A Novel Adversarial FDI Attack and Defense Mechanism for Smart Grid Demand-Response Mechanisms

Guihai Zhang and Biplab Sikdar

***Abstract*—This paper focuses on enhancing the cybersecurity of cyber-physical systems, with a particular emphasis on the False Data Injection (FDI) attack within the Demand Response (DR) mechanism in smart grids. DR seeks to introduce flexibility in consumers' electricity consumption through dynamic pricing or financial incentives, aiming to optimize the equilibrium between supply and demand. The vulnerability of DR to FDI attacks becomes particularly evident when considering its reliance on accurate demand data. In emphasizing the importance of fortifying DR's security against FDI, the Ensemble and Transfer Adversarial Attack (ETAA) based on Adversarial Machine Learning (AML) techniques is proposed. This method facilitates the injection of false data with reduced detectability by existing neural network-based detection method. With the general framework of ETAA, any gradient-based adversarial attack method can be integrated to achieve enhanced attack transferability across diverse detection models. To counteract such attacks, the training process of detection models is refined through three key steps: Gaussian noise injection, latent feature combination and probability margin enlargement. Evaluation results demonstrate that the ETAA method executes FDI attacks with a higher success rate compared to benchmark methods. Furthermore, defensive training contributes to elevating the performance of detection models, ensuring higher standard accuracy, and reducing the success rate of AML attacks. This paper underscores the critical need to enhance the security of DR mechanisms to mitigate the impact of sophisticated FDI attacks on the robustness of smart grids.**

***Index Terms*—Smart grid, cyber-physical security, demand response, false data injection, adversarial machine learning, adversarial data, defense of adversarial data.**

I. Introduction

A smart grid exemplifies a cyber-physical system that seamlessly integrates traditional power systems with modern communication and information technologies, utilizes real-time data for monitoring and control, and leverages advanced technologies to optimize the efficiency and reliability of electrical power systems. The Demand Response (DR) scheme is a key component in the progression of smart grids, playing a central role in proactively aligning power supply with demand and mitigating power fluctuations. DR implements time-based rates, such as Day Ahead Pricing, Time-of-Use rates, and Real-Time Pricing, or by offering various financial incentives. These measures are designed to motivate customers to adjust or reduce their electricity consumption during peak hours. In such initiatives, both wholesale market electricity costs and retail rates are reduced [1]. The integration of DR into smart grids not only brings forth innovative energy management capabilities but also exposes it to potential cyber-attacks that could compromise its integrity and reliability [2]. Embarking on this transformative journey, the vulnerability of the Demand Response mechanism to sophisticated cyber threats, particularly False Data Injection (FDI) attacks, becomes a critical consideration.

The FDI attack represents a prevalent cyber threat to contemporary cyber-physical systems. Attackers can employ FDI attack to modify necessary data by directly injecting fabricated data, compromising smart meters, or manipulating data transmission to induce varying degrees of consequences. Potential impacts encompass power flow fluctuations and inaccurate power flow analyses [3]. More severe outcomes, such as surpassing operational limits and breaching safety constraints, may lead to overloads, power failures, and blackouts. These attacks not only pose significant economic risk to power companies but also adversely impact downstream industries and consumers. Research has explored Deep Learning (DL) neural network-based approaches to enhance the efficacy of FDI attack detection in smart grid systems when compared to conventional mathematical and statistical detection methods. However, Adversarial Machine Learning (AML) techniques have the capability to exploit carefully crafted data, disrupting the network-based detection model and compromising the defense against FDI attacks. When the false data provided to a DL-based detection model is carefully manipulated, the model may be led to generate inaccurate predictions for that data, influencing subsequent decision-making. In response to this pressing concern, this paper seeks to contribute to the enhancement of cybersecurity, with a specific focus on securing the DR mechanism against FDI attacks.

A. Related Work

1) Demand Response Schemes: DR schemes predominantly rely on the interactions between a power utility company and its customers to fulfill both their current and future power demand requirements. Users are primarily presented with two main programs: incentive-based and price-based programs. In incentive-based DR programs, customers receive remunerations for reducing their demand, while in price-based programs, customers encounter varying electricity costs at different time intervals [4]. The typical process flow in price-based programs is shown in Fig. 1. The procedure commences as the power company communicates the original unit price to consumers at a specified time, typically a day ahead. Prices vary by time slot, with higher rates during peak hours. Consumers strategically plan their electricity usage considering factors such as starting times and durations, aligning them with the fluctuating prices with the help of Home Energy Management System. An aggregator may gather demand forecast data, transmitting it to the power company, which then adjusts prices through optimization functions. This iterative process concludes at an optimal point where both the power company and consumers accept the price, leading to the calculation of bills based on actual electricity

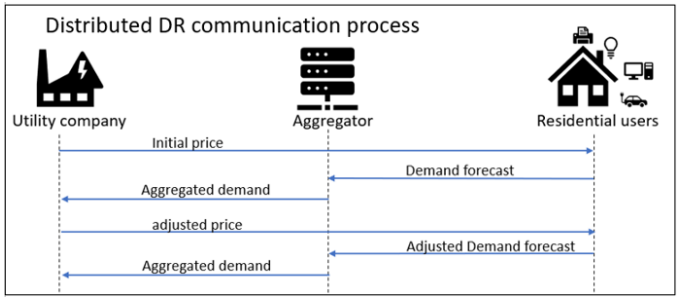

Fig. 1. Distributed DR communication sequence.

usage from devices following the predefined schedule. Several DR optimization methods have been introduced. The work in [5] proposes algorithms to maximize retailer profits and minimize the combined residential costs when plug-in electric vehicles are considered. Similarly, [6] has assessed rule-based and machine learning-based control algorithms for the implementation of demand response strategies in the residential sector and demonstrated their positive contribution to the reduction of electricity usage and utility generation cost.

As demonstrated in [7] and [8], attackers can exploit strategic FDI attacks to gain financial advantages within real-time-based pricing DR schemes. These discoveries emphasize the essential importance of establishing timely detection mechanisms for such attacks in the implementation DR schemes. Specifically, in [9], the integration of Conventional Neural Network and Spectral Residual is introduced for the detection of false data injection attacks. Nevertheless, it is noteworthy that the network-based model is vulnerable to targeted adversarial data, rendering it susceptible to compromise.

2) Adversarial Machine Learning: AML methods can generate fake data in a non-linear and more complex way, which has better performance in eluding identification than traditional data-driven methods like the random matrix approach for grid measurements in [10] and the linear attacked model for DC motor output in [11]. The inception of AML can be traced back to the work in [12], involving imperceptible small alterations to input image pixels and deceiving the network into producing inaccurate output labels. The resulting adjusted input data is termed an adversarial example. The application of AML extends seamlessly to power systems, as highlighted in [13], which explores the vulnerabilities of DL models to AML attacks and proposes the Saliency Map Attack to state estimation. The potential risks of AML attacks on load forecasting networks are scrutinized in [14], emphasizing the exploitation of historical data. The earlier work in [15] introduced a novel iterative FGV method for generating adversarial data specifically for DR application. While iterative methods such as Basic Iterative Method (BIM) and Momentum Iterative Fast Gradient Sign Method (MIM) [16] have advanced attack capabilities by incorporating momentum from previous steps, their efficacy diminishes when applied to models beyond the intended target, resulting in a lower success rate. These methods often generate adversarial data from a single substitute model, lacking assurance in transferability to unknown models. The work in [17] explored improving attack success rates by using ensemble methods. However, the sequential generation of adversarial data from various models hinders attack generalization. Additionally, existing approaches typically employ the same type of network for both data generation and attack detection, which further reduces the attack transferability. Therefore, this paper enhances the generalization of attacks by employing a parallel ensemble and transfer updating process, utilizing fused entropy loss to generate false data injected into DR demand data.

3) Defense from Adversarial Data: Adversarial training is the most common defensive choice in existing works. In adversarial training, the model is trained with an augmented dataset that includes additional adversarial data, enabling the model to learn latent patterns associated with adversarial inputs. The PGD-AT [18] is a representative method of adversarial training, employing BIM perturbations in data modification during training. Another variations of adversarial training is [19], which trains the model with adversarial data slightly crossing the decision boundary. However, these methods only rely on a single model structure and use only the modified data as training data. Therefore, the information in original data are missed, and the trained model would not successfully classify attack data from various other unknown inputs. Moreover, [20] introduces noise generated from gradients to the hidden layer output to improve the model robustness. Although this method further improved the classification accuracy, it lacks generalization.

Besides adversarial training, ensemble training that involves the incorporation of adversarial data from multiple models has been published. The work in [21] exemplifies this approach by using many pre-trained models to prepare the training data. Another study [22] develops a sequential ensemble comprising an autoencoder, convolutional-recurrent, and feed-forward model to identify any energy theft. Nonetheless, these methods have the potential risks of overfitting to the training data, especially in scenarios where the diversity within the normal data is restricted. The adaptive diversity promoting regularizer to encourage diversities of non-maximal predictions from different models utilized in [23] acts as another example for ensemble training. However, this regularizer requires much more computational cost. Building upon existing works, this paper introduces a novel adversarial training process, an ensemble-based method aimed at improving model performance against adversarial data. The processes of noise injection and feature combinations work together to address the abovementioned drawbacks in existing methods.

B. Proposed Attack and Defensive Training

First, this paper proposes the Ensemble and Transfer Adversarial Attack (ETAA). This method introduces a sophisticated approach for generating adversarial data to be injected into demand data for DR application, strategically reducing detectability by existing neural network-based detection methods. The proposed ETAA method tackles the challenges related to transferability and stealthiness present in current methodologies. It does so by employing an ensemble strategy to minimize disparities among various models and integrating a zero-mean projection to enhance the stealthiness of the adversarial attack. Next, to counteract false data generated from adversarial attacks,

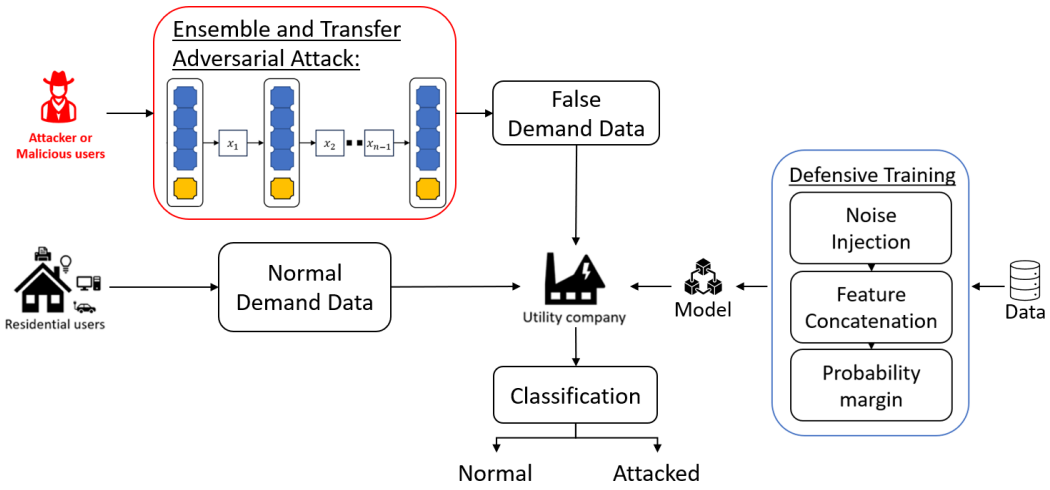

Fig. 2. General framework of the proposed attack (red box) and defense (blue box) methods.

Defensive Training is introduced. This approach utilizes the ensemble concept in model construction, incorporating processes of noise injection and a larger probability margin. These processes solve the issues of overfitting and help to further improve the classification accuracy. Evaluation results demonstrate that the ETAA method allows for a more generalized adversarial attack with a higher success rate. Defensive Training is shown to be effective in creating a model with superior robustness and the ability to defend against adversarial samples compared to benchmark adversarial training methods. The overall framework of the proposed methods are illustrated in Fig. 2. In the normal scenarios users provide the benign demand data to the utility company and the data are classified by the detection model as normal or attacked data before further DR optimization process. The detection model is the Deep Learning network trained by the utility company using historical data. However, there are chances that attackers or malicious users can provide false demand data with purposes like energy theft and disrupt normal processing. The proposed ETAA method is highlighted in the red box to conduct the generation of false data. To improve the security of the DR scheme and capture these finely produced adversarial data, the Defensive Training method is proposed and highlighted in the blue box.

This paper makes the following principal contributions:

- This paper signifies impactful research in the analysis of adversarial FDI attacks within the cybersecurity framework of DR schemes.
- A generalized adversarial attack is proposed for generating false demand data in DR schemes, demonstrating improved transferability across various detection models, named as Ensemble and Transfer Adversarial Attack.
- The Defensive Training is introduced to enhance the robustness of the detection model against adversarial false data.
- Simulation based evaluation is performed to illustrate the superior performance of both the proposed ETAA method and Defensive Training compared to existing benchmark methods specifically in terms of attack success rate and classification accuracy.

The rest of the paper is organized as follows. Section II is the methodology part to describe the details of ETAA to generate adversarial data. Section III explains the process of Defensive Training. Section IV presents the evaluation and comparison results. Finally, Section V gives a summary of this paper.

II. Adversarial Attack

A. Adversarial False Data Injection Attack

The proposed method to perform the FDI attack is to only change the demand data. The goal is to generate adversarial false demand data that can bypass the detection from neural network-based model such as the method in [9]. Note that attacking the communication link for dropping packets or launching denial-of-service attacks etc. is not the focus of this paper. The conventional statistical Bad Data Detection in power systems are not considered, as this work stands as a pioneering effort to illuminate the susceptibilities associated with implementing network-based FDI detection within a DR scheme. If the demand data is falsified and used for further DR optimization, the outcome will change and result in problems like the power supply and demand imbalance, energy theft and even blackouts. For example, as shown in the analysis in [8], an adversary can achieve significant bill cost reductions with a small amount of demand changes, e.g., 1% of demand manipulation can lead an average of 10% bill reduction for the attacker. The emphasis of the proposed attack method lies exclusively on generating adversarial demand data and assessing the performance of various network-based models when exposed to well-tuned adversarial data. Consequently, the goal is to design a method to fortify the model against such adversarial challenges and to correctly capture those adversarial false data.

Utilizing identical configurations as presented in [9], [15], a complete day of 24 hours is divided into 48 time slots, with each slot spanning a duration of 30 minutes. Hence, the vector $D = [d_1, d_2, \cdots, d_n]$ represents the collected demand

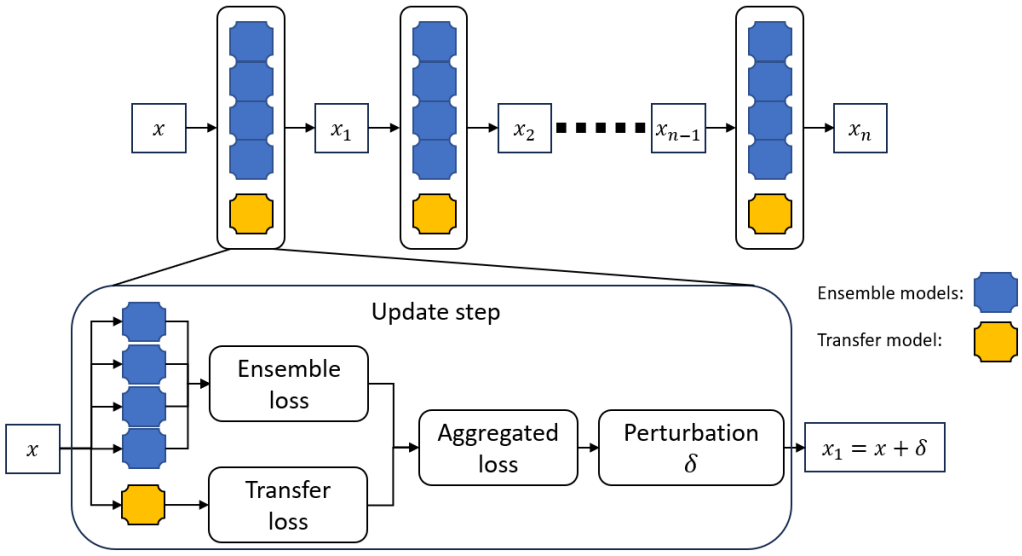

Fig. 3. Ensemble and Transfer Adversarial Attack framework.

forecasts without attacks, where n is 48. The demand forecasts with modified false data are denoted as $\hat{D} = [\hat{d}_1, \hat{d}2, \cdots, \hat{d}n]$. Attackers aim to increase the values at any random time slot i, leading to $\hat{d}_i \geq d_i$. As assessed in [7], the adversary ensures a successful and significant cost benefit of bill reduction by injecting values that increase by more than 0.1% of the overall demand. Consequently, the modified demand data is still labeled as normal ("0") if this 0.1% increment is not met. The demand value vector is labeled as attacked ("1") whenever there is at least one time slot with a value increment greater than 0.1%. Given a network f_θ responsible for classifying the demand values D, the prediction output is denoted as $f_\theta(D)$. Furthermore, for stealthy adversarial data injection, it is advantageous to maintain the sum of modified demand forecasts close to the normal demand forecasts. Mathematically, the FDI attack is formulated as:

$$max : \hat{D} \tag{1}$$

$$subject\ to : f_\theta(\hat{D}) = \text{normal}, \tag{2}$$

$$: \sum(\hat{D}) = \sum(D). \tag{3}$$

B. Gradient AML Attacks

Gradient-based method is the most common way to generate adversarial data where the gradient of the targeted label's loss with respect to the input data of a DL network is computed. Then, the data is moved along the gradient ascent or descent direction. The popular gradient-based methods include:

1) FGSM: Fast Gradient Sign Method is a single-step method. The gradient is calculated with respect to the input and the sign of that gradient is added to the data to get the adversarial data. The formula is:

$$D' = D + \epsilon \cdot sign(\nabla_D L(f_\theta(D), Y)). \tag{4}$$

There is an updating factor ϵ to control the amount of changes and the loss $L(f_\theta(D), Y)$ is any loss function that takes the input D and target label Y. Function ∇_D is the general process to get the gradients with respect to D.

2) BIM: This Basic Iterative Method is an extension of FGSM by making the update steps to be iterative. This can help to further modify the data to get a higher attack success rate than singe-step modification. The formula is:

$$D'_{i+1} = D'_i + \epsilon \cdot sign(\nabla_{D'_i} L(f_\theta(D'_i), Y)). \tag{5}$$

3) MIM: The Momentum Iterative Method further accumulates the previous gradients before current step. The formula is:

$$g_{i+1} = \mu \cdot g_i + \frac{\nabla_{D'_i} L(f_\theta(D'_i), Y)}{\|\nabla_{D'_i} L(f_\theta(D'_i), Y)\|}, \tag{6}$$

$$D'_{i+1} = D'_i + \epsilon \cdot sign(g_{i+1}). \tag{7}$$

The momentum of the computed gradients at current i-th iteration is g_i. The percentage of the momentum gradient to be used is controlled by the factor μ.

C. Ensemble and Transfer Adversarial Attack

In the common adversarial methods, obtaining a substitute model for the target is typically required to generate adversarial data. The efficacy of the attack on unseen black-box models (where the attacker conducts the attack without the knowledge of the target model) hinges on the transferability of the generated adversarial data. Due to disparities in the structures and parameters of substitute and black-box models, such attacks cannot guarantee successful outcomes. Hence, the proposal of ETAA aims to narrow the gap in gradient directions between the white-box model (where the attack knows complete information about the target model) and black-box models, thereby enhancing the transferability of generated adversarial data and increasing the success rate of the attack.

The detailed framework of the ETAA method is shown in Fig. 3. The input to the method is denoted as x. This value is updated to x_1 from the ensemble and transfer process in an

"update step". At first the Cross Entropy loss is calculated for both ensemble and transfer parts. Then the loss is summed to further obtain the perturbations that are necessary to be added to the data. There can be N iterations of "update step" to produce the final adversarial data x_n. Details of the computations of loss and perturbations are explained in remaining parts. Within each "update step", the merged gradient direction is determined through the combination of two components: the ensemble part and the transfer part. The ensemble part has m selected models from a pool of diverse models. The remaining unselected model is used in the transfer part. It is noted that any other DL network may be selected for the model pool. It highlights the advantage of generalization of the proposed method. For illustrative purposes, the ETAA method utilizes a pool of 5 models, namely Conventional Neural Network (CNN), Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). These models are chosen as representatives due to their widespread usage in deep learning networks for classification tasks [23]. Consequently, 4 randomly selected models serve as ensemble models, while 1 model functions as the transfer model. Since the ETAA method is a general framework, it is easy to implement with any gradient-based AML method. For the remaining parts, the details of the ETAA method are explained by using the Basic Iterative Method as an example.

1) Ensemble Part: The objective of the ensemble component is to systematically investigate a broad spectrum of gradient information from various established models, aiming to generate adversarial data with greater generalization. Typically, leveraging a larger number of models enhances the exploration of diverse gradient directions, contributing to an elevated transfer attack rate. The ensemble part is designed to simulate a white-box attack by knowing complete information of targeted models to generate adversarial data. By employing randomly selected ensemble models, a collective loss is computed. As shown in Fig. 3, an input x is passed into ensemble models simultaneously, and the ensemble logits $logits_{ens}$ are obtained as:

$$logits_{ens}(x) = \sum_{m=1}^{4} (w_m \cdot logits_m(x)). \tag{8}$$

The $logits_m(x)$ are the logits output from each model. Here, the weighting factor is w_m for each model and $\sum_{m=1}^{4}(w_m) = 1$. To get the average logits of the ensemble models, w_m is set to 1/4 for all models. Next, the Cross Entropy Loss is calculated:

$$L_{ens}(x) = -Y \cdot \log(softmax(logits_{ens}(x))). \tag{9}$$

The loss from this ensemble part is therefore obtained as $L_{ens}(x)$. At the same time, another loss is obtained from the transfer part.

2) Transfer Part: This is the part to further improve the gradient information searching. In this part, the remaining model acts as the black-box model and is marked as m_{tf}. The input is x and the output logits of this transfer model are $logits_{tf}(x)$. Next, the transfer loss is:

$$L_{tf}(x) = -Y \cdot \log(softmax(logits_{tf}(x))). \tag{10}$$

For the gradient-based attack, the data is updated by moving along the gradient direction which has the maximized loss with respect to the input. The total loss to compute the gradient is:

$$L(x) = L_{ens}(x) + L_{tf}(x). \tag{11}$$

Therefore, the perturbation to be applied to the input x is the gradient of this loss with respect to the input itself:

$$\delta = \epsilon \cdot sign(\nabla_x L(x)). \tag{12}$$

And then, the adversarial data is updated:

$$x_1 = x + \delta. \tag{13}$$

Since the ensemble models and transfer model are selected randomly in each step, different combinations of white-box and black models are achieved by increasing the N. By iteratively looping through the "update step", the gaps of the gradient information between various models are narrowed. Therefore, the final adversarial data x_n has better attack performance in terms of transferability.

D. Zero-mean Projection

To further craft the adversarial data, the sum of the generated x_n should be kept as close as possible to the original data sum as shown in Equation (3). To achieve this goal, for any time slot whose value is increased by a certain degree, there must be values to be decreased by the same amount in other time slots. Therefore, the perturbations δ is projected onto a hyperplane of zero-mean before adding to x directly. The zero-mean hyperplane is defined as the plane whose elements are denoted as a and the average of a is zero: $v^T a = 0$. The vector v is another vector of the same length as a and has all ones, and it can be seen as the normal of the zero-mean hyperplane. Therefore, the projection of perturbation δ is:

$$\delta' = \delta - \frac{\delta^T v}{||v||^2} \cdot v. \tag{14}$$

During the "update step", every obtained perturbation is projected to get the modified δ'. Equation (13) is also updated to be:

$$x_{i+1} = x_i + \delta'. \tag{15}$$

This projection process is done in each step, to ensure the stealthiness of the generated adversarial data.

III. Adversarial Defense

Since the PGD-AT method [18] is the representative work and is regarded as the critical standard way to do the adversarial training, this method is implemented first as an exploration of the possible defense solutions to the adversarial attack in DR applications. After that, a new ensemble-based adversarial training process to further improve the general classification accuracy and increase the true positive rate is proposed. This method is named as Defensive Training.

PGD-AT considers minimizing the maximum loss of the worst-case samples for a given model in adversarial training. The BIM attack was used to create the adversarial samples to solve the inner maximization problem. The model is trained to

be robust to adversarial samples by minimizing the outer loss. The adversarial samples are generated using:

$$x^{t+1} = \Pi_{x+S}(x^t + \alpha \cdot sign(\nabla_x L(f_\theta(x), y))) \quad (16)$$

where t is the current step and Π_{x+S} is the projection of the adversarial data to be bounded by $x+S$. S is the allowed range of changes. Then, instead of using both adversarial and original samples for training, PGD-AT only uses adversarial samples in training.

A. Defensive Training

The proposed method to address the adversarial samples contains three major processes: the random noise injection to the input, the latent feature combination, and the probability margin regulation. The overall process is shown in Fig. 4.

1) Noise Injection: This idea introduces a higher level of variability in the positive samples to allow the model to learn more latent features about them. The reason of keeping negative samples ("normal" labeled data) unchanged is that less attention to the variations of the negative samples is required but more strict examination on the positive samples and their similar copies is important. It is better to let the model classify the data that have even minor differences from the known positive samples to be positive as well. Thus, Gaussian noise is selected because it is a good approximation for the natural noise and unknown distributions. Since the samples fed into the model have been standard scaled based on data mean and variance, the Gaussian noise $N(\mu, \sigma^2)$ should have the mean $\mu \in [0, 1]$. To make the noise have a narrow peak around μ, the variance σ^2 should be small. The training data after scaling is represented as X. Motivated by the aforementioned zero-mean hyperplane in the generation of adversarial data, both positive noise and negative noise will be sampled from Gaussian noise and subsequently incorporated into the positive samples. There are a total of 48 values in X and the noise would be:

$$n_1 = [n_{1,1}, n_{1,2}, ..., n_{1,24}] \in N[\mu, \sigma^2], \quad (17)$$

$$n_2 = [n_{2,1}, n_{2,2}, ..., n_{2,24}] \in N[-\mu, \sigma^2], \quad (18)$$

$$n_0 = [n_1, n_2]. \quad (19)$$

μ is defined based on how much variability is to be given to the values of positive samples. Next, 24 noise values are sampled from the Gaussian distribution of mean μ and another 24 noise values are sampled from the Gaussian distribution of mean -μ. The variances are the same. n_0 contains all the selected values and the mean of this vector is expected to be 0. The crafted adversarial copy X_{adv} of training data is:

$$X_{adv} = X + noise \quad (20)$$

$$noise = \begin{cases} 0, & \text{if } Y = 0 \\ n_0, & \text{if } Y = 1 \end{cases} \quad (21)$$

where X is the original sample. $Y = 0$ is the case of a negative sample and $Y = 1$ is the case of a positive sample. The set of X and X_{adv} will be used to calculate the forward propagation logits of a given model.

2) Features Combination: Existing ensemble-based training methods focus on using the output directly from separate networks to make the final classification or using data generated from separate networks to augment the training dataset. But it is the latent feature spaces in networks that contain more higher-dimension information about the input data, which can be used to make better predictions. Therefore, instead of just adding outputs of "0" or "1" from more networks to modulate the final classification, the latent features from different models are utilized. In this paper, the CNN, MLP and GRU networks are used for illustration. However, more networks are acceptable to enhance the performance because more features are exploited by adding other models that make the trained defense model more robust when it encounters new data. Taking the input data to be X during the training phase, the output after certain hidden layers can be denoted as ψ. As Fig. 4 shows, each network could have its own operational layers to manipulate the data processing and each will have a dense layer to produce the latent feature output. For CNN network, the output is ψ_{CNN} with a length of a. Similarly, ψ_{MLP} with length of b and ψ_{GRU} with length of c are the feature outputs from MLP and GRU, respectively. Then, these outputs are concatenated into a vector as ψ_t, and the length of ψ_t is $d = a + b + c$. This ψ_t is fed into a dense layer to get the final output logits. The loss function is derived from the logits and the parameters of these networks are updated at the same time during the training backpropagation. Each network uses the same loss to modify itself, and all networks are related to the others rather than being independent.

3) Probability Margin: Similar to general adversarial training, the Cross Entropy loss of the training data is used for backpropagation. However, since the variant of the positive samples from noise injection are created, the losses are:

$$logits = f_{def}(X), \quad logits_{adv} = f_{def}(X_{adv}) \quad (22)$$

$$L_1 = L_{CE}(logits, Y), \quad L_2 = L_{CE}(logits_{adv}, Y) \quad (23)$$

where f_{def} is the defense model to be trained and L_{CE} is to compute the Cross Entropy loss. Then, the total loss from the original samples and adversarial samples is $L = L_1 + L_2$.

Besides this, a regulation factor to each L_1 and L_2 loss during the forward propagation is applied. The output logits of the defense model are passed through a sigmoid activation to get a value p in a form of the probability in the range of [0,1]:

$$p = \sigma(logits). \quad (24)$$

This p represents the probability of a given input to be a positive sample (adversarial attacked sample). In the case of input data to be X_{adv}, the output probability is denoted as p_{adv}. To let the model be robust to variations in data and make correct classification with higher confidence, the margin of the probabilities made by the model should be as large as possible. Intuitively, when the model gets the output p or p_{adv} to be opposite to the real label, the loss corresponding to each p and p_{adv} will be penalized more. Therefore, a regulation factor r is introduced to each loss:

$$r = (1 - 2 \cdot Y) \cdot (P - Y) \quad (25)$$

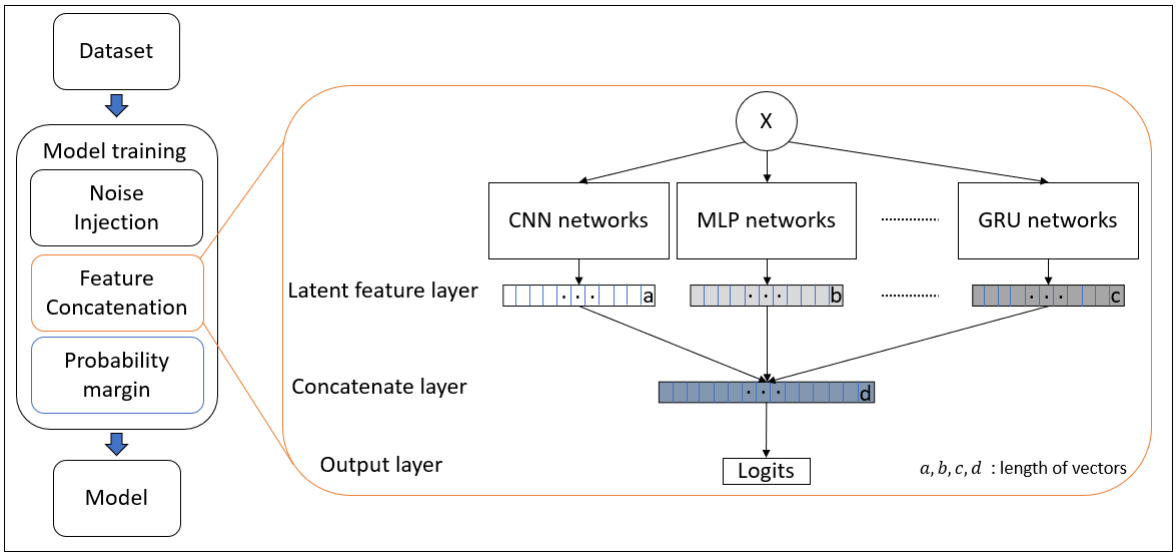

Fig. 4. Defensive Training process.

where Y is the true label of the data and P is the probability output of the data. For an adversarial copy of data X_{adv}, P is p_{adv} and for an original data X, P is p. As Equation (25) shows, if the output probability P is exactly the same as the true label Y, r will be 0 which is appropriate since for a totally confident and correct classification, the loss is 0. The smaller the probability margin, the more loss the model would apply. Overall, the loss function now becomes:

$$L_T = r_1 \cdot L_1 + r_2 \cdot L_2 \tag{26}$$

$$= (1-2Y)\cdot(p-Y)\cdot L_{CE}(f_{def}(X),Y) + (1-2Y)\cdot(p_{adv}-Y)\cdot L_{CE}(f_{def}(X_{adv}),Y). \tag{27}$$

In summary, the defense model is trained by computing the custom loss in the forward propagation and updating the model parameters based on this loss in backpropagation.

IV. Results

To capture the stochastic natures of DR processes when running the experiments, real-life data is used in this paper. The dataset employed for the evaluation is the Pecan Street Dataset, as utilized in [9], [15]. The energy consumption data is extracted and aggregated from randomly selected devices from a total of 168 houses in Austin, Texas between June and August 2017. This random choice of demand patterns from different houses helps to overcome the limitations of data from single house at a single time. Furthermore, to examine the performance of ETAA, adversarial false demand data are generated based on randomly sampled data from those demand patterns. This implies that the attack could happen to any demand data at any time. Existing BIM and MIM adversarial methods with various single substitute models are also replicated for comparison with the proposed method. Both white-box and black-box scenarios are simulated to show the transferability of the adversarial data. Furthermore, the proposed defense method, in conjunction with models trained using the PGD-AT approach, is employed to classify adversarial data generated by ETAA methods, thereby showcasing their respective performances. This assessment helps to evaluate the defense capabilities against adversarial data. From the perspective of conducting attacks on the detection model, a lower accuracy and higher False Negative Rate (FNR) on adversarial data classification indicate better results of attack. Other than those, two more metrics are computed. The first metric is the Attack Success Rate (ASR). It is the percentage of the generated "attack" class data that has been classified to be "normal" among all the generated adversarial data: $ASR = \frac{FN}{n}$, where $n = 10000$, and FN represents the number of data falsely classified as "normal" when they actually belong to the "attack" class. For a more successful adversarial attack, a higher value of the ASR is desired, indicating that a greater proportion of adversarial instances has successfully evaded detection by the model. The second metric is the difference in demand sums between the adversarial demand and the normal demand. The average of the differences for all samples are calculated and denoted as SD. A smaller SD value signifies a greater resemblance between the adversarial data and the original data, highlighting the impacts of the zero-mean projection as elaborated in the previous section.

A. Attack Results

The generated adversarial false demand data from ETAA method are illustrated in Fig. 5. The x-axis is the 48 time slots that equal to 24 hours and y-axis is the demand power. Some generated adversarial false demand data are randomly selected and plotted. They are all positive samples and labeled as "attack". The figure shows that the generated adversarial attack data follows the original demand pattern and there are more introduced false peaks between time slot 30 to 40. Moreover, this figure demonstrates the modifications introduced by the ETAA method, showcasing diversity in the generated adversarial data.

1) Attack Success Rate: Table I shows all the values of ASR when using adversarial data generated from each method to

TABLE I
THE ATTACK SUCCESS RATE (%) OF GENERATED ADVERSARIAL DEMAND DATA

No.	Method	Substitute Model	White-box Models						Black-box Models			
			CNN	MLP	RBF	LSTM	GRU	Average	CNN-E	MLP-B	GRU-B	Average
1	BIM	CNN	69.72	19.33	16.97	35.45	34.00	35.09	19.44	25.38	27.85	24.22
2		MLP	14.83	44.10	16.45	24.00	23.38	24.55	12.21	24.27	22.43	19.64
3		RBF	52.21	66.88	87.49	72.53	76.3	71.08	57.32	71.08	72.81	67.07
4		LSTM	22.12	15.8	11.94	64.86	31.76	29.30	7.32	18.99	30.41	25.01
5		GRU	23.13	22.62	20.43	45.38	58.52	34.02	17.01	32.51	32.61	32.10
6	MIM	CNN	**99.39**	11.72	8.52	36.70	33.17	37.90	13.27	14.93	26.53	18.24
7		MLP	11.53	58.32	14.05	25.52	22.92	26.47	9.15	19.68	22.42	17.08
8		RBF	53.96	71.61	**96.26**	82.08	85.45	77.87	**60.09**	77.61	82.33	73.54
9		LSTM	28.50	22.86	22.03	**95.19**	67.21	47.16	11.69	30.96	60.89	34.51
10		GRU	29.07	23.35	22.07	79.16	**98.88**	50.51	12.02	35.73	60.41	36.05
11	ETAA with BIM	-	88.36	88.28	87.43	88.56	88.53	88.23	56.23	80.09	85.08	73.8
12	ETAA with MIM	-	91.24	**91.35**	89.31	91.72	91.81	**91.12**	57.29	**83.2**	**87.68**	**76.06**

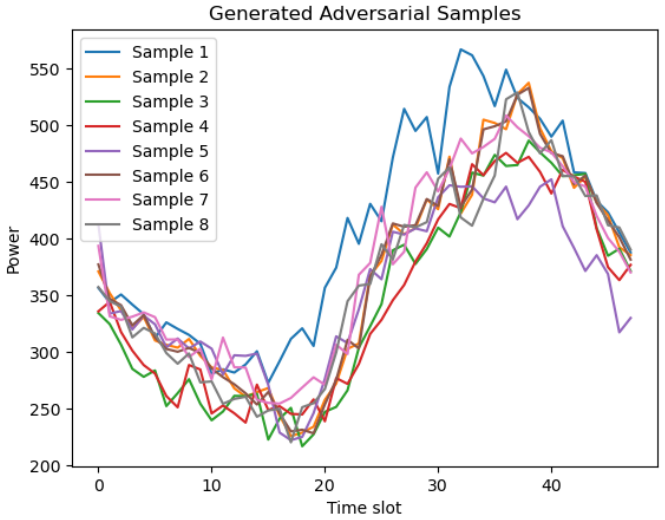

Fig. 5. Some ETAA generated adversarial samples.

attack white-box and black-box detection models. The highest ASR value under each setting is made bold. It can be seen that the highest average ASR for both white-box and black-box settings are obtained using the ETAA method with MIM updating process. The highest average white-box ASR is 91.12% and the highest average black-box ASR is 76.06%. It is expected to see almost all independent MIM methods to have highest ASR when executing the white-box attacks because the generated adversarial data are finely crafted using the identical detection model as the substitute model. However, the ETAA can also obtain the highest ASR of 91.35% when the detector model is MLP. In the black-box scenario, where the detection model differs in layer structure and parameters from the substitute model, ETAA with MIM process achieves the highest ASR of 83.2% for MLP-B and 87.68% for GRU-B. This underscores the versatility of the ETAA method in generating adversarial data for external models. The ensemble and transfer components in ETAA contribute to minimizing differences in gradient directions among various models, ensuring enhanced attack transferability of the generated adversarial demand data.

Specifically, Table II shows the summarized results of generated adversarial data attacking against black-box models. The metrics under BIM and MIM methods are averaged to provide the direct comparison that shows the advances of the proposed method. The lowest accuracy, highest ASR and highest FNR are made bold, which correspond to the most successful adversarial attack on black-box models. It is seen that the method outperforms the BIM and MIM methods in generating the adversarial false demand data against the detection models with higher success rate to mislead the models, with the lowest accuracy of 23.4%, highest ASR of 76.06%, and highest FNR of 83.04%.

TABLE II
THE ADVERSARIAL ATTACK RESULTS IN BLACK-BOX SETTING

Method	Accuracy(%)	ASR(%)	FNR(%)
BIM	51.10	33.63	46.45
MIM	63.52	35.88	36.39
ETAA-BIM	25.04	73.80	**83.04**
ETAA-MIM	**23.40**	**76.06**	82.72

2) Difference in Sum of Demand: Table III shows the results of SD values of each method, with the smallest value made bold. The ETAA with MIM process has the smallest value of 9.49kW and the ETAA with BIM process has a comparable value of 9.84kW. This shows that the projection of perturbations to the zero-mean hyperplane can help to constrain the adversarial data to have the smallest changes in the sum of demand data, and thus to ensure the stealthy attack. It can be noted that although the MIM process with a substitute RBF model can achieve the highest ASR of 60.09% when attacking the black-box CNN-E model, the SD value is much larger than the ETAA method. Therefore, from the results, it can be concluded that the ETAA method is able to generate adversarial data that has higher attack transferability while also allowing the attack to be imperceptible.

Moreover, the relationship between values of SD and ASR in both white-box and black-box settings are displayed in Fig. 6 and Fig. 7. The legend number corresponds to the $No.$ in Table I and Table III for an easier visualization. In both Fig. 6 and Fig. 7, the 1st, 2nd, 6th and 7th cases get the relatively small values for SD about 50kW. However, their corresponding ASR values are still smaller compared to the others. This demonstrates that achieving a smaller SD for adversarial false data through existing methods does not necessarily guarantee a more successful adversarial attack. The effective exploitation of underlying

TABLE III
THE SUM DIFFERENCE (KW) OF GENERATED ADVERSARIAL DATA

No.	Method	Substitute Model	SD(kW)
1	BIM	CNN	43.33
2		MLP	56.84
3		RBF	63.67
4		LSTM	234.17
5		GRU	192.66
6	MIM	CNN	55.91
7		MLP	35.09
8		RBF	55.71
9		LSTM	346.09
10		GRU	353.33
11	ETAA with BIM	-	9.84
12	ETAA with MIM	-	**9.49**

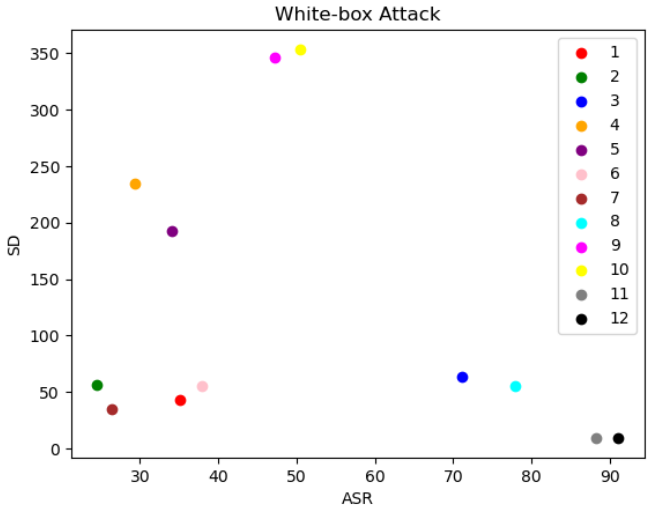

Fig. 6. Correlation between SD and ASR in white-box setting.

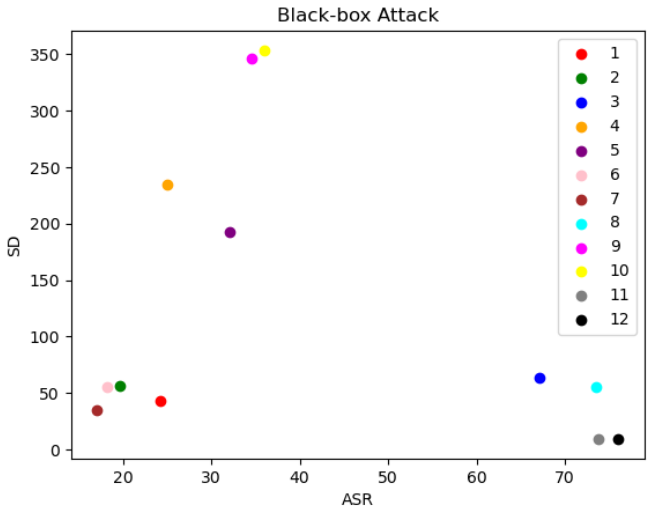

Fig. 7. Correlation between SD and ASR in black-box setting.

demand data characteristics is crucial for generating successful adversarial data, an aspect in which the proposed method has proven its proficiency. The adversarial data generated by ETAA exhibits superior performance in attacking detection models, as evident in both the ASR and SD values.

B. Defense Results

To comprehensively analyze how different levels of injected Gaussian noise influence the performance and robustness of the defense method, means of 0.2, 0.5, and 0.8 are applied. These values help to a wide range of possible noises. By introducing these varying noise levels, the method's ability to maintain accuracy and reliability under diverse and challenging conditions is evaluated. The Defensive Training models trained from those mean values are denoted as $DT-0.2$, $DT-0.5$ and $DT-0.8$. Additionally, the comparison between separate black-box detection models trained using the benchmark PGDAT method and the proposed approach is conducted. The learning curve of the proposed Defensive Training method is shown in Fig. 8. The selection of Gaussian mean does not exert a significant impact on the training loss and accuracy of the training dataset. However, it does result in increased fluctuations in the accuracy of the validation dataset. Nevertheless, the plots illustrate that the proposed training method has converged effectively to obtain high validation accuracy using the improved loss function.

1) Trained Models: The performance of trained detection models are summarized in Table IV. The metrics indicate that the detection model generated by the proposed method exhibits comparable performance to the model derived from an existing method. It effectively predicts regular demand data in the absence of adversarial input, demonstrating high accuracy. Thus, in common scenarios, the proposed method's model can continue to operate reliably.

2) Defense against Adversarial Data: Table V shows the results when the adversarial data generated from the ETAA method is used to attack the new models. The best value of each metric is made bold. The accuracy denotes the overall classification accuracy of predictions on adversarial data, and a higher accuracy is indicative of better performance. From a defensive standpoint, a lower ASR is preferred as it signifies a more effective detection model. The proposed defensive training, incorporating a noise mean of 0.2, achieves the highest accuracy at 78.28%, surpassing the CNN-PGDAT accuracy of 76.27% by 2%. This indicates this model's proficiency in correctly classifying a significant number of both positive and negative samples. In terms of ASR, the defensive training model

TABLE IV
PERFORMANCE OF DETECTION MODELS ON CLEAN TEST DATA

Model	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
CNN-PGDAT	97.93	95.47	64.78	77.18
MLP-PGDAT	97.87	94.63	64.13	76.45
GRU-PGDAT	97.89	92.85	65.89	77.08
DT-0.2	97.88	91.18	67.33	77.43
DT-0.5	98.01	94.12	67.72	78.50
DT-0.8	98.02	94.82	66.94	78.47

TABLE V
ATTACK RESULTS ON DEFENSIVE MODEL

Model	Accuracy(%)	ASR(%)	Precision(%)	Recall(%)	FNR(%)	TN	FP	FN	TP
CNN-PGDAT	76.27	21.55	96.99	76.53	23.42	579	218	2155	7048
MLP-PGDAT	61.28	37.47	**97.76**	59.29	40.72	672	125	3747	5456
GRU-PGDAT	67.16	30.92	96.95	66.40	33.59	605	192	3092	6111
DT-0.2	**78.28**	**18.14**	95.43	**80.29**	**19.71**	438	360	1814	7390
DT-0.5	70.41	26.90	96.03	70.77	29.23	529	269	2690	6514
DT-0.8	54.76	43.80	97.08	52.41	47.59	653	145	4380	4823

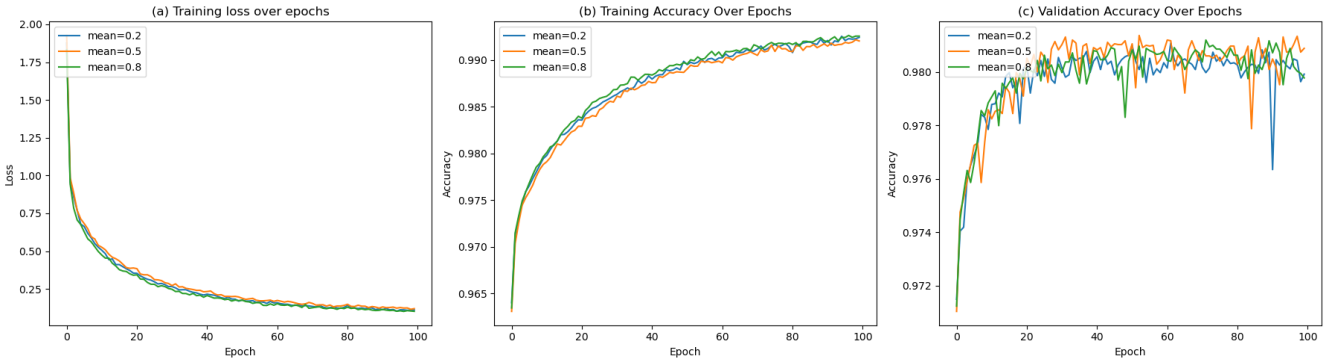

Fig. 8. Model loss and accuracy during the defensive training phase.

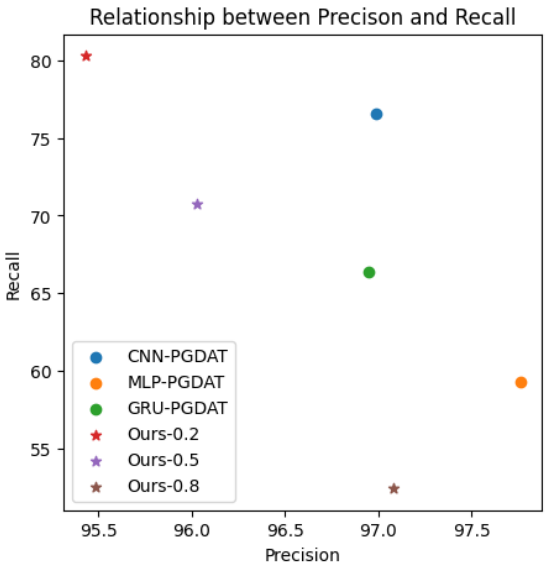

Fig. 9. Scatter plot of precision and recall when facing adversarial data.

records the lowest value at 18.14%, showing its ability to identify more positive samples. The number of missed faults is another important value and the FNR indicates the performance of actual positives that are incorrectly identified as negatives. From this table, the proposed method with noise mean of 0.2 obtains the lowest FNR of 19.71%. It shows that the proposed Defensive Training has the smallest number of missed positives. The noise injection and the feature concatenation processes have successfully captured the underlying characteristics of any attack data and classified them correctly. Additionally, the notable recall of 80.29% underscores the proposed method's capability to accurately identify adversarial data. The highest precision, at 97.76%, is achieved by the MLP-PGDAT model, representing the percentage of true positives among detected positives. However, this model demonstrates lower overall accuracy and higher ASR, indicating worse performance in correctly classifying adversarial data. In addition, the proposed method can obtain a precision value of 97.08%, which is close to the highest value. In summary, the proposed method outperforms the other methods with increased accuracy and reduced ASR, highlighting the success in defending against the adversarial false data.

Moreover, Fig. 9 depicts the relationship between precision and recall, providing a clear understanding of the trends. The plot highlights a trade-off inherent in the proposed Defensive Training method: as the noise mean value increases, precision rises while recall decreases. Conversely, with a smaller noise mean value, precision decreases while recall increases. When the mean of the noise increases from 0.2 to 0.8, the precision exhibits an increase of 1.65%, whereas the recall experiences a substantial decline of 25.88%. Moreover, there is a notable decrease in accuracy, dropping significantly from 78.28% to 54.76% as the mean noise is raised from 0.2 to 0.8. If the noise mean is too high, the model may start to overfit to the noise rather than learning the underlying patterns in the data. This could result in poor generalization to unseen adversarial data. Therefore, the selection of this parameter hinges on the objective of obtaining more precisely classified attacked data or identifying a greater number of potential adversarial instances.

V. CONCLUSION

In conclusion, the integration of DR schemes into smart grids represents a pivotal aspect of modern cyber-physical systems. These schemes, designed to align power supply with demand and optimize energy efficiency, are susceptible to emerging

cyber threats, particularly FDI attacks where demand data are manipulated. Thus, this study presents a multifaceted approach to enhance the security related to the defense against FDI attacks in DR. The ETAA method is introduced, leveraging AML techniques to generate adversarial data for injection into DR demand data. ETAA addresses challenges related to transferability and stealthiness by employing ensemble and transfer strategies and zero-mean projection. Simulations of the attacks have demonstrated that ETAA enhances attack transferability and yields a higher attack success rate. Furthermore, the Defensive Training method, incorporating noise injection, feature combination, and a regulated probability is introduced as means to safeguard the model against adversarial data. The model trained using the proposed method attains higher classification accuracy and successfully identifies more true adversarial data compared to the benchmark method. The innovative nature of the proposed solutions positions this work as a key contributor in paving the way for the secure and widespread implementation of Demand Response in the evolving landscape of smart grids.

VI. Acknowledgment

This work was supported by the Ministry of Education, Singapore under Tier 2 grant T2EP20121-0036.

References

[1] C. Barreto, A. A. Cárdenas, N. Quijano, and E. Mojica-Nava, "Cps: Market analysis of attacks against demand response in the smart grid," in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014, pp. 136–145.

[2] Y. Jiang, S. Wu, R. Ma, M. Liu, H. Luo, and O. Kaynak, "Monitoring and defense of industrial cyber-physical systems under typical attacks: From a systems and control perspective," *IEEE Transactions on Industrial Cyber-Physical Systems*, 2023.

[3] H. Zhu, L. Xu, Z. Bao, Y. Liu, L. Yin, W. Yao, C. Wu, and L. Wu, "Secure control against multiplicative and additive false data injection attacks," *IEEE Transactions on Industrial Cyber-Physical Systems*, 2023.

[4] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 570–582, 2015.

[5] J. Li, H. Li, T. Huang, L. Zheng, L. Ji, and S. Yin, "Model-free reinforcement learning economic dispatch algorithms for price-based residential demand response management system," *IEEE Transactions on Industrial Cyber-Physical Systems*, 2023.

[6] F. Pallonetto, M. De Rosa, F. Milano, and D. P. Finn, "Demand response algorithms for smart-grid ready residential buildings using machine learning models," *Applied energy*, vol. 239, pp. 1265–1282, 2019.

[7] T. Dayaratne, C. Rudolph, A. Liebman, M. Salehi, and S. He, "High impact false data injection attack against real-time pricing in smart grids," in *2019 IEEE ISGT-Europe*. IEEE, 2019, pp. 1–5.

[8] T. Dayaratne, C. Rudolph, A. Liebman, and M. Salehi, "We can pay less: Coordinated false data injection attack against residential demand response in smart grids," in *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, 2021, pp. 41–52.

[9] T. Dayaratne, M. Salehi, C. Rudolph, and A. Liebman, "False data injection attack detection for secure distributed demand response in smart grids," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2022, pp. 367–380.

[10] S. Lakshminarayana, A. Kammoun, M. Debbah, and H. V. Poor, "Data-driven false data injection attacks against power grids: A random matrix approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 635–646, 2020.

[11] S. Gao, H. Zhang, Z. Wang, C. Huang, and H. Yan, "Data-driven injection attack strategy for linear cyber-physical systems: An input-output data-based approach," *IEEE Transactions on Network Science and Engineering*, 2023.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[13] Y. Chen, Y. Tan, and D. Deka, "Is machine learning in power systems vulnerable?" in *IEEE SmartGridComm*. IEEE, 2018, pp. 1–6.

[14] Y. Chen, Y. Tan, and B. Zhang, "Exploiting vulnerabilities of load forecasting through adversarial attacks," in *ACM International Conference on Future Energy Systems*, 2019, pp. 1–11.

[15] Z. Guihai and B. Sikdar, "Adversarial machine learning against false data injection attack detection for smart grid demand response," in *2021 IEEE SmartGridComm*. IEEE, 2021, pp. 352–357.

[16] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[17] G. Zhang and B. Sikdar, "Ensemble and transfer adversarial attack on smart grid demand-response mechanisms," in *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2022, pp. 53–58.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[19] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *International conference on machine learning*. PMLR, 2020, pp. 11 278–11 287.

[20] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 909–913.

[21] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[22] A. Takiddin, M. Ismail, and E. Serpedin, "Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 14, no. 1, pp. 663–676, 2022.

[23] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4970–4979.