2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 979-8-3503-1855-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SmartGridComm60555.2024.10738112

An Adversarial Approach to Evaluating the Robustness of Event Identification Models

Obai Bahwal *Member, IEEE,* Oliver Kosut *Senior Member, IEEE*, and Lalitha Sankar *Senior Member, IEEE*
School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, USA
{obahwal,lalithasankar,okosut}@asu.edu

***Abstract*—Intelligent machine learning approaches are finding active use for event detection and identification that allow real-time situational awareness. Yet, such machine learning algorithms have been shown to be susceptible to adversarial attacks on the incoming telemetry data. This paper considers a physics-based modal decomposition method to extract features for event classification and focuses on interpretable classifiers including logistic regression and gradient boosting to distinguish two types of events: load loss and generation loss. The resulting classifiers are then tested against an adversarial algorithm to evaluate their robustness. The adversarial attack is tested in two settings: the white box setting, wherein the attacker knows exactly the classification model; and the gray box setting, wherein the attacker has access to historical data from the same network as was used to train the classifier, but does not know the classification model. Thorough experiments on the synthetic South Carolina 500-bus system highlight that a relatively simpler model such as logistic regression is more susceptible to adversarial attacks than gradient boosting.**

***Index Terms*—Event identification, machine learning, mode decomposition, grid security, adversarial attacks, robustness.**

I. Introduction

With the increasing need for real-time monitoring of the grid dynamics, machine learning (ML) algorithms are providing viable and highly accurate solutions that support the system operator's requirement in making informed and timely decisions for reliable and safe operation of the system. In particular, such algorithms are invaluable for leveraging high-fidelity synchrophasor data (obtained using phasor measurement units (PMUs)) in real-time for accurate event detection [1], [2]. However, PMUs have been shown to be susceptible to adversarial attacks [3], [4] which in turn can lead to erroneous outcomes from the learned ML models.

In [5], the authors evaluate false data injection (FDI) attacks on ML-based state estimation models that rely on supervisory control and data acquisition (SCADA) network. The authors use data poisoning and gradient-based attacks as the threat models and show that such attacks are very successful in causing the state estimator to fail. More recently, [6] evaluates white box adversarial attacks against event classification models based on deep neural networks. Those models utilize time-series PMU measurements to classify between 'no events', 'voltage-related', 'frequency-related', or 'oscillation-related' events.

The work of Kosut and Sankar is funded in part by the NSF EPCN-2246658 and a DoE BIRD grant. Bahwal is funded by a grant from the Saudi Arabian Cultural Mission in the United States.

In contrast to the above-mentioned recent results, *we focus on real-time event identification using PMU data and physics-based modal decomposition methods along with interpretable ML models*. Our event identification framework leverages the approach in [7] and involves two steps: (i) extract features using physics-based modal decomposition methods; (ii) use such features to learn logistic regression (LR) and gradient boosting (GB) models for event classification. *Our primary goal is to design an algorithmic approach that generates adversarial examples to evaluate the robustness of this physics-based event classification framework*. We evaluate our attack algorithm in two distinct settings: white box and gray box. In the white box setup, we assume that the attacker has full knowledge of the classification framework including the classification model (i.e., knows both (i) and (ii) detailed above), and can only tamper with a subset of PMUs. On the other hand, for the gray box setup, we assume that the attacker does not know the ML classifier used by the system operator or the data that was used for training; however, the attacker has knowledge of the aspect (i) of the framework, has access to historical data from the same network, and can tamper with a subset of PMUs. In either setting, the attack algorithm perturbs event features in the direction of the classifier's gradient until the event is incorrectly classified. Using detailed event-inclusive PSS/E generated synthetic data for the 500-bus South Carolina system, we show that both types of attacks can significantly reduce the accuracy of the event classification framework presented in [7].

II. Setup

We first describe the event identification framework, introduced in [7], and the two classification models we consider.

A. Event Identification Framework

We focus on identifying two classes of events: generation loss (GL) and load loss (LL), denoted by the set $\mathcal{E} \in \{\text{GL, LL}\}$. These events are measured using M PMUs, each of which has access to three channels, namely, voltage magnitude, voltage angle, and frequency, indexed via the set $\mathcal{C} = \{V_m, V_a, F\}$. For a given event in $\mathcal{E}$ and channel $c \in \mathcal{C}$, the collected time-series data from M PMUs yields

a matrix $x^c \in \mathbb{R}^{M\times N}$, where N is the length of the sample window. Thus, for a given event, the data collected is given by $x = [[x^{V_m}]^T, [x^{V_a}]^T, [x^F]^T]^T \in \mathbb{R}^{|C|M\times N}$, where T denotes transpose of a matrix/vector.

In order to evaluate the robustness of this event identification framework, we follow the same feature extraction technique as in [7] by assuming that the system dynamics can be captured by using modal decomposition to extract a small number p of dominant modes that represent the interacting dynamics of power systems during an event. We refer the reader to [8], [9] for more details on modal decomposition used in this context. These dynamic modes are defined by their frequency, damping ratio, and residual coefficients that comprise the presence of each mode in a given PMU [10], [11]. The mode decomposition model is:

$$x_i^c(n) = \sum_{k=1}^{p} R_{k,i}^c \times (Z_k^c)^n + \epsilon_i^c(n), \quad i \in \{1,\dots,M\}, \quad c \in C \tag{1}$$

where $x_i^c(n)$ is the time-series signal for the i^{th} PMU and channel $c \in C$, $R_{k,i}^c$ is the residue for the k^{th} mode and i^{th} PMU, $Z_k^c = \exp(\lambda_k^c T_s)$ is the k^{th} event mode with $\lambda_k^c = \sigma_k^c \pm j\omega_k^c$ and T_s is the sampling period, and $\epsilon_i^c(n)$ is noise. The mode λ_k^c, defined by σ_k^c and ω_k^c, representing the damping ratio and angular frequency of the k^{th} mode, respectively. The residue $R_{k,i}^c$ is denoted by its magnitude $|R_{k,i}^c|$ and angle $\theta_{k,i}^c$. The dynamic response to an event is captured by a subset of the system PMUs ($M' < M$) which are chosen based on the highest PMUs' signal energy for a given channel and event. Finally, by extracting the values described above for a given channel and event, we define the feature vector as

$$X = \left[\{\omega_k^c\}_{k=1}^{p'}, \{\sigma_k^c\}_{k=1}^{p'}, \{|R_{k,i}^c|\}_{k=1}^{p'}, \{\theta_{k,i}^c\}_{k=1}^{p'}\right]_{i\in\{1,\dots,M'\}, c\in C} \tag{2}$$

Here, we select only the first $p' = p/2$ modes, since typically modes are composed of complex conjugate pairs; by choosing the first p' modes, we keep only one of each conjugate pair.

To compose the overall dataset, we assign event class labels as $y_i = -1$ and $y_i = 1$ for LL and GL, respectively. Taking such pairs of event features and their labels, we define the overall dataset as $\mathbf{D} = \{\mathbf{X}_D, \mathbf{Y}_D\}$ where $\mathbf{X}_D = [X_1, ..., X_{n_D}]^T \in \mathbb{R}^{n_D\times d}$, $\mathbf{Y}_D = [y_1, ..., y_{n_D}] \in \mathbb{R}^{n_D}$, and n_D is the total number of events from both classes.

B. Classification Models

We use logistic regression (LR) and gradient boosting (GB) classification models as the ML models for the evaluation of the framework and design of adversarial attacks. For LR, classification requires computing the probability of event y_i as

$$P(\mathbf{Y} = y_i | X_i, w) = \frac{1}{1 + \exp(-y_i w^T X_i)} \tag{3}$$

where w is the separating hyperplane between the two classes that would minimize the average classification error over the training data. The optimum estimator is obtained by minimizing the logistic loss as:

$$w_{\text{LR}} = \arg\min_w \sum_{i=1}^{n} \log(1 + \exp(-y_i X_i^T w)). \tag{4}$$

Gradient boosting is an ensemble learning algorithm which builds on weak learners, that in our case are decision stumps (single level decision trees thresholded on one feature), each based on a single feature. GB models are trained with an iterative greedy approach which minimizes error of each new weak learner by fitting to the residual error made by the previous learned predictors [12]. The output of the GB model is

$$F(X) = \sum_{m=1}^{d'} \text{dt}_m(X), \quad \text{where } \text{dt}_m(X) = \begin{cases} v_{1m}, & X_{j_m} \le \text{th}_m \\ v_{2m}, & X_{j_m} > \text{th}_m. \end{cases} \tag{5}$$

where dt_m is the m^{th} decision tree with its regression output being v_{1m} or v_{2m} and thresholding the j_m feature at th_m. The final GB classifier is obtained by mapping $F(X)$ to the $[0, 1]$ range using a sigmoid function and thresholding at 0.5.

For the purposes of this work, we consider only the two classification models described above. We focus on these two models since they are interpretable, fairly simple, and amenable to the same threat models (see the next section), while including two different levels of complexity.

III. Threat Models

In order to evaluate the vulnerability of the event identification framework, we consider two settings: (i) white box; and (ii) gray box. In the white box attack setting, we assume the following: (a) the attacker has full knowledge of the event identification framework, (b) access to all measurements and their corresponding ground truth event label *but* with restricted ability to only tamper with a subset of PMUs, and (c) knowledge of the ML classifier used by the system operator, including all the parameters of the classifier learned by the operator.

In the gray box attack setting, while assumptions (a) and (b) on the adversarial capabilities still hold, we now assume that the attacker does not know the classification model used by the system operator, but has access to historical data that is not necessarily the same as that used to train the classifier. In either case, our attack algorithm is designed to spoof a specific classifier: in the white box setting, this classifier is the true classifier used by the operator; in the gray box setting, it is a different classifier trained on the adversary's own data.

A third possible threat model would be a *black box attack*, in which the attacker does not even know the event identification framework. However, this would entail establishing an entirely separate event identification framework to use for the attacker, which is beyond the scope of this work.

A. Targeted Adversarial Example Generation

Algorithm 1 (illustrated in Fig. 1) describes how we generate adversarial PMU data. The algorithm utilizes the

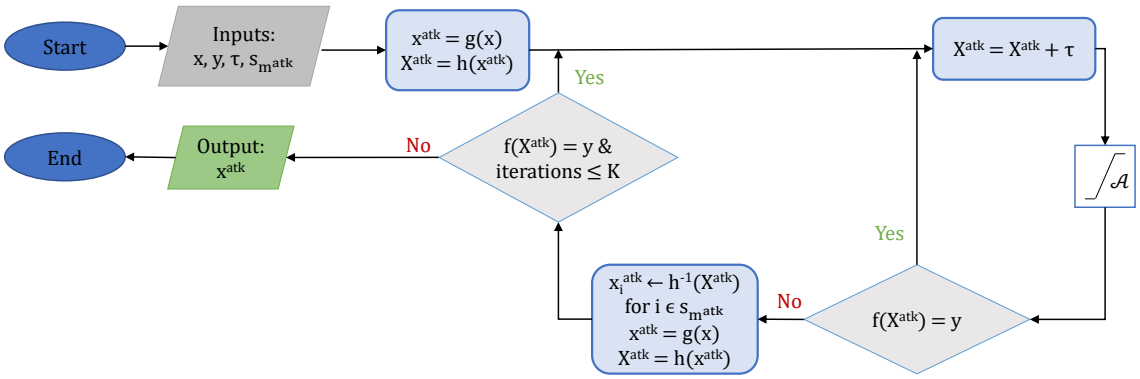

Fig. 1: Attack algorithm using perturbation vector τ on subset of targeted PMUs $S_{M^{\text{atk}}}$ to generate adversarial PMU data x^{atk}. The function g is a signal energy boosting function, h is a modal decomposition conversion function to extract features, f is the classifier, and $\mathcal{A}$ is the feasible set of feature values.

knowledge of classification models to perturb an incoming feature vector such that the direction of the perturbation is chosen to point towards the negative gradient of the classifier. The tampered vector of features is then reconstructed to obtain a time domain signal for each of the PMUs tampered by the attacker which then replace the original measurements at these PMUs. The resulting collated tampered and untampered event data across all PMUs is passed through the learned classifier for reclassification. This entire procedure is repeated until the classifier fails to classify correctly or when a maximum number of iterations K is reached.

We explain the steps of Algorithm 1 as follows. The function h represents the the transform function described in Section II-A, and f is the classification model used by the attacker (in the white box setting, this is the same as that at the control center; in the gray box setting, it is different). First, we check whether $f(h(x)) = y$; that is, whether the event is classified correctly. If not, there is no need to attack it. Next, we start with the untampered time domain data x and boost it so that the PMUs controlled by the adversary are present in the feature vector; this step is represented by the function g, which outputs the initial time domain attack vector x^{atk}. In particular, recall from Section II-A that only for the M' PMUs with highest energy are the modal residues kept in the feature vector X. To ensure that the PMUs controlled by the attacker, denoted by the set $\mathcal{S}^{M_{\text{atk}}}$, are among these M' PMUs, their energy is boosted by applying $x_i^{\text{atk}} \leftarrow \lambda x_i^{\text{atk}}$ iteratively for all $i \in \mathcal{S}^{M_{\text{atk}}}$, where $\lambda > 1$, until the set $\mathcal{S}^{M_{\text{atk}}}$ is included in the set of M' PMUs kept in the feature vector.

The perturbation vector τ, which is designed based on the classification model f and is meant to be a vector such that changing the feature vector in the direction of τ will cause the event to be misclassified. The precise details of designing τ are described in the next subsection. The feature vector X^{atk} is extracted from x^{atk} and perturbed by τ until $f(X^{\text{atk}}) = y'$ where y' is the incorrect event class label. To ensure that the tampered signal remains within reasonable bounds, the feature classes are restricted to lie within a feasible set $\mathcal{A}$, defined as

$$\mathcal{A} = \{X : |\omega_k^c| \le \upsilon_\omega,\ \sigma_k^c > \upsilon_\sigma,\ \upsilon_{|R|,min} \le |R_{k,i}^c| \le \upsilon_{|R|,max}, \text{ for all } k, c, i\}. \tag{6}$$

where υ_ω, υ_σ, and $\upsilon_{|R|,min}$ and $\upsilon_{|R|,max}$ are the bounds for frequency mode, damping mode, and residual amplitude features. Note that θ is not restricted since any numerical value of it will be equivalent to a value in $[-\pi, \pi]$ when performing the modal analysis transformation but allows a larger set of feasible and relevant attacks. After perturbing X^{atk} by τ, it is projected onto $\mathcal{A}$ to ensure it is feasible.

Once the event features are misclassified in the inner loop, the time domain signals for the compromised PMUs are boosted before replacing the original signal replaces the original signal. The resulting tampered time-domain signal is denoted by x_i^{atk}, where $x_i^{\text{atk}} \leftarrow h^{-1}(X^{\text{atk}}, i)$, and h^{-1} denotes the inverse of feature extraction transform that recovers the time domain signal for the i^{th} PMU (given by (1) without the noise term). After reconstructing x_i^{atk}, those time-domain signals are once again boosted via function g. Since the feature vector is related to all PMUs, but the attacker can only control a subset, the resulting time-domain attack vector x^{atk} will not exactly match the feature vector X^{atk}. Thus, X^{atk} is recomputed using feature extraction, and the loop repeats.

B. Designing τ

We describe how to find the perturbation vector τ used to design attacks in Algorithm 1 based on the classifier $f \in \{f_{LR}, f_{GB}\}$. For the LR classifier, we designate the separating hyperplane by its weight vector w^{LR}, as in (3). Thus, we can misclassify an event by perturbing its values towards the hyperplane. To realize this, we let $\tau = -y_i \eta w^{LR}$

for event i, where $\eta \in \mathbb{R}$ is a step size chosen sufficiently small to avoid perturbing event features too much.

For GB, recall that the classifier is composed of a sum of d' decision trees, given by equation (5). The m^{th} decision tree dt_m is applied to the feature j_m and is described by its two values v_{1m}, v_{2m} and the threshold th_m. A crude approximation of the gradient of GB model can be written as:

$$w_m^{GB} = v_{1m} - v_{2m} \tag{7}$$

where w_m^{GB} defines the weight and direction of the approximated gradient from dt_m. Thus, if w_m^{GB} is positive, we increase the value of the j_m^{th} feature if $y_i = 1$ and decrease it if $y_i = -1$. (Vice versa if w_m^{GB} is negative.) By doing so, we are forcing regression trees to output the less favorable value, leading to misclassification. In cases where multiple decision trees act on the same feature, and potentially have opposite signs of w_m^{GB}, the magnitude of w_m^{GB} for a given decision tree will signify its importance on the overall output of GB classification. Now we define the perturbation vector τ as the d-dimensional vector whose j^{th} entry, for $j = 1, \ldots, d$, is

$$\tau_j = \eta\, y_i \sum_{m:\, j_m = j} w_m^{GB}. \tag{8}$$

By (8), If the same feature is used in multiple trees, then this feature will be adjusted in proportion to the tree importance described in (7).

Algorithm 1 Targeted Adversarial Example Generation

Input: x, y: untampered PMU data and true label
f: Classification model
h: Feature extraction transform
g: Signal energy boosting function
τ: Perturbation vector
$\mathcal{A}$: Feasible feature set
$\mathcal{S}^{M_{\text{atk}}}$: Set of PMUs controlled by attacker
If: $f(h(x)) = y$ **do**
Initialize: $x^{\text{atk}} \leftarrow g(x, \mathcal{S}^{M_{\text{atk}}})$
$X_{\text{atk}} \leftarrow h(x^{\text{atk}})$
while $f(X^{\text{atk}}) = y$ and iterations $\leq K$ **do**
  while $f(X^{\text{atk}}) = y$ **do**
    $X^{\text{atk}} \leftarrow X^{\text{atk}} + \tau$
    Project X^{atk} into $\mathcal{A}$
  end while
  for all $i \in \mathcal{S}^{M_{\text{atk}}}$ **do**
    $x_i^{\text{atk}} \leftarrow h^{-1}(X^{\text{atk}}, i)$
  end for
  $x^{\text{atk}} \leftarrow g(x, \mathcal{S}^{M_{\text{atk}}})$
  $X^{\text{atk}} \leftarrow h(x^{\text{atk}})$
end while
Return: x^{atk}

IV. Numerical Results

A. Dataset

The synthetic South Carolina 500-bus grid, consisting of 90 generators, 466 branches, and 206 loads [13], is used to generate synthetic generation loss and load loss events. A dynamic model of the system on PSS/E is used to generate event data by running dynamic simulations for 11 seconds at a sampling rate of 30Hz. The event is applied after 1 second to ensure the system has reached steady-state. Data is collected from PMUs distributed on the largest $M = 95$ generator and load buses of the network (largest in terms of net generation or load). The GL events are generated by disconnecting the largest 50 generators, one per simulation run. For each such generator, 15 different loading scenarios are considered where the overall system loading varies between 90% to 100% of the net load. This is done by varying each individual load in the system randomly within its operational limits. Through this process, we obtain a total of 750 GL events. We create the LL events in a similar manner (i.e., disconnecting the largest 75 loads, one at a time, at 10 different loading scenarios varying between 90% to 100%). Thus the complete dataset has a total of $n_D = 1500$ event samples collected from voltage magnitude, voltage angle, and frequency channels of $M = 95$ PMUs.

In order to train the ML classification models, the dataset is split into three sets: 20% testing set and training sets for LR and GB each consisting of 40% of the dataset. Each set is assured to be nearly balanced across the two classes of events.

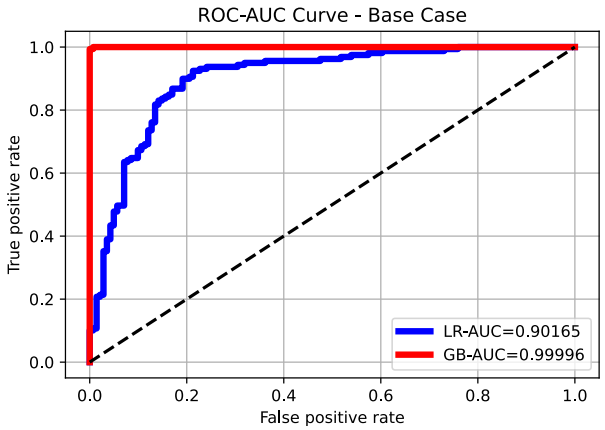

Fig. 2: Base case (untampered) performance of LR and GB classification models evaluated on the testing set.

B. Evaluation of Base Cases

Figure 2 shows the base case performance of both models. Note that the LR and GB classifiers are trained on their respective untampered training set and evaluated on untampered testing set. The resulting test accuracy is shown in the figure. To this end, we use receiver operating characteristic area under the curve (ROC-AUC) as the accuracy metric to evaluate the performance of the base and tampered models. The base models are able to identify unseen data with high accuracy with the GB model approaching 100% accuracy and surpassing the performance of LR.

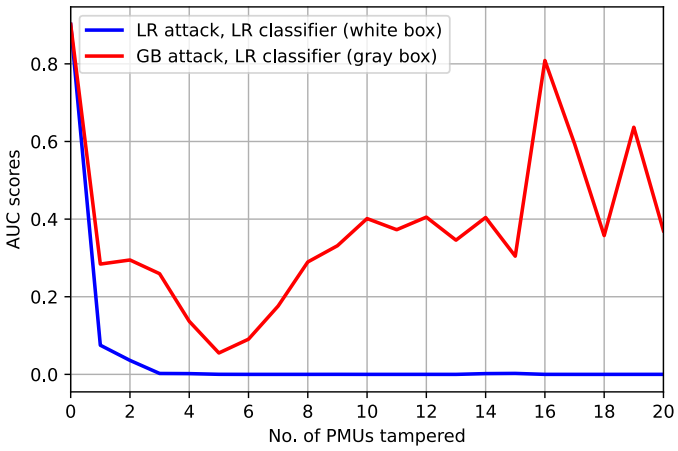

Fig. 3: AUC scores as a function of the number of tampered PMUs for white (blue curve) and gray (red curve) box attacks for the logistic regression (LR) classifier.

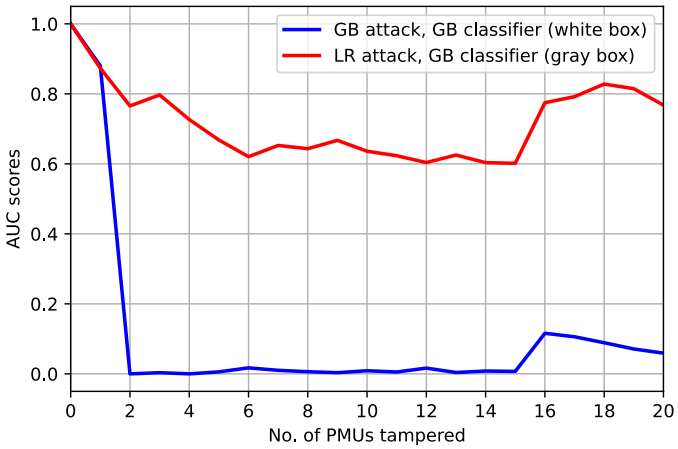

Fig. 4: AUC scores as a function of the number of tampered PMUs for white (red curve) and gray (blue curve) box attacks for the gradient boosting (GB) classifier.

C. Generation and Evaluation of Adversarial Examples

As a first step towards evaluating the white box and gray box attack algorithms, we generate the tampered events as outlined earlier. The average AUC scores are plotted in Figures 3 and 4. In short, we iterate over the original events from the testing set as input to the attack algorithms and choose the feasible set bounds as

$$\upsilon_\omega = 2\omega_0,\ \upsilon_\sigma = 0,\ \upsilon_{|R|,min} = 0.8|R_0|,\ \upsilon_{|R|,max} = 2|R_0| \quad (9)$$

where ω_0 and $|R_0|$ are the untampered values of those features for a given event.

To evaluate the impact of attacks on different numbers of PMUs, we choose 10 random sets $\mathcal{S}_{\text{atk}}$, each consisting of $M' = 20$ PMUs. Denote $\mathcal{S}_{M^{\text{atk}}}$ as the set consisting of the first M^{atk} PMUs in $\mathcal{S}_{\text{atk}}$, where M^{atk} varies from 1 to 20. We then evaluate the attack on $\mathcal{S}_{M^{\text{atk}}}$ for each M^{atk}. Figures 3 and 4 show the average AUC as a function of M^{atk}.

We evaluate white and gray box attacks as follows. Let $f \in \{\text{LR}, \text{GB}\}$ be the classification model used in the attack algorithm. In the white box setup, f is used both in the attack algorithm and as the classifier applied to the generated attack data. In the gray box setup, f is only used in the attack algorithm; and the classification model in $\{\text{LR}, \text{GB}\}$ other than f is used as the classifier. In other words, we run the attack algorithm using the knowledge of both classification models (LR and GB) and evaluate the output from each case using both classifiers.

For white box attacks on both LR and GB classifiers, the accuracy of the classifiers drops to close to 0% even when only 2 or 3 PMUs are attacked. In the gray box setup, attacks show a significant decrease in accuracy; however, they are less successful than white box attacks. This is expected as these attacks are designed to target different models. Moreover, we observe that GB models are more robust against gray box attacks compared to LR. Finally, gray box attacks on LR show a fluctuating behavior as the number of tampered PMUs increases. This is likely a result of the attack being tailored for GB, leading to unpredictable effects on the LR classifier. That is, an attack with the power to control more PMUs will not necessarily be more effective, since it may be pushing in the wrong direction.

V. Conclusion

ML-based event classification techniques can enhance situational awareness, especially with increasing DER penetration and their need for fast dynamic monitoring and response. We have shown that white box attacks for both LR and GB classifiers are highly successful, reducing the AUC score significantly with only a few PMUs tampered. On the other hand, gray box attacks cause a relatively modest reduction in AUC scores with GB being more robust. With our attack showing vulnerabilities in ML classifiers, future work will include developing classifiers to be more robust against attacks, as well as classifiers that are designed to distinguish attacks from legitimate events.

References

[1] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-time identification of dynamic events in power systems using pmu data, and potential applications—models, promises, and challenges," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.

[2] M. K. N. M. Sarmin, N. Saadun, M. T. Azmi, S. K. S. Abdullah, N. S. N. Yusuf, M. M. Vaiman, M. Y. Vaiman, M. Povolotskiy, and M. Karpoukhin, "Implementation of a pmu-based ems system at tnb," in *2021 IEEE Power & Energy Society General Meeting (PESGM)*, 2021, pp. 1–5.

[3] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 3153–3158.

[4] S. Lakshminarayana, A. Kammoun, M. Debbah, and H. V. Poor, "Data-driven false data injection attacks against power grids: A random matrix approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 635–646, 2021.

[5] A. Sayghe, O. M. Anubi, and C. Konstantinou, "Adversarial examples on power systems state estimation," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2020, pp. 1–5.

[6] Y. Cheng, K. Yamashita, and N. Yu, "Adversarial attacks on deep neural network-based power system event classification models," in *2022 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2022, pp. 66–70.

[7] N. Taghipourbazargani, G. Dasarathy, L. Sankar, and O. Kosut, "A machine learning framework for event identification via modal analysis of pmu data," *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4165–4176, 2023.

[8] T. Becejac and T. Overbye, "Impact of pmu data errors on modal extraction using matrix pencil method," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, 2019, pp. 1–8.

[9] T. Sarkar, F. hu, Y. Hua, and M. Wicks, "A real-time signal processing technique for approximating a function by a sum of complex exponentials utilizing the matrix-pencil approach," *Digital Signal Processing*, vol. 4, p. 127–140, 04 1994.

[10] K. Sheshyekani, G. Fallahi, M. Hamzeh, and M. Kheradmandi, "A general noise-resilient technique based on the matrix pencil method for the assessment of harmonics and interharmonics in power systems," *IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp. 2179–2188, 2017.

[11] D. Trudnowski, J. Johnson, and J. Hauer, "Making prony analysis more accurate using multiple signals," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 226–231, 1999.

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 – 1232, 2001. [Online]. Available: https://doi.org/10.1214/aos/1013203451

[13] T. Xu, A. B. Birchfield, K. S. Shetye, and T. J. Overbye, "Creation of synthetic electric grid models for transient stability studies," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:220726884