Detecting Malicious Data in Distribution System State Estimation using Feature Attribution

Afia Afrin, Omid Ardakanian *Member, IEEE,*

***Abstract*—The digital transformation of power distribution system has increased the demand for Distribution System State Estimation (DSSE) techniques that are robust to adversarial perturbations in addition to noise. We design a versatile method for detecting stealthy data manipulation attacks on DSSE, drawing on a model-agnostic attribution method that quantifies the contribution of each input feature to the state estimation result. The key intuition for this work is that data manipulation attacks, including adversarial and false data injection attacks, generally have a discernible effect on this feature saliency measure. Through extensive numerical simulation, we corroborate that the proposed method reliably detects various data manipulation attacks, outperforming the most prominent detection methods from the previous work.**

***Index Terms*—State estimation, bad data detection, false data injection attack, adversarial attack.**

I. INTRODUCTION

Ensuring the reliable operation of power distribution networks has become increasingly challenging owing to the rapid installation of Distributed Energy Resources (DER) that could cause reverse flow, transformer overloading, wide voltage fluctuations, and voltage limit violations. This warrants comprehensive monitoring and control of distribution networks, both of which rely on telemetry data and Distribution System State Estimation (DSSE) to augment this data [1].

Existing DSSE approaches can be broadly classified into model-based and data-driven. Model-based approaches, such as Weighted Least Squares (WLS) [2] use the distribution system model for state estimation. Data-driven approaches, on the other hand, do not require knowledge of the distribution system model. Instead, they train a machine learning model to estimate the state given the sensor data [3]–[5]. Although these approaches differ in terms of accuracy and computation cost, both are found susceptible to data manipulation attacks that occur by comprising sensors, network devices, or servers that store or process the sensor data, allowing the attacker to manipulate this data before it is used for DSSE [6]–[9].

Early work that explored the robustness of state estimation has primarily focused on showing that stealthy false data injection attacks (FDIAs) can bypass the conventional Bad Data Detection (BDD) mechanism [10], [11]. Several detection strategies have been developed in recent years to mitigate these FDIAs [12]–[15].However, the landscape is different for adversarial attacks which can be generated through access to the input and output of DSSE, without even knowing what approach is used for state estimation. While it is established that the conventional BDD is unable to detect such attacks [7], [9], [16], an effective strategy to protect DSSE from adversarial attacks is yet to be developed. The previous work on detecting malicious data uses either statistical techniques, e.g. analyzing the distance between the distribution of measurement variations from a known distribution [17], or learning-based techniques, e.g. feeding the measurement data to a pretrained neural network [7]. Nevertheless, none of these techniques is fully effective against the wide range of stealthy FDIAs and adversarial attacks.

We propose an effective detection-based DSSE safeguarding strategy by analyzing the dispersion of *feature attribution* scores. Specifically, we borrow the idea of attributing the decisions made by a machine learning model to its input features from the explainable Artificial Intelligence (xAI) literature [18], and adapt it to work with DSSE, which is a multivariate regression problem. We use a perturbation-based attribution method [19], because it does not assume differentiability of the DSSE model and can therefore be used to protect both model-based and data-driven DSSE approaches. We show that data manipulation attacks that are designed to circumvent the conventional BDD, under the white-box or black-box assumption, change the *dispersion* of feature attribution scores in a way that is distinct from the effect of measurement noise. We use this insight to detect the presence of malicious data regardless of whether it is crafted using a FDIA or adversarial attack. Our contribution is threefold:

- We propose Detection by Feature Attribution (DEFEAT), a novel safeguarding strategy for both model-based and data-driven DSSE that distinguishes between malicious and benign (but possibly noisy) measurement data by analyzing the dispersion of feature attribution scores and classifying it using a logistic regression model.
- We show the efficiency of DEFEAT against two classes of data manipulation attacks, and analyze its sensitivity to the noise that might be introduced by phasor measurement units and smart meters.
- We compare the performance of DEFEAT with state-of-the-art detection-based DSSE safeguarding strategies, and show that it achieves higher success rate and better generalization across different attack scenarios.

II. RELATED WORK

Safeguarding DSSE can be accomplished via protection and detection methods. The former reduces the attack surface by securing and restricting access to sensors, communication

---

As a part of the University of Alberta's Future Energy Systems research initiative, this research was made possible in part thanks to funding from the Canada First Research Excellence Fund.

A. Afrin (email: aafrin@ualberta.ca) and O. Ardakanian (email: ardakanian@ualberta.ca) are with the Department of Computing Science, University of Alberta, Canada.

TABLE I: Comparative analysis of related work on safeguarding DSSE using detection-based methods.

References	Attack Type	Access to DSSE Model	Access to Detection Model	Detection Strategy	Successful Detection?
[6], [10]	FDIA	•		Conventional BDD	No
[17]	FDIA	•		Statistical Method (KLD)	Yes
[7]	Joint adversarial & FDIA	•	•	Conventional BDD, MLP	No
[20]	Adversarial	•	•	MLP	No
[16]	Adversarial			Conventional BDD	No
[21]	Adversarial			Conventional BDD, Statistical Methods	No
				Learning-based Methods	Partially
				Outlier Graph	Yes
Our work	FDIA	•		DEFEAT	Yes
	Adversarial (untargeted)				
	Adversarial (targeted)	•			

channels, and storage and processing nodes. But protective measures are not entirely foolproof so it is necessary to employ them alongside detection methods to ensure overall system security. Given the focus of this work, we only delve into the literature concerning methods that have been proposed for detecting various types of data manipulation attacks on DSSE.

Table I summarizes the related work on detecting data manipulation attacks. Two access classes have been considered depending on the attacker's knowledge of the DSSE model (aka the *victim model*) and the detection model (in case of learning-based detection methods). In particular, adversarial attacks can be categorized as either white-box or black-box depending on the attacker's level of access to these models. In white-box attacks, the attacker has full access to the victim model, including its architecture and parameters, allowing them to create adversarial examples by using the victim model. In black-box attacks, the attacker's access to the victim model is limited; it is often restricted to querying the victim model to predict the state that corresponds to some input and using this dataset to train a *surrogate model* that is used for generating attack vectors.

The most notable white-box attacks that assume full access to DSSE and detection models simultaneously are [7], [20]. In [7], a joint adversarial and stealthy false data injection attack has been launched against a DC state estimation model protected by two detection-based mechanisms, namely a conventional BDD and a Neural Attack Detector (NAD), which is a fully connected neural network trained to classify bad/malicious data. The authors found that both BDD and NAD are vulnerable to a state-perturbation-based FDIA generated under the white-box assumption. In [20], a similar study has been conducted on a DC state estimation model protected by a multilayer perceptron (MLP) that was trained to distinguish bad measurement samples from good ones. It is also found that the pretrained MLP is not effective against adversarial attacks generated under the white-box assumption.

Turning to black-box attacks, Bhattacharjee et al. [16] establish that conventional BDD is not successful in safeguarding the WLS-based AC state estimation approach against adversarial attacks. In recent work, the vulnerabilities of state estimation techniques to a black-box attack have been investigated [21]. The authors introduced a stealthy black-box attack algorithm capable of deceiving any state estimation model. The attack vector is generated by solving a convex optimization problem that incorporates a surrogate of the state estimation model in its objective function. This approach has been found successful in bypassing multiple detection-based safeguarding strategies, including conventional, statistical, and learning-based methods, when just a subset of measurements are manipulated. Our literature review suggests that there is currently no detection method that is effective against the variety of adversarial attacks, under white-box and black-box settings. Moreover, statistical methods that were found successful in detecting a specific type of FDIA do not attain a high success rate when it comes to adversarial attacks [21].

In recent years, xAI techniques that compute feature attribution score and Shapley value have been adopted to detect adversarial attacks on image classification models [19], [22]. Yet, to our knowledge, the application of a feature attribution method to safeguard DSSE, which is a multivariate regression task, is explored for the first time in our work.

III. BACKGROUND

A. Distribution System State Estimation (DSSE)

Definition. DSSE is the problem of determining the distribution network state, e.g. voltage magnitude and phase angle of some nodes, from potentially noisy and incomplete measurements obtained from various telemetry systems, such as Supervisory Control and Data Acquisition (SCADA), Distribution-level Phasor Measurement Units (D-PMUs), and Advanced Metering Infrastructure (AMI).

Casting state estimation as a WLS problem is the most widely used approach in transmission and distribution systems. Let us denote the vector that collects all n state variables as $\mathbf{x}$ and the vector that collects m independent measurements as $\mathbf{z}$. To estimate the system state at a given time, a WLS-based estimator minimizes the following objective function [2]:

$$\min_{x} J(\mathbf{x}) \text{ where } J(\mathbf{x}) = \sum_{i=1}^{m} \left(z_i - h_i\left(\mathbf{x}\right)\right)^2 / \mathrm{R}_{ii}. \quad (1)$$

Here $h(\cdot)$ is the (non-linear) function that relates state variables, $\mathbf{x}$, to the measurements, $\mathbf{z}$, and $\mathbf{R}$ is a diagonal matrix known as the *covariance matrix of measurement errors* and is constructed as $\mathbf{R} = \mathbf{diag}(\sigma_1^2, \cdots, \sigma_m^2)$ where $\mathbf{R}_{kk} = \sigma_k^2$ is the variance of the k^{th} measurement in $\mathbf{z}$. Writing (1) in vector/matrix form gives

$$\min_{x} \left[\mathbf{z} - h(\mathbf{x})\right]^\top \mathbf{R}^{-1} \left[\mathbf{z} - h(\mathbf{x})\right], \quad (2)$$

where $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$ with $\mathbf{H}$ being the *measurement matrix* and defined as the Jacobian matrix of $h(\cdot)$, i.e. $\mathbf{H} = \delta h(\mathbf{x})/\delta \mathbf{x}$.

B. Data Manipulation Attacks

We consider the following attacks in our evaluation.

Stealthy FDIA: This attack replaces the measurement vector $\mathbf{z}$ by $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$, where $\mathbf{a}$ is the attack vector defined as [11]:

$$\mathbf{a} = h(\mathbf{x_a}) - h(\mathbf{x}) \quad (3)$$

The key objective of the stealthy FDIA formulation is to bypass the conventional BDD mechanism, which thresholds the measurement residue to detect the presence of bad measurement data. Under FDIA, the residue can be written as:

$$\begin{aligned}\mathbf{r_a} &= \mathbf{z_a} - h(\mathbf{x_a}) = \mathbf{z} + \mathbf{a} - h(\mathbf{x_a}) \\ &= \big(\mathbf{z} - h(\mathbf{x})\big) + \mathbf{a} - h(\mathbf{x_a}) + h(\mathbf{x}) \\ &= \mathbf{r} + \mathbf{a} - h(\mathbf{x_a}) + h(\mathbf{x}) = \mathbf{r} \qquad \text{using Equation (3)}\end{aligned}$$

Here, $\mathbf{r}$ is the residue calculated for benign data, $\mathbf{z}$. Thus, generating an attack vector using (3) which requires knowledge of $h(\cdot)$, ensures that the residue under attack remains the same as the residue of benign data, bypassing the conventional BDD.

Adversarial Attack: Two different types of adversarial attacks on DSSE have been proposed in the literature, both capable of striking a balance between effectiveness and stealthiness.

a) Fast Gradient Sign Method (FGSM) [23]: Three variants of FGSM have been proposed to date: (a) the basic FGSM attack in which the sign of the gradient of the surrogate model's loss with respect to the input data is used to construct the attack vector [9]: $\mathbf{a}= -\ \epsilon\cdot\text{sign}(\nabla_{\mathbf{z}}\left[L\left(g_\theta(\mathbf{z}), \mathbf{x}\right)\right])$ with $g_\theta(\mathbf{z})$ representing the surrogate model that is trained on historical measurement data to mimic the victim DSSE model; (b) Sneaky FGSM [9]: a variant of FGSM that selectively adds perturbation to measurement data, thereby increasing the chance of remaining undetected; and (c) Targeted FGSM: a variant of FGSM that allows the attacker to control the direction and amount of error that is being injected [24], e.g. causing only erroneous overvoltage estimation. Unlike FGSM and Sneaky FGSM that can work under the black-box assumption, Targeted FGSM is successful under the white-box assumption only due to its targeted nature [24].[1]

b) Deep Black Box Adversarial Attack (DeeBBAA) [21]: This black-box attack strategy finds the attack vector that maximizes the deviation from the true state estimation result rather than the one that directly maximizes the loss of the state estimation model – the objective used in FGSM and its variants. The attack vector is generated following a two-step process. First, similar to other black-box adversarial attacks (i.e., FGSM and Sneaky FGSM), a surrogate model, denoted as $g_\theta(\mathbf{z})$, is trained on historical measurement data to mimic the victim DSSE model. In the next step, a constrained optimization problem is solved to find the adversarial perturbation vector, $\mathbf{a}$, that has a sufficiently small 2-norm and results in the greatest deviation from the state estimation result using benign data:

$$\begin{aligned}&\max_{\mathbf{a}} \|g_\theta(\mathbf{z}+\mathbf{a}) - g_\theta(\mathbf{z})\|_2 \\ &\text{s.t.} \qquad \|\mathbf{a}\|_2 \le \epsilon\end{aligned}$$

Since the above problem is non-convex, a semi-definite programming (SDP) relaxation is proposed in [21] by incorporating the gradient of the surrogate model's loss function, and solved to generate the attack vector quickly. DeeBBAA is claimed to be successful in bypassing conventional and statistical detection methods, such as chi-squared test, largest normalized residue test, adaptive non-linear cumulative sum, KL-divergence (KLD) based detector, as well as several learning-based detection models that are trained on malicious measurement data crafted by FDIA.

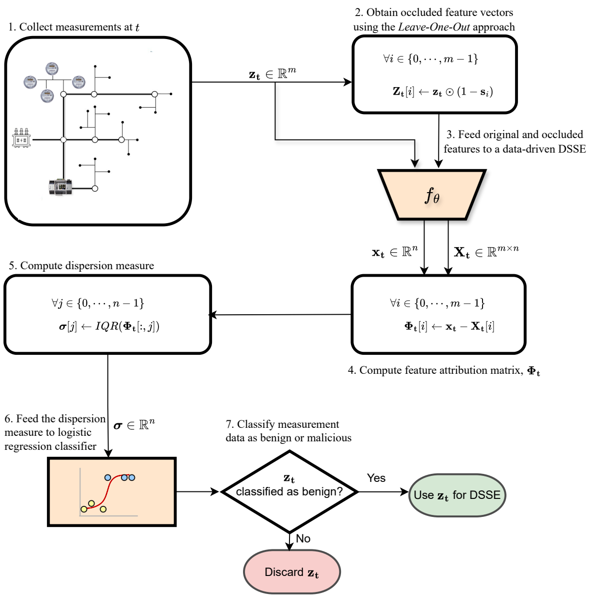

1. Collect measurements at t
$\mathbf{z_t} \in \mathbb{R}^m$
2. Obtain occluded feature vectors using the *Leave-One-Out* approach
$\forall i \in \{0, \cdots, m-1\}$
$\mathbf{Z_t}[i] \leftarrow \mathbf{z_t} \odot (1 - \mathbf{s}_i)$
3. Feed original and occluded features to a data-driven DSSE
f_θ
$\mathbf{x_t} \in \mathbb{R}^n$
$\mathbf{X_t} \in \mathbb{R}^{m\times n}$
5. Compute dispersion measure
$\forall j \in \{0, \cdots, n-1\}$
$\boldsymbol{\sigma}[j] \leftarrow IQR(\mathbf{\Phi_t}[:, j])$
$\forall i \in \{0, \cdots, m-1\}$
$\mathbf{\Phi_t}[i] \leftarrow \mathbf{x_t} - \mathbf{X_t}[i]$
4. Compute feature attribution matrix, $\mathbf{\Phi_t}$
6. Feed the dispersion measure to logistic regression classifier
$\boldsymbol{\sigma} \in \mathbb{R}^n$
7. Classify measurement data as benign or malicious
$\mathbf{z_t}$ classified as benign?
Yes
Use $\mathbf{z_t}$ for DSSE
No
Discard $\mathbf{z_t}$

Fig. 1: Schematic overview of DEFEAT

IV. METHODOLOGY

Figure 1 shows the architecture of DEFEAT, which is a filter through which every measurement vector must pass before being fed to model-based or data-driven DSSE. At any time step t, DEFEAT takes the measurement vector, $\mathbf{z_t}$, and outputs the probability that it is malicious. To do this, it uses two machine learning models: (a) a deep neural network, denoted as f_θ, estimating the state vector, $\mathbf{x}_t$, from the measurement vector, $\mathbf{z}_t$, so that per-feature attribution scores can be calculated (as discussed in Section IV-B), and (b) a classifier to separate malicious data from benign data. We note that f_θ could be the model used in data-driven DSSE or a deep neural network that is trained independently to approximate the DSSE process, i.e., mapping the measurement vector to the DSSE result.[2] The latter allows us to use DEFEAT to also safeguard model-based DSSE approaches, which use a physical model for state estimation.

A. Machine Learning Models

We choose the Stacked ResNetD model proposed for DSSE in [5] as f_θ. This model consists of three base learners, each being a 13-layer dense ResNet model that is trained to estimate the state given the measurement data. The output of these base learners are combined together and fed into a multivariate linear regression model, serving as the meta-learner. We use historical measurement-state pairs, i.e. $\{(\mathbf{z}_t, \mathbf{x}_t)\}_t$, to train the ensemble. Our choice of f_θ is motivated by the strong performance of Stacked ResNetD in DSSE [5] and its better robustness due to the use of ensemble learning [25].

---

[1] FGSM serves as the foundation of basic iterative method (BIM) and projected gradient descent (PGD). For this reason, we consider this particular adversarial attack in our evaluation.

[2] Any neural network that has enough learning capacity to accurately predict the system state can be used as the function approximator in DEFEAT.

We adopt a logistic regression model as our classifier. It takes as input the feature attribution scores (calculated using f_θ as discussed next) and outputs the probability that the measurement data is malicious.

B. Perturbation-based Feature Attribution Method

Perturbation-based feature attribution methods quantify the contribution of each feature of an input sample to the machine learning model's decision by perturbing the value of that feature. A famous perturbation-based attribution method is *leave-one-out*, which creates a new sample by occluding one feature in the input sample, i.e., replacing its value by zero. Both samples are then fed to the machine learning model and the amount of change in the output of this model is used to calculate the importance or *score* of that feature. Concretely, when the machine learning model is a C-class classifier that maps an input sample $\mathbf{z} \in \mathbb{R}^m$ to a probability vector of size C, Yang et al. [19] proposed calculating the score of the i^{th} feature, denoted as $\boldsymbol{\phi}(\mathbf{z})_i$, using the following equation:

$$\boldsymbol{\phi}(\mathbf{z})_i = f_\theta(\mathbf{z})_c - f_\theta(\mathbf{z} \odot (1 - \mathbf{s}_i))_c, \tag{4}$$

where $c = \arg\max_{j \in C} f_\theta(\mathbf{z})_j$ with $f_\theta(.)_j \in [0, 1]$ being the j^{th} element of the probability vector returned by the classifier f_θ, $\mathbf{s}_i$ is the one-hot encoding of the i^{th} feature of $\mathbf{z}$, and $\odot$ is the element-wise product. Repeating this process for all features yields the attribution vector $\boldsymbol{\phi}(\mathbf{z}) \in \mathbb{R}^m$. Then, the interquartile range (IQR) of data in $\boldsymbol{\phi}(\mathbf{z})$ is used to characterize statistical dispersion of the feature attribution scores:

$$\text{IQR}(\boldsymbol{\phi}(\mathbf{z})) = Q_3(\boldsymbol{\phi}(\mathbf{z})) - Q_1(\boldsymbol{\phi}(\mathbf{z})) \tag{5}$$

Here Q_3 and Q_1 denote the upper and lower quartile of the data, respectively.

As shown in [19], statistical dispersion of malicious data crafted by an adversarial attack is consistently larger than benign data. This implies that malicious data can be detected either by thresholding $\text{IQR}(\boldsymbol{\phi}(\mathbf{z}))$ or by fitting a logistic regression for the dispersion of the feature attribution scores on a training set, which eliminates the need for setting a threshold. We adopt this idea to detect data manipulation attacks. But, since DSSE is a multivariate regression problem rather than a classification problem, with f_θ being the Stacked ResNetD model, we have to make fundamental modifications to the approach described above. First, we use a feature attribution matrix, denoted as $\boldsymbol{\Phi}(\mathbf{z})$, instead of a feature attribution vector, because we have multiple state variables. Second, we update Equation (4) as DSSE predicts state variables rather than assigning probabilities to classes. Specifically, we define the i^{th} row of the feature attribution matrix, denoted as $\boldsymbol{\Phi}(\mathbf{z})_i \in \mathbb{R}^n$, as the difference between the output of the DSSE model on the true measurement data $\mathbf{z}$ and its output on $\mathbf{z} \odot (1 - \mathbf{s}_i)$ in which the i^{th} feature of the measurement data is masked:

$$\boldsymbol{\Phi}(\mathbf{z})_i = f_\theta(\mathbf{z}) - f_\theta(\mathbf{z} \odot (1 - \mathbf{s}_i)) \tag{6}$$

It follows from this definition that any column j of this matrix contains the attribution scores of all m features with respect to the j^{th} state variable ($\boldsymbol{\Phi}(\mathbf{z})^\top{}_j \in \mathbb{R}^m$). Notice that analyzing the dispersion of all data in $\boldsymbol{\Phi}(\mathbf{z})$ is not meaningful as they pertain to different state variables, so we calculate statistical dispersion of the data in each column separately, using Equation (5). Finally, we fit a logistic regression for the vector of dispersion measures obtained for the n columns. This model is used to classify data as malicious or benign. The main steps of DEFEAT are shown in Figure 1.

Remark. Since the attack strategy would not be known at the time of training the logistic regression classifier, it is not realistic to train this model on malicious data generated by the specific attack strategy that we evaluate it on. Thus, we train the logistic regression classifier on malicious data generated by the basic FGSM with the perturbation factor $\epsilon = 0.1$ and a convolutional neural network (CNN) used as the surrogate model (see [9] for details). Once trained, we evaluate its performance on various attack strategies.

V. EXPERIMENTAL SETUP

A. Baseline Detection Approaches

We briefly introduce three detection methods that have been proposed for safeguarding DSSE, namely residual-based BDD with hypothesis testing, Kullback–Leibler Divergence (KLD)-based detector, and Neural Attack Detector (NAD). These methods serve as our baseline.

Residual-based Bad Data Detection (BDD): Residual-based BDD is a widely used strategy to identify the presence of bad measurement data that might occur due to sensor drift and bias, interference, and incorrect topology information [2]. This method typically runs on top of the WLS-based state estimator by processing the measurement residuals. Once the state is estimated, it is utilized in the residual-based BDD mechanism to get an estimate of the measurement vector, denoted as $\hat{\mathbf{z}}{=}\mathbf{H}\hat{\mathbf{x}}$. Then, the measurement error is calculated as $\mathbf{e}{=}\mathbf{z}-\hat{\mathbf{z}}$.

From (1), the residual can be rewritten as:

$$J(\mathbf{x}) = \sum_{i=1}^{m} \frac{e_i^2}{R_{ii}} = \sum_{i=1}^{m} \left(\frac{e_i}{\sigma_i}\right)^2 \tag{7}$$

Notice that Equation (7) is of the form $y = \sum_{i=1}^{d} \chi^2$, resembling the chi-squared distribution with d degrees of freedom. Given the assumption $m > n$, at most $(m-n)$ of the measurement residuals will be linearly independent, resulting in $d = m-n$. Thus, to detect the presence of bad measurement data, $J(x)$ is compared to the critical chi-square value at the degree of freedom d, and a pre-specified level of significance α. If $J(x) < \chi^2_{d,\alpha}$, then the estimated state, $\hat{\mathbf{x}}$, can be trusted. Otherwise, the measurement is assumed to contain bad data.

KL Divergence-based Detection (KLD): This effective strategy to mitigate FDIA on state estimation is proposed in [17]. The basic idea is to compare the KLD between two distributions $\mathbf{p}_k$ and $\mathbf{q}$, denoted as $D_{KL}(\mathbf{p}_k \| \mathbf{q})$, against a predefined threshold, τ, at any time step k, to decide if the measurement sample $\mathbf{z}_k$ contains bad data. Let $\mathbf{q}$ be the distribution of average *measurement variations* obtained from the historical data and $\mathbf{p}_k$ be the distribution of measurement variations at time step k, defined as $\mathbf{p}_k{=}\mathbf{z_k}-\mathbf{z_{k-1}}$. When all measurements are expressed in the per unit system, most of the measurement variations, i.e., the elements of $\mathbf{p}_k$, should be close to zero, even though not all measurements are of the same type and they are from sensors installed at different locations in the network. In other words, regardless of the type of the measurement, the difference between two consecutive measurements obtained from any sensor can be treated as the realization of one random variable. If no bad data is injected into $\mathbf{z_k}$, then the distribution of measurement variation at time step k, i.e. $\mathbf{p}_k$, should be similar to the distribution of average measurement variations obtained from historical data, i.e. $\mathbf{q}$. Thus, $D_{KL}\left(\mathbf{p}_k \| \mathbf{q}\right)$ should be smaller than τ.

Neural Attack Detection (NAD): Malicious data can be detected using a deep neural network trained on a mixture of benign and malicious data. This neural network, called NAD, was initially introduced in [7] to protect transmission system state estimation. It has six hidden layers. ReLU is used as the activation function of hidden layers and Softmax is the activation function of the output layer. We train this NAD model using adversarial data generated by the basic FGSM attack crafted using a CNN surrogate with ϵ=0.1, and evaluate it using a more diverse and larger set of adversarial samples.

B. Simulation Scenario

Distribution Network: To obtain training and test datasets, we adopt the IEEE 33-bus system presented in [26], which was extended by using the IEEE European low voltage test feeder[3] as the secondary networks. Each of the primary buses, except the first one, is connected to a low-voltage feeder (i.e. the secondary network), powering 55 single-phase loads. We utilize the Multifamily Residential Electricity Dataset (MFRED) [27] to model these loads. This dataset comprises daily load profiles from 390 US apartments, recorded at 15-minute intervals over a span of 12 months (from January 2019 to December 2019). The load data is segmented into 26 groups, with each group encompassing the average real and reactive power usage of 15 apartments. We augment the original dataset by adding Gaussian noise with standard deviations of $1\%, 2\%, \cdots, 10\%$ to each of the 26 household load data to generate 286 distinct apartment load data including the original 26 households. Then, we create 500 hypothetical buildings, each consisting of 1 to 10 apartments chosen randomly from the 286 apartments. These hypothetical buildings are then connected to the secondary buses, serving as the system loads. We assume that each building is equipped with smart meters, providing load data at 15-minute intervals. Additionally, we assume six buses in the primary bus system are equipped with D-PMUs providing phasor information. The number of D-PMUs is chosen following [26] as this level of observability led to reasonable state estimation performance. Note that determining the optimal number of measurement devices and their location is outside the scope of this work, so we just tried one reasonable sensor placement strategy. We conduct an AC power flow analysis on this network using the Open Distribution Simulator Software (OpenDSS) [28] to generate the training and test datasets. In the real world, the training dataset can be constructed in a similar manner by utilizing historical load and generation data to solve power flow equations and determine the states.

State Variables: Our approach to defining measurements and states aligns with the method outlined in [29]. Specifically, state variables are represented by bus voltage phasors denoted as $\mathbf{x_t}$=$[\mathbf{v}_t^1, \cdots, \mathbf{v}_t^b, \boldsymbol{\theta}_t^1, \cdots, \boldsymbol{\theta}_t^b]$, with b denoting the number of buses not equipped with D-PMUs, and $\mathbf{v_t^i}$ and $\boldsymbol{\theta}_t^i$ denoting the vectors that contain the three-phase voltage magnitudes and phase angles of bus i at time step t, respectively. The system state is represented by the three phase voltage magnitudes and phasor angles of 27 buses that are not equipped with D-PMUs. Thus, the state vector is of size 27×3×2=162. On the other hand, various combinations of redundant network data, including bus voltage phasors, real and reactive power consumption, and branch flows, can serve as measurements for the DSSE process. We define the measurement vector as a combination of real and reactive power consumption obtained from the MFRED dataset (in real scenarios this data can be obtained from smart meters connected to the household loads) and the voltage magnitude measurements obtained from the buses equipped with D-PMUs. Considering the first bus as slack bus, we have three-phase real and reactive power measurements from 32 load buses and three-phase voltage magnitude values from the six D-PMU installed buses. Thus, the measurement vector is of size $(32 \times 3 \times 2) + (6 \times 3) = 210$.

We split the entire dataset into three portions– (a) training data for the learning-based detection methods, i.e. DEFEAT and NAD, (b) training data for the surrogate models used in adversarial attack generation, and (c) test data. We utilize the data from the first half of each month to train machine learning models for DEFEAT and NAD. Given the dataset's 15-minute resolution, this results in a total of 17,280 training samples. From the remaining days, we randomly select three to construct the test dataset. This process yields 3,456 test samples, organized into 12 groups of 288 consecutive measurements. These groups, evenly distributed across the one-year window, comprise data from three consecutive days per month, with 96 samples per day. The remaining samples, spanning 12 days in the second half of each month, are reserved for training the surrogate models during adversarial attack generation.

VI. RESULTS

To evaluate DEFEAT and baseline detection methods, we create malicious data using the attack strategies presented in Section III-B. Specifically, one batch of malicious data is generated according to the FDIA strategy. For each of the black-box, untargeted attacks, nine batches of adversarial data are generated using three different surrogate models and three ϵ values for each surrogate model. The architecture of these surrogate models is described in [9]. For the white-box, targeted attack, the attacker does not need to train a surrogate model as they know the architecture and parameters of the victim DSSE model, i.e. Stacked ResNetD. Thus, we just create three batches of adversarial data using three ϵ values.

To evaluate each detection method, we plot the receiver operating characteristic (ROC) curve by adjusting the threshold used in each detection method to separate the two classes. The ROC curve is the plot of the true positive rate (TPR) against the false positive rate (FPR), with TRP being the proportion

---

[3] https://cmte.ieee.org/pes-testfeeders/resources/

TABLE II: Comparing performance of DEFEAT with baseline detection methods on different data manipulation attacks.

Attack	Surrogate Model	ϵ	AUC			
			BDD	NAD	KLD	DEFEAT
FDIA	N/A	N/A	0.507	0.944	1.000	0.999
FGSM	Stacked ResNetD	0.05	0.595	0.576	0.999	0.999
		0.15	0.825	0.799	0.999	1.000
		0.30	0.942	0.882	1.000	1.000
	CNN	0.05	0.680	0.999	0.782	0.999
		0.15	0.872	1.000	0.855	1.000
		0.30	0.899	1.000	0.874	1.000
	MLP	0.05	0.653	0.999	0.884	1.000
		0.15	0.884	1.000	0.925	1.000
		0.30	0.943	1.000	0.929	1.000
Sneaky FGSM	Stacked ResNetD	0.05	0.536	0.526	0.947	0.952
		0.15	0.766	0.567	0.961	0.963
		0.30	0.900	0.608	0.962	0.970
	CNN	0.05	0.519	0.650	0.591	0.889
		0.15	0.521	0.618	0.635	0.940
		0.30	0.577	0.618	0.653	0.946
	MLP	0.05	0.512	0.959	0.513	0.892
		0.15	0.718	0.971	0.518	0.944
		0.30	0.848	0.977	0.519	0.957
Targeted FGSM	N/A	0.05	0.565	0.337	0.938	1.000
		0.15	0.734	0.760	0.995	1.000
		0.30	0.923	0.956	0.997	1.000
DeeBBAA	Stacked ResNetD	0.4	0.612	0.918	0.508	0.998
	CNN	0.4	0.535	0.930	0.873	0.994
	MLP	0.4	0.561	0.949	0.656	0.999

of adversarial measurement data that are classified correctly and FPR being the proportion of benign measurement data that are misclassified as malicious. We use the area under the curve (AUC) of the ROC curve to compare different detection methods. For each attack, the detection method that has the highest AUC is the most successful one. Table II summarizes the results for all 21 cases. Figure 2 also shows the ROC curves of DEFEAT and baseline detection methods for 5 of these cases. The plots for the remaining 16 cases are omitted to save space.

It is evident from the table that DEFEAT outperforms the three baseline methods in detecting various types of adversarial attacks generated with different perturbation factors, ϵ. While NAD shows comparable performance in detecting some of the attacks, its performance is not consistent across all the attacks, especially in case of Sneaky FGSM and Targeted FGSM with smaller ϵ values. This variable performance can be attributed to the fact that NAD was trained on adversarial data generated solely by the basic FGSM attack, highlighting the inherent limitation of neural attack detectors as they are constrained by the data used for their training. On the other hand, the KLD-based detection method has subpar performance, although it is slightly better than BDD. Overall, our analysis reveals that *unlike traditional FDIA, surface-level analysis of measurement data (e.g. applying a threshold to their residual or statistical properties of their distribution) is not sufficient to detect adversarial attacks.* We expand on this in the next section.

VII. DISCUSSION

A. Why Does the Dispersion of Feature Attribution Scores Reveal Malicious Data?

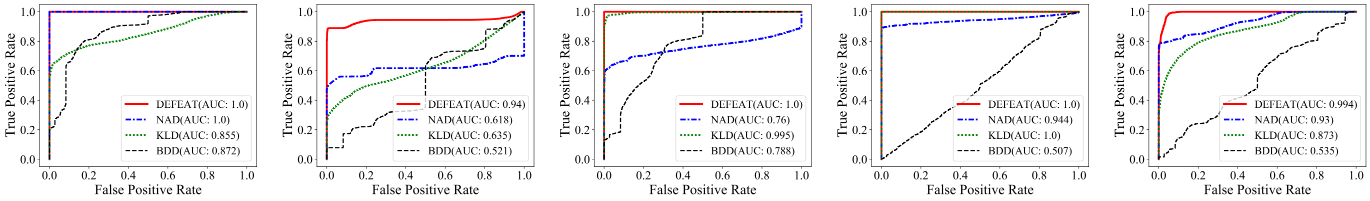
Adversarial data are crafted by adding small perturbations that maximally confuse the victim model. Hence, they are often indistinguishable from the original data in terms of statistical properties, such as mean, variance, etc. Conventional and statistical detection methods, such as BDD and KLD, rely on the assumption that the injected data follows a specific pattern, which is violated by adversarial attacks. As a result, despite being effective in detecting FDIA, they cannot offer a reliable approach for detecting adversarial attacks. This is evident from Figure 3a, which shows the histogram of KLD values for the benign and adversarial measurements, $D\left(p_{benign}\|q\right)$ and $D\left(p_{adv}\|q\right)$, respectively. Here, q represents the measurement variation distribution obtained from the historical data. It can be readily seen that there is a significant overlap between the two histograms, making it impossible to find a good threshold (τ) to achieve high TPR and low FPR simultaneously. Adversarial attacks and stealthy FDIAs, however, leave a trace on the representations used by neural networks for classification or regression. Thus, the key to identifying adversarial data lies in examining how the neural network's decision-making process is affected by these data samples, which can be done using an attribution method. Figure 3b compares the histogram of the probability produced by the logistic regression classifier used in DEFEAT. Observe that there is no overlap between the two histograms, making it easy to classify the data.

To get a better understanding of what DEFEAT does, the feature attribution scores of each of the 210 measurements (rows) for each of the 162 estimated states (columns) for a particular measurement sample $\mathbf{z_t}$ have been presented in Figure 4. A notable contrast emerges between the benign and malicious versions. Inspecting the plots that pertain to malicious data reveals that perturbations introduced by the attacker result in the misidentification of some features that are not actually important as highly influential. Consequently, this leads to erroneous estimations, which explains the susceptibility of the DSSE process to malicious data.

Apart from its superior performance, a major advantage of DEFEAT is its *generalizability* across different victim models. Unlike the conventional BDD that uses the states estimated by the victim model to calculate the measurement residual, DEFEAT uses raw sensor data, $\mathbf{z}_t$, collected at time step t and classifies it as benign or malicious. Thus, its performance remains consistent across different victim models.

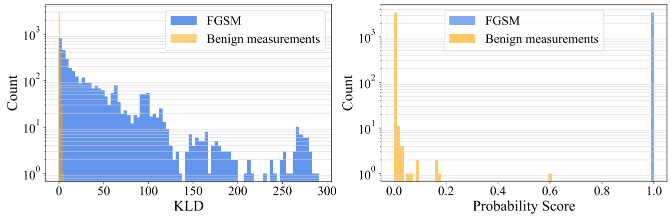
B. Sensitivity to Measurement Noise

In the previous experiments, we considered an ideal scenario with no measurement noise. However, electrical measurement



(a) FGSM (b) Sneaky FGSM (c) Targeted FGSM (d) FDIA (e) DeeBBAA

Fig. 2: ROC curves of detection methods under different attacks. Note that for the black-box adversarial attacks (i.e. FGSM, Sneaky FGSM, and DeeBBAA), a CNN consisting of 3 convolutional layers and 3 dense layers with ReLU activation is used as the surrogate model.



(a) (b)

Fig. 3: Histogram of (a) KLD and (b) output of DEFEAT approach for the benign and adversarial samples crafted with vanilla FGSM ($\epsilon = 0.15$) using the MLP surrogate. Note the y-axis is in logarithmic scale.

devices do not measure the physical quantity with 100% accuracy. To understand the robustness of the proposed DEFEAT approach in a more realistic scenario, we evaluate DEFEAT on noisy test data samples. We generate the noisy data by adding Gaussian white noise with standard deviation $\sigma = 0.01$ to the test dataset. The standard deviation is chosen such that the error is in the range of $\pm 3\%$ in the vast majority of cases.[4] Figure 5 compares the logistic regression output on benign samples, noisy samples, and samples created by Stealthy FDIA, FGSM, and Sneaky FGSM (representing stealthy adversarial attacks). It can be readily seen that measurement noise has little impact on DEFEAT's performance. Despite being trained on noise-free data, DEFEAT successfully classifies noisy data as benign, while still identifying malicious data generated by FDIA and FGSM. This is because DEFEAT assigns relatively higher probability scores to malicious data than noisy data, leading to a clear separation between the two classes. Even in the case of Sneaky FGSM, which is the most difficult adversarial attack to detect as evident from Table II, one can still find a threshold that results in an acceptable trade-off between TPR and FPR. This confirms that data manipulation attacks leave a trace on the representations used by neural networks, and therefore, are distinguishable from the effect of measurement noise through the perturbation-based feature attribution method used in this work.

VIII. CONCLUSION

We introduce DEFEAT, a robust feature attribution-based detection strategy aimed at safeguarding DSSE by detecting the presence of malicious data. Through comprehensive

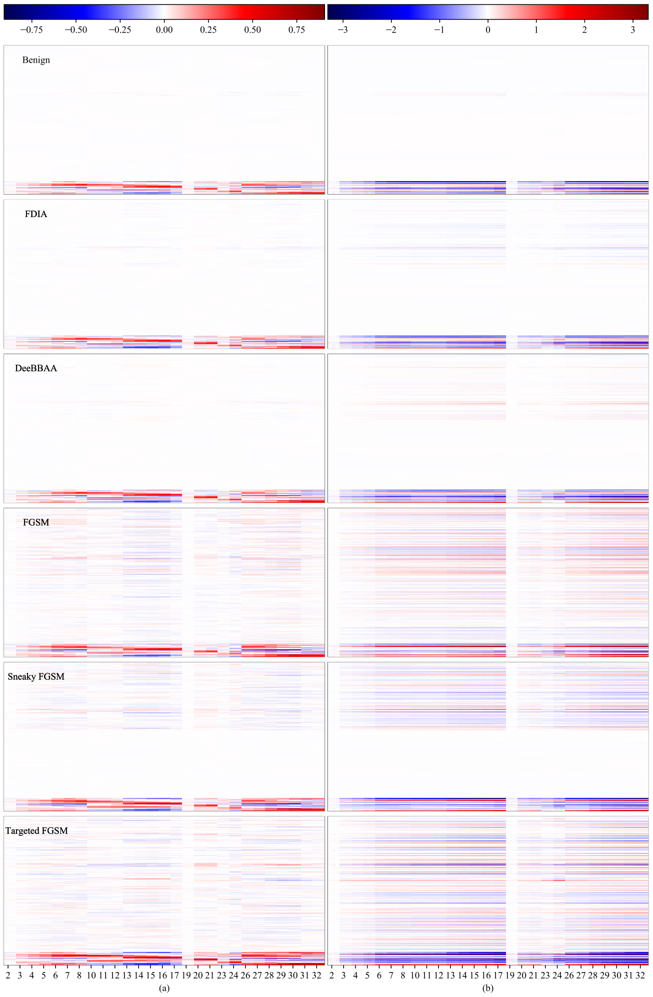


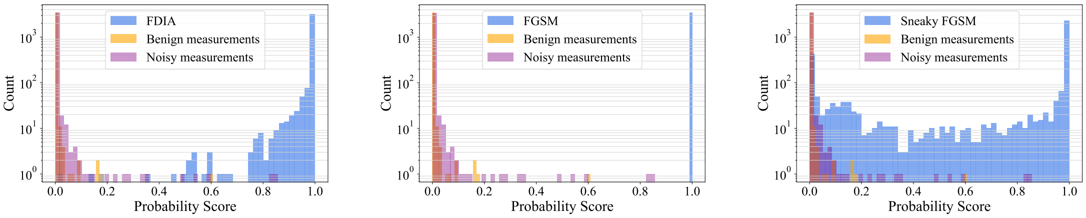
Fig. 4: Feature attribution score of each measurement feature for (a) voltage magnitude and (b) phase angle estimation at different buses. The x-axis labels are bus numbers.

evaluation, we demonstrate the effectiveness of this approach in identifying FDIAs and adversarial attacks generated using various surrogates, even in the presence of measurement noise. We explain why conventional and statistical detection methods often fail to detect adversarially crafted data and shed light on the superior performance of DEFEAT.

This study lays the groundwork for several promising research directions in the areas of data-enabled optimization and cybersecurity in the smart grid. One intriguing direction involves evaluating DEFEAT under a stronger assumption

---

[4] The European standard for energy meters (EN 50470-3:2006) allows measurement error of up to $\pm 2.5\%$ [30].



(a) (b) (c)

Fig. 5: Output of the logistic regression model for benign, noisy (natural measurement noise) and malicious data generated using (a) FDIA, (b) FGSM, and (c) Sneaky FGSM. Note the y-axis is in logarithmic scale.

where adversaries have access to the structure or inner working of the detection method. Another direction is to evaluate its performance under sparse attacks where the adversary can compromise only a subset of sensors. Designing algorithms to identify the attack point and developing data-driven DSSE techniques that are inherently robust to malicious data are also deferred to future work.

REFERENCES

[1] C. Lu, J. Teng, and W.-H. Liu, "Distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 10, no. 1, pp. 229–240, 1995.

[2] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

[3] Y. Weng *et al.*, "Robust data-driven state estimation for smart grid," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1956–1967, 2016.

[4] Z. Cao *et al.*, "Scalable distribution systems state estimation using long short-term memory networks as surrogates," *IEEE Access*, vol. 8, pp. 23 359–23 368, 2020.

[5] N. Bhusal *et al.*, "Deep ensemble learning-based approach to real-time power system state estimation," *Int J. Electr. Power Energy Syst.*, vol. 129, p. 106806, 2021.

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

[7] J. Tian *et al.*, "Joint adversarial example and false data injection attacks for state estimation in power systems," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13 699–13 713, 2021.

[8] ——, "Adversarial attack and defense methods for neural network based state estimation in smart grid," *IET Renew. Power Gener.*, vol. 16, no. 16, pp. 3507–3518, 2022.

[9] A. Afrin and O. Ardakanian, "Adversarial attacks on machine learning-based state estimation in power distribution systems," in *Proc. 14th ACM Int. Conf. Future Energy Syst.*, 2023, pp. 446–458.

[10] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–33, 2011.

[11] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks against nonlinear state estimation in smart power grids," in *2013 IEEE Power & Energy Society General Meeting*. IEEE, 2013, pp. 1–5.

[12] J. James, Y. Hou, and V. O. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Trans. Industr. Inform.*, vol. 14, no. 7, pp. 3271–3280, 2018.

[13] B. Li *et al.*, "Detecting false data injection attacks against power system state estimation with fast go-decomposition approach," *IEEE Trans. Industr. Inform.*, vol. 15, no. 5, pp. 2892–2904, 2019.

[14] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2019.

[15] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Trans. Industr. Inform.*, vol. 13, no. 1, pp. 198–207, 2015.

[16] A. Bhattacharjee, S. Mishra, and A. Verma, "Deep adversary based stealthy false data injection attacks against AC state estimation," in *14th Asia-Pacific Power Energy Eng. Conf.* IEEE, 2022, pp. 1–7.

[17] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in AC state estimation," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[18] W. Samek *et al.*, "Explaining deep neural networks and beyond: A review of methods and applications," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 247–278, 2021.

[19] P. Yang *et al.*, "ML-LOO: Detecting adversarial examples with feature attribution," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, 2020, pp. 6639–6647.

[20] A. Sayghe, J. Zhao, and C. Konstantinou, "Evasion attacks with adversarial deep learning against power system state estimation," in *2020 IEEE Power & Energy Society General Meeting*. IEEE, 2020, pp. 1–5.

[21] A. Bhattacharjee *et al.*, "Deebbaa: A benchmark deep black box adversarial attack against cyber-physical power systems," *arXiv preprint arXiv:2303.09024*, 2023.

[22] G. Fidel, R. Bitton, and A. Shabtai, "When explainability meets adversarial learning: Detecting adversarial examples using shap signatures," in *Proc. Int. Joint Conf. Neural Networks*. IEEE, 2020, pp. 1–8.

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Int. Conf. Learn. Represent.*, 2015.

[24] A. Afrin and O. Ardakanian, "On brittleness of data-driven distribution system state estimation to targeted attacks," in *Proc. 15th ACM Int. Conf. Future Energy Syst.*, 2024.

[25] F. Tramr *et al.*, "Ensemble adversarial training: Attacks and defenses," in *Int. Conf. Learn. Represent.*, vol. 1, 2018, p. 2.

[26] M. M. Haji and O. Ardakanian, "Practical considerations in the design of distribution state estimation techniques," in *2019 IEEE Int. Conf. Commun., Control, Comput. Techn. Smart Grids*. IEEE, 2019, pp. 1–6.

[27] C. Meinrenken, "MFRED (public file, 15/15 aggregate version): 10 second interval real and reactive power in 390 US apartments of varying size and vintage," *Harvard Dataverse, V1*, 1 2024.

[28] R. C. Dugan and T. E. McDermott, "An open source platform for collaborating on smart grid research," in *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–7.

[29] A. Primadianto and C.-N. Lu, "A review on distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3875–3883, 2016.

[30] R. Q. Cetina, A. J. Roscoe, and P. Wright, "A review of electrical metering accuracy standards in the context of dynamic power quality conditions of the grid," in *2017 52nd International Universities Power Engineering Conference (UPEC)*. IEEE, 2017, pp. 1–5.