Feature Collusion Attack on PMU Data-driven Event Classification

1st Amir Ghasemkhani
Department of Computer Engineering and Computer Science
California State University Long Beach
Long Beach, USA
amir.ghasemkhani@csulb.edu

2nd Rutuja Sanjeev Haridas
School of Computer Science and Engineering
California State University San Bernardino
San Bernardino, USA
007433264@coyote.csusb.edu

3rd Seyed Mahmoud Sajjadi Mohammadabadi
Department of Computer Science and Engineering
University of Nevada Reno
Reno, USA
mahmoud.sajjadi@nevada.unr.edu

4th Lei Yang
Department of Computer Science and Engineering
University of Nevada Reno
Reno, USA
leiy@unr.edu

***Abstract*—Event classification is a critical task to ensure the reliability of the power system. Recent developments in event classification methods leverage data-driven techniques with fine-grained Phasor Measurement Units (PMU) data. However, the existing event classification methods are vulnerable to different types of adversarial attacks that can significantly degrade the event classification performance. In this paper, we evaluate the vulnerability of the classification models against feature collision attacks on PMU data. Feature collusion attack leverages a surrogate model to learn the victim's classification model which in turn makes it a plausible attack strategy for both black-box and white-box settings. Specifically, this attack strategy undermines the accuracy of the classification models by crafting poisonous samples that share common features with benign samples which in turn changes the decision boundaries of the classification models. The experimental results on real-world PMU data in a black-box setting show that generating and adding poisonous samples into the model training dataset can significantly degrade the accuracy of current event classification methods.**

***Index Terms*—Phasor Measurement Units (PMUs), Adversarial Attacks, Data Poisoning Attacks, and Feature Collusion Attack.**

I. Introduction

Phasor Measurement Unit (PMU) is an important tool used on electric transmission systems to improve operators' visibility into what is happening across the vast grid network [1]. One of the main applications of using PMUs is to develop accurate and robust event classification frameworks which are considered a crucial tool for improving power transmission system reliability [2]. Correctly classifying the PMU events from one another and differentiating them from a normal condition of the network, is very important in identifying the best remedial actions and acquiring insights for post-event analysis applications. The large-scale real-world PMU dataset and associated event logs provided by the U.S. Department of Energy and Pacific Northwest National Lab (PNNL) has paved the way for developing event diagnostic and classification frameworks for analyzing PMU data [3–6]. However, the proposed event classification models are vulnerable to adversarial attacks where small changes to the PMU data may result in the misclassification of the events rendering them, unreliable in real-world applications.

Adversarial attacks are malicious activities aimed at manipulating the output of a machine-learning model by introducing subtle perturbations to the input data. The goal of these attacks is to cause the model to make incorrect or biased predictions, leading to potentially harmful security consequences. There are various types of adversarial attacks, including evasion attacks [7], where an attacker tries to manipulate input data to evade detection, and poisoning attacks [8], [9], where an attacker tries to inject malicious data into the training set to undermine the integrity of the model. Since the PMU data streams can be easily manipulated by injecting false data, power system applications are more susceptible to data poisoning attacks [10], [11].

Data poisoning attacks can be categorized into backdoor and clean-label attacks. Backdoor attacks [12] involve embedding hidden malicious behaviors into classification models, resulting in misclassifications and activation only on inputs that contain a specific "trigger". These attacks are relatively easy to detect because they require triggers to be placed on the test samples. In contrast, clean-label attacks (e.g., [8], [9]) manipulate the training instances without utilizing triggers. The goal is to misclassify a single test sample by introducing perturbations that disrupt the feature region of the targeted sample. The feature collusion attack [8] is a form of clean-label attack where the attacker manipulates the feature representation of the training data to affect the model's output. This attack is not exclusive to neural networks but can also be applied to any other types of machine-learning models. Additionally, it can be executed in either a white-box or black-box setting, making it a highly effective technique for adversaries.

Despite the significant amount of research on cyberattacks in power systems, such as false data injection attacks [13], [14], there has been relatively little focus on adversarial attacks

This work is supported in part by NSF under Grants IIS-1838024, CNS-1950485, and OIA-2148788.

against power system's machine learning applications [15], [16]. The existing studies mainly leveraged attack strategies that are only applicable to neural network-based models which in turn limits their applicability to non-neural network classifiers. The vulnerability of the deep learning-based event classification models against several adversarial attack mechanisms is evaluated in [15] where small data perturbations are tailored and added to the PMU signals. Their findings indicate that the current deep learning-based event classifiers in power systems are highly susceptible to adversarial attacks which in turn pose a significant risk to the reliability of power systems. However, the scope of their work is limited to neural network-based classification models with white-box settings where the attacker has complete knowledge of the targeted model, including its architecture, parameters, and training data.

In this paper, we leverage a data poisoning scheme based on the feature collusion attack. The current attack strategies on PMU data are limited to white-box settings whereas the feature collusion attack can be used in both black-box and white-box settings. This is due to using a surrogate model to learn the victim's classification model, making it a viable strategy in both scenarios. Moreover, we examine how slight modifications to the PMU data can lead to incorrect predictions by both non-neural network (i.e., random forest model) and neural network event classification models. Finally, we conduct a large-scale case study using real PMU data from power system events in the Western Interconnection of the U.S. transmission grid to demonstrate the vulnerability of the event classification models to feature collusion attacks. Based on our findings, the current event classification models are vulnerable to poisoning attacks, as the addition of small perturbations to the PMU data leads to a significant reduction in the models' accuracy.

II. Feature collusion attack on PMU data

A. Overall Attack Framework

Figure 1 illustrates the overall framework of how an attacker can generate poisonous PMU data through a feature collision attack. We assume the attacker has full access to the PMU dataset and event logs. In the first step, the attacker starts with data preprocessing, which includes executing event detection [17] to capture PMU data around event time. The labels for the captured events are obtained from the system event logs. Once the data preprocessing procedure is complete, a feature collusion attack is applied to the event dataset to create poisonous samples specific to each event type. The generated poisonous samples are then fed into the feature extraction algorithm, which extracts the necessary features for building event classification models (e.g., the random forest classification model [4]). The feature extraction step is placed to enhance the event classifier's interpretability and performance by identifying event characteristics based on patterns of various event types. Note that this step can be skipped for the neural network-based classifiers as they have an automated feature extraction embedded in their model. Finally, the poisonous PMU features are combined with the benign features in the training dataset to train the event classification models.

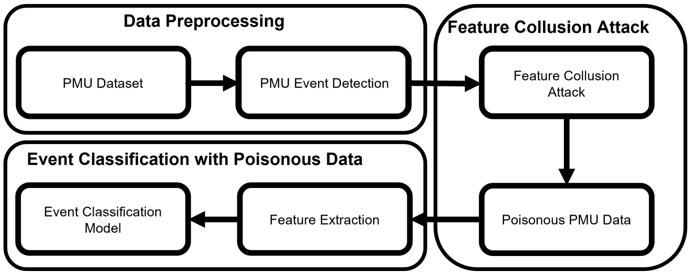

Fig. 1: Feature collusion attack on power system event classification models with PMU data.

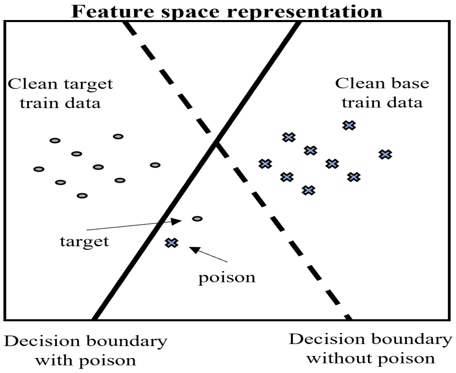
Feature space representation
Clean target train data
Clean base train data
target
poison
Decision boundary with poison
Decision boundary without poison

Fig. 2: Feature space representation for feature collusion attack.

B. Overview of Feature Collusion Attack

In the feature collusion attack, the goal is to undermine the accuracy and reliability of the machine learning model by introducing false data points that share common features. Figure 2 shows how the adversary changes the decision boundary of the classification model by adding poisonous instances to the training data. Specifically, the attacker selects an instance from the test set and modifies it in a subtle manner so that it will be misclassified by the model during testing. This altered instance is called the "target instance". In the next step, the attacker chooses a "base instance" from a different class and alters it imperceptibly to create a "poison instance". The poison instance is added to the training data with the goal of tricking the model into mislabeling the target instance with the base label during testing. Finally, the model is trained on the poisoned dataset, which includes both the clean and the poison instances. The attack is considered successful if the model misclassifies the target instance as the base class during testing.

C. Attack Model

Denote $\mathbf{X} = \{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_N, \mathbf{Y}_N)\}$ as the event dataset for PMU data where $\mathbf{X}_i$ is the extracted PMU data features for the i^{th} event and $\mathbf{Y}_i$ is the associated event label. Each event contains data features for P streaming PMUs, i.e., $\mathbf{X}_i = [x_i^1, x_i^2, \dots, x_i^P]$, where $x_i^p \in \mathbb{R}^{M \times T}$ is the measurement matrix for the p^{th} PMU with T samples around the event time and M measurements. We assume the attacker is capable of detecting the events through the PMU data streams. Therefore, their goal is to craft the poisonous samples around the event time and incorporate them into the training dataset to fool the event classification models. In order to craft the poisonous samples via the feature collusion

Algorithm 1 Generate Poisonous Samples based on Feature Collusion Attack on PMU Data

```
**Input:** Select target instance $\mathbf{X}_i^t \in \mathbf{X}_i$, base instance $\mathbf{X}_i^b \in \mathbf{X}_i$, learning rate $\lambda$, maximum number of iterations $maxIters$, and weight factor $\beta$
Initialize $\mathbf{X}_i(0) \leftarrow \mathbf{X}_i^b$.
Define $L_{\mathbf{X}_i}(\mathbf{X}_i) = ||f(\mathbf{X}_i) - f(\mathbf{X}_i^t)||^2$
**for** $k = 1$ **to** $maxIters$ **do** :
  Forward step: $\bar{\mathbf{X}}_i(k) = \mathbf{X}_i(k-1) - \lambda \cdot \nabla L_{\mathbf{X}_i}(\mathbf{X}_i(k-1))$
  Backward step: $\mathbf{X}_i(k) = (\bar{\mathbf{X}}_i(k) + \lambda \cdot \beta \cdot \mathbf{X}_i^b)/(1 + \beta \cdot \lambda)$
**end for**
```

attack, we solve the following optimization problem:

$$\bar{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} ||f(\mathbf{X}_i) - f(\mathbf{X}_i^t)||_2^2 + \beta||\mathbf{X}_i - \mathbf{X}_i^b||_2^2, \tag{1}$$

where $\bar{\mathbf{X}}_i$ is the generated poisonous instance, $\mathbf{X}_i^t$ is the target class instance, $\mathbf{X}_i^b$ is the base class instance, and $f(\mathbf{X}_i)$ denotes the surrogate classification model which maps the input signal $\mathbf{X}_i$ to the event label Y_i. Note that the surrogate model utilized in the feature collusion attack can be a type of neural network model designed to replicate the behavior of the original classification model. By incorporating a surrogate model, it becomes possible to generate poisonous samples in the black-box setting and is applicable for non-neural network classifiers. Moreover, the second term on the right-hand side of Equation (1) ensures the poisonous instance $\bar{\mathbf{X}}_i$ to appear very similar to the base class instance. We will use a combination of original and poisonous samples, $[\mathbf{X}_i \ \bar{\mathbf{X}}_i]$, to train the event classification models. Note that incorporating the poisonous samples in the training dataset through a feature collusion attack not only alters the decision boundaries of the event classifier but also decreases the classifier's accuracy by increasing the misclassification of the benign samples, leading to an increased rate of false positives. In security-critical applications such as event classification in power systems, the consequences of a compromised model can be severe resulting in extended blackouts and increased maintenance costs.

D. Optimization Procedure

Algorithm 1 illustrates the optimization procedure which is based on the forward-backward splitting method [18] to solve (1). The initial (forward) step involves performing a gradient descent update to reduce the L2 distance to the target instance in the feature space. The subsequent (backward) step employs a proximal update to minimize the Frobenius distance from the base instance in the input space. The weight factor β tunes the degree of similarity between the poison and base class instances.

III. Experiments

A. Dataset and Event Description

The dataset for this project is a real-world dataset collected from the western interconnection transmission grids in the United States given by the Pacific Northwest National Laboratory (PNNL). There are 23 PMUs with a sampling rate of 60 frames per second. The dataset under consideration spans two years, 2016 and 2017. In addition to the raw measurements (i.e, voltage and current magnitude of positive sequence and frequency), we have event logs that can be utilized as labels for event classifiers. After collecting PMU data, we use the event detection method to accurately localize the event times [17]. After detecting the exact event time index, we crop the event data over a time frame of 15 seconds (300 samples before and 600 samples after the event index, i.e., 900 samples of event data). In total, 7389 events were detected with three types of events, frequency, line outage, and transformer outage events. The dataset is split into 1903 testing samples and 5486 training samples including 1760 frequency, 2084 line outage, and 1624 transformer outage events.

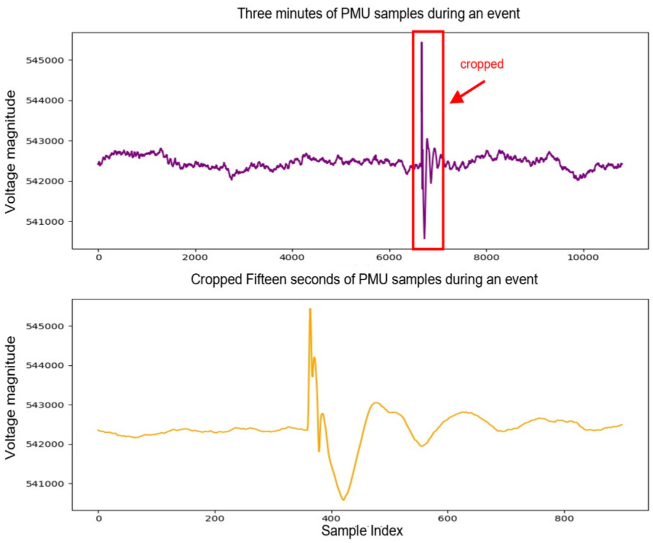

Fig. 3: Cropping PMU around the detected event time (cropped 15 seconds, i.e., 900 samples, around an event)

B. Experimental Setup

We discuss each component of the proposed framework in Figure 1 in detail to establish a concrete setup for our experiments.

1) Event Detection: Upon analyzing the PMU measurements collected from PNNL dataset, it has been observed that direct utilization of the PMU data is challenging due to its poor data quality, which does not meet the requirements of standard machine learning techniques. To overcome this limitation, an existing real-time event detection scheme is leveraged to capture the data around the event time [17]. The proposed event detection scheme constructs rank signatures using the relative change in the ratio of the two largest singular values of the PMU measurement matrix. The average relative change of this ratio across a short time window is computed to detect events using a threshold-based rule applied to different signals. An event is detected when one of the signals exceeds a specific threshold. After the event is detected, event data within a short time frame (e.g., 300 samples before and 600 samples after the event time index) is extracted. Figure 3 illustrates the cropped fifteen seconds of voltage magnitude data (i.e., 900 samples) for a sample PMU around the event time.

2) Feature Extraction: The goal of feature extraction is to extract crucial PMU characteristics that improve the performance and interpretability of event classifiers. We use a pre-existing approach from [4] for feature extraction that generates event characteristics based on the patterns of distinct PMU event

TABLE I
PERFORMANCE OF THE RANDOM FOREST CLASSIFIER ON POISONOUS PMU EVENTS

No. of poisonous samples in the training dataset	Base class: Line event		Base class: Frequency event		Base class: Transformer event	
	Target class: Frequency event	Target class: Transformer event	Target class: Line event	Target class: Transformer event	Target class: Line event	Target class: Frequency event
0	92.87 %	92.87 %	92.87 %	92.87 %	92.87 %	92.87 %
500	61.90 %	70.74 %	84.14 %	75.95 %	74.61 %	76.45 %
1000	50.89 %	61.56 %	72.42 %	70.74 %	66.19 %	72.74 %

TABLE II
PERFORMANCE OF THE NEURAL NETWORK MODEL ON POISONOUS PMU EVENTS

No. of poisonous samples in the training dataset	Base class: Line event		Base class: Frequency event		Base class: Transformer event	
	Target class: Frequency event	Target class: Transformer event	Target class: Line event	Target class: Transformer event	Target class: Line event	Target class: Frequency event
0	75.23 %	75.23 %	75.23 %	75.23 %	75.23 %	75.23 %
500	35.60 %	48.90 %	64.44 %	57.89 %	56.31 %	60.90 %
1000	24.21 %	33.87 %	55.47 %	49.12 %	41.48 %	52.86 %

types. Note that the feature extraction is applied on both clean and poisonous samples. The feature extraction block generates a total of 57 features, of which 6 are used for capturing the shape features for each event type such as Amplitude above average, Amplitude below average, Ramp-up rate, Ramp-down rate, Area above average, Area below average; 9 are for computing signal similarities between different PMUs, including minimum, maximum, and mean for frequency, current, and voltage magnitude; and three are auxiliary ratio features. After extracting the PMU features for event time samples, these features are used as training data for the event classification models.

3) Event classification Models: We adopt an existing random forest event classification model with 100 decision trees for performing event classification as proposed in [4]. To train an event classification model, we leverage the extracted features from the event time data instead of the original time-series data. Our experiments indicate that using the original time-series data for training the model would yield lower classification accuracy compared to the case in which we use the extracted features. To evaluate the vulnerability of the classification model against the feature collusion attack, we train the model with a combination of clean and poisonous samples.

To better compare the impact of the poisonous samples in the performance of the event classification problem, our approach also involves utilizing a fully connected neural network for the purpose of training the event classifier. However, it is worth noting that when using the original time series data to train the neural network-based classifier, the classification accuracy is found to be lower than that of the random forest classifier. This is attributed to a lack of adequate training samples and overfitting issues [4]. To further improve the classification accuracy, we add a custom activation layer based on the feature extraction function as the initial layer of the neural network event classifier. This layer is kept non-trainable and serves solely to provide the 57 features necessary for training the classifier. Using the extracted features instead of the original time-series data improves the accuracy of the event classification model.

C. Experimental Results

We choose different combinations of event types as target and base instances and craft the poison samples using Algorithm 1 with default values of the *maxIters* and β from IBM Adversarial robustness toolbox [19].

1) Performance of Event Classification Models against Feature Collusion Attack: To investigate the impact of poisonous samples on the performance of the event classifiers, we incorporate the generated poisonous samples in the training process of random forest and neural network-based classifiers. We first train the classification models with non-poisonous samples, then evaluate their performance after progressively adding a variable number of poisonous samples to the training dataset. Note that the maximum number of poisonous samples which can be added to the training dataset is equal to the number of base class samples. For example, if the line outage event is the base class, we can generate up to 2084 poisonous samples.

Table I shows the classification accuracy of the random forest classifier in presence of the poisonous samples. The accuracy of the random forest event classification model after training with non-poisonous samples is 92.87%. However, the accuracy is dropping significantly as the number of poisonous samples is increased in the training process. The results also illustrate the accuracy of different combinations of the base and target classes. In all scenarios, the random forest classifier's accuracy degrades considerably compared to the scenario in that we have no poisonous samples in the training dataset. We also investigate the performance of the neural network-based classifier in presence of the poisonous samples as shown in Table II. We leverage a fully connected neural network in which the hyperparameters have been optimally tuned using the scikit-learn grid search module. The results also indicate that classification accuracy drastically deteriorates as the number of poisonous samples is increased in the training process.

2) Signal Similarity of Poisonous and Original Data: Introducing data perturbations into the original signals through the feature collision attack can lead to the misclassification

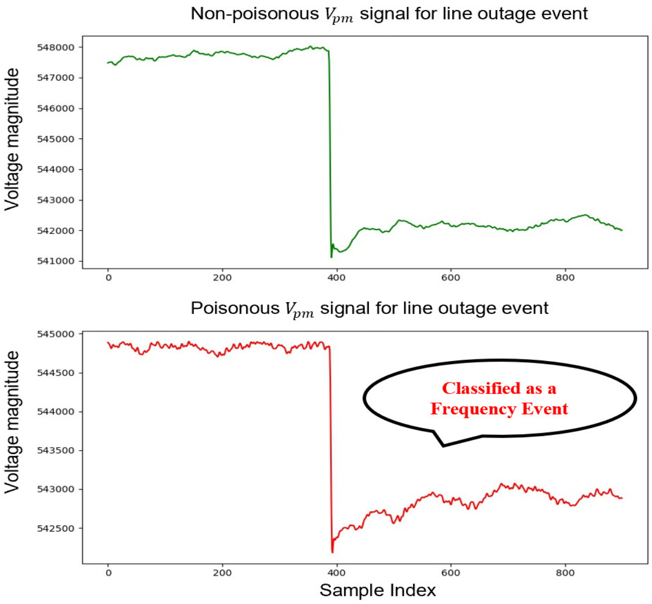

Fig. 4: Original and poisonous signals after the feature collusion attack

of events. Figure 4 shows how adding perturbations to the frequency (i.e., f) and voltage magnitude of positive sequence (i.e., V_{pm}) signals during a line outage even alters the original signals and cause the event classification model to misclassify the event as a frequency event. The figure highlights the striking similarity between the poisonous and original samples, making it challenging for bad data detection frameworks to differentiate the malicious data. To apply this attack on the PMU data, the attacker needs to gain unauthorized access to PMUs or the communication infrastructure used to transmit PMU data or the data storage units. Once inside, they can manipulate the data or inject poisonous samples by changing the section of measurement data during the event time and replace it with the poisonous time series data. This is similar to replay attacks where the attacker intercepts the communication line and inserts the previously recorded measurements instead of the actual sensor data [10], [11]. However, in our case, the attacker will insert poisonous data to impersonate the original data. While acknowledging the importance of developing defense mechanisms against such attacks, this paper does not focus on the implementation of appropriate security measures. Therefore, further research and investigation are needed in this area.

IV. Conclusions

This paper implemented a feature collision attack on phasor measuring unit (PMU) data and examined how the generation of poisonous samples in a black-box setting influenced the accuracy of event classifiers in power systems. By leveraging the feature collision attack, it was possible to generate poisonous samples for different types of classifiers such as neural network and non-neural network-based classifiers. Additionally, leveraging a surrogate model to learn the victim's classification model enables the attack to be applied in both white-box and black-box settings. The experimental results on a real-world PMU dataset revealed that the event classifiers' performance was drastically hampered after adding perturbations to the original PMU data. These results highlight a significant weakness in power system event classifiers when it comes to data poisoning attacks. The outcomes of this study will be utilized to develop robust defense mechanisms against poisoning attacks for PMU data.

Acknowledgment

We thank Pacific Northwest National Laboratory (PNNL) and Department of Energy (DOE) for offering the data in this study.

References

[1] A. Ghasemkhani, H. Monsef, A. Rahimi-Kian, and A. Anvari-Moghaddam, "Optimal design of a wide area measurement system for improvement of power network monitoring using a dynamic multiobjective shortest path algorithm," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2303–2314, 2017.

[2] M. Vaiman, R. Quint, A. Silverstein, M. Papic, D. Kosterev, N. Leitschuh, A. Faris, S. Yang, B. Blevins, S. Rajagopalan, P. Gravois, O. Ciniglio, S. Maslennikov, E. Litvinov, X. Luo, and P. Etingov, "Using synchrophasors to improve bulk power system reliability in north america," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, 2018, pp. 1–5.

[3] J. Shi, B. Foggo, and N. Yu, "Power system event identification based on deep neural network with information loading," *IEEE Transactions on Power Systems*, pp. 1–1, 2021.

[4] Y. Liu, L. Yang, A. Ghasemkhani, H. Livani, V. A. Centeno, P.-Y. Chen, and J. Zhang, "Robust event classification using imperfect real-world pmu data," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[5] I. Niazazari, Y. Liu, A. Ghasemkhani, S. Biswas, H. Livani, L. Yang, and V. A. Centeno, "Pmu-data-driven event classification in power transmission grids," in *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2021, pp. 1–5.

[6] Y. Yuan, Y. Guo, K. Dehghanpour, Z. Wang, and Y. Wang, "Learning-based real-time event identification using rich real pmu data," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5044–5055, 2021.

[7] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," 2020.

[8] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," 2018.

[9] A. Turner, D. Tsipras, and A. Madry, "Clean-label backdoor attacks," 2019. [Online]. Available: https://openreview.net/forum?id=HJg6e2CcK7

[10] A. Ashok, M. Govindarasu, and V. Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1636–1646, 2018.

[11] M. Ghaderi, K. Gheitasi, and W. Lucia, "A blended active detection strategy for false data injection attacks in cyber-physical systems," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 168–176, 2021.

[12] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," 2019.

[13] J. Liang, L. Sankar, and O. Kosut, "Vulnerability analysis and consequences of false data injection attack on power system state estimation," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3864–3872, 2016.

[14] X. Liu, Z. Li, and Z. Li, "Optimal protection strategy against false data injection attacks in power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1802–1810, 2017.

[15] 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.

[16] X. Zhou, R. Canady, Y. Li, X. Koutsoukos, and A. Gokhale, "Overcoming stealthy adversarial attacks on power grid load predictions through dynamic data repair," in *Dynamic Data Driven Applications Systems*, F. Darema, E. Blasch, S. Ravela, and A. Aved, Eds. Cham: Springer International Publishing, 2020, pp. 102–109.

[17] A. Ghasemkhani, Y. Liu, and L. Yang, "Real-time event detection using rank signatures of real-world pmu data," in *2022 IEEE Power Energy Society General Meeting (PESGM)*, 2022, pp. 1–5.

[18] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a fasta implementation," 2016.

[19] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.2.0," *CoRR*, vol. 1807.01069, 2018. [Online]. Available: https://arxiv.org/pdf/1807.01069