessment Model

SA plays a critical role in power system control and operation, including different stability criteria such as transient stability, frequency stability, or (long-term and short-term) voltage stability. Without loss of generality, this work takes transient stability based on rotor angles and short-term voltage stability (STVS) as use cases, while other stability problems can be assessed following an approach similar to that described in Section III. In general, SA problems considered in this work intend to predict the system stability in a few seconds after the fault clearance [9]. For example, transient stability is the ability of generators to maintain synchronism after the fault clearance. The status of transient stability is commonly classified by using a stability index [29],

$$\eta = \frac{360^\circ - \Delta\bar{\theta}}{360^\circ + \Delta\bar{\theta}} \tag{1}$$

where $\Delta\bar{\theta}$ denotes the absolute value of the maximum post-fault difference between generator rotor angels. One can conclude that the system is stable if $\eta > 0$ and unstable if $\eta \leq 0$. Note that the use of a different transient stability index will not affect the performance of the proposed ROSA, and different indexes can be adopted based on specific applications.

An STVS is defined as the ability of a power system to maintain steady acceptable bus voltages under normal conditions and after disturbances. In particular, an acceptable level of bus voltage amplitudes should be maintained before interference under stable conditions. Otherwise, the system is unstable if the acceptable level is not satisfied at any bus(es), or the voltage may collapse in the transition window (i.e., 10 s). Please note that different voltage stability indices (e.g., transient voltage severity index and voltage sag severity index [9]) can also be used to evaluate the voltage performance.

Note that ML-based methodologies have widely addressed the above-discussed assessment problems. Such ML-based assessment models can predict the stability status using (complete or incomplete) PMU measurements in real-time. Given a SA problem, an ML-based model learns a parametric function f_σ to map the system states to the stability status with a training dataset (x, y), where x is the sampled state set and y is the label set. The mapped relationship is represented by the parameter σ using ML-based techniques (e.g., multi-layer NNs). In predicting stability, x can be PMU measurements (e.g., bus voltage data) and y is the associated stability status (i.e., stable and unstable).

SA can be reformulated as a classification problem (e.g., 0 → unstable and 1 → stable). The ML-based model is trained to minimize the difference between the predicted value and the ground-truth status by minimizing a predefined loss function L_f. The parameters σ of the ML-based model are usually updated by a back-propagation procedure via gradient descent algorithms such as

$$\sigma_{k+1} = \sigma_k - \eta \nabla_\sigma L_f \tag{2}$$

with the learning rate η at the k learning iteration until converging. Once the ML-based model is well-trained, it can predict the stability status accurately but lack relevant robustness guarantees.

B. Stability Assessment Under Adversaries

Although the ML-based approaches have been widely developed for power system stability assessment, they assume that the dataset (e.g., PMU measurements) is clean and there is no adversarial disturbance when training/implementing these approaches. As aforementioned, the robust accuracy of an ML assessment model can be diminished when the adversarial disturbances contaminate clean PMU measurements and shift the correct representations towards false outputs. As found in [14], even a slight disturbance to input can be amplified over NNs and lead to a false classification output (e.g., wrong stability assessment).

Both white-box and black-box attack algorithms are available to cyber adversaries. The former assumes attackers have complete knowledge while the latter further assumes none or limited knowledge of the original ML model.

1) White-Box Attack Algorithm: the first-order algorithms that utilize $\nabla_x L_f$, the gradient of L_f w.r.t. x as information are the most common methods for generating adversarial attacks. There are different first-order algorithms, e.g., fast gradient sign method (FGSM) [30], projected gradient descent (PGD) [14] and Carlini & Wagner (C&W) attack [23].

FGSM [30], f_{FGSM}, is the simplest but an efficient first-order algorithm that generates attacks for maximizing the value of the loss function,

$$x' = x + \epsilon^s \text{sign}\left(\nabla_x L_f\right) \tag{3}$$

where ϵ^s is the attack strength and sign is the sign function for a given tensor.

PGD [14], f_{PGD}, as an extension of FGSM produces

$$x_{i+1} = \text{Bound}_{l_p}(f_{\text{FGSM}}(x_i), x_0) \tag{4}$$

where x_0 is the original input data and $0 < i < k - 1$ given the overall k iterations. In (4), it ensures the adversarial sample x' within a l_p range of x_0.

In [23], it introduce a strong first-order algorithm that solves an optimization problem to discover a perturbation δ given input x,

$$\min \|\delta\|_p + \beta f'(x + \delta), \quad s.t. \;\; x + \delta \in [0, 1]^n \tag{5}$$

where p is the norm distance and a constant β is used to balance two terms in (5). f' is an objective function driving the perturbed sample to false classes (ideally $f' \leq 0$) [23].

2) Black-Box Attack Algorithm: this type of attacks assumes no or limited knowledge of the ML model and only has access to the model by queries or sending inputs and receiving associated outputs to determine the inner model knowledge. To falsify/confuse an ML model, the well-known algorithms include surrogate network-based algorithms [31] and gradient estimation-based algorithms [32]. The former trains a new ML model that mimics the behaviors of the target model [31] while the latter estimates the gradients of the target directly instead of substituting networks (Zeroth order optimization or ZOO [32]).

Once applied to the input data to the trained SA model, these attack algorithms can lead to misclassification. Currently, one common mitigation approach is adversarial training that improves NN robustness by augmenting adversarial samples to the original data set [14]. However, its implementation is limited by the computation expense for discovering adversarial samples. In addition, the adversarial training sacrifices the performance when the input is clean. Another natural method to address adversarial attacks is to clean the testing samples by shifting the adversarial representations back to the correct ones, namely purification, which reveals a promising way to avoid solving complex inner optimization problems [24].

In this context, this paper investigates an online purification approach to enhance robustness of SA models. Let $f^{\text{enc}}_{\sigma_{\text{enc}}}(x)$ be an encoder that maps system states (e.g., PMU measurements) to an embedding representation h_x, $f^{\text{sa}}_{\sigma_{\text{sa}}}(h_x)$ be an SA model that outputs the system stability prediction given h_x and f^{com} be a composition of f^{enc}, and f^{sa}, i.e., $f^{\text{enc}} \circ f^{\text{sa}}$, that defines the entire SA model. The purification problem is described as follows: for a clean sample (x) (unknown) and its adversarial counterpart (x'), we need to develop a purification solution, i.e., Π, that attempts to discover

$$\tilde{x} = \Pi(x') \tag{6}$$

where $\tilde{x}$ is a purified sample satisfying $\tilde{x} \rightarrow x$. Note that an adversarial sample (x') can be shared among different clean samples that makes the purification problem underdetermined. This problem can be relaxed following [33] and rewritten as

$$\min_{sa} L_{\text{sa}}\left(f^{\text{com}}(\tilde{x}, y)\right) \quad s.t. \; \|\tilde{x} - x'\| \leq \epsilon', \;\; \tilde{x} = \Pi\left(x'\right) \tag{7}$$

where y is the label and $\mathcal{L}_{\text{sa}}$ is the cross entropy loss for SA. ϵ' is the adversarial budget reflecting maximum changes in measures [24]. However, this problem cannot be solved without the knowledge of y and ϵ' in testing. This work will investigate an alternative approach to develop a solution for robust online SA.

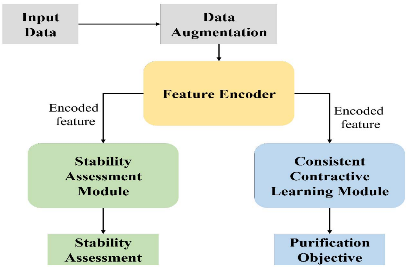


Fig. 1. The scheme of ROSA training.

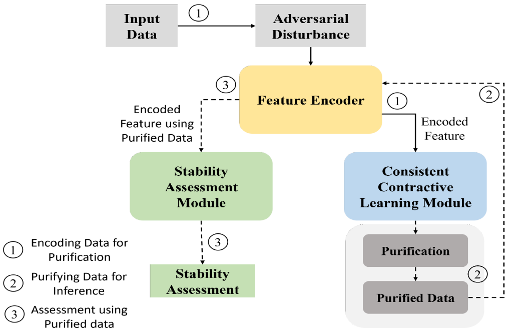

Input Data
1
Adversarial Disturbance
3
Feature Encoder
2
1
Encoded Feature using Purified Data
Encoded Feature
Stability Assessment Module
Consistent Contractive Learning Module
3
Purification
2
Purified Data
Stability Assessment
1 Encoding Data for Purification
2 Purifying Data for Inference
3 Assessment using Purified data

Fig. 2. The scheme of ROSA testing.

III. Self-Supervised Robust Online Stability Assessment

As discussed above, an accurate and robust model for SA is the key to guaranteeing real-time system operation. To this end, this paper first introduces an alternative formulation of (7) based on self-supervised learning. An auxiliary network is then designed to generate self-supervised signals for online purification. Lastly, a purification strategy is developed to shift the adversarial samples to the clean ones as close as possible.

A. Pipeline

Solving the problem defined in Section II requires the explicit knowledge of y and ϵ' and its acquisition is intractable. To overcome this, we leverage the powerful capability of self-supervised learning for representing unlabeled data [24] and propose an online purification approach, namely, ROSA, to iteratively and adaptively mitigate the impacts of adversarial attacks on the SA results. ROSA includes an SA module and a CCL module and its conceptual diagram for training and testing is illustrated in Figs. 1 and 2. The SA module utilizes an ML-based approach and outputs SA results (i.e., stable or unstable). In the CCL module, it follows self-supervised learning with the objective of counter-shifting the adversarial samples back to the clean representation. Loss functions are defined for these two modules that will be trained jointly to achieve multiple objectives. Every adversarial sample will be purified during testing to counter its adversarial representation instead of feeding it directly to the inference and improving the solution robustness. Note that a feature encoder is used to map system state measurements to an embedding representation.

B. SA Module

Given an SA problem as in Section II.A, the existing ML-based models can be adopted to fulfill the corresponding classification problem. Taking TSA as an example, either an autoencoder-based model or a convolutional neural network (CNN)-based model can be embedded in the SA module. The SA module uses the PMU measurements as input and predicts the system stability as output. The SA module is therefore a placeholder that can accommodate different NN models. Interested readers may refer to [34] for more details about the ML-based TSA and STVS models. In Section IV, we incorporate ROSA with several widely adopted ML-based SA models to test its performance. To simplify the notation, we denote $f^{\rm ta}_{\sigma_{\rm sa}}$ as the overall SA model.

C. CCL Module

The major challenge is to purify the adversarial samples using real-time measurements since it is hard to label raw PMU data. Thus, we propose to take advantage of the label-independent capability of the self-supervised learning technique to assist the purification process. Self-supervised learning frames a supervised learning task in a form that extracts only a subset of representations using the rest. Rather than focusing on the final SA performance, we are interested in learning some intermediate representations that can carry meaningful semantic or structural information and benefit the SA task. From this perspective, we develop a CCL-based method that purposes to extract the consistency between different data augmentations of the original sample x and provide pre-knowledge from such representations as the basis to perform purification.

1) Alternative Formulation: This concept of self-supervised learning motivates us that the adversarial samples not only affect the performance of SA models, but also indirectly hinder the self-supervised tasks. Specifically, adversarial samples will mislead self-supervised tasks, eventually hindering the stability classification. Building upon this, these adversarial examples can be purified by using a self-supervised learning-based approach as the performance of either self-supervised tasks or the SA tasks is highly determined by adversarial samples.

Given this observation, we first reformulate the original problem (7) to an alternative form based on the self-supervised learning. Given h_x, the hidden representation of x from the encoder $f^{\rm enc}_{\sigma_{\rm enc}}(x)$, we denote the model of the CCL module and its target as $f^{\rm ccl}_{\sigma_{\rm ccl}}$ and $L_{\rm ccl}$, respectively. The CCL objective is given by

$$\min_{sa} L_{\rm ccl}\left(f^{\rm ccl}_{\sigma_{\rm ccl}} \circ f^{\rm enc}_{\sigma_{\rm enc}}\right)(\tilde{x})) \quad \text{s.t. } \|\tilde{x}-x'\| \le \tilde{\epsilon},\ \ \tilde{x}=\Pi\left(x'\right). \tag{8}$$

where $\tilde{\epsilon}$ is the budget of purification [33]. In (8), the SA objective function $L_{\rm sa}$ is replaced by $L_{\rm ccl}$, and therefore, one can remove the need for labels y. Such label information can be derived from the data itself by leveraging label-independent self-supervised learning. The intuition underlying the CCL module is that the alternative formulation is expected to learn useful feature information from the data, which can thereby assist the purification process.

2) Purification Objective: Note that the output of the ML-based model using different data augmentations should be consistent with the one using the original sample x [28], namely label consistency. This finding motivates us to design a purification signal independent of data augmentations to assure consistent output. To this end, this work adopts the CCL to distinguish different samples statistically by learning a space where the similar samples stay close to each other while those dissimilar ones are far apart.

Let x^a and x^b be two distorted samples of x that are generated via a distribution of data augmentations. We borrow the Barlow Twins [28] from the concept of contrastive learning to make the CCL module invariant to sample distortions. It feeds two identical NNs (i.e., shared network parameters) with two distorted samples to calculate the cross-correlation matrix between different NNs, and attempts to make this matrix close to the identity. The objective function for purification is defined by

$$L_{\text{ccl}} = \sum_i (1 - M_{ii})^2 + \nu \sum_i \sum_{j \neq i} M_{ij}^2 \tag{9}$$

where ν is a positive coefficient balancing the two loss terms and M is the cross-correlation matrix between two identical NNs with i, j being the vector dimension of their output (i.e., z^a and z^b). It is calculated by

$$M_{ij} = \frac{\sum_c z_{c,i}^a z_{c,i}^b}{\sqrt{\sum_c (z_{c,i}^a)^2}\sqrt{\sum_c (z_{c,i}^b)^2}} \tag{10}$$

where c is the index of batch samples. Thus, M is a square matrix that comprises values in $[-1, 1]$ indicating the relationship from the perfect anti-correlation $[\rightarrow -1]$ to the perfect-correlation $[\rightarrow 1]$. Intuitively, increasing the diagonal elements $(\rightarrow 1)$ of M makes the embedding z invariant to the distorted samples while decreasing its off-diagonal elements $(\rightarrow 0)$ decorrelates the different components and reduces the output redundancy [28].

D. Data Augmentation

Effective data augmentation is needed for creating distorted samples of the original ones that are critical for learning generalizable embedding features for the purification objective. Such augmentation techniques have been widely adopted in applications of image and video data [35]. However, they are not designed for time-series PMU measurements (e.g., cropping and flipping) and must be modified properly. In what follows, several augmentation methods are presented in detail. To simplify the notation, we denote $x_{i,t}$ and $\hat{x}_{i,t}$ as the element and its counterpart at the tth step in the i^{th} sequence, respectively.

1) Random Smoothing: In order to smooth the time-series measurements, a Finite Impulse Response (FIR) filter is used to achieve varying representation while keeping the overall trend and shape unchanged.

2) Random Noise: The noise is applied individually to each time-series PMU sequence based on its standard deviation σ_i. Let r_i be a value between -1 and 1 and λ_i be an additional parameter for tuning the noise level, we have

$$\hat{x}_{i,t} = x_{i,t} + r_i \lambda_i \sigma_i. \tag{11}$$

3) Magnitude Warping: This method modifies the measurement by slightly increasing or decreasing its amplitude in different fragments. We apply a sine wave to scale the measurement. Each sine wave with the amplitude λ_M is shifted with a random parameter κ_i.

$$\hat{x}_{i,t} = x_{i,t} + \lambda_M \sin \frac{2\pi t}{T} + \kappa_i \tag{12}$$

where T is the time horizon.

4) Time Warping: For time warping, we compress and stretch the original data. In particular, given two selected fragments, compressing discards the measurements in one fragment while stretching utilizes a linear interpolation for another one. We randomly select two fragments for individual sequences and sample a binary variable deciding on compressing (0) or stretching (1). The stretched elements are further scaled by a parameter used to tune the augmentation strength, i.e., between -30% and 30%. Equal-length random samples of both stretching and compressing data sequences are used to maintain the exact shape of different sequences.

E. Training Procedure

Given the augmented data, training the ROSA requires a joint optimization of the parameters of ML-based models in both SA and CCL modules. An encoder $f_{\sigma_{\text{enc}}}^{\text{enc}}$ first maps sequences of PMU measurements to a low-dimension space. Its embeddings are fed to the models in the SA and CCL modules (i.e., $f_{\text{sa}\sigma_{\text{sa}}}$ for the SA objective and $f_{\sigma_{\text{ccl}}}^{\text{ccl}}$). Lastly, the SA model outputs the stability prediction, while the purification model outputs the similarity between distorted samples. With these outputs, we jointly minimize the classification loss for SA and the CCL loss for the purification objective, i.e.,

$$L_{\text{sum}} = \alpha_1 L_{sa} + \alpha_2 L_{\text{ccl}} \tag{13}$$

where α_1 and α_2 are the weight parameters for the two objectives. L_{sa} is the cross entropy loss and L_{ccl} is defined in (9). Algorithm 1 summarizes the details of training procedure.

During training, the feature of a training sample is first extracted and embedded using an encoder. These embedded features will be fed to SA Module and CCL Module at the same time. The objective of SA Module is to provide stability prediction while CCL Module is to purify the data based on a self-supervised learning objective. Each module will produce the output individually. Their outputs will be collected and used to calculate the cross-entropy loss for stability assessment and the loss of the purification objective defined in (9)–(10). After the two losses are obtained, they will be added together and weighted by α_1 and α_2 in (13), which will be used to calculate the gradient to optimize the neural networks in both the SA Module and the CCL Module via back-propagation.

Algorithm 1: Training Procedure.

```
Input: PMU measurements and sequence length: $x$, $T$
Output: Stability predication
Initialize all the NN weights randomly;
repeat
    Sample a batch of $x$ from the PMU measurements;
    Obtain augmented data $\hat{x}$ randomly using the
      augmentation methods;
    Feed $\hat{x}$ into the encoder $f^{\text{enc}}_{\sigma_{\text{enc}}}$ and output the embedding
      $z$ ;
    Feed $z$ into the SA model $f^{\text{sa}}_{\sigma_{\text{sa}}}$ and the CCL model $f^{\text{ccl}}_{\sigma_{\text{ccl}}}$
      and output the predictions ;
    Compute the loss function $L_{\text{sum}}$;
    Update parameters $\sigma_{\text{enc}}$, $\sigma_{\text{sa}}$ and $\sigma_{\text{ccl}}$
until reach the maximum training iteration;
return The trained SA and CCL models
```

Algorithm 2: Purification in Test Time.

```
Input: Trained ROSA model, normalized sample $x$, iteration
       numbers $N_t$, step size $\gamma$ and $\epsilon$ in (8)
Output: Stability predication and sample similarity
Assign $x \rightarrow x^{\text{pur}}$;
repeat
    $\Delta^{\text{pur}} = \nabla L_{\text{ccl}}(x^{\text{pur}})$   (14a)
    $x^{\text{pur}} = x^{\text{pur}} - \gamma \text{sign}(\Delta_{\text{pur}})$   (14b)
    $x^{\text{pur}} = \min(\max(x^{\text{pur}}, x - \epsilon), x + \epsilon)$   (14c)
    $x^{\text{pur}} = \min(\max(x^{\text{pur}}, 0), 1)$   (14d)
until reach the maximum training iteration $N_t$;
return Purified samples $x^{pur}$
```

F. Online Purification

The ultimate goal of adversarial disturbances is to maximize the SA loss L_{sa} and, in turn, mislead the prediction. To mitigate this, a goal is defined to reduce the adversarial impact on the SA prediction by minimizing the CCL loss L_{ccl}. This goal is achieved by purifying these adversarial samples into the samples that are as close to the clean ones, by which their impacts are mitigated. To this end, given a trained ROSA model with fixed NN parameters, an iterative gradient sign method is adopted to solve the CCL objective in (8) taking advantage of its label-independence at test time. The purification will be processed for N_t iterations, and at each iteration, given any normalized input sample x, it will be purified by the procedures (14) in Algorithm 2.

Unlike the adversarial budget ϵ', the introduction of ϵ makes the problem trackable since this parameter is obtained from data self instead of unknown adversarial disturbances. Intuitively, it is used to constrain the purified sample within the ϵ-ball of an input sample so as to avoid altering the output of the SA network.

G. Adaptive Selection of ϵ

Note that given an unknown ϵ', setting ϵ is still challenging for the online implementation. An insufficient ϵ may not be able to neutralize the adversarial disturbances. Empirically, following [25], ϵ is set to a slightly large value to make the ROSA network robust to new noise samples in training time. During the testing, ϵ is set adaptively based on the CCL loss since this loss is the key factor that affects the purification performance, as shown in (14). In particular, we empirically decide the value of ϵ adaptively as below. For the first batch of testing samples, the value of ϵ is the same as the one used for training. We monitor the CCL loss in several testing iterations (e.g., N_k) and decrease the value of ϵ adaptively if the loss value does not change as we expected, i.e.,

$$\epsilon \longleftarrow \tau \epsilon \tag{15}$$

where τ is a tunable parameter that gives us more flexibility to adjust the change of ϵ. Then the next test starts with setting its value as the current value of ϵ. The above procedure determines the lowest bound of ϵ that has the best ROSA performance for online implementation.

Remark 3.1: Since the proposed framework, especially for data augmentation methods, relies on the PMU data, the PMU associated data quality issues, e.g., time delay and data dropout, need to be considered. It should be noted that the data augmentation methods in Section III.D could be possibly taken the time delay and data dropout issues into account by treating them as a case of time wrapping and magnitude wrapping, respectively. Case studies reveal the potential of the proposed solution to address the exact time delay and data dropout issues. However, the impact of either time delay or data dropout on the performance of the proposed solution may be significant and should be carefully investigated in the future.

IV. Case Studies

This section demonstrates the proposed ROSA framework for purifying adversarial disturbances in online implementation by two SA applications using the IEEE 118 bus system [19]. The first application considers a TSA problem based on rotor angles, while the second one investigates an STVS problem. In each case, we compare two cases using the proposed ROSA framework, i.e., a benchmark case with no defense and a case with adversarial training. For a fair comparison, all the NNs for SA share exactly the same hyperparameters in each application. Case Study I takes the PMU measurements at generator buses, i.e., generator bus voltage magnitude, rotor angle, frequency deviation, active power, and reactive power, as the inputs. In Case Study II, the voltage trajectories of buses are taken as inputs.

A. Data Generation

To acquire realistic datasets for training and testing, only a limited number of buses is assumed to be equipped with PMUs and in this study, we deploy 20 PMUs with 120 frames/s sampling frequency to ensure observability [19]. A wide range of system operating conditions is simulated to generate the responses as the post-fault PMU measurements. Various loading conditions (i.e., 0.8–1.2 of the base value) are simulated together with different transient contingencies. The contingencies include three-phase faults located on either a bus or a transmission line that is randomly selected, which are cleared after 5 cycles. We generate a balanced training dataset (i.e., 50% stable and

TABLE I
AVERAGE ACCURACY OF ML-BASED MODELS FOR TSA WITHOUT DEFENSES

		LSTM	FCN	CNN
No adversarial disturbances		97.54%	94.35%	97.03%
White-box scenarios	FGSM	9.02%	8.13%	8.65%
	PGD	7.56%	6.82%	8.14%
	CW	7.25%	6.65%	6.93%
Black-box scenario	Surrogate algorithm [2]	14.81%	11.14%	12.82%

TABLE II
AVERAGE TESTING ACCURACY OF ML-BASED MODELS FOR STVS WITHOUT DEFENSES

		LSTM	FCN	CNN
No adversarial disturbances		96.89%	95.63%	96.74%
White-box scenarios	FGSM	7.45%	7.02%	7.24%
	PGD	6.25%	5.84%	6.78%
	CW	6.43%	5.47%	6.24%
Black-box scenario	Surrogate algorithm [2]	13.42%	9.88%	12.02%

TABLE III
AVERAGE ACCURACY OF ML-BASED MODELS FOR ONLINE TSA WITH DIFFERENT DEFENSE METHODS

Network	Attack	Adversarial Training Method			
		FGSM Training	PGD Training	Black-box Training	ROSA
FCN	FGSM	81.26%	53.74%	\	**87.85%**
	PGD	24.59%	62.57%	\	**78.39%**
	CW	31.18%	26.36%	\	**80.81%**
	Black-box	\	\	90.36%	**91.55%**
CNN	FGSM	89.26%	90.17%	\	**90.22%**
	PGD	35.59%	**93.86%**	\	92.44%
	CW	18.26%	86.62%	\	**89.91%**
	Black-box	\	\	92.52%	**95.55%**
LSTM	FGSM	**91.26%**	88.94%	\	91.15%
	PGD	45.32%	**94.15%**	\	92.19%
	CW	32.49%	84.21%	\	**90.05%**
	Black-box	\	\	94.25%	**97.24%**

TABLE IV
AVERAGE ACCURACY OF ML-BASED MODELS FOR ONLINE STVS WITH DIFFERENT DEFENSE METHODS

Network	Attack	Adversarial Training Method			
		FGSM Training	PGD Training	Black-box Training	ROSA
FCN	FGSM	82.34%	61.52%	\	**88.62%**
	PGD	28.75%	72.57%	\	**80.29%**
	CW	38.98%	43.29%	\	**83.22%**
	Black-box	\	\	91.78%	**93.42%**
CNN	FGSM	89.56%	90.47%	\	**91.22%**
	PGD	52.59%	**93.46%**	\	93.04%
	CW	24.96%	88.62%	\	**90.02%**
	Black-box	\	\	92.52%	**94.52%**
LSTM	FGSM	92.38%	91.33%	\	**93.05%**
	PGD	59.42%	**95.25%**	\	95.14%
	CW	42.82%	89.41%	\	92.19%
	Black-box	\	\	97.14%	**98.48%**

TABLE V
AVERAGE ACCURACY FOR ONLINE TSA WITH THE PROPOSED METHOD

Network	Attack	ROSA
LSTM	FGSM	90.04%
	PGD	91.23%
	CW	89.75%
	Black-box	93.46%

50% unstable) to reduce the unfavorable effects of an imbalanced dataset. Given each transient, the data labels are obtained using either the index (1) for the TSA problem or the index in [36] for the STVS problem. The number of samples is given in each case study. The entire dataset is divided into two subsets, i.e., 80% as the training data and 20% as the testing data.

For each test case, an additive Gaussian noise is included in the raw PMU measurement to model the measurement noise. Then, the data augmentation methods described in Section III.D are employed to generate distorted samples during training. In particular, for each training episode, we first sample a batch of training data (e.g., 128 samples) and distort each individual sample by randomly selecting one of the augmentation methods. In sequence, the distorted samples will be used to train the ROSA model. It should be noted that in training, cyberattack scenarios are not applied to the training set, which, however, will be only deployed in testing. For the sake of simplification, the combination of different augmentation methods is not adopted in this study. After the model is validated, the proposed ROSA is implemented online using real-time measurements of other operating conditions where we consider different time windows (0.3 s, 0.6 s and 0.9 s) after the fault clearance to test its performance comprehensively.

It should be noted that rather than creating adversarial samples, the data augmentation is only used to generate distorted samples in training. The adversarial disturbances/samples are only applied during real-time testing, and therefore, to the trained model, all tested adversarial samples are unknown to the model in testing.

B. Basic Settings

State-of-the-art ML-based SA models, including fully-connected network (FCN), long short-term memory (LSTM) and convolutional neural network (CNN), are adopted to demonstrate the SA performance under different adversarial attack scenarios for both cases. Both TSA and STVS problems utilize an MLP as the encoder and adopt FCN, LSTM, or CNN as the model in the SA module. Note that designing the model in the SA module is not the focus of this work, and interesting readers please refer to [34] for more detailed information. The ML-based model in the CCL model shares the same as the SA model but with a different objective.

We train the network using the Adam optimizer with the initial learning rate is 0.001 and 0.005 for TSA and STVS, respectively. The learning rate is decreased by 0.1 every 100 iterations. All numerical studies are performed using a personal desktop with a 2.5 GHz Intel Core i7 and 32 GB of RAM, where the proposed ROSA is trained and tested in Python.

C. Vulnerability Analysis

This section demonstrates the vulnerability of ML-based SA models under the attack scenarios described in Section II.B that simulate the adversarial disturbances. For white-box attacks, the model information is known to the adversaries while for black-box attacks we assume that adversaries have no access to the model and use surrogate network-based algorithms [31] to generate the adversarial samples.

Tables I–II summarize the performance of different SA models under various adversarial disturbances. The results are calculated by averaging the prediction accuracy for 100 testings. It can be seen that, although the original ML-based SA models have excellent performance, their performance (i.e., accuracy) degrade significantly for the adversarial disturbances under either the white-box or black-box scenarios. It is shown that these white-box scenarios would have higher negative impacts on the performance than the black-box scenario since they target and attack the ML model directly instead of a surrogate model. In conclusion, the performance of the original ML-based SA models deteriorates sharply for adversarial disturbances.

D. ROSA Results

We demonstrate the ability of ROSA to purify the adversarial disturbance online by two common SA problems, i.e., TSA and STVS.

1) Case I - TSA Problem: For this problem, a total number of 10,000 samples are obtained, and each sample is corrupted by an additive Gaussian noise ($\mu = 0, \sigma_G = 0.5$). The maximum training iteration is 500 and the trade-off parameters are $\alpha_1 = 0.9$ and $\alpha_2 = 1.05$. The purification step is $T = 10$ with the step size $\gamma = 0.1$. Given these settings, we first need to find ϵ that has the best performance for training. It will be used to initialize the value of $\epsilon_{\max}$ in testing and adaptively find the best $\epsilon_{\min}$. Based on the procedure described in Section III.G, we set $\epsilon = 0.43$ for online implementation and Fig. 3 shows its performance for different NNs under various adversarial scenarios.

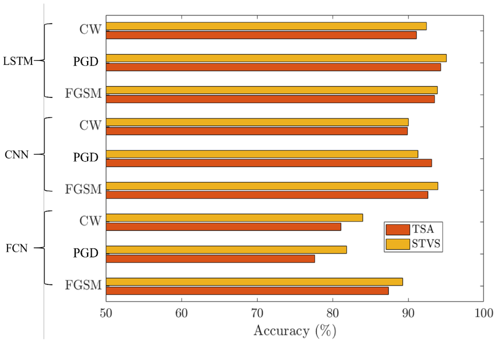
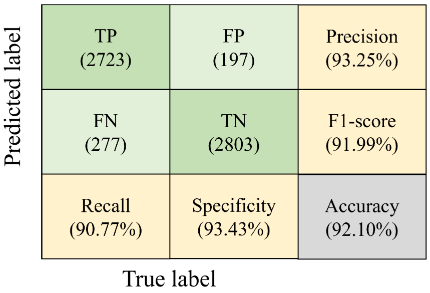


Fig. 3. Testing Accuracy with $\epsilon = 0.43$.

We compare the proposed ROSA with the SA model using adversarial training under different disturbance scenarios. Table III gives the qualitative accuracy of different methods, where FGSM, PGD and Black-box training methods mean adversarial training based on the samples generated by the associated adversarial disturbances. For a fair comparison, black-box disturbances are only tested under black-box-based adversaries training. The adversarial training method is only effective for the trained model using the corresponding scenarios, while the proposed ROSA is universally applicable and effective. The results verify the ability of ROSA to make the ML models much robust against various adversarial disturbances in TSA.



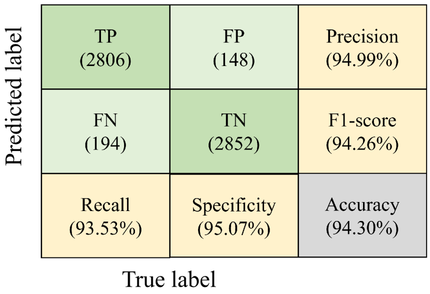
Predicted label	TP (2723)	FP (197)	Precision (93.25%)
	FN (277)	TN (2803)	F1-score (91.99%)
	Recall (90.77%)	Specificity (93.43%)	Accuracy (92.10%)
	True label		

Fig. 4. Confusion matrix of PGD adversarial training.

2) Case II - STVS Problem: In this case, we apply the ROSA framework to the STVS problem. A total number of 10,000 samples are obtained, and each sample is corrupted by an additive Gaussian noise ($\mu = 0, \sigma_G = 0.1$). The maximum training iteration is 600 and the trade-off parameters are $\alpha_1 = 0.95$ and $\alpha_2 = 1.02$. Again, the purification step is $T = 10$ with the step size $\gamma = 0.1$. Following a similar procedure to Case I, the best performance is given with $\epsilon = 0.52$. The obtained ROSA model is then compared with adversarial training with various adversarial scenarios. The comparative results in Table IV summarize the averaged prediction accuracy in testing.

The findings are aligned with the results in Table III, and one can conclude that ROSA has the best performance for most situations. Although PGD-based adversarial training slightly outperforms PGD disturbances, the proposed ROSA can deal with various adversarial scenarios while holding outstanding performance.

3) Performance Analysis: To further demonstrate the performance, we evaluate the ROSA and adversarial training according to performance indices in [37]. In particular, a confusion matrix is first built based on true positive (TP), true negative (TN), false positive (FP) and false negative (FN) that are widely used for classification problems, where for SA problems, the FP value is a key factor indicating the number of misclassifications (i.e., unstable → stable). In sequence, four performance indices are calculated following [37]. The results are given in Figs. 4 and


Predicted label
True label

TP (2806)	FP (148)	Precision (94.99%)
FN (194)	TN (2852)	F1-score (94.26%)
Recall (93.53%)	Specificity (95.07%)	Accuracy (94.30%)

Fig. 5. Confusion matrix of ROSA.

TABLE VI
AVERAGED COMPUTATIONAL TIME OF THE PROPOSED METHOD

	Offline Training	Online Testing
Computation Time	3.6 hours	0.086s

5, where the PGD adversarial training is selected based on its performance in Tables III–IV. For all of the performance indices, the larger number means the better classification performance. Overall, it is shown that the proposed ROSA can be a very promising solution for SA problems under adversarial disturbances.

Remark 4.1: The sensitivity to adversarial disturbances of different stability problems could be different. However, different adversarial scenarios would generate various adversarial samples according to the original datasets of different applications. Meanwhile, since all the adversarial disturbances target the ML models, different stability assessment models may have different neural network models, which could have different vulnerabilities [20]. Therefore, the sensitivity can be application dependent and NN model specific, which makes it hard to have a general conclusion. We agree that the investigation of disturbance sensitivity for different stability assessments is an important and interesting topic and could be one of our future directions.

E. Scalability Analysis

To verify the effectiveness and scalability, we applied the proposed solution for TSA to the Illinois 200-bus system which includes 200 buses, 49 generators, and 160 loads. The data source can be found in [38]. The approach as in Section IV.B is applied to prepare the training data. In this case, the trade-off parameters are $\alpha_1 = 0.98$ and $\alpha_2 = 1.02$, and the Gaussian noise is ($\mu = 0$, $\sigma_G = 1.04$). The testing results are included in the revised paper and shown below. It should be noted that increasing the system scale only affects the computational time during the training stage, and once the ROSA model is trained/converged, it can perform in an online fashion, as shown in Table V. Meanwhile, the computational time is given in Table VI, which verifies its potential capability of online implementation.

TABLE VII
NN ARCHITECTURES FOR TSA

FCN	CNN	LSTM
Fully-connected, (128)	3x3 Conv., (32)	LSTM layer, (128)
Fully-connected, (64)	3x3 Conv., (64)	LSTM layer, (64)
Fully-connected, (2)	Fully-connected, (64)	Fully-connected, (64)
	Fully-connected, (2)	Fully-connected, (2)

TABLE VIII
NN ARCHITECTURES FOR STVS

FCN	CNN	LSTM
Fully-connected, (256)	3x3 Conv., (48)	LSTM layer, (256)
Fully-connected, (128)	3x3 Conv., (96)	LSTM layer, (128)
Fully-connected, (2)	Fully-connected, (96)	Fully-connected, (64)
	Fully-connected, (2)	Fully-connected, (2)

V. CONCLUSION

This paper develops a novel ROSA framework for mitigating adversarial disturbances in real-time. Vulnerability analysis reveals that the ML-based SA models are easily compromised to produce inaccurate/erroneous output under either white-box or black-box attack scenarios. We leverage the merits of self-supervised learning and design a suitable objective to assist the original SA objective and improve model robustness. Training the self-supervised learning-based models requires proper data augmentation to learn meaningful representations of the data itself. Efficient data augmentation methods are therefore introduced to time-series PMU measurements. After the modules in ROSA are trained jointly, an online purification algorithm is developed to reduce the success probability of an adversarial disturbance tremendously. The results show that the ROSA framework provides an effective way to relieve computational burdens of adversarial learning-based mitigation and a promising solution to robustifying the existing ML-based models in power system applications.

Future works would include the investigation of applying different and even more complex adversarial disturbances.

APPENDIX

A. Parameters for NNs

Tables VII and VIII summarize the parameters of NNs that are used to train the ROSA model for each application case.

REFERENCES

[1] G. S. Chawda et al., “Comprehensive review on detection and classification of power quality disturbances in utility grid with renewable energy penetration,” *IEEE Access*, vol. 8, pp. 146807–146830, 2020.

[2] 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.

[3] L. Duchesne, E. Karangelos, and L. Wehenkel, “Recent developments in machine learning for energy systems reliability management,” *Proc. IEEE Proc. IRE*, vol. 108, no. 9, pp. 1656–1676, Sep. 2020.

[4] L. S. Moulin, A. P. A. Da Silva, M. A. El-Sharkawi, and R. J. Marks, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 818–825, May 2004.

[5] H. Yang, W. Zhang, J. Chen, and L. Wang, "PMU-based voltage stability prediction using least square support vector machine with online learning," *Electric Power Syst. Res.*, vol. 160, pp. 234–242, 2018.

[6] R. Diao et al.,, "Decision tree-based online voltage security assessment using PMU measurements," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 832–839, May 2009.

[7] S. Liu et al., "An integrated scheme for online dynamic security assessment based on partial mutual information and iterated random forest," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3606–3619, Jul. 2020.

[8] 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.

[9] Y. Zhang, Y. Xu, Z. Y. Dong, and R. Zhang, "A hierarchical self-adaptive data-analytics method for real-time power system short-term voltage stability assessment," *IEEE Trans. Ind. Inform.*, vol. 15, no. 1, pp. 74–84, Jan. 2019.

[10] H. Wang, Q. Chen, and B. Zhang, "Transient stability assessment combined model framework based on cost-sensitive method," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 12, pp. 2256–2262, 2020.

[11] S. K. Azman, Y. J. Isbeih, M. S. El Moursi, and K. Elbassioni, "A unified online deep learning prediction model for small signal and transient stability," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4585–4598, Nov. 2020.

[12] H. Hagmar, L. Tong, R. Eriksson, and L. A. Tuan, "Voltage instability prediction using a deep recurrent neural network," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 17–27, Jan. 2021.

[13] 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.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[15] D. Selsam, P. Liang, and D. L. Dill, "Developing bug-free machine learning systems with formal mathematics," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3047–3056.

[16] Z. Li and D. Hoiem, "Improving confidence estimates for unfamiliar examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2686–2695.

[17] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.

[18] G. Ravikumar and M. Govindarasu, "Anomaly detection and mitigation for wide-area damping control using machine learning," *IEEE Trans. Smart Grid*, early access, May 18, 2020, doi: 10.1109/TSG.2020.2995313.

[19] R. Ma, S. Basumallik, and S. Eftekharnejad, "A PMU-based data-driven approach for classifying power system events considering cyberattacks," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3558–3569, Sep. 2020.

[20] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[21] K. R. Mestav and L. Tong, "Universal data anomaly detection via inverse generative adversary network," *IEEE Signal Process. Lett.*, vol. 27, pp. 511–515, 2020.

[22] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4910–4920, Nov. 2019.

[23] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.

[24] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 262–271.

[25] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," 2017, *arXiv:1710.10766*.

[26] C. Ren, X. Du, Y. Xu, Q. Song, Y. Liu, and R. Tan, "Vulnerability analysis, robustness verification, and mitigation strategy for machine learning-based power system stability assessment model under adversarial examples," *IEEE Trans. Smart Grid*, vol. 13, no. 2, pp. 1622–1632, Mar. 2022.

[27] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang, "Adversarial robustness: From self-supervised pre-training to fine-tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 699–708.

[28] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12310–12320.

[29] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient Stability of Power Systems: A Unified Approach to Assessment and Control*, vol. 581. Berlin, Germany: Springer, 2000.

[30] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[31] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[32] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[33] C. Shi, C. Holtz, and G. Mishne, "Online adversarial purification based on self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=_i3ASPp12WS

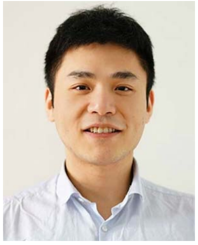
[34] O. A. Alimi, K. Ouahada, and A. M. Abu-Mahfouz, "A review of machine learning approaches to power system security and stability," *IEEE Access*, vol. 8, pp. 113512–113531, 2020.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[36] D. Shoup, J. Paserba, and C. Taylor, "A survey of current practices for transient voltage dip/sag criteria related to power system stability," in *Proc. IEEE PES Power Syst. Conf. Expo.*, 2004, pp. 1140–1147.

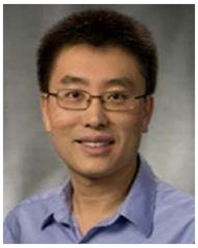
[37] D. M. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

[38] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overby, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, 2017, doi: 10.1109/TPWRS.2016.2616385.



Tianqiao Zhao (Member, IEEE) received the B.Eng. degree in automatic control from North China Electric Power University, Beijing, China, in 2013, and the Ph.D. degree in electrical and electronic engineering from the University of Manchester, Manchester, U.K., in 2019. From September 2018 to August 2019, he was a Postdoctoral Associate with the Department of Electrical and Electronic Engineering, University of Manchester. From September 2019 to February 2021, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX, USA. He is currently working with Brookhaven National Laboratory, Upton, NY, USA. His research interests include machine learning applications, distributed optimization and control, smart grid, and energy storage systems.

Meng Yue (Member, IEEE) received the Ph.D. degree in electrical engineering from Michigan State University, East Lansing, MI, USA. He is currently with Interdisciplinary Science Department, Brookhaven National Laboratory, Upton, NY, USA. His research interests include power system dynamics and control, uncertainty modeling and probabilistic applications in grid planning and operation and data analytic in damage forecasting, and anomaly detection.



Dr. Jianhui Wang (Fellow, IEEE) is currently a Professor with the Department of Electrical and Computer Engineering with Southern Methodist University, Dallas, TX, USA. His research interests include smart grid, microgrids, power system operation and control, renewable integration, grid resilience, and cybersecurity. He is the Past Editor-in-Chief of IEEE TRANSACTIONS ON SMART GRID and an IEEE PES Distinguished Lecturer. He is also the Guest Editor of the Proceedings of the IEEE special issue on power grid resilience. He is the recipient of the IEEE PES Power System Operation Committee Prize Paper Award in 2015, the 2018 Premium Award for Best Paper in IET Cyber-Physical Systems: Theory & Applications, the Best Paper Award in IEEE Transactions on Power Systems in 2020, and the IEEE PES Power System Operation, Planning and Economics Committee Prize Paper Award in 2021. He is a Clarivate Analytics highly cited Researcher of production of multiple highly cited papers that rank in the top 1% by citations for field and year in Web of Science (2018–2022).