Title

Poisoning Attack against Event Classification in Distribution Synchrophasor Measurements

Authors

Kamal, Mohasinina
Shahsavari, Alireza
Mohsenian-Rad, Hamed

Publication Date

2021

Peer reviewed

Poisoning Attack against Event Classification in Distribution Synchrophasor Measurements

Mohasinina Kamal, *Student Member, IEEE*, Alireza Shahsavari, *Member, IEEE*,
and Hamed Mohsenian-Rad, *Fellow, IEEE*

***Abstract*— Distribution-level phasor measurement units (D-PMUs), a.k.a., micro-PMUs, have received a growing attention in recent years to support various applications in power distribution systems. Many of the applications of micro-PMUs work based on the analysis of *events* in the stream of synchrophasor measurements to achieve situational awareness. A key step in almost every *event-based* method in this emerging field is to classify the type of the event, where classification can be done with respect to various factors. However, if the task of event classification is compromised, then an adversary can highly affect the perception of the utility operator and undermine any event-based application that makes use of the event classification results. In this paper, we explore a new cyber-threat against data-driven event classification in micro-PMU measurements. In particular, we model the *poisoning attack* against support vector machine (SVM) as the method of event classification; which has been used in practice to study distribution synchrophasors. We apply the new attack model to an event classifier that uses *real-world* micro-PMU data. In addition to conducting vulnerability analysis, we also propose a novel *attack detection* method which can detect and evaluate the changes in the decision boundary of the SVM due to the poisoning attack. The proposed attack detection method is also able to identify the number of poisoned data points in the training dataset.**

***Keywords*: D-PMUs, Micro-PMUs, event classification, cyber attack, poisoning attack, attack model, attack detection, machine learning, power distribution, distribution synchrophasors.**

I. Introduction

A growing class of smart grid sensors are called distribution-level phasor measurement units (D-PMUs), a.k.a., micro-PMUs. They provide a continuous stream of GPS-synchronized voltage and current phasor measurements. The typical reporting rate of micro-PMUs is 120 phasor readings per second [1].

Many of the existing data-driven methods in the field of distribution synchrophasors work based on the analysis and classification of *events* that are observed in micro-PMU measurements. In this context, *event classification* may identify whether the root cause of the event is in the under-study power distribution feeder; or in the up-stream power transmission network [2]. Event classification may also involve identifying the *type* of the event, such as transformer tap-changer operation, load switching, capacitor bank switching, device malfunction, etc. [3]–[6].

Different methods have been developed in the literature to conduct event classification in micro-PMU measurements. Machine Learning (ML) methods are particularly popular in this area; including supervised learning [7], [8] and unsupervised learning methods [9], [10]. Once an event is classified, it can be used in various *event-based applications*, such as asset monitoring [11], fault location [12], state estimation [13], etc.

In this paper, we are concerned with the potential vulnerability of the *event classification* task in distribution synchrophasor measurements to cyber-attacks. We are particularly interested in addressing this open problem for the cases where event classification is done by using machine learning techniques.

Generally speaking, most ML methods were not originally developed to operate in an adversarial environment. If an attack can compromise the *learning stage* in the development of an event classifier, then it can undermine our ability to analyze the events correctly. This in turn can prompt incorrect actions that make use of the event classification results.

Accordingly, it is necessary to not only conduct a vulnerability analysis of the attacks against event classification in micro-PMU measurements, but also develop proper countermeasures.

A. Literature Review

Different types of attacks against PMUs have been investigated in the literature, mainly in power transmission systems. Examples in this area include: packet drop attacks [14], denial of service (DoS) attacks [15], and GPS signal spoofing [16].

The literature on attacks against micro-PMU data and cyber-security challenges in distribution system is still evolving. In [17], [18], the authors developed new methods to detect false data injection attacks against distribution system state estimation. In [19], the authors developed a geometric method to model the attacks against events in micro-PMU measurements. In [20], a source authentication method is proposed to detect data spoofing attacks in power distribution synchrophasor measurements. A flexible Bayes classifier is proposed to train spatio-temporal patterns of distribution systems data which can distinguish attacked data from non-attacked ones [21]. In [22], micro-PMU measurements are used by data-driven methods to detect data integrity attacks against PV inverter sensors.

The studies in [17]–[22] do *not* discuss the attacks *against* machine learning methods. They either are not related to machine learning, as in [17]–[19], or use machine learning to detect the attack, as in [20]–[22]. On the contrary, our focus is on the cases where the *target* of the attack is a machine learning method that conducts event classification based on micro-PMU data.

In practice, ML methods continuously train the learning model components and parameters through new measurements, so as to adapt to the changes in the system. However, this can open up a new attack surface against such methods [23].

Outside the field of distribution synchrophasors, adversarial attacks against ML-based frameworks have been studied in

The authors are with the University of California, Riverside, CA, USA. A. Shahsavari is also affiliated with the San Diego Gas & Electric, San Diego, CA, USA. This work is supported by UCOP grant LFR-18-548175. The corresponding author is Hamed Mohsenian-Rad. E-mail: hamed@ece.ucr.edu.

computer science. The initial efforts were in the field of spam filtering [24], showing that linear classifiers can be tricked by few carefully-crafted changes in the content of spam emails, without significantly affecting the readability of the spam message. Attacks against ML systems can happen during *testing (evasion)* or *training (poisoning)*. In evasion attacks, the attacker manipulates test data to have them misclassified, i.e., to evade detection by the learning algorithm, e.g., see [25].

On the other hand, poisoning attack refers to manipulating the training data, mainly by injecting adversarial points into the training set, which changes the decision function of the underlying classifier to compromise the result, c.f., [26], [27].

While the concept of adversarial attacks against ML in power systems is fairly new, it has drawn some attention recently. The studies in [28]–[30] have all focused on *evasion* attacks. To the best of our knowledge, the only study on *poisoning attacks* in power systems is done in [31], where the impact of poisoning attacks against *short term load forecasting* is studied.

B. Summary of Technical Contributions

To the best of our knowledge, this paper is the first study to analyze the vulnerability of event classification in micro-PMU measurements to poisoning attacks in the field of adversarial machine learning in power distribution systems. In addition to formulating the attack, we use data from real-world field measurements to identify most vulnerable feature scenarios. Furthermore, we investigate the choice of kernels to achieve more robust event classifiers in the presence of the poisoning attack. Next, we develop a new *attack detection* method which does *not* require access to any prior information on ground truth. The performance of the proposed method is verified in a case study. It is shown that a baseline method, that works based on pre-filtering of the synchrophasor measurements using spatial clustering is not capable of detecting the attack under the same circumstances. Finally, this study also includes sensitivity analysis and it also confirms how the proposed method can also provide an estimate on the number of poisoned data points, in addition to detecting the presence of the attack.

II. Threat Model

Consider a power distribution feeder with two micro-PMUs, e.g., as shown in Fig. 1. Suppose the synchrophasor measurements are used to train an ML model for event classification. In this section, we develop the model to conduct a poisoning attack against the training of the event classification model.

A. Event Classification

For the sake of this study, we consider the machine learning-based event classification method that was designed in [2] as the target of possible poisoning attack. Importantly, this event classification method has been implemented and tested in practice by using real-world micro-PMU data to classify real-world distribution synchrophasor events; therefore, it serves well for the purpose of developing a realistic threat model.

The event classification method that was developed in [2] uses a set of training power system events, that are already labelled

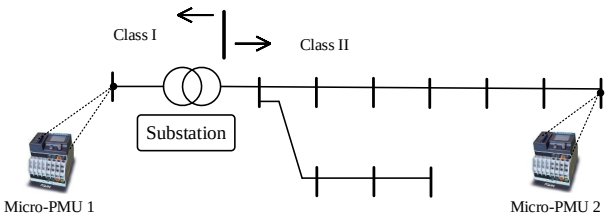

Fig. 1: The basic setup in the event classification problem using the synchronized measurements from a pair of micro-PMUs.

according to the field knowledge and utility event logs. The events are assigned into two broad categories corresponding to their root causes as shown in Fig. 1, as follows:

1) Class I: Events that are initiated from somewhere outside of the under-study feeder, i.e., transmission system or other distribution feeders; see the *left*-arrow in Fig. 1.
2) Class II: Events that are initiated from the under-study power *distribution* system; see the *right*-arrow in Fig. 1.

A binary-SVM classifier is trained with the features that are extracted from the micro-PMU measurements during each event. For each training event $i = 1, \cdots, m$, let x_i denote the vector of extracted features X corresponding to event i; and let y_i denote the assigned label for event i. We define $y_i \in \{-1, 1\}$, where $y_i = -1$ indicates that the event belongs to Class I; and $y_i = 1$ indicates that the event belongs to Class II. In case the training samples are linearly separable, we aim to find a separating hyperplane in feature space as $W^T X + b = 0$ to separate the two classes, where W is the vector of coefficients, and b is the intercept. In this paper, we also use non-linear separation, i.e., we use $\phi(x)$, where ϕ takes an input feature X and maps it into some other feature space, usually to a higher dimensional space. The SVM model is trained by solving the following primal optimization problem:

$$\begin{aligned} \underset{W,b,\zeta}{\textbf{minimize}} \quad & \frac{1}{2}||W||_2^2 + C\sum_{i=1}^{m}\zeta_i \\ \textbf{subject to} \quad & y_i(W^T\phi(x_i) + b) \geq 1 - \zeta_i, \\ & \zeta_i \geq 0, \quad i = 1, \cdots, m. \end{aligned} \tag{1}$$

where ζ_i is a slack variable corresponding to training event i to allow some samples to be at a distance ζ_i from their correct margin boundary and C is a model tuning parameter.

B. Poisoning Attack

The objective of the poisoning attack is to *compromise the training data* such that the trained SVM model in Section II-A results in *incorrect* classification of events, i.e., to significantly misplace the non-linear boundaries of the two classes. Here we assume a white box attack, where the attacker has full knowledge of the target classifier, including the training data, the feature set, the learning algorithm, the objective function to be minimized during training, and the parameters learned after training the model. This aligns with the *worst-case* vulnerability analysis, providing empirical upper bounds on the performance degradation that may be incurred by the system under attack.

In practice, the attack could be achieved to a certain extent by using a surrogate training set drawn from the same underlying data distribution to approximate classifier function [32].

Poisoning refers to the process of manipulating the training data, in which a number of specially crafted attack points are injected into the training data set. The poisoning attack can be formulated as a *bi-level* optimization problem in which the outer optimization maximizes the attacker's objective, i.e., hinge loss $\mathcal{L}$ in the case of SVM, while the inner optimization amounts to learning the classifier, i.e., L corresponding to the dual SVM learning problem, on the poisoned training data. Here, the attacker's goal is to find additional training sample (x^p, y^p), to add into the training dataset $\mathcal{X}_{tr}$ to maximally decrease the SVM's classification accuracy. This can be expressed as

$$\underset{x^p}{\textbf{maximize}} \quad \mathcal{L}(\mathcal{X}_{val}, W_p) \tag{2a}$$

$$\textbf{subject to} \quad W_p \in \underset{W}{\textbf{arg min}} \quad L(\mathcal{X}_{tr} \cup (x^p, y^p), W), \tag{2b}$$

where $\mathcal{X}_{tr}$ and $\mathcal{X}_{val}$ are the training data set and the validation data set available to the attacker, respectively. The former, along with the poisoning training sample (x^p, y^p), is used to train the attacked SVM classifier, while the latter is used to evaluate the performance of the trained model on untainted data, through the loss function $\mathcal{L}(\mathcal{X}_{val}, W_p)$. Notably, the objective function implicitly depends on x^p through the parameters in W_p of the poisoned classifier. The hinge loss for SVM is expressed as

$$\mathcal{L}(\mathcal{X}_{val}, W_p) = \sum_i max(0, 1 - y_i({W_p}^T \phi(x_i) + b)). \tag{3}$$

The process to solve the bi-level problem in (2) is explained in details in [32], [33]. It involves applying the gradient-ascent method to the dual formulation of the SVM problem, as:

$$\begin{aligned} &\underset{\alpha}{\textbf{minimize}} \quad \frac{1}{2}\sum_{i,j} \alpha_i Q_{ij} \alpha_j - \sum_i \alpha_i \\ &\textbf{subject to:} \quad \sum_i \alpha_i y_i = 0 \\ &\qquad\qquad 0 < \alpha_i < C, \quad i = 1, \cdots, m, \end{aligned} \tag{4}$$

where Q is an $m \times m$ positive semidefinite matrix:

$$Q_{ij} = y_i y_j \, K(x_i, x_j) = y_i y_j \, \phi(x_i)^T \phi(x_j). \tag{5}$$

The function $K(x_i, x_j)$ is the kernel. The terms α_i are the dual coefficients; and they are upper bounded by C.

From (2)-(5), the attacker selects the poisoning points by solving the following revised optimization problem [33]:

$$\underset{x^p}{\textbf{maximize}} \quad \sum_i \left(- \sum_j Q_{ij} \alpha_j - y_i b + 1 \right)^+. \tag{6}$$

III. Vulnerability Analysis

The key in the vulnerability analysis of poisoning attacks is to examine which features are more prone to cause misclassification if a poisoning attack occurs; and which kernels are more robust to prevent misclassification in the presence of poisoning attacks. Importantly, the results may vary depending on each specific classification problem in each field of research. In this paper, the specific classification problem of interest is the problem of event classification in micro-PMU measurements. Therefore, in this section, we conduct a vulnerability analysis based on the real-world micro-PMU measurements that have been previously studied in event classificaion in practice.

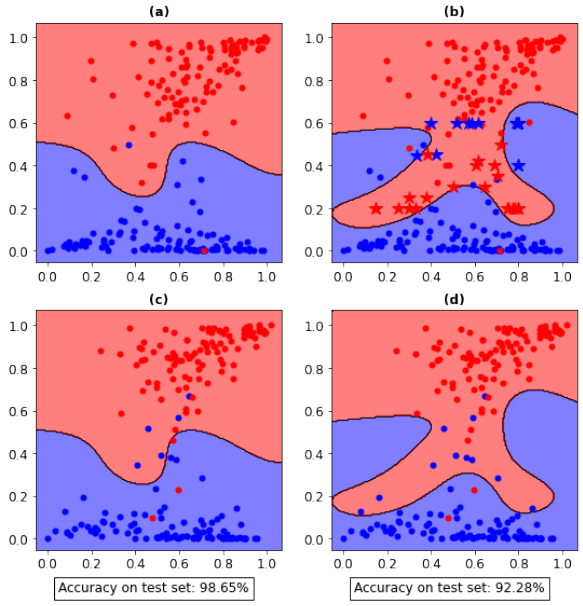

Fig. 2: The decision boundaries of the event classifier during: (a) the original training dataset; (b) the poisoned training dataset; (c) the test set being tested on the original classifier; (d) the test set being tested on the poisoned classifier.

For each event, the features that are used in event classification problem that we laid out in Section II-A include various combinations of two synchronized data sequences $D_1, D_2 \in \{I, V, P, Q\}$, which come from micro-PMU 1 and micro-PMU 2, respectively. Here, I, V, P, and Q denote the current magnitude, the voltage magnitude, the active power, and the reactive power, respectively. As in [2], we also consider the correlation between the measurements of the four available data sequences from each micro-PMU to construct some additional multi-stream features $corr(D_1, D_2)$; thus creating a total of 16 correlation features.

An example for the outcome of a poisoning attack against event classification based on real-world micro-PMU data is shown in Fig. 2. In this example, the features are $corr(V_1, Q_2)$ and $corr(Q_1, V_2)$ with 16% poisoned data points in the training dataset. Subscripts 1 and 2 stand for micro-PMU-1 and micro-PMU-2, respectively. Here we use the radial basis function (RBF) kernel with $\gamma = 10$ and $C = 1$. Parameter γ defines how far the influence of a single training data point reaches, with low values meaning *far reaching* and high values meaning *close reaching*.

Fig. 3 shows the vulnerability analysis based on real-world data. We consider three scenarios which make use of the most prominent features that are best capable of helping in event classification in the absence of an attack:

- Feature Scenario 1: $corr(Q_1, I_2)$ and $corr(Q_1, V_2)$
- Feature Scenario 2: $corr(I_1, I_2)$ and $corr(I_1, V_2)$
- Feature Scenario 3: $corr(V_1, Q_2)$ and $corr(Q_1, V_2)$.

As we can see in Fig. 3, the *least vulnerable* feature scenario in the presence of the poisoning attack is Feature Scenario 1. The *most vulnerable* feature scenario is Feature Scenario 3.

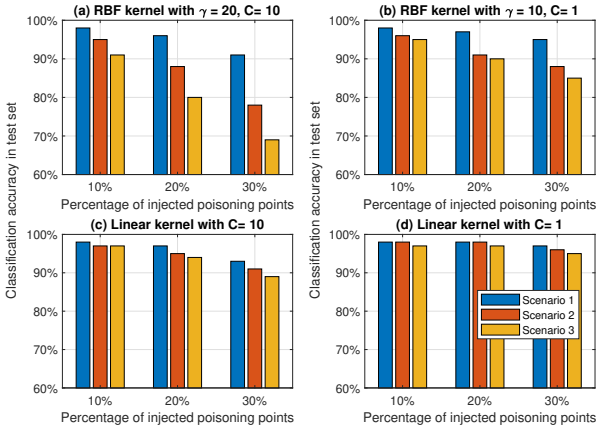

Fig. 3: Vulnerability analysis, in terms of the drop in event classification accuracy, for the three predominant feature scenarios and different kernels.

Importantly, our analysis also shows that using RBF as the kernel makes event classification more prone to error in the testing dataset, than using the linear kernel. Using the linear kernel with a lower C can provide a larger margin against the poisoning attack. Thus, while using nonlinear kernel is desirable in the *absence* of the poisoning attack, using linear kernel is more robust in the *presence* of poisoning attack.

It is worth adding that, conducting the above vulnerability analysis did not require using the raw measurements from the micro-PMUs. The key was rather to only use the features that are extracted from the raw measurements. In fact, we used the exact same features that were extracted in [2]; therefore, we were able to replicate the real-world event classification process in [2] to conduct a realistic vulnerability analysis.

IV. Attack Detection Method

Based on the nature of the features in micro-PMU measurements, we propose an attack detection method that does *not* require access to any ground truth. This is important; because the ground truth is usually not available to the utility in practice. The rational and the details of this method are explained next.

A. Basic Mathematical Concepts

Before we can explain the attack detection method, we need to first define some basic concepts. Suppose $\mathcal{D}$ denotes a *subset* of the set of all available training data $\mathcal{S}$. That is, we have:

$$\mathcal{D} \subset \mathcal{S}, \tag{7}$$

where

$$\mathcal{S} = \{1, \ldots, m\}. \tag{8}$$

Clearly, if we use the data points in set $\mathcal{D}$ to train the event classification model, then the obtained event classifier may not be exactly the same as the event classifier that we obtain if we use *all* the available training data points in set $\mathcal{S}$. An example is shown in Fig. 4. This example is constructed based on the poisoned event classification model in Fig. 2(b). The event classifier that is shown in Fig. 2(b) is the event classifier that is obtained by using all the $m = 250$ training data points in set $\mathcal{S}$. Now suppose we construct set $\mathcal{D}$ by *randomly dropping* 40 training data points

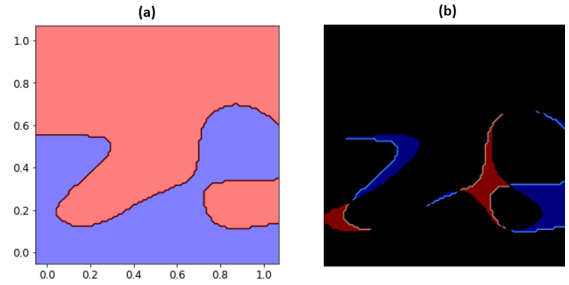

Fig. 4: An example to obtain classifier difference: (a) a new classifier that is obtained by dropping 40 data points from the poisoned data points in Fig. 2(b); (b) the difference between the classifier in Fig. 2(b) and the new classifier.

in set $\mathcal{S}$. The new event classifier that is trained by using the data points in set $\mathcal{D}$ are shown in Fig. 4(a). If we compare the classifier in Fig. 2(b), that is based on training data set $\mathcal{S}$ with the classifier in Fig. 4(a), that is based on the training data set $\mathcal{D}$, we can obtain the *classifier difference*, as shown in Fig. 4(b).

In Fig. 4(b), the points in *black* are classified the same by both classifiers; the points in *blue* are classified in Class I by the initial classifier but in Class II by the new classifier; and the points in *red* are classified in Class II by the initial classifier but in Class I by the new classifier. Here, the initial classifier refers to the classifier that is trained by using all the training data points in set $\mathcal{S}$; and the new classifier refers to the classifier that is trained by using the data points in set $\mathcal{D}$. Based on the classifier difference that is shown in Fig. 4(b), we can define:

$$\mathscr{P}(\mathcal{S}, \mathcal{D}) = \frac{\text{Blue Area} + \text{Red Area}}{\text{Total Area}}, \tag{9}$$

which is between 0 and 1. A higher $\mathscr{P}(\mathcal{S}, \mathcal{D})$ means a *more significant* difference between the event classifier that is obtained by using the training data points in set $\mathcal{S}$ and the event classifier that is obtained by using the training data points in set $\mathcal{D} \subset \mathcal{S}$.

B. Attack Detection Problem Formulation

Next, we use the index that we defined in Section IV-A to introduce the proposed attack detection method.

Based on the definition of set $\mathcal{D}$ in (7), let us define n as a parameter that indicates how many training data points we may randomly drop to create set $\mathcal{D}$. For the example in Section IV-A, we assumed that $n = 40$. We define set $\mathbb{D}$ as follows:

$$\mathbb{D} = \text{Set of all subsets of set } \mathcal{S} \text{ with } m - n \text{ members.} \tag{10}$$

Clearly, set $\mathbb{D}$ is a subset of the *super set* (i.e., the set of all subsets) $\mathcal{S}$. From (10), set $\mathbb{D}$ includes all the subsets of set $\mathcal{S}$ that can be created by dropping n training data points from the m total available training data points.

For a given training data set $\mathcal{S}$ and a given n, we define

$$\Gamma = \max_{\mathcal{D} \in \mathbb{D}} \mathscr{P}(\mathcal{S}, \mathcal{D}) \tag{11}$$

as the maximum difference between the initial event classifier based on the training data points in $\mathcal{S}$ and any possible event classifier that is obtained by dropping n training data points.

In practice, Γ is never zero. However, if set $\mathcal{S}$ is *clean*, i.e., if none of the available training data points is poisoned, then Γ is generally small. On the contrary, if set $\mathcal{S}$ is *poisoned*, i.e., if

some of the available training data points in set $\mathcal{S}$ are poisoned to the extent that they have considerable impact on the resulting model for the event classifier, then Γ is considerably large.

Accordingly, we detect a poisoning attack if we have:

$$\Gamma > \delta, \tag{12}$$

where δ is a detection threshold parameter. As we will see in Section V, selecting δ is not challenging in practice; because there is often a significant difference between Γ for a clean training data set and Γ for a poisoned training data set.

It is worth noting that set $\mathbb{D}$ can potentially be a large set, depending on the values of m and n. Therefore, in practice, one may obtain Γ based on only a *randomly selected* subset of the sets in $\mathbb{D}$. This can reduce computational complexity. In fact, this is exactly how we run our case studies in Section V.

V. Case Studies: Attack Detection

In this section, we evaluate the accuracy of the proposed attack detection method. We also compare the proposed method with a baseline method that works based on pre-filtering and spatial clustering. In addition, we conduct sensitivity analysis.

A. Attack Detection Accuracy

In this case study, we assume that $m = 250$ and $n = 40$. A total of 16% of the training data is assumed to be poisoned. The poisoning points are generated by using the code in [34]. Fig. 5 shows the value of $\mathscr{P}(\mathcal{S}, \mathcal{D})$ for 100 random scenarios. In each scenario, we create a new set $\mathcal{D}$ by randomly dropping n training data points. We can see that, the classifier differences are much higher for the poisoned case than for the clean case. From (11), we obtain $\Gamma = 0.054$ for the poisoned case and $\Gamma = 0.014$ for the clean case. Thus, there is a clear distinction between the poisoned case and the clean case. As a result, the proposed method can accurately detect the poisoning attack. In this example, δ can be anywhere between 0.03 to 0.05. We will further discuss the selection of the parameters in Section V-C.

B. Comparison with Baseline Method

In computer science, a popular approach to detect poisoning attacks is pre-filtering [35], [36]. In this approach, the training data set is examined *before* it is used in classification. On the contrary, we take into consideration the outcome of the event classifier training as the means to detect the poisoning attack.

We implemented the following pre-filtering technique as the baseline method: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [37]. The goal here is to find clusters of arbitrary shape to identify the presence of *outliers* in the training data set; which are deemed to be the poisoning data points. The results of applying DBSCAN to the real-world micro-PMU event features data are shown in Fig. 6. As we can see, this method identifies several outliers (black dots) and clusters (colored circles) in the poisoned training set.

If we were to use DBSCAN to pre-filter the training data set, then we would mark too many benign data points as outliers. For example, notice how the benign data points are placed into outlier clusters in the bottom red left corner in Fig. 6. Thus, as far as the features in micro-PMU measurements are concerned, DBSCAN is not the right tool to detect the attack.

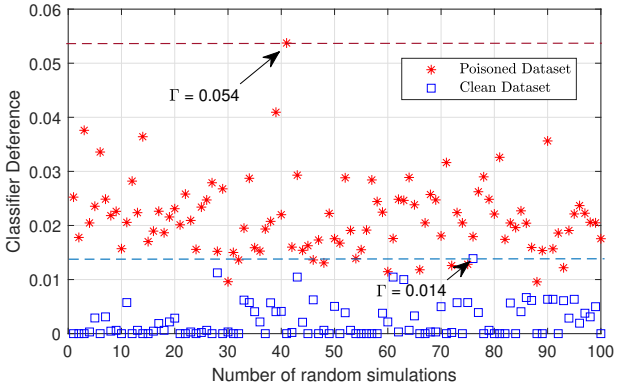

Fig. 5: Clear distinction between Γ for the poisoned dataset and Γ for clean dataset. The proposed detection method can accurately detect the attack.

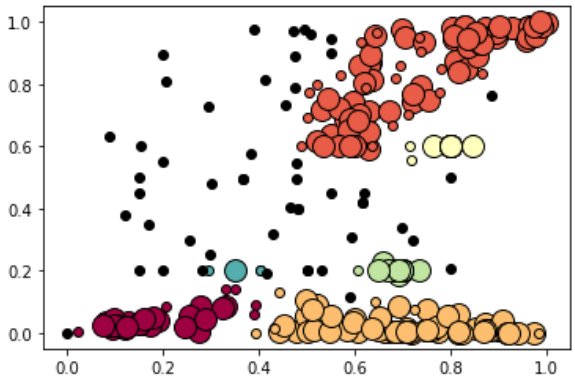

Fig. 6: The baseline method incorrectly clusters benign data points as outlier.

C. Sensitivity Analysis

Fig. 7(a) shows the sensitivity of obtaining Γ against the number of randomly simulated cases, i.e., the size of the random version of set $\mathbb{D}$. As we can see, we reach saturation at about 300. This is very promising; because it suggests that a relatively small number of random simulated cases is sufficient to obtain Γ. Therefore, the proposed method is computationally tractable.

Fig. 7(b) shows Γ for different choices of n in percentages. As expected, Γ increases as we randomly drop more training data points; whether from the poisoned training set or from the clean training set. Also, we can make two novel observations. First, regardless of the choice of n, there is a clear distinction between Γ for the poisoned case versus for the clean case. Therefore, the proposed detection method is *not* sensitive to the choice of n. Second, there is a breaking point in the *slope* of the red curve at $n = 16\%$. This is because, as we pass the number of poisoned data points, which is 16%, the slope reduces to almost the same slope as in the clean case. This is important; because we can check the slope of the curve for the poisoned case to *estimate the number of poisoned data points in the training dataset*.

VI. Conclusions

A new adversarial attack model, based on the concept of poisoning attack, is introduced against machine learning-based event classification in distribution synchrophasor measurements. On one hand, vulnerability analysis is conducted; and it is shown which features and kernels are more prone to deviate

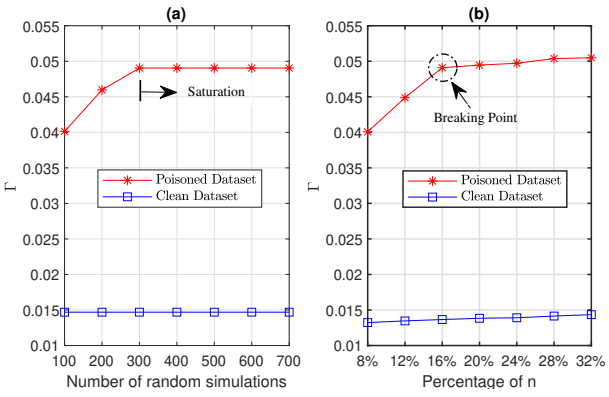

Fig. 7: Sensitivity analysis of the proposed method over two parameters: (a) the number of random simulated cases in set $\mathbb{D}$; and (b) parameter n.

event classification results in case of the attack, from adversarial attackers' or operators' perspective. On the other hand, a novel attack detection method is proposed that does not require any ground truth level to detect the attack. The proposed method is not sensitive to various parameters. Importantly, it can not only detect the attack; but also estimate the number of poisoned data points in the training data set. The results in this paper are insightful to the field of smart grid cyber-security and they can stimulate more future work; such as to examine other machine learning platforms for event classification as the target of the attack; or other data-driven applications of distribution synchrophasor measurements, i.e., beyond event classification.

REFERENCES

[1] H. Mohsenian-Rad, E. Stewart, and E. Cortez, "Distribution synchrophasors: Pairing big data with analytics to create actionable information," *IEEE Power and Energy Magazine*, vol. 16, no. 3, pp. 26–34, May 2018.

[2] A. Shahsavari, M. Farajollahi, E. M. Stewart, E. Cortez, and H. Mohsenian-Rad, "Situational awareness in distribution grid using micro-PMU data: A machine learning approach," *IEEE Trans. on Smart Grid*, vol. 10, no. 6, pp. 6167–6177, Feb 2019.

[3] M. Saini and R. Kapoor, "Classification of power quality events–a review," *Int. J. of Electrical Power & Energy Sys.*, vol. 43, pp. 11–19, Dec. 2012.

[4] O. Samuelsson, M. Hemmingsson, A. H. Nielsen, K. O. H. Pedersen, and J. Rasmussen, "Monitoring of power system events at transmission and distribution level," *IEEE Trans. on Power Systems*, vol. 21, no. 2, pp. 1007–1008, May 2006.

[5] H. Mohsenian-Rad, *Smart Grid Sensors: Principles and Applications*. Cambridge University Press, 2021.

[6] A. Shahsavari, A. Sadeghi-Mobarakeh, E. Stewart, E. Cortez, L. Alvarez, F. Megala, and H. Mohsenian-Rad, "Distribution grid reliability versus regulation market efficiency: An analysis based on micro-pmu data," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2916–2925, Nov. 2017.

[7] H. Ren, Z. J. Hou, B. Vyakaranam, H. Wang, and P. Etingov, "Power system event classification and localization using a convolutional neural network," *Frontiers in Energy Research*, vol. 8, p. 327, Nov 2020.

[8] R. Yadav, S. Raj, and A. K. Pradhan, "Real-time event classification in power system with renewables using kernel density estimation and deep neural network," *IEEE Trans. on Smart Grid*, vol. 10, no. 6, pp. 6849–6859, April 2019.

[9] A. Aligholian, A. Shahsavari, E. Stewart, E. Cortez, and H. Mohsenian-Rad, "Unsupervised event detection, clustering, and use case exposition in micro-pmu measurements," *IEEE Trans. on Smart Grid*, March 2021.

[10] Y. Chen, L. Xie, and P. R. Kumar, "Power system event classification via dimensionality reduction of synchrophasor data," in *IEEE Sensor Array and Multichannel Signal Proc. Workshop*, Coruna, Spain, Aug. 2014.

[11] A. Shahsavari, M. Farajollahi, E. Stewart, C. Roberts, F. Megala, L. Alvarez, E. Cortez, and H. Mohsenian-Rad, "Autopsy on active distribution networks: A data-driven fault analysis using micro-PMU data," in *Proc. of the IEEE PES NAPS*, Morgantown, WV, Nov. 2017.

[12] M. Farajollahi, A. Shahsavari, and H. Mohsenian-Rad, "Location identification of high impedance faults using synchronized harmonic phasors," in *Proc. of the IEEE PES ISGT*, Washington, DC, Apr. 2017.

[13] A. Akrami, S. Asif, and H. Mohsenian-Rad, "Sparse distribution system state estimation: An approximate solution against low observability," in *Proc. of the IEEE PES ISGT*, Washington, DC, May 2020.

[14] S. Pal, B. Sikdar, and J. Chow, "Real-time detection of packet drop attacks on synchrophasor data," in *Proc. of IEEE SmartGridComm*, Venice, Italy, Nov. 2014.

[15] A. Chawla, P. Agrawal, A. Singh, B. K. Panigrahi, K. Paul, and B. Bhalja, "Denial-of-service resilient frameworks for synchrophasor-based wide area monitoring systems," *Computer*, vol. 53, no. 5, pp. 14–24, May 2020.

[16] X. Fan, L. Du, and D. Duan, "Synchrophasor data correction under GPS spoofing attack: A state estimation-based approach," *IEEE Trans. on Smart Grid*, vol. 9, no. 5, pp. 4538–4546, Sept. 2018.

[17] R. Deng, P. Zhuang, and H. Liang, "False data injection attacks against state estimation in power distribution systems," *IEEE Trans. on Smart Grid*, vol. 10, no. 3, pp. 2871–2881, Mar 2018.

[18] B. Li, G. Xiao, R. Lu, R. Deng, and H. Bao, "On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices," *IEEE Trans. on Ind. Informatics*, June 2019.

[19] M. Kamal, M. Farajollahi, H. Nazaripouya, and H. Mohsenian-Rad, "Cyberattacks against event-based analysis in micro-PMUs: Attack models and counter measures," *IEEE Trans. on Smart Grid*, vol. 12, no. 2, pp. 1577–1588, Mar. 2021.

[20] W. Qiu, Q. Tang, Y. Wang, L. Zhan, Y. Liu, and W. Yao, "Multi-view convolutional neural network for data spoofing cyber-attack detection in distribution synchrophasors," *IEEE Trans. on Smart Grid*, vol. 11, no. 4, pp. 3457–3468, July 2020.

[21] M. Cui, J. Wang, and B. Chen, "Flexible machine learning-based cyber-attack detection using spatiotemporal patterns for distribution systems," *IEEE Trans. on Smart Grid*, vol. 11, no. 2, pp. 1805–1808, Mar. 2020.

[22] Q. Li, F. Li, J. Zhang, J. Ye, W. Song, and A. Mantooth, "Data-driven cyberattack detection for photovoltaic (PV) systems through analyzing micro-pmu data," in *Proc. of IEEE ECCE*, Detroit, MI, Oct. 2020.

[23] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. of ACM Symposium on Information, Computer and Comm. Security*, Taipei, Taiwan, Mar. 2006.

[24] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proc. of the ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, WA, Aug. 2004.

[25] P. Laskov and N. Rndic, "Practical evasion of a learning-based classifier: A case study," in *IEEE symposium on security and privacy*, May 2014.

[26] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar, "Antidote: understanding and defending against poisoning of anomaly detectors," in *Proc. of the ACM SIGCOMM Conference on Internet Measurement*, New York, NY, Nov. 2009.

[27] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. of the 29th AAAI Conference on Artificial Intelligence*, Austin, TX, Jan. 2015.

[28] A. Sayghe, J. Zhao, and C. Konstantinou, "Evasion attacks with adversarial deep learning against power system state estimation," in *IEEE PES General Meeting*, Mar. 2020.

[29] Y. Chen, Y. Tan, and D. Deka, "Is machine learning in power systems vulnerable?" in *Proc. of IEEE SmartGridComm*, Aalborg, Denmark, Oct. 2018.

[30] I. Niazazari and H. Livani, "Attack on grid event cause analysis: An adversarial machine learning approach," in *Proc. of IEEE PES ISGT*, Washington DC, Feb. 2020.

[31] Y. Liang, D. He, and D. Chen, "Poisoning attack on load forecasting," in *Proc. of IEEE PES ISGT Asia*, Sichuan, China, May 2019.

[32] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," in *28th USENIX Security Symposium*, Santa Clara, CA, Aug. 2019.

[33] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[34] https://gitlab.com/secml/secml.

[35] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," *arXiv preprint arXiv:1802.03041*, 2018.

[36] R. Laishram and V. V. Phoha, "Curie: A method for protecting SVM classifier from poisoning attack," *arXiv preprint arXiv:1606.01584*, 2016.

[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proc. of the Conf. on Knowledge Discovery & Data Mining*, Portland, OR, 1996.