Robustness Verification for Machine Learning-based Power System Dynamic Security Assessment Models under Adversarial Examples

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

***Abstract*—Based on machine learning (ML) technique, the data-driven power system dynamic security assessment (DSA) has received significant research interests. Yet, the well-trained ML-based models with high training and testing accuracy may be vulnerable to the adversarial example, which is a modified version of the original sample that is intentionally perturbed but retains very close to the original one. Such adversarial examples can mislead the DSA results and lead to catastrophic consequences. Thus, the accuracy index alone is not enough to represent the performance of the ML-based DSA models. To evaluate the ML-based DSA models and provide formal robustness guarantee for real-time DSA, this paper proposes an adversarial robustness verification method to quantify the ability of ML-based DSA models against all kinds of adversarial examples. A model-free and attack-independent robust index is defined for both differentiable and non-differentiable attack scenarios. Simulation results have verified the effectiveness of the proposed adversarial robustness verification method and the superiority of robust index compared with the upper bound of the adversarial perturbations computed by existing adversarial attack methods.**

***Index Terms*—Adversarial example, adversarial robustness verification, data-driven, power system dynamic security assessment, machine learning, robust index.**

I. Introduction

POWER system dynamic security assessment (DSA) is to assess the stability of an electric power system, i.e., for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact [1]. The loss of dynamic security can lead to catastrophic consequences such as cascading failure and even wide-spread blackout. Therefore, maintaining power system dynamic security has long been an essential requirement for secure and continuous electricity supply to the customers.

The popular DSA methods include time-domain simulation (TDS), direct methods, and data-driven methods. The TDS needs an iterative calculation to solve the large set of different algebraic equations [2], which suffers from high computation burden. For a real-world power system with thousands of buses, it could take TDS several hours to complete its simulations on all the postulated disturbances. With such high computation burden, TDS is not difficult for online application. For the direct methods, they aim to accelerate the stability assessment by simplifying the highly nonlinear stability problem into an energy-type function, Lyapunov function or algebraic equations. They include transfer energy function-based methods, extended equal area criterion (EEAC), and single machine equivalent paradigms. However, their calculations are based on simplified modelling of multi-machine power systems, which suffer from conservative and/or approximated results [3], [4]. Another limit is that these methods only focus on transient stability assessment, but lack assessment ability on other stability problems, such as voltage and frequency stability, that are increasingly prominent in the smart grid [5].

Machine learning (ML)-based data-driven models have been identified as a promising approach to achieve real-time DSA of electric power grids [6]. The principle of the data-driven DSA method is that: at the offline stage, an ML-based model is trained by a dynamic security database with the strategically selected features; at the online stage, with the real-time measurements, such trained ML model can instantaneously deliver the DSA results. Compared with existing analytical methods (e.g., time-domain simulation and direct methods), it is advantageous for its much faster assessment speed, less data requirement, and stronger generalization capability, etc. For pre-fault DSA, the input of the ML-based DSA model is power generation, load demand, bus voltage, branch flow, etc. and the output is the corresponding security status or degree. Traditional ML algorithms such as artificial neural network, support vector machine (SVM), decision tree (DT), extreme learning machine, etc. have been successfully used in the literature [7].

More recently, with the continuous advancement of artificial intelligence research, deep learning technique has shown higher accuracy performance, such as deep belief networks (DBN), convolutional neural network (CNN), and recurrent neural network (RNN), etc. Based on their deep structures, the learning model tends to be more accurate [8].

In the literature, DBN is applied for transient voltage stability assessment with the extracted features using kernel principal component analysis [9]. The work in [10] uses a local linear interpreter to constraint the DBN but cannot give a detailed explanation for the whole DSA model. The work in [11] uses a twin convolutional SVM network to predict transient stability status, which can mine the internal structure of time-varying features. In [12], a CNN-based ensemble model is used to train a transient stability predictor, which can be updated rapidly

The work in this paper was supported in part by Ministry of Education (MOE), Republic of Singapore, under grant AcRF TIER 1 2019-T1-001-069 (RG75/19).

C. Ren is with Interdisciplinary Graduate School, Nanyang Technological University, Singapore (email: renc0003@e.ntu.edu.sg).

Y. Xu is an associate professor at School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (email: eeyanxu@gmail.com).

with the informative and representative samples before the operating conditions or topologies change greatly. In [13], a hierarchical and self-adaptive CNN-based model is designed to determine the post-disturbance transient stability of the system via integrated decision-making rule for multiple CNNs. In [14], the cascaded CNNs combining with TDS can improve the computational efficiency for pre-fault transient stability assessment via extracting features from different TDS time intervals. In [15], power system snapshots are represented as the images, hence can be directly applied CNN to predict the stability status. RNNs can use their internal state to process variable length sequences of inputs, which considers spatial and temporal correlations, such as long short-term memory (LSTM) and gated recurrent units (GRU). In [16], a LSTM-based method is used to judge the transient stability status with multiple time-step algebraic variables. In [17], a LSTM based self-adaptive learning model is applied for online transient stability assessment, which can extract both spatial and temporal dependency from the feature inputs. GRU-based methods in [18] and [19] are applied for real-time transient instability prediction, which are robust to measurement noise and topology changes. Other latest DL technique, such as generative adversarial networks (GAN) [20], is applied for online DSA with incomplete PMU measurements.

While the above data-driven DSA models have achieved excellent accuracy performance, their *robustness* and *security* have not been well examined [21]. Due to some practical issues, according to IEEE C37.118 standard [22], the total communication errors and noise manipulation of the received PMU measurements can be 1%. Besides, through the false data injection [23], [24], or even with cyber-attack by adversarial attack algorithms [25], [26], the ML-based DSA model may be vulnerable. 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 [27]. As a result, the adversarial examples can mislead the ML-based DSA models to the wrong/inaccurate DSA results, which cause a severe consequence. E.g., if an ML-based DSA model is perturbed, the unstable status is predicted to be stable and therefore the system would miss the preventive 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 preventive control, leading to unnecessary costs and customer interruption. It is clear that *a high DSA accuracy is not equal to high robustness against such adversarial examples*.

In practice, several defensive methods against adversarial examples have been shown either partially or completely broken after stronger adversarial attack algorithms are designed [28]. Thus, there is a pressing need to provide an attack-agnostic robustness evaluation index. Evaluating the robustness ability of the ML model can be done via generating adversarial examples with the specific adversarial attack algorithm [29], [30]. Taking [29] as an example, a linear programming (LP) formulation is proposed to find adversarial examples, which uses the perturbations as the robustness metric. They find that the LP formulation can find adversarial examples with smaller perturbations than other gradient-based adversarial attacks. However, these methodologies have a common shortcoming as the resilience of the ML model to existing attacks is not ensured to be extended to other adversarial attacks. Existing robustness verification methods aim to solve the exact minimal adversarial perturbations (independent of adversarial attack algorithms), but it is very hard to find a non-trivial lower bound for minimal adversarial perturbations due to expensive cost and suffering from computational burden [31]. In order to solve such problem, the work in [32] found that Lipschitz constant can be used to compute the lower bound of "safe region" to explain the robustness issue, the principle is that Lipschitz constant bounds the change of output with respect to small input perturbations, but the global Lipschitz constant often provides a very loose bound. Recently, the works in [33] and [34] have further improved the robustness lower bound using a local Lipschitz continuous condition and derived the closed-form bound. However, these robustness verification methods only work for differentiable scenario, which are not generalized.

To accurately quantify the robustness of an ML-based model under adversarial perturbations on safety-critical power system DSA problem, this paper proposes an adversarial robustness verification method for data-driven DSA models. Besides, a model-free and attack-independent robust index are defined to evaluate the ability of the ML-based DSA models against adversarial examples, which can be used to select the candidate model and provide formal robustness guarantee for DSA application in practice. Rigorous mathematic proof is provided for the robust index under both differentiable and non-differentiable scenarios.

II. Preliminaries

For pre-fault DSA, each instance $\mathbf{x} \in \mathbb{R}^m$ contains m input features to the DSA model, which can be power generation, load demand, and bus voltage magnitudes, etc.; the output y is the stability status under a given fault. To proceed, the following definitions are given below.

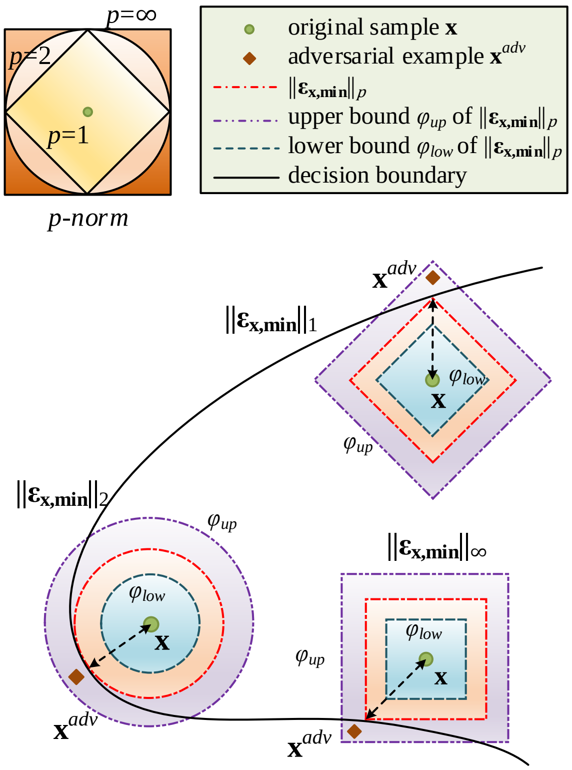
Definition 1 (ML-based DSA model): Given a DSA database, the ML-based DSA model aims to learn a function $f^{\theta}(\cdot)$ by an ML algorithm as (1), which maps the relationship from $\mathbf{x}$ to y with the model parameters θ.

$$f^{\theta}(\mathbf{x}) = f^{(l)}(..f^{(2)}(f^{(1)}(\mathbf{x}))) \tag{1}$$

where $\{f^{\theta}(\cdot)\big|\mathbb{R}^m \to \mathbb{R}^z\}$ denotes a z-class classifier, and $f^{(l)}$ represents the function of the l-th layer of the neural network. The training process of an ML-based DSA model is to minimize the difference between the predicted class c and the ground truth label y as (2).

$$\begin{gathered}\min_{\theta}\ L_{f^{\theta}}(c, y)\\ \text{where}\quad c = \arg\max_{1\le i\le z} f_i^{\theta}(\mathbf{x})\end{gathered} \tag{2}$$

where $L(\cdot,\cdot)$ is the pre-defined loss function of ML-based DSA model $f_{\theta}(\cdot)$. The gradient descent algorithms can be utilized to



p=∞
p=2
p=1
p-norm
original sample $\mathbf{x}$
adversarial example $\mathbf{x}^{adv}$
$\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_p$
upper bound φ_{up} of $\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_p$
lower bound φ_{low} of $\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_p$
decision boundary
$\mathbf{x}^{adv}$
$\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_1$
φ_{low}
$\mathbf{x}$
φ_{up}
$\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_2$
φ_{up}
φ_{low}
$\mathbf{x}$
$\mathbf{x}^{adv}$
$\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\|_\infty$
φ_{up}
φ_{low}
$\mathbf{x}$
$\mathbf{x}^{adv}$

Fig. 1. Illustration of the upper and lower bounds of minimal adversarial perturbations under different p-norm distance conditions. (p=1, 2, ∞)

update the model parameters.

Definition 2 (Adversarial example $\mathbf{x}^{adv}$ and adversarial attack): Given a trained classifier $f^{\theta}(\cdot)$ and a sample $\mathbf{x}$ with its ground truth label y, an adversarial example $\mathbf{x}^{adv}$ with the adversarial perturbation $\boldsymbol{\varepsilon}_{\mathbf{x}}$ can be generated via solving the optimization problem as (3):

$$\min_{\boldsymbol{\varepsilon}_{\mathbf{x}}} \|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p$$
$$\text{s.t.:} \begin{cases} \arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x}) = c = y \\ \arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x}^{adv}) = \arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) = t \\ t \neq c \end{cases} \tag{3}$$

where c and t represent the corresponding predicted 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 $\boldsymbol{\varepsilon}_{\mathbf{x}}$ with p-norm distance. Different p-norm have different properties: for $p = 1$ case, the sparse solution can be obtained up to extreme condition, i.e., only a single feature is tampered; for $p = 2$ case, it measures the Euclidean distance between the adversarial examples and the original samples, the solving process is more spread but may results in localized perturbations; for $p = \infty$ case, it denotes the maximum change among all dimensions in the adversarial examples, the adversarial perturbation is small and can influence all the features. Such optimization problem in (3) is non-convex and thus intractable.

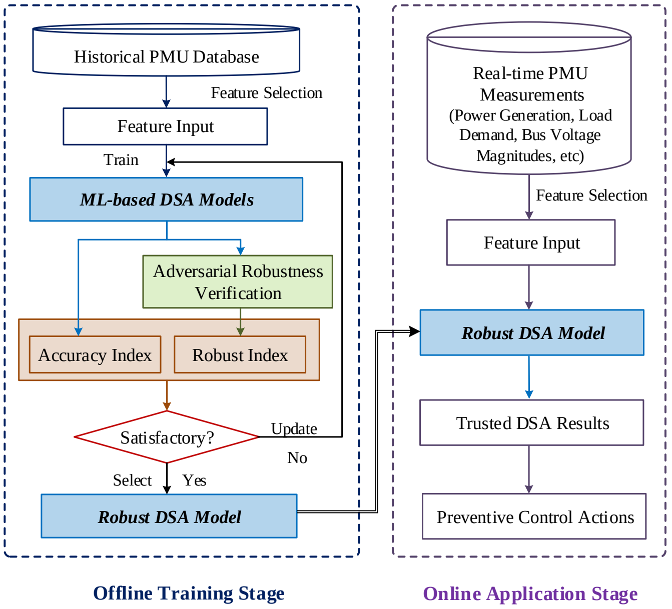


Fig. 2. Implementation of the proposed method.

In other words, $\mathbf{x}^{adv} = \mathbf{x} + \boldsymbol{\varepsilon}_{\mathbf{x}}$ is an adversarial example of original sample $\mathbf{x}$ if exists $\boldsymbol{\varepsilon}_{\mathbf{x}} \in \mathbb{R}^m$ with small p-norm $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p$ that changes $\arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x})$. The successful *untargeted* adversarial attack aims to find an adversarial example $\mathbf{x}^{adv}$ misleading $\arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x}^{adv}) \neq \arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x})$, while the successful *targeted* adversarial attack aims to find an adversarial example $\mathbf{x}^{adv}$ with the target class t, misleading $\arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x}^{adv}) = t \neq \arg\max\limits_{1\le i\le z} f_i^{\theta}(\mathbf{x})$.

Definition 3 (Minimal adversarial perturbation $\left\|\boldsymbol{\varepsilon}_{x,min}\right\|_p$): Given an original sample $\mathbf{x}$ with its trained classifier $f^{\theta}(\cdot)$, the minimal p-norm adversarial perturbation of $\mathbf{x}$, denoted by $\left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$, is considered as the smallest p-norm distortion $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p$ for all possible adversarial examples of the original sample $\mathbf{x}$, as shown in Fig. 1.

Definition 4 (Upper bound φ_{up} of minimal adversarial perturbation): Suppose $\left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$ represents the minimal adversarial perturbation of original sample $\mathbf{x}$. The upper bound of $\left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$, denoted by φ_{up} where $\varphi_{up} \ge \left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$, is defined that there exists an adversarial example for the original sample $\mathbf{x}$ with $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \ge \varphi_{up}$, as shown in Fig. 1.

Definition 5 (Lower bound φ_{low} of minimal adversarial perturbation): Suppose $\left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$ represents the minimal adversarial perturbation of original sample $\mathbf{x}$. The lower bound of $\left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$, denoted by φ_{Low} where $\varphi_{low} \le \left\|\boldsymbol{\varepsilon}_{\mathbf{x,min}}\right\|_p$, can guarantee that, for original sample $\mathbf{x}$, there does not exist any adversarial example within p-norm distortion $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \le \varphi_{low}$. In other words, the classifier is robust to any possible adversarial example within the lower bound φ_{Low}, as shown in Fig. 1.

III. Adversarial Robustness Verification

The framework of this work is shown in Fig. 2. The objective of the adversarial robustness verification is to firstly quantify the robustness of the candidate ML-based DSA models under all kinds of adversarial perturbations at the offline stage. Then, according to practical requirements, based on the robustness and accuracy performances of different ML-based DSA models, the most suitable ML-based DSA model can be selected for online DSA application. Besides, existing ML-based DSA model can also be updated if the robustness performance is not satisfactory. Since the proposed robust index considers both the differentiable and non-differentiable classification scenarios, the selected ML-based DSA method can provide an accurate and trusted DSA results.

A. Robust Index of ML-based DSA Models by The Bound of Minimal Adversarial Perturbations

Fig. 1 shows the upper and lower bound of minimal adversarial perturbations under different *p*-norm distances. The upper and lower bounds of minimal adversarial perturbation are attack-independent but are specific to the instance. φ_{up} can be easily acquired by using any adversarial attack method to find the adversarial example of original sample $\mathbf{x}$ (i.e., *Limited-memory Broyden–Fletcher–Goldfarb–Shanno* algorithm (L-BFGS) [32], iterative fast gradient method (I-FGM) [35], etc.). However, φ_{low} is not easy to obtain due to the different adversarial attack algorithms and *p*-norm distances. φ_{low} ensures that the classifier is robust to any perturbation with $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \le \varphi_{low}$, so it can be utilized as a robust index to certify the robustness of the ML-based DSA models.

B. Adversarial Robustness Verification with Robust Index

This section provides the tight mathematic proof for the proposed adversarial robustness verification under both the differentiable and non-differentiable scenarios. Adversarial robustness verification can be formulated via satisfying the lower bound φ_{low} of $h(\mathbf{x}^{adv})$ to be non-negative as follows:

$$h_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) = f_c(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) - f_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) \ge 0, \quad \forall\ \|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \le \varphi_{low} \tag{4}$$

where $h_t(\mathbf{x}^{adv}) = f_c(\mathbf{x}^{adv}) - f_t(\mathbf{x}^{adv})$ represents the margin function at $\mathbf{x}^{adv}$ for class t. When $h_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) = 0$, it means that an adversarial example $\mathbf{x}^{adv}$ of original sample $\mathbf{x}$ can be obtained. For the differentiable scenario, implicitly assuming that *Lipschitz Continuity* holds, *Hölder's Inequality* and *Mean Value Theorem* are applied to prove; another approach with a mild assumption only needs to use *Lipschitz Continuity and its relationship with gradient norm Lemma* in [36]; for the non-differentiable scenario, with *Backward Pass Differentiable Approximation* algorithm [28], the feasibility of the proposed adversarial robustness verification is proved. Next, the solving process of robust index φ_{low} under differentiable and non-differentiable scenarios are listed as below.

Theorem 1 (Differentiable scenario with tight assumption): For $\mathbf{x} \in \mathbb{R}^m$ and classifier $\{f(\cdot)|\mathbb{R}^m \to \mathbb{R}^z\}$ with continuously differentiable function, let $c = \arg\max_{1\le i\le z} f_i^{\theta}(\mathbf{x})$ be the predicted class of original sample $\mathbf{x}$. Define the hyper-ball with center $\mathbf{x}$ and radius r as $Ball_p(\mathbf{x}, r) = \{\hat{\mathbf{x}} \in \mathbb{R}^m | \|\hat{\mathbf{x}} - \mathbf{x}\|_p \le r\}$ and $\hat{\mathbf{x}}$ is over the fixed ball. For all $\boldsymbol{\varepsilon}_{\mathbf{x}} \in \mathbb{R}^m$ with:

$$\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \le \min\left\{\min_{t\ne c} \frac{f_c(\mathbf{x}) - f_t(\mathbf{x})}{\max\limits_{\hat{\mathbf{x}}\in Ball_p(\mathbf{x},r)} \|\nabla f_c(\hat{\mathbf{x}}) - \nabla f_t(\hat{\mathbf{x}})\|_q}, r\right\} := \varphi_{low} \tag{5}$$

it holds $c = \arg\max_{1\le i\le z} f_i^{\theta}(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$ with $\frac{1}{p}+\frac{1}{q} = 1$, $1 \le p, q \le \infty$, which means the predicted decision does not change within $Ball_p(\mathbf{x}, \varphi_{low})$. In other words, φ_{low} is the lower bound φ_{low} of the minimal adversarial perturbation $\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\|_p$.

Proof. Based on the theorem of calculus [37], $f_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$ is formalized as below:

$$f_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) = f_t(\mathbf{x}) + \int_0^1 \langle \nabla f_t(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}), \boldsymbol{\varepsilon}_{\mathbf{x}}\rangle\, dj, \quad \text{for } i = 1, \dots, z \tag{6}$$

Then, in order to achieve $h_t(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}}) \le 0$ as (4), (7) should hold as below:

$$\begin{aligned} f_c(\mathbf{x}) - f_t(\mathbf{x}) &\le \int_0^1 \langle \nabla f_t(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}) - \nabla f_c(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}), \boldsymbol{\varepsilon}_{\mathbf{x}}\rangle\ dj \\ &\le \|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \cdot \int_0^1 \|\nabla f_t(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}) - \nabla f_c(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}})\|_q\ dj \end{aligned} \tag{7-a}$$

where *Hölder's Inequality* is applied with the fact that the *q*-norm is dual to the *p*-norm, that is $\frac{1}{p}+\frac{1}{q} = 1$, $1 \le p, q \le \infty$. Then, $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p$, satisfying (4), holds as (7-b).

$$\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \ge \frac{f_c(\mathbf{x}) - f_t(\mathbf{x})}{\int_0^1 \|\nabla f_t(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}) - \nabla f_c(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}})\|_q\ dj} \tag{7-b}$$

In view of (7-b), the upper bound of the denominator is fixed over in $Ball_p(\mathbf{x}, r)$. Then, we can make assertion of adversarial perturbation $\boldsymbol{\varepsilon}_{\mathbf{x}} \in Ball_p(0, r)$ and the guaranteed upper bound is at most *r*, which holds for

$$\sup_{\boldsymbol{\varepsilon}_{\mathbf{x}}\in Ball_p(0,r)} \int_0^1 \|\nabla f_t(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}}) - \nabla f_c(\mathbf{x}+j\boldsymbol{\varepsilon}_{\mathbf{x}})\|_q\ dj \le \max_{\hat{\mathbf{x}}\in Ball_p(\mathbf{x},r)} \|\nabla f_t(\hat{\mathbf{x}}) - \nabla f_c(\hat{\mathbf{x}})\|_q \tag{8}$$

Then, the lower bound for the minimal norm of the adversarial perturbations as (9) which can change the classification result from class c to class t.

$$\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_p \ge \min\left\{\frac{f_c(\mathbf{x}) - f_t(\mathbf{x})}{\max\limits_{\hat{\mathbf{x}}\in Ball_p(\mathbf{x},r)} \|\nabla f_c(\hat{\mathbf{x}}) - \nabla f_t(\hat{\mathbf{x}})\|_q}, r\right\} \tag{9}$$

Considering the worst case for all $t \neq c$, the lower bound φ_{low} of the minimal adversarial perturbation $\left\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\right\|_p$ is obtained as (5), which holds for any fixed $r > 0$.

It is clear that the above mathematic proof is only available for differentiable scenario [34]. In fact, such adversarial perturbation verification can be considered as a local Lipschitz constant estimation problem for both the differentiable and non-differentiable scenarios, because the lower bound φ_{low} has relationship with the maximum norm of the local gradients with respect to the original sample. To efficiently compute the lower bound φ_{low} under both the differentiable and non-differentiable scenarios, *Extreme Value Theory* and *Backward Pass Differentiable Approximation* are applied, only with the mild assumption on *Lipschitz Continuity and its relationship with gradient norm as Lemma 1*.

Lemma 1 (Lipschitz Continuity and its relationship with gradient norm in* [36]*): Let $S \in \mathbb{R}^m$ be a convex bounded closed set and let $h(\mathbf{x})$: $S \to \mathbb{R}$ be a continuously differentiable function on an open set containing S. When the following inequality holds for any: $\mathbf{x}, \hat{\mathbf{x}} \in S$

$$|h(\mathbf{x}) - h(\hat{\mathbf{x}})| \leq L_q \cdot \|\mathbf{x} - \hat{\mathbf{x}}\|_p \tag{10}$$

then $h(\mathbf{x})$ belongs to a Lipschitz function with Lipschitz constant $L_q = \max\{\|\nabla h(\mathbf{x})\|_q : \mathbf{x} \in S\}$ with $\frac{1}{p} + \frac{1}{q} = 1$, $1 \leq p, q \leq \infty$, where the gradient of $h(\mathbf{x})$ is $\nabla h(\mathbf{x}) = \left(\frac{\partial h(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial h(\mathbf{x})}{\partial x_m}\right)^T$.

Based on Lemma 1, Theorem 2 can be proved.

Theorem 2 (Differentiable scenario with mild assumption): For $\mathbf{x} \in \mathbb{R}^m$ and classifier $\{f(\cdot)|\mathbb{R}^m \to \mathbb{R}^z\}$ with continuously differentiable function, let $c = \arg\max_{1\leq i\leq z} f_i^\theta(\mathbf{x})$ be the predicted class of original sample $\mathbf{x}$. For all $\boldsymbol{\varepsilon}_\mathbf{x} \in \mathbb{R}^m$ with:

$$\|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \leq \min_{t\neq c} \frac{f_c(\mathbf{x}) - f_t(\mathbf{x})}{L_q^t} := \varphi_{low} \tag{11}$$

it holds $c = \arg\max_{1\leq i\leq z} f_i^\theta(\mathbf{x} + \boldsymbol{\varepsilon}_\mathbf{x})$ with $\frac{1}{p} + \frac{1}{q} = 1$, $1 \leq p, q \leq \infty$, where L_q^t is the Lipschitz constant of the $h_t(\mathbf{x}) = f_c(\mathbf{x}) - f_t(\mathbf{x})$ in p-norm. In other words, φ_{low} is the lower bound φ_{low} of the minimal adversarial perturbation $\left\|\boldsymbol{\varepsilon}_{\mathbf{x},\mathbf{min}}\right\|_p$.

Proof. Based on the *Lipschitz continuity and its relationship with gradient norm Lemma* in [30], for $\mathbf{x} \in \mathbb{R}^m$ and classifier $\{f(\cdot)|\mathbb{R}^m \to \mathbb{R}^z\}$ with continuously differentiable function, the lower bound φ_{low} is guaranteed if (5) holds for any $\mathbf{x}$ with corresponding $\mathbf{x}^{adv}$:

$$|h_t(\mathbf{x} + \boldsymbol{\varepsilon}_\mathbf{x}) - h_t(\mathbf{x})| \leq L_q^t \cdot \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \tag{12}$$

Then (12) is expanded into (13).

$$h_t(\mathbf{x}) - L_q^t \cdot \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \leq h_t(\mathbf{x} + \boldsymbol{\varepsilon}_\mathbf{x}) \leq h_t(\mathbf{x}) + L_q^t \cdot \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \tag{13}$$

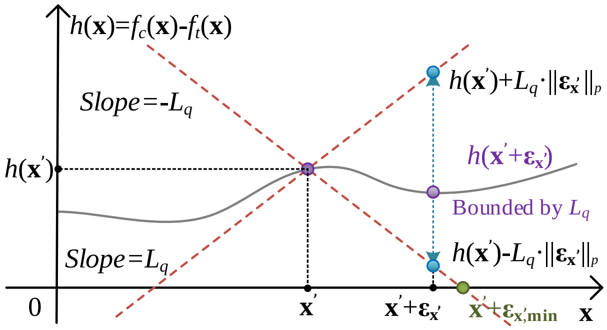

$h(\mathbf{x})$=$f_c(\mathbf{x})$-$f_t(\mathbf{x})$
Slope=-L_q
$h(\mathbf{x}')$+L_q·$\|\boldsymbol{\varepsilon}_{\mathbf{x}'}\|_p$
$h(\mathbf{x}')$
$h(\mathbf{x}'$+$\boldsymbol{\varepsilon}_{\mathbf{x}'})$
Bounded by L_q
Slope=L_q
$h(\mathbf{x}')$-L_q·$\|\boldsymbol{\varepsilon}_{\mathbf{x}'}\|_p$
0
$\mathbf{x}'$
$\mathbf{x}'$+$\boldsymbol{\varepsilon}_{\mathbf{x}'}$
$\mathbf{x}'$+$\boldsymbol{\varepsilon}_{\mathbf{x}',\mathbf{min}}$
$\mathbf{x}$

Fig. 3. Illustration of (13) to prove Theorem 2.

It can be seen that $h_t(\mathbf{x}) - L_q^t\|\boldsymbol{\varepsilon}_\mathbf{x}\|_p$ denotes the lower bound of $h_t(\mathbf{x} + \boldsymbol{\varepsilon}_\mathbf{x})$. As shown in Fig. 3, the value of $h_t(\cdot)$ around $\mathbf{x}$, (e.g., $h_t(\mathbf{x} + \boldsymbol{\varepsilon}_\mathbf{x})$), is bounded via $h_t(\mathbf{x})$, $\boldsymbol{\varepsilon}_\mathbf{x}$ and Lipschitz constant L_p, with slopes equal to $\pm L_p$. Based on the analysis of (13), if $\|\boldsymbol{\varepsilon}_\mathbf{x}\|_p$ is small enough as (14), which is equal to (11), no adversarial examples can be found.

$$h_t(\mathbf{x}) - L_q^t \cdot \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \geq 0 \tag{14}$$

Then (15) is obtained as follow:

$$\|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \leq \frac{h_t(\mathbf{x})}{L_q^t} = \frac{f_c(\mathbf{x}) - f_t(\mathbf{x})}{L_q^t} \tag{15}$$

Combining (12)-(15), the robust index φ_{low} for untargeted adversarial attack is formalized as (11). Based on (11), the prediction of classifier will never be misclassified and all of adversarial attacks will be invalid. A higher robust index φ_{low} indicates the better ML-based model robustness ability.

Corollary 1: Theorem 1 is the special case of the Theorem 2 by Lipschitz continuity assumption when requiring $f_t(\mathbf{x})$ to be continuously differentiable. Let L_q^t be local Lipschitz constant of $h_t(\mathbf{x}) = f_c(\mathbf{x}) - f_t(\mathbf{x})$ at $\mathbf{x}$ over the fixed $Ball_p(\mathbf{x}, r) = \{\hat{\mathbf{x}} \in \mathbb{R}^m | \|\hat{\mathbf{x}} - \mathbf{x}\|_p \leq r\}$ and let $\boldsymbol{\varepsilon}_\mathbf{x} \in Ball_p(\mathbf{0}, r)$, L_q^t is obtained as:

$$\begin{aligned} L_q^t &= \max_{\hat{\mathbf{x}} \in Ball_p(\mathbf{x},r)} \|\nabla h(\mathbf{x})\|_q \\ &= \max_{\hat{\mathbf{x}} \in Ball_p(\mathbf{x},r)} \|\nabla f_c(\hat{\mathbf{x}}) - \nabla f_t(\hat{\mathbf{x}})\|_q \end{aligned} \tag{16}$$

Then, Theorem 2 can be rewritten as Theorem 1.

Corollary 2: For a special case of non-differentiable classification functions with Rectified Linear Unit (ReLU), the proposed adversarial robustness verification method is also available by replacing the Lipschitz constant as the maximum norm of directional derivative. Define the neural network with ReLU as $g_\theta(\mathbf{x})$ in (17).

$$g_\theta(\mathbf{x}) = Relu(\theta^{(l)}(\ldots Relu(\theta^{(2)} Relu(\theta^{(1)}\mathbf{x})))) \tag{17}$$

where $\theta^{(l)}$ is the weight of the l-th layer of the neural network; $Relu(\cdot) = \max(0, \cdot)$. Here, the biases terms are omitted due to no effect on the gradient. Based on the maximum norm of the directional derivative, denote a univariate function $k(u) = g(\mathbf{x} + u\mathbf{v})$ via $g_\theta(\mathbf{x})$, where the unit vector $\mathbf{v}$ and unit distance d from $\mathbf{x}$ to corresponding $\mathbf{x}^{adv}$ as (18):

$$\mathbf{v} = \frac{\mathbf{x}^{adv} - \mathbf{x}}{\|\mathbf{x}^{adv} - \mathbf{x}\|_p}$$

$$\text{where } d = \|\mathbf{x}^{adv} - \mathbf{x}\|_p \tag{18}$$

Thus, $k(0) = g(\mathbf{x})$ and $k(d) = g(\mathbf{x} + d\mathbf{v}) = g(\mathbf{x}^{adv})$. Then the two side directional derivatives of $k(u)$ is calculated. Only if right-hand derivative $g'(\mathbf{x} + u\mathbf{v}; \mathbf{v})$ is equal to left-hand derivative $g'(\mathbf{x} + u\mathbf{v}; -\mathbf{v})$, inequality (19) holds [34].

$$k'(u) = g'(\mathbf{x} + u\mathbf{v}; \mathbf{v}) \le L_q^t \tag{19}$$

ReLU network only has limited points in $u \in (0, d)$ such that $k'(u)$ does not exist, since ReLU activation function can cause some discontinuous u. Based on the standard theorem of calculus and mean value theorem, there exists $\widetilde{u_i} \in (u_{i-1}, u_i)$ which holds for each u as (20).

$$\begin{aligned} k(d) - k(0) &\le \sum\nolimits_i |k(u_i) - k(u_{i-1})| \\ &\le \sum\nolimits_i |k'(\widetilde{u_i})(u_i - u_{i-1})| \\ &\le \sum\nolimits_i L_q^t \cdot |(u_i - u_{i-1})|_p \\ &= L_q^t \cdot \|\mathbf{x}^{adv} - \mathbf{x}\|_p \end{aligned} \tag{20}$$

where (20) is equal to (21) as below.

$$g(\mathbf{x}^{adv}) - g(\mathbf{x}) \le L_q^t \cdot \|\boldsymbol{\varepsilon}_\mathbf{x}\|_p \tag{21}$$

Therefore, Theorem 2 still holds for non-differentiable ReLU scenario with $L_q^t = \sup_\mathbf{x}\left\{\left|\sup_{\|\mathbf{v}\|_p=1} g'(\mathbf{x}; \mathbf{v})\right|\right\}$. Hence, the robust index φ_{low} in (11) is still valid for non-differentiable classifiers.

Since such robustness verification can be considered as a local Lipschitz constant estimation problem. Theorem 2 can be extended to non-differentiable scenario as Theorem 3 by applying *Backward Pass Differentiable Approximation.*

Theorem 3 (Non-differentiable scenario): For a neural network classifier $\{f(\cdot)|\mathbb{R}^m \to \mathbb{R}^z\}$ with non-differentiable function $g(\mathbf{x})$ to input $\mathbf{x} \in \mathbb{R}^m$, the function $f(g(\mathbf{x}))$ belongs to non-differentiable classifier. *Backward Pass Differentiable Approximation* algorithm usually holds that $g(\hat{\mathbf{x}}) \approx \hat{\mathbf{x}}$, when $g(\hat{\mathbf{x}})$ is non-differentiable. Thus, in backpropagation process, the shattered gradient of non-differentiable function is approximately replaced as follow:

$$\nabla_\mathbf{x} f(g(\mathbf{x}))|_{\mathbf{x}=\hat{\mathbf{x}}} \approx \nabla_\mathbf{x} f(\mathbf{x})|_{\mathbf{x}=g(\hat{\mathbf{x}})} \tag{22}$$

In this way, the approximated gradient is similar as the original model. By simply collecting $\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})$ as the gradient and calculating its norm as a sample for Lipschitz constant estimation, the similar gradient of the original model is estimated. Hence, the lower bound φ_{low} as Theorem 2 do not change too much and still holds in most cases of non-differentiable scenario.

Algorithm 1: Computing process for robust index φ_{low}

```
Input: For $\mathbf{x} \in \mathbb{R}^m$ and classifier $\{f(\cdot)|\mathbb{R}^m \to \mathbb{R}^z\}$ , $c = \arg\max_{1\le t\le z} f_\theta^t(\mathbf{x})$, the quantity of samples for each batch $n_s$, batch size $n_b$, perturbation norm $p$, maximum value of perturbation $r$.
Output: Robust index $\varphi_{low}$.
Initialize: Set $E \leftarrow \{\emptyset\}$, $h_t(\mathbf{x}) = f_c(\mathbf{x}) - f_t(\mathbf{x})$, $q = \frac{p}{p-1}$.
begin
  for i = 1 to $n_b$ do
    for j = 1 to $n_s$ do
      Randomly select the point $\mathbf{x}^{(i,j)}$ within the $Ball_p(\mathbf{x}, r)$.
      Calculate $e_{ij} \leftarrow \|\nabla h(\mathbf{x}^{(i,j)})\|_q$ via backpropagation.
    end for
    Collect set $E \leftarrow E \cup \{\max_j\{e_{ij}\}\}$.
  end for
  Calculate $\mu \leftarrow$ maximum likelihood estimation of location parameters for Reverse Weibull distribution on set $E$.
  Obtain Lipschitz constant $L_q^t$.
  Obtain robust index $\varphi_{low}$ as (11).
end
```

C. Lipschitz Constant Estimation via Extreme Value Theory

The adversarial robustness verification with the lower bound φ_{low} can be considered as Lipschitz constant estimation issue. Thus, based on above analysis in Theorem 2 and 3, if want to obtain the lower bound φ_{low}, Lipschitz constant L_q^t should be effectively computed. As shown in (16), L_q^t can be computed via backpropagation. However, this calculating process needs high computational burden, since it requires to obtain the maximum value of $\|\nabla h(\mathbf{x})\|_q$ in a ball region, which is infeasible for high-dimensional data.

The sampling-based method is applied to calculate Lipschitz constant L_q^t within a $Ball_p(\mathbf{x}, r)$ by getting the maximum value of $\|\nabla h(\mathbf{x})\|_q$. However, an accurate and feasible estimation of the maximum value of $\|\nabla h(\mathbf{x})\|_q$ needs a large number of samples. Thus, Extreme Value Theorem [38], which aims to guarantee the maximum value of random variables should only follow one of three extreme value distributions, is applied to estimate the maximum value of $\|\nabla h(\mathbf{x})\|_q$ with only a tractable number of samples.

Lemma 2 (Extreme Value Theorem/Fisher-Tippett-Gnedenko Theorem in* [38]*): If there exists a sequence of independent and identically distributed random variables with cumulative distribution function (CDF), F_Z^n. Suppose that there exist two sequences of real numbers (a_n, b_n), where $a_n > 0$ and $b_n \in \mathbb{R}$ such that the following limits converge to a non-degenerate distribution function $G(z) = \lim_{n\to\infty} F_Z^n(a_n z + b_n)$, then the limit distribution $G(z)$ belongs to either the Gumbel distribution, the Fréchet distribution or the Reverse Weibull distribution with corresponding CDF.

TABLE I
CONTINGENCY SET

Contingency ID	1	2	3	4	5	6	7	8	9	10
Fault Setting	Fault bus 1, trip 1-39	Fault bus 39, trip 1-39	Fault bus 3, trip 3-4	Fault bus 4, trip 3-4	Fault bus 14, trip 14-15	Fault bus 15, trip 14-15	Fault bus 15, trip 15-16	Fault bus 16, trip 15-16	Fault bus 16, trip 16-17	Fault bus 17, trip 16-17
No. of secure instances	4438	4429	4387	4375	4352	4473	4451	4482	4501	4486
No. of insecure instances	2689	2698	2740	2752	2775	2654	2676	2645	2626	2641

Corollary 3: Based on Lemma 2, when sampling the point in within a $Ball_p(\mathbf{x},r)$, $\|\nabla h(\mathbf{x})\|_q$ is considered as the random variable characterized by CDF as Z, the probability distribution of Z is discrete and its CDF is piecewise constant within at most $Region = \sum_{i=0}^{m} \binom{n_{hidden}}{i}$ pieces for m-dimensional space, where n_{hidden} represents the number of hidden nodes for a hidden layer. Without loss of generality, assuming there exist several distinct values for Z, and denote them in the increasing order as $\{re_1, \dots, re_{Region}\}$, the CDF of F_Z^n is formulated as below:

$$F_Z^n(re_i) = F_Z^n(re_{i-1}) + \frac{\mathbb{V}_m(Ball_p(\mathbf{x},r)) \cap \mathbb{V}_m(\{\mathbf{x}|\|\nabla h(\mathbf{x})\|_q = re_i\})}{\mathbb{V}_m(Ball_p(\mathbf{x},r))} \quad (23)$$

where $F_Z^n(re_0) = 0$, and $\mathbb{V}_m$ represents the volume in a m-dimensional space. Thus, for any neural networks, suppose there exist n_s samples over a $Ball_p(\mathbf{x},r)$ uniformly and independently in each batch with n_b batches, and denote $\{\|\nabla h(\mathbf{x})\|_q\}$ as a sequence of independent and identically distributed random variables. Then, $\|\nabla h(\mathbf{x})\|_q$ is computed and the maximum value of $\|\nabla h(\mathbf{x})\|_q$ of each batch should be collected. For Reverse Weibull distribution parameters as (24):

$$G(z;\mu,\sigma,\xi) = \begin{cases} \exp\left\{-\left(\frac{\mu - z}{\sigma}\right)^{\xi}\right\} & , if\ z < \mu \\ 1 & , if\ z \geq \mu \end{cases} \quad (24)$$

where $\mu \in \mathbb{R}, \sigma > 0, \xi > 0$ represent the location, scale and shape parameters of Reverse Weibull distribution, respectively. Thus, by performing the maximum likelihood estimation of Reverse Weibull distribution parameters, the estimated location parameter μ is utilized as Lipschitz constant L_q^t. The complete computing process is shown in Algorithm 1.

IV. SIMULATION RESULTS

The proposed method is tested on the New England 10-machine 39-bus system (shown in Fig. 4), which is a widely used benchmark testing system for stability analysis. The numerical simulation is conducted on a high-performance computer with an Intel Core i7 CPU of 3.3-GHz, 16-GB RAM and GPU with NVIDIA GeForce GTX 1060. TDS is implemented in Transient Security Assessment Tool (TSAT) software. The proposed method is implemented in the Python with Pytorch framework.

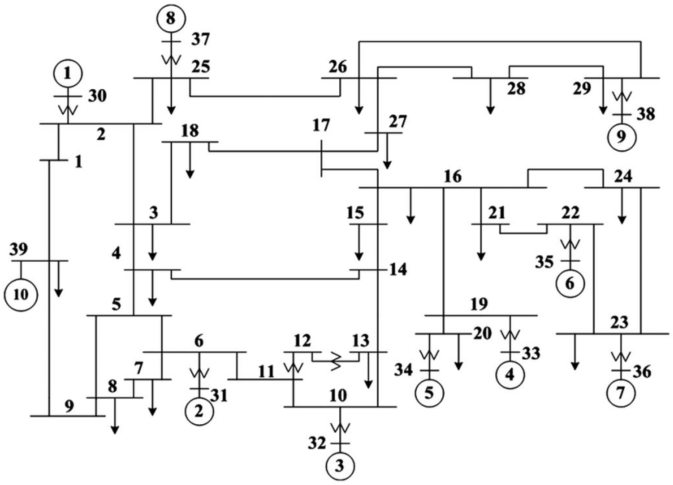


Fig. 4. New England 10-machine 39-bus system.

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

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

A. Database Generation

For the testing system, the operating points with their corresponding security conditions are generated via Monte-Carlo method [39], which is run to sample uncertain power variations under the forecasted load demand level for each bus. The contingencies are the three-phase faults with inter-area

TABLE III
UPPER BOUND CALCULATED BY ADVERSARIAL ATTACK AND THE PROPOSED ROBUST INDEX FOR FOUR DIFFERENT ML-BASED DSA MODELS UNDER THREE DIFFERENT *P*-NORM DISTANCES

ML-based DSA models	**Original DSA accuracy**	*1-norm distance*		*2-norm distance*		*∞-norm distance*	
		Upper bound φ_{up}	Robust index φ_{low}	Upper bound φ_{up}	Robust index φ_{low}	Upper bound φ_{up}	Robust index φ_{low}
FCNN	97.68%	0.237	0.122	0.132	0.084	0.044	0.023
LSTM	98.22%	0.168	0.103	0.123	0.071	0.028	0.021
BPNN	97.37%	0.142	0.094	0.094	0.062	0.028	0.019
ReLU network	97.45%	0.127	0.062	0.088	0.048	0.036	0.012

corridor trip and cleared 0.25 s after their occurrences. 10 typical three-phase faults at inter-area corridors are simulated on TSAT, listed in Table I, and all of the contingencies can also be considered depending on practical needs. In total, 320 relevant and commonly used operating variables (including, 136 bus features and 184 branch features) are selected as the primary features, listed in Table II. Then, given the contingency set, the security conditions subject to the selected contingency are also obtained by running a TDS using the TSAT. 7127 operating instances with their corresponding security conditions were obtained. The number of secure and insecure instances induced by each contingency are also listed in Table II. The average ratio between secure and insecure instances of different faults is around 3:2, which is reasonable to train the ML-based stability model. Based on the previous studies and experience, 5345 (75%) instances were randomly selected for model training, and the remaining 1782 (25%) instances are utilized for testing.

B. Testing Results

In this case study, given the DSA database, we select the three different *p*-norms (*p*=1, 2, ∞) that denote the level of changes for various dimension of the adversarial perturbation. To obtain the accurate robust index, radius *r* is set as a large value. In order to demonstrate the validity and feasibility of the proposed adversarial verification method with its robust index φ_{low} for ML-based DSA models, four different state-of-the-art ML algorithms, LSTM, fully convolutional neural network (FCNN), back-propagation neural network (BPNN) and ReLU based network are applied to calculate the upper bound φ_{up} and robust index φ_{low} under three *p*-norm adversarial perturbations. The results of upper bound φ_{low} computed by I-FGM attack are utilized to check how tight the proposed robust index is. Note that all the testing results (including the DSA accuracy, the upper bound and the proposed robust index) are the average performance values under different fault settings for the ten contingencies as Table I. All of the ML-based DSA models under comparison have been well trained and tuned for the best DSA accuracy performances. Also, other ML algorithms can also be considered if necessary.

The original DSA accuracy, the upper bound φ_{up} and the robust index φ_{low} of ML-based DSA models shown in Fig. 2 are compared in Table III. It can be seen that for all of the four ML-based DSA models, φ_{low} is always smaller than φ_{up}, which means the defined robust index is valid and tight. By comparing the four ML-based DSA models, the LSTM model has the highest DSA accuracy rate, but its upper bound φ_{up} and robust index φ_{low} are smaller than FCNN, which means it is less robust with respect to adversarial perturbations. Therefore, the accuracy alone is not enough to represent the overall performance of an ML-based DSA model considering the adversarial perturbation risks. By using the proposed robust index, the most robust ML-based model, FCNN model, can be selected from several different models. Besides, the proposed robust index is also utilized to check whether existing ML-based model need to be updated in order to improve the robustness ability against adversarial examples.

TABLE IV
DSA ACCURACY OF DIFFERENT MODELS WITH ∞-NORM ADVERSARIAL PERTURBATIONS

ML-based DSA models	DSA accuracy performance under adversarial attack with ∞-norm perturbation $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_\infty$			
	0.01	0.015	0.020	0.022
FCNN	97.62%	97.53%	97.33%	97.12%
LSTM	98.03%	97.72%	97.16%	92.44%
BPNN	97.13%	96.97%	92.76%	87.98%
ReLU network	97.07%	86.51%	80.73%	71.45%

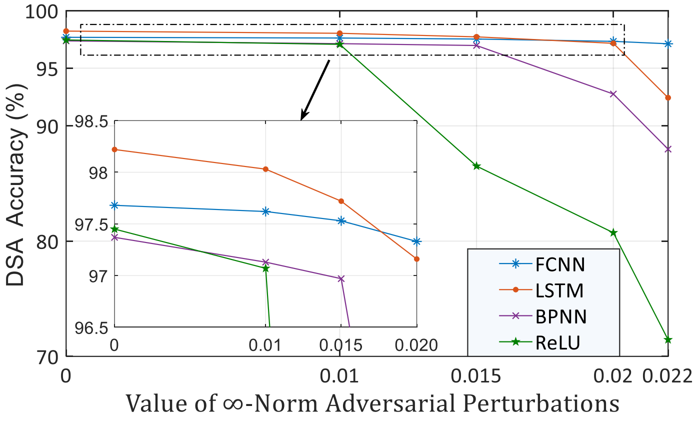


Fig. 5. DSA accuracy performance of four different ML-based DSA models under ∞-norm adversarial perturbations.

In order to further verify the effectiveness of the proposed robust index, the DSA accuracy of different ML-based DSA models under different ∞-norm adversarial perturbations are listed in Table IV and shown in Fig. 5. According to Fig. 5, initially, when the ∞-norm perturbation $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_{\infty}$ is small (0.01), the four ML-based DSA models do not significantly drop in DSA accuracy compared with the original DSA accuracy. However, with the increase of perturbations $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_{\infty}$, four ML-based DSA models show different accuracy degradation trend, especially when the perturbation $\|\boldsymbol{\varepsilon}_{\mathbf{x}}\|_{\infty}$ is larger than the corresponding robust index φ_{low} of different ML-based DSA models in Table III. As listed in Table III, the FCNN model can maintain a high DSA accuracy, which means it has a high robustness level, while the other three show a very fast DSA accuracy decrease which means they are not robust under large adversarial perturbations.

As for how to use the proposed robust index in practice, the users/system operators can design the suitable strategies according to the practical requirements. In this paper, we provide a basic strategy that is to select the more reliable ML-based DSA model for online use, which means to select the ML-based DSA model with the highest robust index value and the FCNN model in this case. Except for this strategy, the users/system operators can balance the DSA accuracy performance and robust index of the different ML-based DSA models. E.g., if the users/system operators think the DSA accuracy performance is more important than the robustness or such the ML-based DSA model will not be attacked, they can select the ML-based DSA model with the highest DSA accuracy with no need to consider the robust index; if the users/system operators think the robustness of the ML-based models is more important than the DSA accuracy performance, they can select the ML-based DSA model with the highest value of the robust index; if the users/system operators think both the DSA accuracy performance and robust index of the different ML-based DSA models are important, the users/system operators can balance the DSA accuracy performance and robust index via multi-objective optimization to show the Pareto Frontier.

V. Conclusions and Future Works

In this paper, an adversarial robustness verification method with a model-agnostic and attack-independent robust index is proposed to characterize the robustness properties of ML-based DSA models against adversarial examples, hence, to select the more reliable ML-based DSA models or judge whether need to be updated. The proposed method is based on the estimating the local Lipschitz constant of the ML-based DSA models. Such approximation is computed via observing $\boldsymbol{p}$-norm of the gradient at several random sampled points. Then, these observations can be utilized to calculate the statistical estimation of the local Lipschitz constant. For non-differentiable scenarios, backward pass differentiable approximation algorithm is used to approximately replace the shattered gradient. Besides, this paper provides the tight mathematical proof for the proposed robust index under both differentiable and non-differentiable scenarios. Simulation results have demonstrated its effectiveness by comparing with upper bound of adversarial perturbations computed by adversarial attacks.

Future works are discussed as follows:

1) As shown in the proof of Theorem 2, the robust index belongs to a first-order approximation. Thus, the robust index can be further derived to provide the second-order formal robustness guarantee for ML-based DSA models.
2) As mentioned in last paragraph of Section IV.C, the practical strategy to select the ML-based DSA model should be considered, such as balancing the DSA accuracy and robustness via multi-objective programming.
3) Besides, one important direction is that how to defense the adversarial attack based on the proposed robust index, the popular method is to make the adversarial training.
4) Finally, it is valuable to involve the extension of the proposed robustness verification method and the robust index to measure the robustness ability of other similar ML-based models for other safety-critical or reliability-sensitive applications in power engineering.

References

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

[2] M. Pavella, D. Ernst, and D. Ruiz-Vega, Transient Stability of Power Systems: A Unified Approach to Assessment and Control. Norwell, MA: Kluwer, 2012.

[3] Y. Xue, T. Van Cutsem and M. Pavella, “Extended equal area criterion justifications, generalizations, applications,” in *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 44–51, Feb. 1989.

[4] T. L. Vu and K. Turitsyn, "Lyapunov Functions Family Approach to Transient Stability Assessment," in *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1269-1277, Mar. 2016.

[5] M. Pavella and F. J. Evans, “Direct methods for studying dynamics of large-scale electric power systems – a survey,” *Automatica*, vol. 21, no. 1, pp. 1-21, 1985.

[6] Y. Xu, Y. Zhang, Z.Y. Dong, and R. Zhang, "Intelligent Systems for Stability Assessment and Control of Smart Power Grids: Security Analysis, Optimization, and Knowledge Discovery” CRC Press, 2020.

[7] L. Duchesne, et al., "Recent developments in machine learning for energy systems reliability management." *Proceedings of the IEEE*, 2020.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] Z. Guoli, et al, “Power system transient voltage stability assessment based on kernel principal component analysis and DBN,” in *International Conference of Artificial Intelligence, Medical Engineering, Education*. Springer, 2018, pp. 713–728.

[10] S. Wu, et al. "Improved Deep Belief Network and Model Interpretation Method for Power System Transient Stability Assessment," *Journal of Modern Power Systems and Clean Energy*, 2019.

[11] A. Bashiri Mosavi, A. Amiri and H. Hosseini, "A Learning Framework for Size and Type Independent Transient Stability Prediction of Power System Using Twin Convolutional Support Vector Machine," in *IEEE Access*, vol. 6, pp. 69937-69947, 2018.

[12] 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,” *International Journal of Electrical Power and Energy Systems*, vol. 107, no. December 2018, pp. 379–394, 2019.

[13] 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.

[14] R. Yan, G. Geng, Q. Jiang, and Y. Li, “Fast transient stability batch assessment using cascaded convolutional neural networks,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2802–2813, 2019.

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

[16] L. Zheng, W. Hu, et al, “Real-time transient stability assessment based on deep recurrent neural network,” *2017 IEEE Innovative Smart Grid*

Technologies-Asia: ISGT-Asia 2017, pp. 1–5, 2017.

[17] J. J. Yu, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, “Intelligent time-adaptive transient stability assessment system,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1049–1058, 2018.

[18] A. Gupta, G. Gurrala, and P. S. Sastry, “Instability prediction in power systems using recurrent neural networks,” *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1795–1801, 2017.

[19] M. Barati, “Faster than real-time prediction of disruptions in power grids using PMU: Gated recurrent unit approach,” *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference*, pp.1–5, 2019.

[20] C. Ren and Y. Xu, "A Fully Data-Driven Method Based on Generative Adversarial Networks for Power System Dynamic Security Assessment with Missing Data," in *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 5044-5052, Nov. 2019

[21] A. Kurakin, I. J. Goodfellow, S. Bengio. "Adversarial examples in the physical world.", *International Conference on Learning Representations (ICLR)*, 2017.

[22] IEEE Standard for Synchrophasors for Power Systems, IEEE Std. C37. 118-2005, 2005.

[23] G. Liang, J. Zhao, et al, "A review of false data injection attacks against modern power systems." *IEEE Transactions on Smart Grid,* vol. 8, no. 4, pp. 1630-1638, 2016.

[24] K. Manandhar, X. Cao, F. Hu and Y. Liu, "Detection of Faults and Attacks Including False Data Injection Attack in Smart Grid Using Kalman Filter," in *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370-379, Dec. 2014

[25] M. J. Khojasteh, A. Khina, M. Franceschetti and T. Javidi, "Learning-based Attacks in Cyber-Physical Systems," in *IEEE Transactions on Control of Network Systems*, doi: 10.1109/TCNS.2020.3028035.

[26] D. Deka, S. Backhaus and M. Chertkov, "Structure Learning in Power Distribution Networks," in *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1061-1074, Sept. 2018.

[27] X. Yuan, P. He, Q. Zhu, et al. "Adversarial examples: Attacks and defenses for deep learning. " *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805-2924, 2019.

[28] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in 35th *International Conference of Machine Learning (ICML)*, 2018.

[29] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, “Measuring neural net robustness with constraints,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[30] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *IEEE Symposium on Security and Privacy (SP*), 2017, pp. 39–57.

[31] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, “Reluplex: An efficient smt solver for verifying deep neural networks,” in *International Conference on Computer Aided Verification*, Springer, 2017.

[32] Szegedy C, Zaremba W, Sutskever I, et al. "Intriguing properties of neural networks", *International Conference on Learning Representations (ICLR)*, 2013.

[33] M. Hein, and A. Maksym. "Formal guarantees on the robustness of a classifier against adversarial manipulation." *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[34] T. Weng, H. Zhang, et al. "Evaluating the robustness of neural networks: An extreme value theory approach." *International Conference on Learning Representations (ICLR)*, 2018.

[35] Kurakin A, Goodfellow I, Bengio S. "Adversarial examples in the physical world." arXiv preprint arXiv:1607.02533, 2016.

[36] Paulavičius, Remigijus, and Julius Žilinskas., "Analysis of different norms and corresponding Lipschitz constants for global optimization." *Technological and Economic Development of Economy* vol.12, no.4, pp: 301-306, 2006.

[37] G. B. Thomas and R. L. Finney, *Calculus and Analytic Geometry*, 8th ed. Reading, MA: Addison-Wesley, 1996.

[38] Kotz, S., & Nadarajah, S. (2000). Extreme value distributions: theory and applications. World Scientific.

[39] Y. Xu, Z. Y. Dong, J. H. Zhao, "A reliable intelligent system for real-time dynamic security assessment of power systems," *IEEE Transactions on Power Systems,* vol. 27, no. 3, pp. 1253-1263, 2012.

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

Yan Xu (Senior Member, IEEE) received the B.E. and M.E degrees in electrical engineering from South China University of Technology, Guangzhou, China in 2008 and 2011, respectively, and the Ph.D. degree in electrical engineering from The University of Newcastle, NSW, Australia, in 2013, all in electrical engineering. He is currently an Associate Professor at School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), and a Cluster Director at Energy Research Institute @ NTU (ERI@N), Singapore. Previously, he held The University of Sydney Postdoctoral Fellowship in Australia. His research interests include power system stability and control, microgrid, and data-analytics for smart grid applications. Dr Xu is an Editor for IEEE TRANSACTIONS ON SMART GRID, IEEE TRANSACTIONS ON POWER SYSTEMS, CSEE Journal of Power and Energy Systems, and an Associate Editor for IET Generation, Transmission & Distribution.