Vulnerability Analysis, Robustness Verification, and Mitigation Strategy for Machine Learning-Based Power System Stability Assessment Model Under Adversarial Examples

Chao Ren, *Student Member, IEEE*, Xiaoning Du, *Member, IEEE*, Yan Xu, *Senior Member, IEEE*, Qun Song, *Student Member, IEEE*, Yang Liu, *Senior Member, IEEE*, and Rui Tan, *Senior Member, IEEE*

***Abstract*—Based on machine learning (ML) technique, the data-driven power system stability assessment has received significant research interests in recent years. Yet, the ML-based models may be vulnerable to the adversarial examples, which are very close to the original input but can lead to a different (wrong) assessment result. Taking short-term voltage stability (STVS) assessment problem as the case study, this paper firstly analyzes the vulnerability of the ML-based models under both the white-box and the black-box attack scenarios, where adversarial examples are generated to falsify the STVS assessment model into the wrong outputs without noticeable changes of the input values. Then, an empirical index is proposed to quantitatively measure the robustness of ML-based models under adversarial examples. After that, an adversarial training-based mitigation strategy is proposed to enhance the ML-based model against the adversarial examples under both the white-box and the black-box scenarios. Simulation results have clearly illustrated the threat of the adversarial examples to the ML-based models and verified the effectiveness of the proposed mitigation strategy.**

***Index Terms*—Adversarial attack, adversarial examples, mitigation strategy, machine learning, robustness verification, short-term voltage stability, vulnerability analysis.**

I. Introduction

MACHINE learning (ML)-based data-driven models have been identified as a promising approach to achieve real-time stability assessment of power grids [1]. The principle of the data-driven method is that: at the offline stage, an ML-based model is trained by a stability database with the strategically selected features; at the online stage, with the real-time or online measurements, such a well-trained ML-based model can directly calculate the stability assessment result, which can provide the advantages of much faster assessment speed, stronger generalization capability, and less data requirement. Taking post-fault short-term voltage stability (STVS) as an example, the input of the ML-based model is the real-time voltage trajectories and the output is the stability status or degree [2].

In the literature, a variety of data-driven models have been reported with satisfactory performance [3]. For example, support vector machine (SVM) is used for transient stability [4] and voltage stability assessment [5]. Single decision tree (DT) is used to predict the voltage stability state [6]. Besides, ensemble of DT, such as random forests [7], [8], are proposed to enhance the accuracy. In European *iTesla* project, DT is used for online static and dynamic security assessments [9]. Another optimal classification trees [10] optimizes the tree structure to explain the dynamic security assessment. The ensemble model with extreme learning machine (ELM) and randomized vector functional link network (RVFL), are used to predict the voltage stability status [2], [11], [12]. The feed-forward neural networks are utilized to for the load stability margin [13], etc. Moreover, deep learning technique based on deep neural networks have shown the better performance in terms of both assessment accuracy and speed. The convolutional neural networks (CNNs) [14]–[16] have been applied to cope with transient stability assessment. In [17], authors propose to represent the power system stability state as an image to take merits of the CNN algorithms for small-signal stability. In [18], deep belief neural network (DBN) is applied for transient stability assessment. Besides, recurrent neural networks (RNNs) are also utilized, since they have the better ability to consider spatial and temporal correlations, such as long short-term memory (LSTM) units [19], [20] and gated recurrent units (GRU) [21]. Besides, generative adversarial networks (GAN) [22] is to deal with incomplete PMU measurements, which can be easily updated during real-time operations. A transfer learning-based method [23] is utilized to solve the across domain faults classification.

Although these data-driven models have achieved excellent performance in power system stability assessment, their

Manuscript received October 7, 2020; revised May 6, 2021 and September 27, 2021; accepted October 30, 2021. This work was supported in part by the Ministry of Education (MOE), Republic of Singapore, under Grant AcRF TIER 1 2019-T1-001-069 (RG75/19). Paper no. TSG-01500-2020. *(Corresponding author: Yan Xu.)*

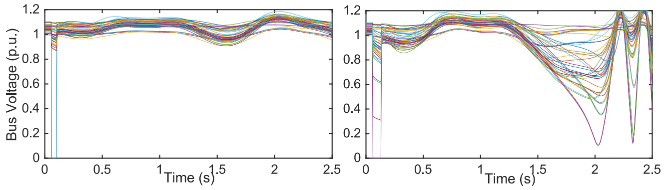
Chao Ren and Qun Song are with the Interdisciplinary Graduate School, Nanyang Technological University, Singapore.

Xiaoning Du, Yang Liu, and Rui Tan are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2021.3133604.

Digital Object Identifier 10.1109/TSG.2021.3133604



(a) stable (b) unstable

Fig. 1. Examples of stable and unstable situations. (a) stable; (b) unstable.

robustness and security have not been well examined [24]. Generally, the existing data-driven models always assumed that the model inputs are intact. However, due to potential cyber-attacks, such as false data injection [25], noise manipulation [26], and communication error [27], the data-driven models designed for power systems may be vulnerable to the adversarial perturbations. In this regard, an adversarial example is defined as a modified version of the original sample that is intentionally perturbed but retains very close to the original one [28]. As a result, the adversarial examples can mislead the data-driven models to a wrong/inaccurate output. For example, if a data-driven STVS model is perturbed, it may occur that the unstable status is predicted to be stable, and therefore the system would miss the emergency control actions, resulting in cascading failure or even wide-spread blackout; or the stable status is predicted to be unstable, and then the system would activate the emergency control (e.g., load shedding), leading to unnecessary costs and customer interruption.

This paper aims to firstly reveal the threat of the adversarial examples to the ML-based model, then systematically evaluate the robustness of the ML-based model under the adversarial examples, and finally develop a mitigation strategy against the adversarial examples. The technical contributions in this paper are three-fold.

1) The threat of the adversarial examples for the ML-based model under both the white-box and the black-box scenarios is illustrated using an adversarial example generation strategy. It reveals that the adversarial example can obviously lead to ML accuracy degradation.
2) To accurately quantify the vulnerability of the ML-based models and instances, two robust indices are proposed for the empirical robustness evaluation, which can be used to select the robust ML-based models and measure the anti-noise ability of instances in practice.
3) To systematically counteract the adversarial examples, a mitigation strategy is designed via adversarial training and the empirical robustness evaluation, which can maintain the accuracy and improve the robustness of the ML-based model against the adversarial examples.

II. Problem Description

A. Real-Time STVS Assessment

This paper considers the STVS assessment as a use case. Note that other stability criteria such as transient stability or frequency stability can also be considered without loss of generality. According to the time frame of the study, voltage stability can be divided into long-term and short-term phenomena [29]. This paper mainly studies STVS, which concerns on a few seconds after the fault clearance. As shown in Figs. 1(a–b), there are different modes of voltage propagation after disturbance in stable and unstable conditions [2]. Under the stable condition, the voltage amplitude of all buses in the system can be restored to an acceptable level (i.e., no less than 90% of the voltage level before interference [30]). However, if any bus voltage is maintained at an unacceptable low level, or the voltage collapses within the transition time (i.e., 10 seconds), the power system is considered as unstable, which may lead to cascading failure or even large-scale outage. Based on PMU measurement of the post-fault voltage trajectories, the STVS status can be predicted by the data-driven model and activate the emergency control if necessary.

B. ML-Based Data-Driven Model

Given a paired training sample $\mathbf{x} = [x_t, t = 1, \ldots, T]$ and label y, the ML-based data-driven model aims to learn a function $f_\theta(\cdot)$ as Eq. (1), which can map the relationship from $\mathbf{x}$ to y with the model parameters θ. For STVS problem, $\mathbf{x}$ represents the voltage trajectories value and y represents the corresponding stability status.

$$f_\theta(\mathbf{x}) = f^{(m)}\Big(\ldots f^{(2)}\Big(f^{(1)}(\mathbf{x})\Big)\Big) \tag{1}$$

where $f^{(h)}$ represents the function of the h-th layer of the neural network, $h = 1, 2, \ldots m$. The training process of ML-based model is to minimize the difference between the predicted $f_\theta(\mathbf{x})$ and the ground truth label y as Eq. (2).

$$\min_{\theta} \; L_f(f_\theta(\mathbf{x}), y) \tag{2}$$

where $L(\cdot,\cdot)$ is the pre-defined loss function of model $f_\theta(\cdot)$. A typical neural network can solve such back-propagation procedure via gradient descent algorithms to update the model parameters as Eq. (3).

$$\theta_{i+1} = \theta_i - \eta \cdot \nabla_\theta L_f(f_\theta(\mathbf{x}), y) \tag{3}$$

where η is the learning rate; i represents the i-th iteration step; θ_i represents the model parameters. Once training sample $\mathbf{x}$ and label y are available, the accurate ML-based model can be obtained by various neural network training algorithms [1].

III. Adversarial Example Generation Strategy

A. Problem Formulation

Different from ML-based model training process, adversarial example generation strategy is based on an already trained ML-based model with the parameters θ, and aims to misguide the model. Given a trained classifier $f_\theta(\cdot)$ and a sample $\mathbf{x} = [x_t, t = 1, \ldots, T]$ with its ground truth label y, an adversarial example $\mathbf{x}^{adv} = [x_t^{adv}, t = 1, \ldots, T]$ can be generated via solving the optimization problem

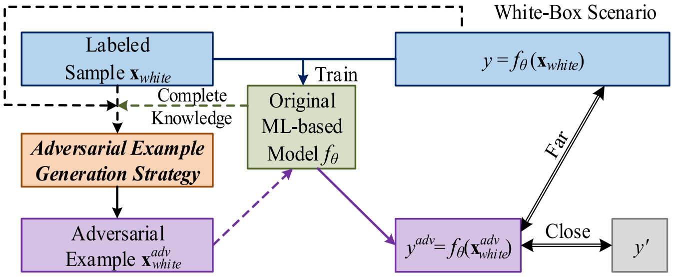


Fig. 2. Adversarial example generation strategy process of the white-box scenarios.

formalized in Eq. (4):

$$\min_{\mathbf{x}^{adv}} \left\| \mathbf{x}^{adv} - \mathbf{x} \right\|_p$$

$$s.t.: \begin{cases} f_\theta(\mathbf{x}) = y \\ f_\theta\left(\mathbf{x}^{adv}\right) = y' \neq y \end{cases} \tag{4}$$

where y and y' represent the corresponding output label of $\mathbf{x}$ and $\mathbf{x}^{adv}$, respectively; $\|\cdot\|_p$ denotes the distance between $\mathbf{x}$ and $\mathbf{x}^{adv}$, and p measures the magnitude of adversarial perturbation by p-norm distance. Note that the generated adversarial examples cannot always be feasible, since such optimization problem is non-convex and we can only calculate the approximate solution of the adversarial examples. In other words, it may be the case that the solution is not the exact global solution due to the non-convexity and complexity of the problem. In that case, the attack may be infeasible.

Based on gradient algorithms, the adversarial example $\mathbf{x}^{adv}$ can be quickly calculated. For a popular fast method called fast gradient sign method [31], it only needs to update gradient for one-step along the direction of the sign of the gradient as follow:

$$\mathbf{x}^{adv} = \mathbf{x} + \delta \cdot sign(\nabla_{\mathbf{x}} L(f_\theta(\mathbf{x}), y)) \tag{5-a}$$

where $sign(\cdot)$ is the sign function; δ specifies the boundary of perturbation; $L(\cdot,\cdot)$ represents the loss function of model $f_\theta(\cdot)$. Besides, we can also utilize multi-step gradient iterative method to generate the adversarial examples as follows:

$$\mathbf{x}_{i+1}^{adv} = \mathbf{x}_i^{adv} + (\delta/i) \cdot sign\Big(\nabla_{\mathbf{x}} L\Big(f_\theta\left(\mathbf{x}_i^{adv}\right), y\Big)\Big) \tag{5-b}$$

According to different scenarios, assumptions, and the requirements of the attackers' knowledge, the adversarial examples can be generated under either the *white-box* or the *black-box* scenarios. The former assumes the complete knowledge of the original trained ML-based model (e.g., original training database, training algorithm, model structure, parameter settings, etc.) is available to cyber attackers. The latter assumes none or limited knowledge of the original trained ML-based model is available to cyber attackers. Under both the white-box and the black-box scenarios, the cyber attackers have the access to use the original trained ML-based model. The generation process under the white-box and black-box scenarios are summarized in Algorithm 1, and also illustrated in Fig. 2 and Fig. 3, respectively (solid lines represent generation/calculation and dashed lines represent transmission).

Algorithm 1: Vulnerability Analysis Process Under the White-Box Scenarios and Black-Box Scenarios

Input: Labeled sample $\mathbf{x}_{white}$ with ground truth label y for the white-box scenarios, unlabeled sample $\mathbf{x}_{black}$ for the black-box scenarios.
Output: Generated adversarial example $\mathbf{x}_{white}^{adv}$ or $\mathbf{x}_{black}^{adv}$.
Initialize: Original well-trained ML-based model f_θ, magnitude of perturbation distance p.
begin
  if *Complete knowledge of ML-based model f_θ White-box* **then**
    # Generate $\mathbf{x}_{white}^{adv}$ under the original ML-based model f_θ.
    Calculate and obtain the adversarial example $\mathbf{x}_{white}^{adv}$ as Eq. (5).
  else
    No knowledge of ML-based model f_θBlack-box
    # Calculate the sample $\mathbf{x}_{black}$ corresponding label y^{close}.
    Feed the sample $\mathbf{x}_{black}$ into the original ML-based model f_θ.
    Obtain the corresponding label y^{close}.
    # Train a surrogate ML-based model $\hat{f}_\theta$.
    Train a surrogate ML-based model $\hat{f}_\theta$ with sample $\mathbf{x}_{black}$ and corresponding label y^{close} via an effective ML algorithm.
    Obtain a surrogate ML-based model $\hat{f}_\theta$.
    #Generate $\mathbf{x}_{black}^{adv}$ under such surrogate ML-based model $\hat{f}_\theta$.
    Calculate and obtain the adversarial example $\mathbf{x}_{black}^{adv}$ as Eq. (5).
  end
  #Analysis the original ML-based model f_θ via $\mathbf{x}_{white}^{adv}$ and $\mathbf{x}_{black}^{adv}$.
  Feed the generated $\mathbf{x}_{white}^{adv}$ and $\mathbf{x}_{black}^{adv}$ into the ML-based model f_θ.
end

B. Adversarial Examples Under the White-Box Scenario

Under the white-box scenarios, the original sample $\mathbf{x}_{white}$, corresponding to the ground truth label y, and the original ML-based model $f_\theta(\cdot)$ are known. The adversarial example $\mathbf{x}_{white}^{adv}$ can be calculated via solving Eq. (4), such as Eq. (5-a/b).

C. Adversarial Examples Under the Black-Box Scenario

The black-box scenarios assume no or limited knowledge of the original well-trained ML-based model but can use the model. In this case, we firstly feed the unlabeled sample $\mathbf{x}_{black}$ into the original well-trained ML-based model $f_\theta(\cdot)$ in order to obtain the corresponding label y^{close} that is extremely close to the ground truth label. Then, a surrogate ML-based model $\hat{f}_\theta(\cdot)$ is trained to emulate the original ML-based model $f_\theta(\cdot)$ via an effective ML algorithm with such unlabeled sample $\mathbf{x}_{black}$ and the close corresponding label y^{close}. Finally, the adversarial example $\mathbf{x}_{black}^{adv}$ can be generated through such surrogate

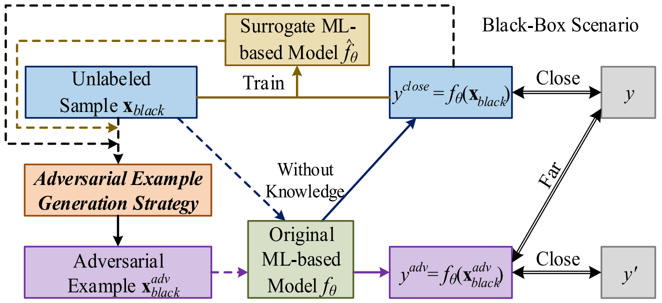


Fig. 3. Adversarial example generation strategy process of the black-box scenarios.

ML-based model $\hat{f}_\theta(\cdot)$ by Eq. (5). Although the adversarial example generation strategy does not necessarily depend on the original trained ML-based model $f_\theta(\cdot)$, the adversarial example $\mathbf{x}^{adv}_{black}$ generated by the surrogate ML-based model $\hat{f}_\theta(\cdot)$ can still mislead the original ML model $f_\theta(\cdot)$. Thus, the adversarial examples have the transferability to falsify/confuse other ML-based models trained by different effective ML algorithms [28].

IV. Empirical Robustness Evaluation

A. Proposed Robustness Indices

In order to quantify the robustness of a well-trained ML-based model to adversarial perturbations, an empirical robustness evaluation process is proposed in this paper. It is equivalent to calculating the average minimal adversarial perturbation for a successful adversarial attack.

Formally, given a trained ML-based classifier, we define the adversarial perturbation $\boldsymbol{\varepsilon}_\mathbf{x}$ added on the original sample as follows:

$$\mathbf{x}^{adv} = \boldsymbol{\varepsilon}_\mathbf{x} + \mathbf{x} \tag{6}$$

Then the optimization problem formalized in Eq. (4) can be transformed as Eq. (7), which minimizes the adversarial perturbation while misclassifying the results.

$$\begin{aligned} &\min_{\boldsymbol{\varepsilon}_\mathbf{x}} \ \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \\ &s.t.\!: \begin{cases} f_\theta(\mathbf{x}) = y \\ f_\theta(\mathbf{x}) \neq f_\theta(\mathbf{x}+\boldsymbol{\varepsilon}_\mathbf{x}) \end{cases} \end{aligned} \tag{7}$$

where $\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}} = \arg\min_{\boldsymbol{\varepsilon}_\mathbf{x}} \|\boldsymbol{\varepsilon}_\mathbf{x}\|_2$ represents the minimal adversarial perturbation of the original sample under the classifier $f_\theta(\cdot)$. For the L2-norm distance, $\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\|_2$ represents the minimal distance from the original sample $\mathbf{x}$ to the classification boundary, which can be used to quantify the *robustness index for instance* (RII) of the classifier $f_\theta(\cdot)$ for the original sample $\mathbf{x}$.

$$\mathrm{RII}(\mathbf{x}) = \left\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\right\|_2 \tag{8}$$

Besides, the *robustness index for classifier* (RIC) can be defined as Eq. (9).

$$\mathrm{RIC}(f_\theta(\cdot)) = \mathop{\mathbb{E}_\mathbf{x}}_{\mathbf{x}\in\mathcal{D}} \frac{\left\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\right\|_2}{\|\mathbf{x}\|_2} \tag{9}$$

where $\mathbb{E}_\mathbf{x}$ denotes the expectation over the distribution of data.

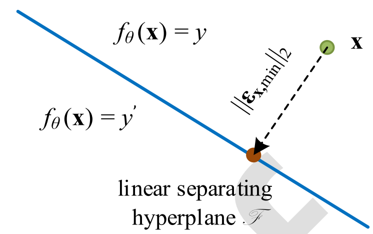


Fig. 4. The adversarial perturbation for a linear binary classifier.

B. Adversarial Perturbations for Linear Binary Classifiers

Firstly, we consider the binary classifier $f_\theta(\cdot)$ as a linear classifier, and then derive the standard algorithm for nonlinear classifiers. For the binary classifier $f_\theta(\cdot)$, we denote the separating affine hyperplane $\mathcal{F} \triangleq \{\mathbf{x} : f_\theta(\mathbf{x}) = 0\}$ as the level set at zero of model $f_\theta(\cdot)$.

For the former case (linear binary classifier), RII is equal to the distance between the original instance $\mathbf{x}$ and the separating hyperplane $\mathcal{F}$, that is, the minimal adversarial perturbation $\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}$ formalized in Eq. (10) corresponds to the orthogonal projection of original instance $\mathbf{x}$ onto the separating hyperplane $\mathcal{F}$ as Fig. 4.

$$\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}} = \arg\min_{\boldsymbol{\varepsilon}_\mathbf{x}} \|\boldsymbol{\varepsilon}_\mathbf{x}\|_2 = -\frac{f_\theta(\mathbf{x})}{\|\theta\|_2^2}\theta \tag{10}$$

This formula can be explained as multiplying the shortest distance $f_\theta(\mathbf{x})/\|\theta\|_2$ from the instance $\mathbf{x}$ to the classification separating hyperplane $\mathcal{F}$ by the unit vector $\theta/\|\theta\|_2$ in the normal direction. The negative sign indicates that the direction always points to the separating hyperplane $\mathcal{F}$.

C. Adversarial Perturbation for Nonlinear Binary Classifiers

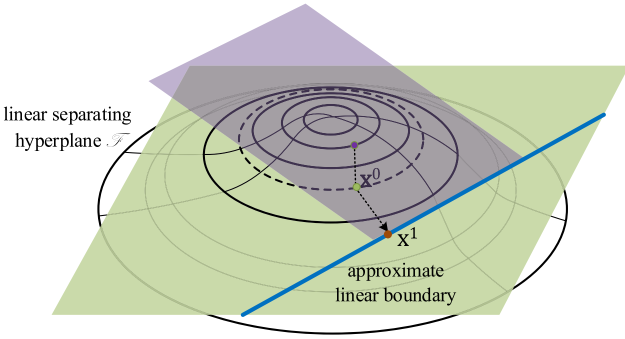
For the nonlinear classification problem, when the separating hyperplane $\mathcal{F}$ is nonlinear, we can simplify the nonlinear problem into the iterative approximate linear decision function [32], formalized in Eq. (11). Assuming that when the shift distance is very small, the hyperplane $\mathcal{F}$ can be considered as a linear separating hyperplane relative to this instance as shown in Figs. 5(a–b).

$$\begin{aligned} &\min_{\boldsymbol{\varepsilon}^i_\mathbf{x}} \ \left\|\boldsymbol{\varepsilon}^i_\mathbf{x}\right\|_2 \\ &s.t.\!: \ f_\theta\left(\mathbf{x}^i\right) + \nabla f_\theta\left(\mathbf{x}^i\right)^T \boldsymbol{\varepsilon}^i_\mathbf{x} = 0 \end{aligned} \tag{11}$$

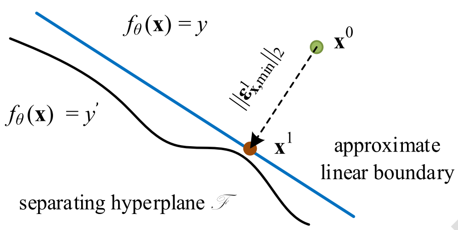
where i represents the iteration steps of the model parameters. In each iteration, the instance $\mathbf{x}^i$ continuously approaches the approximate linear classification boundary with a very small shift distance, and the minimal adversarial perturbation vector can be always calculated based on the current i-th iteration value as Eq. (12).

$$\boldsymbol{\varepsilon}^i_{\mathbf{x},\mathbf{min}} = \arg\min_{\boldsymbol{\varepsilon}^i_\mathbf{x}} \left\|\boldsymbol{\varepsilon}^i_\mathbf{x}\right\|_2 = -\frac{f_\theta\left(\mathbf{x}^i\right)}{\left\|\nabla f_\theta\left(\mathbf{x}^i\right)\right\|_2^2}\nabla f_\theta\left(\mathbf{x}^i\right) \tag{12}$$

The next iteration value $\mathbf{x}^{i+1}$ is updated by calculating the adversarial perturbation $\boldsymbol{\varepsilon}^i_\mathbf{x}$ according to current approximate linear classification boundaries. The continuous procedure superposes the $\boldsymbol{\varepsilon}^i_\mathbf{x}$ of each iteration value as Eq. (13), to



(a) Illustration of Eq. (11). The purple plane is the graph of the constraint, which is tangent to the separating hyperplane. The blue line shows $f_\theta(\mathbf{x}^0) + \nabla f_\theta(\mathbf{x}^0)^T(\mathbf{x} - \mathbf{x}^0) = 0$, and $\mathbf{x}^1$ is acquired via projection of instance $\mathbf{x}$ onto separating hyperplane. The whole process needs i step iterations.



(b) Illustration of two-dimension as gray graph in (a). The black line represents the true separating hyperplane. The blue line represents the approximate linear boundary.

Fig. 5. (a-b). The adversarial perturbation for nonlinear binary classifier.

obtain the minimal adversarial perturbation $\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}$ until the perturbed instance $(\mathbf{x}+\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}})$ makes the different target from the original instance $\mathbf{x}$, that is, $f_\theta(\mathbf{x}) \neq f_\theta(\mathbf{x}+\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}})$.

$$\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}} = \sum_i \boldsymbol{\varepsilon}^i_{\mathbf{x},\mathbf{min}} \tag{13}$$

According to Eq. (13), the adversarial perturbation vector $\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}$ can only make the instance reach the separating hyperplane, but not be enough to cross it. Thus, when generating the adversarial examples, the final adversarial perturbation vector is usually multiplied by a small constant σ, that is, $\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}(1+\sigma)$, $\sigma \ll 1$, in order to prevent the algorithm from converging to the separating hyperplane.

D. Robustness Indices Calculations

Based on the above analysis, RII and RIC can be calculated as Eq. (14) and Eq. (15), respectively.

$$\mathrm{RII}(\mathbf{x}) = \left\|\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}\right\|_2 \tag{14}$$

$$\mathrm{RIC}(f_\theta(\cdot)) = \frac{1}{|N|}\sum_{\mathbf{x}_n \epsilon \mathcal{D}} \frac{\left\|\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}\right\|_2}{\|\mathbf{x}_n\|_2} \tag{15}$$

where $\mathbf{x}_n$ is the n-th instance from the database $\mathcal{D}^N$. The empirical robustness indices (RII and RIC) can be utilized to measure the robustness of instances and classifiers under the adversarial attack. The larger RII and RIC values indicate stronger abilities of instances and classifiers against the adversarial perturbations.

V. Mitigation Strategy

After quantification of the robustness of the instance and the classifier against adversarial samples, the ultimate goal is to train a robust ML-based model that is immune to adversarial examples. This paper proposes an adversarial training-based mitigation strategy for both white-box and black-box scenarios.

A. Adversarial Training-Based Mitigation Strategy

Adversarial training [33] is one effective defensive mitigation method against adversarial examples. The principle is to use a mixture of adversarial examples and original samples to train the ML-based STVS models, rather than using only the original samples.

The adversarial training-based mitigation strategy aims to train a robust ML-based model $g_{\widetilde{\theta}}(\cdot)$ with model parameters $\widetilde{\theta}$. The stability status y can be predicted by the robust ML-based model $g_{\widetilde{\theta}}(\cdot)$ with the input $\mathbf{x}$, such that $y = g_{\widetilde{\theta}}(\mathbf{x})$. The optimization objective function $L_g(g_{\widetilde{\theta}}(\mathbf{x}), y)$ can be formalized as Eq. (16).

$$\begin{aligned} L_g\big(g_{\widetilde{\theta}}(\mathbf{x}), y\big) &= (1-\alpha)\cdot L_f(f(\mathbf{x}), y) \\ &\quad + \alpha\cdot L_f\Big(f\big(\mathbf{x}+\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}(1+\sigma)\big), y\Big) \end{aligned} \tag{16}$$

where α represents the ratio of the adversarial examples; the adversarial training of ML-based model is to minimize the difference between predicted $f(\mathbf{x})$, $f(\mathbf{x}+\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}(1+\sigma))$ and true label y as Eq. (17).

$$\min_{\widetilde{\theta}} \; L_g\big(g_{\widetilde{\theta}}(\mathbf{x}), y\big) \tag{17}$$

Then, a typical neural network can solve such back-propagation procedure by gradient descent algorithms to update the model parameters as Eq. (18).

$$\widetilde{\theta}_{i+1} = \widetilde{\theta}_i - \eta \cdot \nabla_{\widetilde{\theta}} L_g\big(g_{\widetilde{\theta}}(\mathbf{x}), y\big) \tag{18}$$

where η is the learning rate; i represents the iteration steps; $\widetilde{\theta}_i$ represents parameters of the robust ML-based model $g_{\widetilde{\theta}}(\cdot)$ at the i-th iteration.

With the increase of the training samples (including the adversarial examples), the robustness of the ML-based model against the adversarial attack will be improved. At each step of the training, the adversarial training-based mitigation strategy should use both the original samples and the adversarial examples. Due to different structure of the white-box scenarios and the black-box scenarios, adversarial training-based mitigation strategy should be trained via different ways to acquire the best performance.

B. Specific Single Adversarial Training-Based Mitigation Strategy Under the White-Box Scenarios

For the white-box scenarios (shown in Fig. 6), original trained ML-based model $f_\theta(\cdot)$ is completely known. Thus, the adversarial training-based mitigation strategy only needs to utilize the original samples and specific adversarial examples under the known adversarial perturbations to make the specific single adversarial training as Eq. (16), which can acquire the robust ML-based model $g_{\widetilde{\theta}}(\cdot)$ against the adversarial attack.

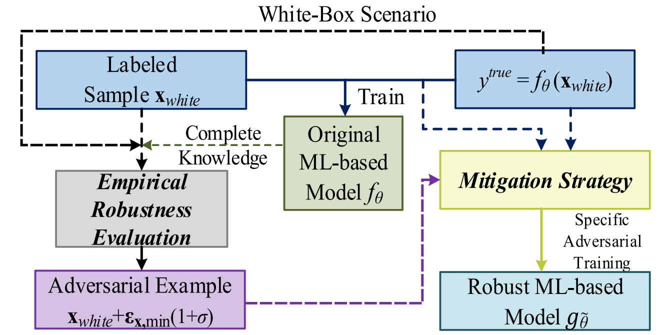


Fig. 6. The process of specific mitigation strategy for the white-box scenarios.

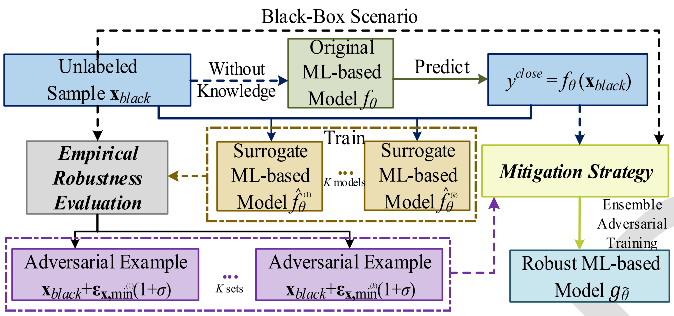


Fig. 7. The process of ensemble mitigation strategy for the black-box scenarios.

For the robust ML-based STVS model $g_{\tilde{\theta}}(\cdot)$ against the adversarial examples under the *white-box* scenarios, it is trained by the original voltage trajectories $\mathbf{x}$, the generated adversarial perturbation vector $\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}$ via the proposed empirical robustness evaluation strategy, and stability status label (the ground truth label y under the *white-box* scenarios), then the output is the predicted stability status.

C. Ensemble Adversarial Training-Based Mitigation Strategy Under the Black-Box Scenarios

In [34], it reported that the robust ML-based model $g_{\tilde{\theta}}(\cdot)$ trained by a specific single adversarial training-based mitigation strategy for the white-box scenarios is more robust than for the black-box scenarios. This is because that, single adversarial training aims to learn the knowledge from the specific adversarial examples generated by a specific known ML-based model, such ML-based model is bound to be more targeted, so it would have higher error when the original model faces the adversarial examples generated by other ML-based models (surrogate ML-based model $\hat{f}(\cdot)$).

In order to solve the threat under the black-box scenarios, based on the single adversarial training-based mitigation strategy, the ensemble adversarial training-based mitigation strategy is proposed to train the robust ML-based model $g_{\tilde{\theta}}(\cdot)$ with the original samples and multiple adversarial examples (shown in Fig. 7), which are generated by different surrogate ML-based models on the basis of the adversarial example generation strategy and the empirical robustness evaluation. For the robust ML-based STVS model $g_{\tilde{\theta}}(\cdot)$ against the adversarial examples under the *black-box* scenarios, it is trained by the original voltage trajectories $\mathbf{x}$, the generated adversarial perturbation vector $\widehat{\boldsymbol{\varepsilon}}_{\mathbf{x},\mathbf{min}}$ via the proposed empirical robustness evaluation strategy, and the stability status label (the close corresponding label y^{close} under the *black-box* scenarios), then the output is the predicted stability status. The optimization objective function $L_g(g_{\tilde{\theta}}(\mathbf{x}), y^{close})$ of ensemble adversarial training-based mitigation strategy for the black-box scenarios can be formalized as Eq. (19).

$$L_g\left(g_{\tilde{\theta}}(\mathbf{x}), y^{close}\right) = \sum_{k=1}^{K}\Big\{(1-\alpha)\cdot L_{\hat{f}^{(k)}}\left(\hat{f}^{(k)}(\mathbf{x}), y^{close}\right) + \alpha \cdot L_{\hat{f}^{(k)}}\left(\hat{f}^{(k)}\left(\mathbf{x}+\widehat{\boldsymbol{\varepsilon}}^{(k)}_{\mathbf{x},\mathbf{min}}(1+\sigma)\right), y^{close}\right)\Big\} \tag{19}$$

where α is the ratio of the adversarial examples; K denotes the total number of the surrogate ML-based model; $\hat{f}^{(k)}$ represents the k-th surrogate ML-based model; $\widehat{\boldsymbol{\varepsilon}}^{(k)}_{\mathbf{x},\mathbf{min}}$ represents the minimal adversarial perturbation of the k-th surrogate ML-based model. Based on above Eq. (17-18), the parameters of robust model $g_{\tilde{\theta}}(\cdot)$ can be obtained.

In summary, the ensemble adversarial training-based mitigation strategy for the black-box attack scenarios is to use more diverse adversarial examples and weaken the over-fitting of single model during the adversarial training process in order to improve the robustness of the model.

VI. Simulation Results

The proposed strategies are tested on the IEEE New England 10-machine 39-bus system. The numerical simulation is conducted on a high-performance computer with an Intel Core i7 CPU of 3.3-GHz, 16-GB RAM, and GPU with NVIDIA GeForce GTX 1060. The post-fault voltage trajectories for each operating condition are obtained via TDS using industry-standard software DSATools [35]. The proposed strategies are realized in Python 3.6 with the PyTorch framework.

A. Database Generation

To acquire a comprehensive database for post-fault STVS assessment, different physical faults are employed on a wide range of system operating conditions, which simulates various post-disturbance voltage trajectories. In the database generation process, the operating conditions are generated via randomly changing the load level between 0.8 and 1.2 of its base values. The different load components and the load dynamics have been considered in generating the training data, from which the ML-based model can lean the impact of load models on STVS assessment. Considering that the high penetration of induction motor loads is the leading driving force of STVS issue for current power systems, composite load model [36] is adopted in this simulation test.

In this paper, we use the industry-standard composite load model "CLOD" [37] to model different load components including motor loads. In PSS/E software, "CLOD" consists of six load types, including small motors, large motors, discharge lighting, transformer saturation, and voltage-dependent loads, all of which are typical load components in practical

TABLE I
VULNERABILITY ANALYSIS FOR STVS ASSESSMENT ACCURACY OF ADVERSARIAL EXAMPLES GENERATION STRATEGY

Observation Windows (0.8s, 1.0s, 1.2s)	Without Adversarial Examples			White-Box Scenarios			Black-Box Scenarios (using LSTM)		Black-Box Scenarios (using FCNN)		Black-Box Scenarios (using BPNN)	
	LSTM	FCNN	BPNN	LSTM	FCNN	BPNN	FCNN (Surrogate)	BPNN (Surrogate)	LSTM (Surrogate)	BPNN (Surrogate)	LSTM (Surrogate)	FCNN (Surrogate)
Average	98.78%	97.83%	97.22%	6.37%	6.22%	6.03%	19.02%	17.03%	14.63%	16.62%	13.53%	15.37%

substations. For each generated operating point, the portion of motor loads is randomly sampled between 0% and 80%, and the different load components share can be obtained via measurement-based load modeling methods [38]. The detailed factors considered in the simulation part can refer to Reference [39]. The generation level from each synchronous generator is determined by optimal power flow [12]. Three-phase faults are considered in the simulation with the random fault duration (0.1s-0.3s). Moreover, the fault location is either a bus or a transmission line which is randomly selected from the system topology. Considering the actual industry scenarios, the fault would be cleared either with the transmission line tripping or without loss of power grid component. In doing so, various fault-induced topology changes have been considered.

Finally, we select the 6536 operating samples, and the ratio between the stable and unstable samples is 1:1, which is reasonable to train the ML-based STVS models. Based on the previous studies and experience, 4536 samples with equal stable and unstable cases were randomly selected for model training, and the remaining 2000 samples with equal stable and unstable cases are used for testing.

For practical application, the longer the response time, the more measurements can be obtained, so the STVS assessment results tend to be more accurate. However, if the observation window is too long, there is less time for activating emergency control timely, and hence instability cannot be avoided. Thus, it is necessary to predict the stability state as early as possible to leave enough time for the control action. In order to comprehensively test the performance of the proposed method in a variety of situations, we chose three different observation windows (0.8s, 1.0s, and 1.2s after the fault clearance) to test for STVS assessment performance.

B. *Vulnerability Analysis*

In this case study, we select the L2-norm that denotes the level of change for the adversarial perturbation $\boldsymbol{\varepsilon}_{\mathbf{x}}$. In order to demonstrate the vulnerability of the ML-based STVS models, the adversarial example generation strategy is applied to state-of-the-art data-driven models for all three observation windows under different adversarial attack scenarios, including LSTM, fully convolutional neural network (FCNN), and back-propagation neural network (BPNN). In this case study, we only consider the neural network-based ML algorithms since the proposed methods focus on gradient based methods. Other algorithms can also be considered with specific modifications. Note that all of the data-driven STVS models under comparison have been well trained and tuned for the best performances.

For Table I, the average accuracy for three observation time-windows (0.8s, 1.0s, 1.2s after the fault clearance) under each attack scenarios are listed, from left to right, the columns are original ML-based STVS models accuracy performance without adversarial examples, ML-based STVS models accuracy performance under the white-box scenarios, and ML-based STVS models accuracy performance under the black-box scenarios, respectively. For the white-box setting, the knowledge of ML-based STVS models is completely known, so the adversarial examples can be directly generated via original ML-based STVS models; for the black-box setting, the surrogate ML-based STVS model is trained by other effective ML algorithms to generate the adversarial examples.

It can be seen that, although the original ML-based STVS models have the excellent performance, the STVS accuracy of the models would drop sharply with the generated adversarial examples under both the white-box scenarios (down to 6.03% to 6.37%) and the black-box scenarios (down to 13.53% to 19.02%). Moreover, it can be seen that the accuracy of the original ML-based STVS models degrades much more significantly in the case of the white-box scenarios than in the black-box scenarios, because the former is more targeted and attack is directly on the original ML-based STVS models, while the latter is not directly targeted and the surrogate models are used to generate the adversarial examples.

Then, this paper employs LSTM as the ML-based STVS model, since its performance has been widely demonstrated. From Figs. 8(a–d), the online testing analysis results under the 1.0s observation window are performed, and four different scenarios are shown (i.e., misclassify into stable/unstable under the white-box/black-box scenarios). The existing ML-based STVS model is trained by LSTM, for the white-box setting, the knowledge of LSTM model is completely known; for the black-box setting, a surrogate ML-based model is trained by FCNN to generate the adversarial examples. For the white-box scenarios and the black-box scenarios, original LSTM model assessment accuracy decreases respectively by 93.42% and 78.04%. Besides, three observation windows of STVS assessment accuracy are compared in Fig. 9(a), where ML-based data-driven model has the bad performance with the adversarial examples for both the white-box and the black-box scenarios, which cannot maintain original STVS assessment accuracy.

C. *Empirical Robustness Evaluation Performance*

For further analysis, the empirical robustness evaluation can evaluate the ability of trained ML-based STVS models against

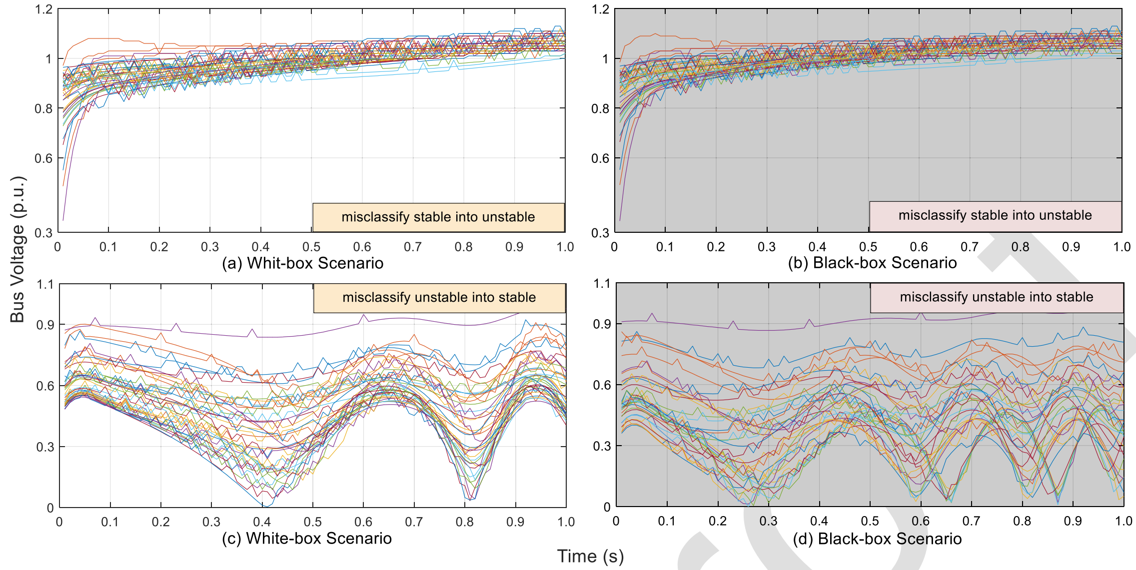


Fig. 8. Online testing results with the adversarial examples. (a) misclassify stable into unstable under the white-box scenarios; (b) misclassify stable into unstable under the black-box scenarios; (c) misclassify unstable into stable under the white-box scenarios; (d) misclassify unstable into stable under the black-box scenarios.

TABLE II
RIC PERFORMANCE FOR ML-BASED STVS MODELS

Observation Windows (0.8s, 1.0s, 1.2s)	Original ML-based Models without Adversarial Examples			Specific Adversarial Training-based Mitigation Strategy under White-Box Scenarios			Ensemble Adversarial Training-based Mitigation Strategy under the Black-Box Scenarios		
	LSTM	**FCNN**	**BPNN**	**Specific LSTM (against LSTM)**	**Specific FCNN (against FCNN)**	**Specific BPNN (against BPNN)**	**Ensemble FCNN&BPNN (against LSTM)**	**Ensemble LSTM&BPNN (against FCNN)**	**Ensemble LSTM&FCNN (against BPNN)**
Average	0.020	0.017	0.016	0.046	0.043	0.041	0.039	0.036	0.035

adversarial examples via calculating the minimal adversarial perturbation vectors and the corresponding distances between original sample and separating hyperplane. A larger empirical robustness evaluation values (RIC and RII) imply the higher STVS robustness against adversarial examples. Table II and Fig. 9(b) separately list the average and each windows RIC results of original ML-based STVS model, robust ML-based STVS models after the specific and ensemble adversarial training-based mitigation strategy, respectively.

For original ML-based models, LSTM models always have the higher RIC values than others, which means that LSTM model is more suitable for STVS problems. However, it can be seen that the robust ML-based models by adversarial training always have the higher values than the original ML-based models, and thus can further verify the vulnerability of the original ML-based models and improvement of the adversarial training-based mitigation strategy under both the white-box and the black-box scenarios among all the ML-based STVS models. Besides, the RIC value validates the adversarial training-based mitigation strategy for the white-box scenarios is more suitable than the black-box scenarios. For RII, the larger the RII value, the greater the adversarial perturbation needed to successfully attack the original sample as Figs. 8(a–d). Based on practical requirements, the system operators can select the more robustness ML-based models via RIC value; also quantify the ability of samples to resist adversarial attack and make the precautions if necessary.

D. Mitigation Strategy Performance

In order to solve the threat of the adversarial examples, the adversarial training-based mitigation strategy combining the robustness index is utilized to maintain the accuracy of ML-based models against the adversarial examples. Table III and Fig. 9(a) show the adversarial training-based mitigation strategy accuracy performances with the original clean samples and adversarial examples under three different observation windows for both the white-box and the black-box scenarios. For the original clean samples, the robust ML-based models via the proposed adversarial training-based mitigation strategy can maintain the satisfactory STVS assessment accuracy performance under both the white-box and black-box scenarios. For the white-box setting with adversarial examples, adversarial training-based mitigation strategy only needs to train with the original samples and specific generated adversarial examples, the ML-based models are not significantly influenced, and still can maintain a relatively high STVS accuracy (on average 95.69%-97.68%). For the black-box setting with adversarial examples, adversarial training-based mitigation strategy should ensemble different surrogate ML-based models to craft adversarial examples,

TABLE III
ACCURACY PERFORMANCE OF ADVERSARIAL TRAINING-BASED MITIGATION STRATEGY AGAINST ADVERSARIAL EXAMPLES

Testing Samples with Observation Windows (0.8s, 1.0s, 1.2s)	White-Box Scenarios			Black-Box Scenarios		
	Specific LSTM (against LSTM)	Specific FCNN (against FCNN)	Specific BPNN (against BPNN)	Ensemble FCNN & BPNN (against LSTM)	Ensemble LSTM & BPNN (against FCNN)	Ensemble LSTM & FCNN (against BPNN)
Clean Samples	98.23%	97.07%	97.05%	96.75%	96.68%	96.22%
Adversarial Examples	97.68%	96.07%	95.69%	95.37%	94.92%	94.70%

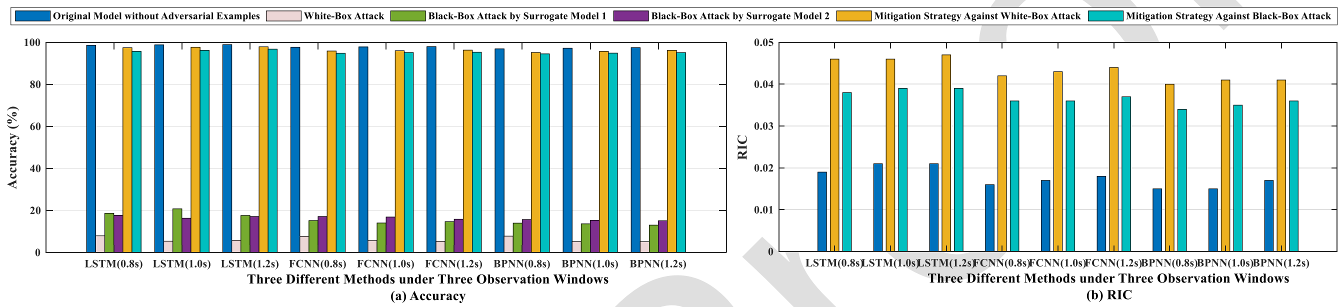


Fig. 9. Accuracy and RIC results for three different methods under the three different observation window (0.8s, 1.0s, 1.2s). (a) STVS accuracy; (b) RIC.

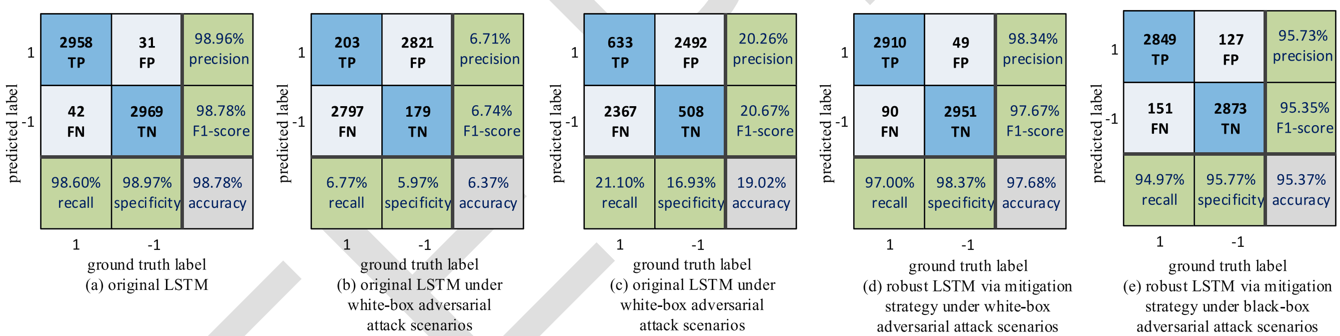


Fig. 10. Confusion matrix of original LSTM model and robust LSTM model with three different observation window under different scenarios. (a) original LSTM with original clean samples; (b) original LSTM under white-box adversarial attack scenarios; (c) original LSTM under black-box adversarial attack scenarios via surrogate FCNN; (d) robust LSTM via mitigation strategy under white-box adversarial attack scenarios; (e) robust LSTM via mitigation strategy under black-box adversarial attack scenarios.

which can increase the diversity of the adversarial examples. Hence, the increased diversity of adversarial examples for ensemble adversarial training-based mitigation strategy can provide marginal improvements.

From Table III, it can be seen that the adversarial training-based mitigation for the white-box scenarios (on average 95.69%-97.68%) have the better performance than the black-box scenarios (on average 94.70%-95.37%); the ensemble adversarial training under the black-box scenarios have the improvement for single adversarial training. Besides, Fig. 9(b) shows that the proposed adversarial training-based mitigation strategy can improve RIC value of the ML-based models, verifying the improvement for the robustness of the ML-based models.

E. STVS Assessment Confusion Matrix Performance

To further verify the vulnerability and the superiority of the proposed mitigation strategy, we have added the tests to show the confusion matrix, including true positive (TP), true negative (TN), false positive (FP) and false negative (FN). For STVS problem, the most important concept is the number of FP cases, which means the unstable case is misclassified as the stable case, resulting in cascading failure or even wide-spread blackout. Then, based on FP, four relevant indices are utilized as Reference [40], including precision, specificity, accuracy and F1-score. The larger the value of these four indices means the better the STVS assessment performance.

Table IV shows the average STVS assessment performance of above different ML-based models with different observation windows under different scenarios. As shown in Table IV, the original ML-based STVS models will be ineffective under the adversarial examples, but the robust ML-based STVS models via the proposed mitigation strategy can still work and maintain the STVS assessment performance in terms of the four indices, i.e., achieving 96.48% and 94.99% under the white-box and black-box scenarios, respectively. Since this paper employs LSTM as the ML-based STVS model, we also display

TABLE IV
AVERAGE PERFORMANCE OF DIFFERENT ML MODELS WITH DIFFERENT OBSERVATION WINDOWS UNDER DIFFERENT SCENARIOS

Models		**Average STVS Assessment Performance**			
		Accuracy	**Precision**	**Specificity**	**F1-Score**
Original Models with Clean Samples		97.94%	98.19%	98.20%	97.93%
Original Models	**White-Box Adversarial Attacks**	6.21%	6.29%	6.12%	6.29%
	Black-Box Adversarial Attacks	16.03%	16.43%	15.42%	16.53%
Robust Models	**White-Box Adversarial Attacks**	96.48%	96.70%	96.71%	96.47%
	Black-Box Adversarial Attacks	94.99%	95.47%	95.52%	94.97%

the detailed information of the original and robust LSTM models for the plot of the confusion matrix in Figs. 10(a–e), where the columns represent the ground truth label and the rows represent the predicted label. Overall, we would like to emphasize that the STVS assessment performance in Table IV and Figs. 10(a–e) is for adversarial attacks, which can be a very promising performance.

VII. CONCLUSION

In this paper, we firstly demonstrate the vulnerability of the ML-based data-driven power system stability assessment model under adversarial examples. An adversarial example generation strategy is proposed for both the white-box and the black-box attack scenarios. Analysis results reveal the threat of the adversarial examples which can significantly reduce the assessment accuracy. Then, an empirical robustness evaluation process is proposed to assess the robustness of instances and ML-based models against adversarial attack via quantifying the distance between samples and separating hyperplane. Finally, an adversarial training-based mitigation strategy is designed to defense the adversarial examples under the white-box and the black-box attack scenarios. To the best of our knowledge, similar works have not been systematically studied in the literature, the proposed adversarial training-based mitigation strategy and empirical robustness evaluation can be a very promising method to measure and improve the robustness of other similar ML-based model in power engineering.

REFERENCES

[1] Z. Y. Dong, Y, Xu, P. Zhang, and K. P. Wong, "Using intelligent system to assess an electric power system real-time stability," *IEEE Intell. Syst.*, vol. 28, no. 4, pp. 60–66, Jul./Aug. 2013.

[2] C. Ren, Y. Xu, Y. Zhang, and R. Zhang, "A hybrid randomized learning system for temporal-adaptive voltage stability assessment of power systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3675–3684, Jun. 2020.

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

[4] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, "Power system transient stability assessment based on big data and the core vector machine," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2561–2570, Sep. 2016.

[5] H. Mohammadi, G. Khademi, M. Dehghani, and D. Simon, "Voltage stability assessment using multi-objective biogeography-based subset selection," *Int. J. Electr. Power Energy Syst.*, vol. 103, pp. 525–536, Dec. 2018.

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

[7] H. Supreme, L.-A. Dessaint, I. Kamwa, and A. Heniche-Oussédik, "Development of new predictors based on the concept of center of power for transient and dynamic instability detection," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3605–3615, Jul. 2018.

[8] C. Liu and C. L. Bak, "An accurate online dynamic security assessment scheme based on random forest," *Energies*, vol. 11, no. 7, p. 1914, 2018.

[9] I. Konstantelos *et al.*, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1417–1426, May 2017.

[10] J. L. Cremer, I. Konstantelos, and G. Strbac, "From optimization-based machine learning to interpretable security rules for operation," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3826–3836, Sep. 2019

[11] Y. Zhang, Y. Xu, Z. Y. Dong, Z. Xu, and K. P. Wong, "Intelligent early warning of power system dynamic insecurity risk: Toward optimal accuracy-earliness tradeoff," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2544–2554, Oct. 2017.

[12] 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. Informat.*, vol. 15, no. 1, pp. 74–84, Jan. 2019.

[13] S. M. Ashraf, A. Gupta, D. K. Choudhary, and S. Chakrabarti, "Voltage stability monitoring of power systems using reduced network and artificial neural network," *Int. J. Electr. Power Energy Syst.*, vol. 87, pp. 43–51, May 2017.

[14] 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. Electr. Power Energy Syst.*, vol. 107, no. 12, pp. 379–394, 2019.

[15] 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.

[16] 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.

[17] J.-M. H. Arteaga, F. Hancharou, F. Thams, and S. Chatzivasileiadis, "Deep learning for power system security assessment," in *Proc. 13th IEEE PowerTech*, 2019, pp. 1–6.

[18] S. Wu, L. Zheng, W. Hu, R. Yu, and B. Liu, "Improved deep belief network and model interpretation method for power system transient stability assessment," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 1, pp. 27–37, Jan. 2020.

[19] 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.

[20] L. Zheng, W. Hu, K. Hou, X. Xu, and G. Shao, "Real-time transient stability assessment based on deep recurrent neural network," in *Proc. IEEE Innov. Smart Grid Technol.*, 2017, pp. 1–5.

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

[22] 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

[23] C. Ren and Y. Xu, "Transfer learning-based power system online dynamic security assessment: Using one model to assess many unlearned faults," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 821–824, Jan. 2020.

[24] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[25] 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.

[26] V. Ajjarapu, *Computational Techniques for Voltage Stability Assessment and Control*. Boston, MA, USA: Springer, 2007. AQ2

[27] W. Du, Z. Chen, H. F. Wang, and R. Dun, "Feasibility of online collaborative voltage stability control of power systems," *IET Gener. Transm. Distrib.*, vol. 3, no. 2, pp. 216–224, 2009.

[28] 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.

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

[30] D. J. Shoup, J. J. Paserba, and C. W. 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.*, vol. 2, 2004, pp. 1140–1147, doi: 10.1109/PSCE.2004.1397688.

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[32] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2019, pp. 2574–2582.

[33] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, p. 35, Jan. 2016.

[34] F. Tramér, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*.

[35] *Transient Security Assessment Tool User Manual*, Powertech Labs, Inc., Surrey, BC, Canada, 2013.

[36] *PSS®E 33.0 Program Application Guide: Volume-II*, Siemens Power Technol. Int., Schenectady, NY, USA, Mar. 2013.

[37] H. Renmu, M. Jin, and D. J. Hill, "Composite load modeling via measurement approach," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 663–672, May 2006.

[38] R. Zhang, Y. Xu, and Z. Y. Dong, "Measurement-based dynamic load modelling using time-domain simulation and parallel-evolutionary search," *IET Gener. Trans. Distrib.*, vol. 10, no. 15, pp. 3893–3900, Nov. 2016.

[39] Y. Zhang, Y. Xu, R. Zhang, and Z. Y. Dong, "A missing-data tolerant method for data-driven short-term voltage stability assessment of power systems," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5663–5674, Sep. 2019.

[40] D. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.