Letter

Achieving 500X Acceleration for Adversarial Robustness Verification of Tree-Based Smart Grid Dynamic Security Assessment

Chao Ren, Chunran Zou, Zehui Xiong, Han Yu, Zhao-Yang Dong, and Niyato Dusit

Dear Editor,

This letter presents a novel and efficient adversarial robustness verification method for tree-based smart grid dynamic security assessment (DSA). Based on tree algorithms technique, the data-driven smart grid DSA has received significant research interests in recent years. However, the well-trained tree-based DSA models with high accuracy are always vulnerable caused by some physical noises or attacks, which can misclassify the DSA results. Only with the accuracy index is not enough to represent the performance of the tree-based DSA models. To provide formal robustness guarantee and select the trusted tree-based DSA models, this letter proposes an efficient adversarial robustness verification strategy with a sound robust index to quantify the ability of tree-based DSA models against any adversarial attack. Analysis results verifies the proposed strategy can achieve up to ~564X speedup.

Tree-based data-driven models have been identified as a promising approach to achieve real-time stability assessment of power grids [1]. With the real-time measurements, the well-trained tree-based model in the offline process can directly deliver the highly accurate stability assessment result [2]. However, due to some practical issues, such as false data injection, noise manipulation, communication error, or even with cyber-attack by adversarial attack algorithms [3]−[6], the tree-based models designed for power systems may be vulnerable to these scenarios and make the misclassification, which can verify that high accuracy is not equal to good robustness and tree-based models only with the high accuracy performances cannot guarantee to be valid all the time. Thus, the robustness and security of the tree-based models have become a severe concern [7].

This letter proposes an efficient adversarial robustness verification strategy to evaluate the ability of tree-based models to resist adversarial attack algorithms, and systematically analyzes the acceleration for ensemble trees. Besides, an attack-independent sound robust index is designed, which can provide formal robustness verification for safety-critical applications, such as DSA.

Related work: For DSA problem, given the fault database, the data-driven DSA models can be trained by various tree algorithms. The feature inputs to the DSA models are P/Q power generation, load demand, and bus voltage magnitudes; the output is the corresponding stability status [6]. Based on tree structures, tree algorithms can be divided into single decision tree and ensemble tree. For tree with ensemble learning, it consists of bagging and boosting methods. The bagging aims to train weak learners based on bootstrap sampling set in parallel, and such method includes random forest (RF) and extra tree (ET). The core idea of boosting is to promote weak learners to strong learners, and it includes adaptive boosting (AdaBoost), gradient boosting (GBDT) and extreme gradient boosting (XGBoost).

Notations and problem description: In order to provide formal robustness guarantee for the different tree-based DSA models (single or ensemble), an efficient adversarial robustness verification strategy with a precise robust index is proposed to quantify the ability of tree-based DSA models against any adversarial attack. For ease of notation, the proposed adversarial robustness verification tree-based (ARVT) strategy for single tree-based and ensemble tree-based DSA model are referred as ARVT-S and ARVT-E, respectively. The purpose of ARVT is to measure the distance from the original input to the closest box decision boundary with the high computational efficiency, especially for the ensemble tree-based DSA models. We firstly introduce ARVT-S how to exactly calculate for verifying the single tree-based model, then we convert robustness verification for ensemble tree-based model into the max-clique problem on a multipartite graph with bounded boxicity.

Adversarial perturbation and adversarial attack: Denote the database with n training instances $\mathrm{X} = \{\mathrm{x}_1,\ldots \mathrm{x}_n|\mathrm{x}_n \in \mathbb{R}^m\}$ as the input vector with m features, and the binary tree-based classification model $\{f(\cdot)|\mathbb{R}^m \rightarrow \{-1,1\}\}$, where the correct predicted label of x can be formalized as $f(\mathrm{x}) = y$. Given x and a well-trained tree-based model, the adversarial perturbation is defined as $\varepsilon_{\mathrm{x}} \in \mathbb{R}^m$, misleading to $f(\mathrm{x}+\varepsilon_{\mathrm{x}}) \neq y$. Then, adversarial attack algorithms aim to find the minimal adversarial perturbation as (1).

$$z = \min_{\varepsilon_{\mathrm{x}}} \|\varepsilon_{\mathrm{x}}\|_\infty \;\; \text{s.t.} \;\; f(\mathrm{x}+\varepsilon_{\mathrm{x}}) \neq y. \tag{1}$$

Note that directly solving (1) cannot ensure to achieve the minimal adversarial perturbation due to the non-convexity. Therefore, adversarial attack algorithms can only obtain the upper bound of z, which can not provide a sound safety guarantee, even if the attack fails to obtain the adversarial examples, it does not mean no adversarial example exists.

Adversarial robustness verification: It aims to determine whether exists the adversarial examples within a radius z fixed ball region around x, $Ball_\infty(\mathrm{x},z) = \{\hat{\mathrm{x}} \in \mathbb{R}^m|\|\hat{\mathrm{x}}-\mathrm{x}\|_\infty \leq z\}$. It can be seen to determine whether (2) is true.

$$f(\mathrm{x}+\varepsilon_{\mathrm{x}}) = y \;\; \forall \|\varepsilon_{\mathrm{x}}\|_\infty \leq z. \tag{2}$$

Note that (2) are designed to calculate the global optimal exact value, which can be regarded as the lower bound of z, implying that no adversarial example exists within $Ball(\mathrm{x},z)$. Through giving the exact "Yes/No" answer, a binary search can obtain the value of z. Hence, it can provide a sound safety guarantee solution against any adversarial attack.

Proposed ARVT-S method: Distinguish from neural network-based models, tree-based models are non-continuous step functions, so existing robustness verification for neural networks [4] are not suitable for tree-based models. Assuming that the single decision tree has n leaf nodes, for each given instance x with m features, starting from the root node, x will traverse several internal nodes until it reaches a leaf node. Each internal node i decides x to pass to the left or right child node via comparing the feature value and its threshold. Each leaf node has a value v_i, which indicates the predicted class label for a classification tree. The purpose of ARVT-S is to calculate a m-dimensional box for each leaf nodes such that any instance in this box will definitely exist in this leaf node. Based on the tree structure, the box of leaf node i is defined by the Cartesian product [8], formalized in (3).

$$B^i = (l_1^i, r_1^i] \times \cdots \times (l_m^i, r_m^i]. \tag{3}$$

Each B^i denotes the decision boundary of a leaf node as Fig. 1.

Theorem 1: Given an instance $\mathrm{x} \in \mathbb{R}^m$ and a box $B = (l_1, r_1] \times \cdots \times (l_m, r_m]$. The smallest ∞-norm distance from x to B can be calculated as

$$dist_\infty(B,\mathrm{x}) = \min_b \|b-\mathrm{x}\|_\infty \tag{4}$$

$$b = \begin{cases} r_j, & x_j \geq r_j \\ l_j, & x_j \leq l_j \\ x_j, & r_j > x_j > l_j. \end{cases} \tag{5}$$

Proof for Theorem 1: Given the constraint on b_j, the minimal dis-

Corresponding author: Chunran Zou.

Citation: C. Ren, C. Zou, Z. Xiong, H. Yu, Z.-Y. Dong, and N. Dusit, "Achieving 500X acceleration for adversarial robustness verification of tree-based smart grid dynamic security assessment," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 3, pp. 800–802, Mar. 2024.

C. Ren, H. Yu, Z.-Y. Dong, and N. Dusit are with Nanyang Technological University, Singapore 639798, Singapore (e-mail: renc0003@e.ntu.edu.sg; han.yu@ntu.edu.sg; zy.dong@ntu.edu.sg; dniyato@ntu.edu.sg).

C. Zou is with Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: zoucr@nuaa.edu.cn).

Z. Xiong is with Singapore University of Technology and Design, Singapore 487372, Singapore (e-mail: zehui_xiong@sutd.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.124053

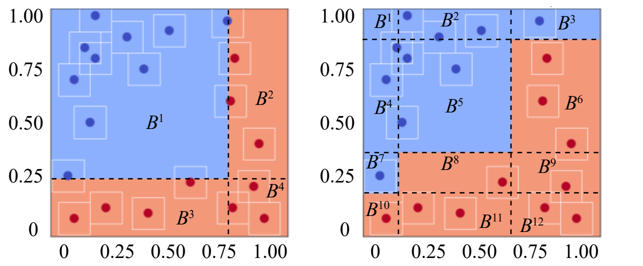

Fig. 1. Illustration of B^i for the tree-based models (Left: exist adversarial examples; Right; no adversarial example).

tance can be obtained in (4).

$$dist_\infty(B,\mathrm{x}) = \min_b \|b-\mathrm{x}\|_\infty = \min_b \sum_{j=1}^{m} \left|b_j - x_j\right| \tag{6}$$

$$\text{s.t. } r_j > b_j > l_j \ \forall j \in [m]. \tag{7}$$

■

Corollary 1: Denote $\cap$ as intersection between $Ball_\infty(\mathrm{x},z)$ and B, $B \cap Ball_\infty(\mathrm{x},z) \neq \emptyset$, if and only if $dist_\infty(B,\mathrm{x}) \leq z$.

In order to prove whether there are misclassification points under $\|\varepsilon_\mathrm{x}\|_\infty \leq z$, we can enumerate the boxes for all n leaf nodes and choose $dist_\infty(B^i,\mathrm{x})$ according to Theorem 1.

$$z = \min_{i:v_i \neq y} \left\| dist_\infty\left(B^i,\mathrm{x}\right)\right\|_\infty. \tag{8}$$

Noted that ARVT-S can be solved within $O(nm)$ time referred to [8].

Proposed ARVT-E method: For ensemble tree-based models, it is NP-complete for directly solving (2), which is intractable for ARVT-E. We propose an efficient strategy to solve ARVT-E by converting ARVT-E into the max-clique problem on a multi-partite graph with bounded boxicity. In order to judge whether a group of leaf nodes from G different single trees can form a G-size clique, it needs to check the intersections among the G different single tree leaf node boxes and the perturbation ball $Ball_\infty(\mathrm{x},z)$.

Assuming that the ensemble tree has G single trees, for each given instance x with m features, starting from the root node, x will traverse several internal nodes until it reaches a leaf node. Given the instance x, the ensemble tree will pass x to each of G single tree leaf nodes independently as i^g for all $g = 1,\ldots,G$, and each leaf node has a predicted label value v_{i^g} as Fig. 2.

Theorem 2: Given an instance $\mathrm{x} \in \mathbb{R}^m$ and G single trees with corresponding $B^{(g)} = \{B^1,B^2,\ldots,B^G\}$, if $B^p \cap B^q \neq \emptyset$ and $B^p \cap Ball_\infty(\mathrm{x},z) \neq \emptyset$ for all $p,q \in [g]$. Then, $\tilde{B} = B^1 \cap B^2 \cap \cdots \cap B^G \cap Ball_\infty(\mathrm{x},z)$ will also be a box and $\tilde{B} \neq \emptyset$.

Proof for Theorem 2: Assuming $l_1 \leq l_2 \leq \cdots \leq l_G$ without loss of generality, $(l_s,r_s] \cap (l_G,r_G] \neq \emptyset$ implies $l_G < r_s$ for each $s < G$. Then, $(l_G,\min(r_1,r_2,\cdots,r_G)]$ will be a nonempty set which is contained in $(l_1,r_1],(l_2,r_2],\cdots,(l_G,r_G]$. Thus, $(l_1,r_1] \cap (l_2,r_2] \cap \cdots \cap (l_G,r_G] \neq \emptyset$ and it is another interval. Hence, it can be generalized to $B^{(g)}$. Assuming $B^1,B^2,\ldots,B^G$ such that $B^p \cap B^q \neq \emptyset$ for all $p,q \in [g]$, for each m-dimensional boxes, the above can proof that $\tilde{B} = B^1 \cap B^2 \cap \cdots \cap B^G \cap Ball_\infty(\mathrm{x},z) \neq \emptyset$ and $\tilde{B}$ will also be another box. ■

Based on the boxicity property and Theorem 2, the intersection $\tilde{B}$ can be represented as G-size cliques in a graph K=(V, E) where $\mathrm{V} = \{i | B^i \cap Ball_\infty(\mathrm{x},z) \neq \emptyset\}$ and $\mathrm{E} = \{(p,q) \in [g| B^p \cap B^q \neq \emptyset\}$. In this graph, nodes in each layer indicate the leaf nodes of each single tree, and they don not include the empty intersection with $Ball_\infty(\mathrm{x},z)$. The edge (p,q) exists between node p and node q when their boxes intersect. Such graph belongs to G-partite graph because all edges connect the different single trees, and this graph belongs to G-maximum clique graph and each clique represents a reachable output for x after perturbation. Thus, we can accurately solve ARVT-E by using the maximum clique searching algorithm in G-partite graph.

Algorithm 1 (maximum *G*-cliques searching algorithm on a *G*-partite graph): For a G-partite graph, we firstly check connectivity of any first two parts $\mathrm{V}^{(1)}$ and $\mathrm{V}^{(2)}$ to find 2 cliques and enumerates all 2-cliques. Then, all the found 2-cliques are converted into the new nodes (possible as Theorem 2) and replace $\mathrm{V}^{(1)}$ and $\mathrm{V}^{(2)}$ as $\mathrm{V}^{(2)\prime}$. After that, continue to find 2-cliques between $\mathrm{V}^{(2)\prime}$ and $\mathrm{V}^{(3)}$. Repeat above steps until reach the final tree layer G. Based on the boxicity property, all the G-cliques C on a G-partite graph can be obtained with the vertices form. We show a basic enumeration process in Fig. 3.

With all the obtained G-cliques C on a G-partite graph, ARVT-E can be solved via verifying their prediction class label values to calculate the robust index (verification bound).

Corollary 2: The basic ARVT-E strategy aims to verify if the sign of the summation can be flipped with $Ball_\infty(\mathrm{x},z)$ for all G-cliques C, we can calculate $\tilde{v}$ as (9).

$$\tilde{v} = \begin{cases} \max\limits_{c}\left\{\operatorname{sign}\left(\sum_G v_{i,c}^{(g)}\right)\right\}, & y=-1 \\ \min\limits_{c}\left\{\operatorname{sign}\left(\sum_G v_{i,c}^{(g)}\right)\right\}, & y=1. \end{cases} \tag{9}$$

However, such strategy is sensitive to the number of cliques and the number of ensemble trees. By extension, the improved ARVT-E strategy is to calculate the limit of $\tilde{v}$ as (10).

$$limit(\tilde{v}) = \begin{cases} \operatorname{sign}\left(\sum_G^{|\mathrm{V}|} \max\limits_{i\in \mathrm{V}^{(g)}} v_{i,c}^{(g)}\right), & y=-1 \\ \operatorname{sign}\left(\sum_G^{|\mathrm{V}|} \min\limits_{i\in \mathrm{V}^{(g)}} v_{i,c}^{(g)}\right), & y=1. \end{cases} \tag{10}$$

It can be seen that such limit of $\tilde{v}$ indicate the upper bound for "−1" label and lower bound for "1" label, since all the G-cliques are still considered if adding more edges to the graph, even it becomes a fully G-partite graph. Thus, ARVT-E will not be particularly affected via the number of trees and cliques, which can provide the sound and efficient solution for (2). Noted that ARVT-E can be solved within $\min\{O(n^G),O((Gn)^m)\}$ time, referred to [9].

Based on Corollarys 1 and 2, combining the binary search, we can compute the value of z in (2) via ARVT-S and ARVT-E. The average value of z with overall n instances can be considered as the robust index for tree-based models (RIT).

$$RIT(f(\cdot)) = \frac{1}{n}\sum_n z_n. \tag{11}$$

A higher RIT typically indicates the better robustness of tree-based model, since it is equal to have a larger decision boundary.

Simulation results: The proposed method was implemented using the MATLAB programming language, and the experimental evaluations were conducted on a computer with the following specifications: An Intel(R) Xeon(R) W-2133 CPU with a clock speed of 3.6-GHz, 16-GB RAM, and a GPU with NVIDIA GeForce RTX 3070. The analysis process is on the New England 10-machine 39-bus system to validate the performance. In the database generation process, massive operating points are generated by randomly sampling generation and load within a certain range based on Monte-Carlo method. The detailed database description can refer to [6]. Eight different faults are studied which are the three-phase faults with inter-area corridor trip. Transient stability criterion is utilized to label the instances, where 60% of samples are randomly sampled for training and the remaining 40% serve as testing data for each fault.

In this case study, the maximum number of nodes in a clique is set as 2 that can ensure all G trees are enumerated; the maximum number of binary searches for finding the largest z is set as 10 that the proposed ARVT can be verified. Verification error is the upper bound of errors under any attack, indicating that no attack can achieve more than a certain percentage error on the testing set within $\|\varepsilon_\mathrm{x}\|_\infty = 0.5$. If only want to get verification errors at a certain $\|\varepsilon_\mathrm{x}\|_\infty$, just need to disable binary search via setting the maximum number of binary searches as 1 to be computed. To demonstrate the validity of the proposed ARVT with RIT for tree-based DSA models, we applied six state-of-the-art tree algorithms (DT, RF, ET, GBDT, XGBoost and AdaBoost) to compare the DSA accuracy, RIT, computational efficiency and verification error as Table 1. Among them, single DT is considered as the baseline, the number of ensemble tree is set as 200 and 1000. In order to guarantee the best DSA performance and fair comparison, all the tree-based DSA models have been well-trained and the hyper-parameters have been fine-tuned to avoid underfitting and overfitting.

It can be seen that ensemble tree-based models always have the better DSA accuracy than single DT. Although bagging ensemble methods have the better DSA accuracy than boosting ensemble methods, the RIT and verification errors of bagging ensemble methods are worse than boosting ensemble methods. It can proof that the accuracy index is not enough to represent the performance of the tree-based DSA models. Besides, it is clear that the computational efficiency of ARVT strategy can achieve up to ~564X speedup com-

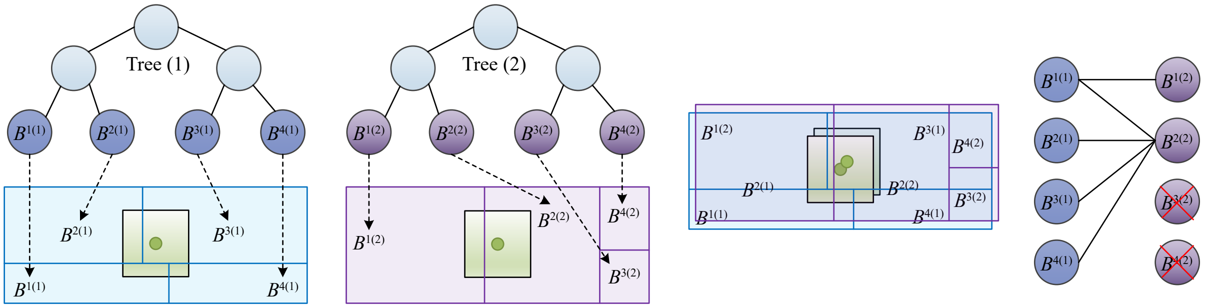

Fig. 2. Illustration of ARVT-S. Combine the different boxes of different single trees, and then convert them into the graph layer form.

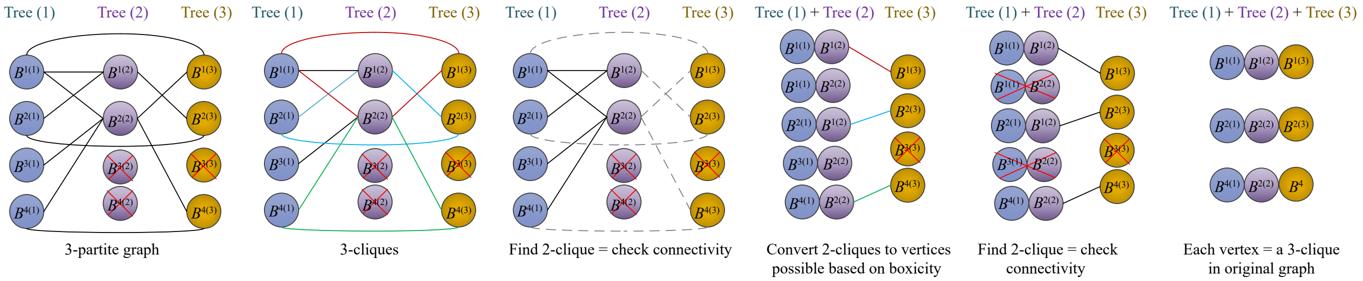
Tree (1) Tree (2) Tree (3)
3-partite graph
3-cliques
Find 2-clique = check connectivity
Tree (1) + Tree (2) Tree (3)
Convert 2-cliques to vertices possible based on boxicity
Find 2-clique = check connectivity
Tree (1) + Tree (2) + Tree (3)
Each vertex = a 3-clique in original graph

Fig. 3. Illustration of Algorithm 1: Maximum *G*-cliques searching algorithm on a *G*-partite graph. (A simple example of 3-partite graph to find all the 3-cliques).

Table 1. ARVT Performance of Different Tree-Based Methods

Tree-based DSA models		No. of trees	DSA accuracy	RIT	Computational efficiency		Verification error at 0.5
					ARVT time	Speedup	
Single	DT	1	97.03%	0.542	18.79 ms	base	48.2%
Ensemble (Bagging)	RF	200	98.54%	0.608	24.98 ms	**150X**	39.1%
		1000	98.83%	0.633	51.15 ms	**367X**	35.2%
	ET	200	97.99%	0.602	25.45 ms	**148X**	39.6%
		1000	98.79%	0.628	58.76 ms	**320X**	35.9%
Ensemble (Boosting)	GBDT	200	97.94%	0.766	20.12 ms	**187X**	17.9%
		1000	98.48%	0.799	33.32 ms	**564X**	15.3%
	XGBoost	200	98.22%	0.796	25.78 ms	**146X**	16.1%
		1000	98.69%	0.832	37.12 ms	**506X**	13.3%
	AdaBoost	200	97.85%	0.788	25.15 ms	**149X**	19.4%
		1000	97.89%	0.806	35.32 ms	**532X**	16.2%

pared with linear sum of several baseline. With the increasing of the number of ensemble trees, Table 1 shows the different degree of increase in RIT and speedup and decline in verification errors.

Conclusions: In this letter, an ARVT strategy is proposed to characterize the robustness of both single and ensemble tree-based DSA models against any adversarial attack, hence, to select the more trusted tree-based DSA models. Analysis results have demonstrated ARVT strategy can achieve up to ~500X speed up. To the best of our knowledge, similar works have not been reported in the literature, and the proposed ARVT can also provide the formal robustness verification for other safety-critical data-driven problems in power engineering.

Acknowledgments: This work was supported in part by the Internal Talent Award with Wallenberg-NTU Presidential Postdoctoral Fellowship 2022, the National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Program (AISG2-RP-2020-019), the Joint SDU-NTU Centre for AI Research (C-FAIR), the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund, Singapore (A20G8b0102), and NOE Tier 1 Projects (RG59/22 & RT9/22).

References

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

[2] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.

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

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

[5] Z. Zhang and D. K. Y. Yau, "CoRE: Constrained robustness evaluation of machine learning-based stability assessment for power systems," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 2, pp. 557–559, 2023.

[6] C. Ren and Y. Xu, "A universal defense strategy for data-driven power system stability assessment models under adversarial examples," *IEEE Internet of Things J.*, vol. 10, no. 9, pp. 7568–7576, May 2023.

[7] M. Andriushchenko and M. Hein, "Provably robust boosted decision stumps and trees against adversarial attacks," in *Proc. Advances in Neural Inform. Processing Systems*, 2019, pp. 12997–13008.

[8] Y. Wang, H. Zhang, H. Chen, *et al.*, "On LP-norm robustness of ensemble decision stumps and trees," in *Proc. Int. Conf. Machine Learning*, 2020, pp. 10104–10114.

[9] H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh, "Robustness verification of tree-based models," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 12317–12328.