Ensemble and Transfer Adversarial Attack on Smart Grid Demand-Response Mechanisms

Guihai Zhang and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Demand Response (DR) mechanisms aim to balance power supply and demand in smart grids by modulating consumers' demand and adjusting electric price based on power consumption patterns and forecasts. Deep Learning (DL) networks have been proved to have better detection of False Data Injection (FDI) attacks in such DR system than traditional statistical methods. Adversarial Machine Learning (AML) attacks can generate finely perturbed data that can mislead or disrupt the normal performance of a DL network and bypass DL-based attack detection in DR systems. However, existing AML attack methods in DR systems require a substitute model to generate the adversarial data and rely on the transferability of the data to attack the target DL models or the others. In this paper, a novel attack method called Ensemble and Transfer Adversarial Attack (ETAA) is proposed to improve the transferability of adversarial attacks across different DL models. This method has a general framework and is able to work with various existing gradient-based attacks. Moreover, to reduce the power company's awareness of FDI attack in the demand data, a zero-mean plane projection is applied to limit the perturbations during adversarial data generation. The evaluation results show that the proposed ETAA method can achieve higher attack success rate across different models and the zero-mean projection method can keep the final total adversarial power demand to be closer to the original normal demand.**

***Index Terms*—Smart grid, demand response, false data injection, deep learning, adversarial machine learning.**

I. Introduction

Power grids strive to maintain a match between power supply and customer demand and Demand-Response (DR) schemes have been proposed to achieve this objective. Under a typical DR scheme, different types of price strategies like Time-of-Use rates, Real-Time Pricing and Day Ahead Pricing may be applied, and customers schedule their power usages according to the unit price with the purpose to reduce the total electricity bill by decreasing usage during peak hours [1]. Also, the utility company can be better prepared for the upcoming power demand and therefore reduce operational costs while maintaining and operating the grid system [2].

Considering the typical FDI attacks where attackers implement strategies like directly injecting data, compromising transmitted data or hacking smart devices, the consequences of successful attacks could lead to different levels of impacts in the DR system. These impacts include the fluctuations in the power flow and incorrect power flow analysis of a grid, physical damages to the devices due to the working states exceeding safe limits and insufficient power generation that cause blackout. Thus, such attacks can cause economic losses for the power company and disrupt the normal operations of affected industries [3], [4]. Besides the traditional mathematical and statistical attack detection methods, Deep Learning (DL) networks have been proven to be applicable in smart grid systems [5]–[9], and are able to protect the DR schemes from FDI attacks. Recently, Adversarial Machine Learning (AML) techniques have been studied and the results show that DL networks have the risk of misbehaving when AML techniques are used on them [10]. AML could add perturbations to the input of a DL network to deceive that DL network by leading to false predictions. If the attackers have the knowledge of the targeted DL network parameters, the attack is named as white-box attack. In reality, most attacks cannot access the information of the targeted model, and the attack is called black-box attack.

In previous work [11], the AML techniques against the DL-based FDI detection in DR systems have been explored and the vulnerabilities of existing systems to AML attacks have been evaluated. However, the work in [11] only considered using a single DL model to generate adversarial demand data and to believe that the generated data itself has the transferability to allow them to attack the other DL networks successfully. Moreover, it did not consider the stealthiness of the generated data. The sum of adversarial power demand may have large difference from the normal power demand, which may draw attention from the utility company. Therefore in this paper, we propose a new attack framework called Ensemble and Transfer Adversarial Attack (ETAA) to generate the adversarial FDI data for the DR schemes. This ETAA is able to work with any gradient-based adversarial attack method and to reduce the discrepancy between different models to improve the transferability of attacks. Moreover, the method of projecting perturbations to a zero-mean plane is applied to ensure that the sum of total adversarial demand power is close to normal demand which helps to hide it from the operator's notice.

The main contributions of this paper are as follows:

- We propose the Ensemble and Transfer Adversarial Attack (ETAA) framework which is able to reduce the gap of gradient directions between white-box and black-box models.
- We evaluate ETAA combined with existing gradient-based attacks and show that the attack success rate increases with better transferability than standalone method.
- We introduce the zero-mean projection method to limit the perturbations and therefore to make the final adversarial demand data closer to the original normal data.

The authors are with the National University of Singapore, Singapore. This work was supported by Ministry of Education, Singapore under Tier 2 grant A-8000109-01-00.

978-1-6654-3254-2/22/$31.00

The organization of the rest of this paper is as follows. Section II reviews existing DR schemes, DL techniques, and AML methods. Section III describes the attack model proposed for this paper and the details of existing gradient-based AML methods and our ETAA method. Section IV presents the evaluation results of the proposed attack and comparison with existing methods. Section V concludes the paper.

II. Related Work

A. Demand Response Schemes

Figure 1 shows the typical communication process in a DR scheme. To initiate the process, the utility company sends the original unit price to the customer side ahead of the real electricity usage (e.g., a day). This is a list of prices for each time slot of a day. Then, at user side, customers schedule their appliances' consumption (e.g., starting time and running duration) in different time slots based on the price. Next, an aggregator collects the demand forecasts from all customers. After the utility company receives the accumulated demand forecasts, its uses its optimization process to adjust the unit price. Then, this adjusted new unit price is sent to customers again. This communication process is repeated and stops when both customers and utility company arrive at their optimization goal. On the day of real usage, the households follow their schedule to run appliances and the billing is calculated based on the actual consumption.

In this paper, we consider an attack scenario where attackers want to reduce their electricity bill by providing fake data into the demand forecasts. We assume that attackers can manipulate some households' demand data or are able to hack into the network and modify the accumulated demand forecasts. As stated in [3], when the attacker has access to the aggregated demand forecasts, he can perform FDI attacks to modify the power demand forecasts and gain monetary benefits by introducing false predicted peak consumption at specific time slots to the utility company during the prediction phase. With the general assumption that the unit price increases with the demand and usage of a pricing slot, the attacked time slot will have higher unit price. During the prediction phase, because of this higher unit price, the other users will consider rescheduling their consumption by moving their usage from original time slot to another time slot with cheaper unit price. This action of rescheduling will cause the real power usage during attacked time slots to be lower and the unit price during real usage is actually lower than the optimized unit price during prediction phase. Therefore, the attackers can have a lower bill if they run their appliances in those attacked time slots.

B. Deep Learning

Deep Learning (DL) is subset of Machine Learning (ML) and has better performance than general ML to handle large amounts of data and to extract features in higher dimensions. Since smart grids generate a large amount of data, DL methods are able to solve problems like load forecasting, event classification, fault detection and attack detection. The use of DL models for load forecasting has been demonstrated in [12], [13]. DL methods

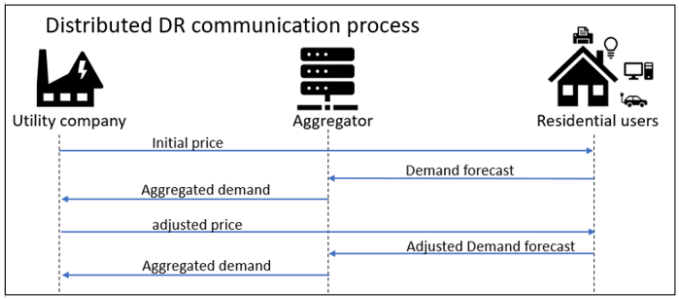

Fig. 1. Distributed DR communication sequence.

such as Convolutional Neural Networks (CNN), Recursive Neural Network (RNN) and Long-Short-Term Memory (LTSM) can also perform classification and identification tasks (e.g., classify the power quality disturbances) [14] or to detect malicious devices [15] in power systems. In addition, it is more common to use DL methods to detect the FDI attacks as proven by [5]–[8]. Specifically, [9], [16] also focused on the stealthy and covert FDI attacks. Besides the FDI detection in state estimation and power flow monitoring, there are few researches on the DR applications. The latest work that uses CNN to detect FDI attacks in DR schemes is [17], and it can obtain higher accuracy.

C. Adversarial Machine Learning

Adversarial Machine Learning (AML) was first introduced in [18] as a deceptive method against computer vision neural networks. The input pixel values of an image are perturbed by AML method without being detected by human eyes and the network is fooled by the adversarial data to give a wrong label for the image. When this is considered in a general DL case, AML methods modify the inputs of a DL model and thereafter the model would give wrong predictions. Since then, AML techniques have gained interest among researchers and various methods have been developed. Fast Gradient Sign Method (FGSM) [19], Fast Gradient Value (FGV) [20], and DeepFool [21] are the well-known AML algorithms. Moreover, AML applications in power systems have been considered in recent studies. The vulnerabilities of DL applications and AML applications have been discussed in [22]. In [23], the authors used Saliency Map Attack to conduct AML against power grid state estimation. In [24], the vulnerabilities in load forecasting model based on historical data are exploited to show the possible AML attack. There is the latest work in [11] that an iterative FGV method is used to generate the adversarial FDI data for the DR schemes. However, this work is done only using a CNN model to generate the adversarial data to attack another black-box CNN model. There is lack of attack transferability for the others models. Similar to the iterative FGV method, there are other state-of-the-art iterative gradient methods like the Basic Iterative Method (BIM) [19] which is an extension of FGSM, and Momentum Iterative Fast Gradient Sign Method (MIM) [25] which further uses the accumulated momentum of gradients to update the direction. These methods have better black-box attack transferability, but further modifications are required to perform the stealthy FDI attack in DR schemes.

III. Attack Details

A. FDI Attack

Consider the same settings and attack purpose as in [11], [17], where one whole day of 24 hours is divided into 48 time slots and each time slot is 30 minutes. The normal collected demand forecasts is represented by $D = [d_1, d_2, \cdots, d_n]$ and $n = 48$. The attacked demand forecasts now become $\hat{D} = [\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_n]$. In this paper, we assume the attackers have no specific time slot goal, which means they may want to increase the values at any random time slot i. Therefore, $\hat{d}_i \geq d_i$. Moreover, to perform successful FDI attacks that result in bill reduction, the modified demand data should have enough increment. Based on the evaluations in [3], the injected false demands are started from 0.1% of the overall demand to show the successful significant cost benefits to the adversary. If the 0.1% increment is not met, the perturbed data is still labeled as normal. On the other hand, generated adversarial data that has at least one demand data at any time slot whose increment is larger than 0.1% of the overall demand, will be labeled as attacked. For a DL detection model f_θ, the prediction output is $f_\theta(D)$. In addition, to make the adversarial data to be stealthy, the sum of modified demand forecasts should not change significantly compared to the normal demand forecasts. Therefore, $\sum(D) \approx \sum(\hat{D})$. Thus, the attackers should try to modify the values in D to become higher and at the same time, the modified values should be able to bypass the detection model to achieve a hidden FDI attack. In summary, the attack is modeled as follows:

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

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

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

B. Gradient Attacks

Most of the existing AML attacks use the gradient methods. The gradient of the targeted label's loss with respect to the input of a model is computed. Based on the gradient ascent or gradient descent method, the input data is modified along the gradient direction. Some of the gradient-based attacks are:

1) FGSM: Fast Gradient Sign Method is a fast method which only does a one-step modification to the input data. The gradient of the loss with respect to the input is calculated and the sign of gradient would be the perturbation added to the input data. The update equation is:

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

where ϵ is the updating factor, $L(f_\theta(D), Y)$ is the loss of input D with respect to target label Y and ∇_D is the function to compute the gradient of loss with respect to the D.

2) BIM: BIM is the iterative extension of FGSM, and has higher success rate than the single step FGSM. The formula is:

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

with $D'_0 = D$ and ϵ is the updating factor.

3) MIM: In MIM, the momentum of previous computed gradients is accumulated to decide the update direction in the iterative process. The equations are:

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

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

where g_i is the momentum of the computed gradients in the i-th iteration. The factor μ controls the ratio of the momentum to be accumulated.

C. Ensemble and Transfer Adversarial Attack

Most of the existing gradient-based attacks require a substitute model of the targeted model to generate adversarial data, and then use the transferability of the adversarial data to conduct attacks on unseen black-box models. Such attacks may not perform well because of the differences between substitute and black-box models in terms of their structures and parameters. To improve the transferability of such transfer-based adversarial attacks, we propose the Ensemble and Transfer Adversarial Attack (ETAA) method. This method is able to generate the adversarial data that has better transferability to attack various models by narrowing the gap of gradient directions between substitute/white-box model and black-box models.

The overall process of ETAA method is shown in Figure 2. Basically, there are K iterations to generate the adversarial DR demand data. In each iteration, there are two main stages: the ensemble stage and the transfer stage. The ensemble stage is the simulation of white-box attack to generate adversarial data based on the ensemble of 5 different models. The adversarial data is modified along the gradient direction computed from the fused entropy loss. The modification of adversarial data in ensemble stage can repeat N times. Next, the transfer stage acts like a black-box attack where a single model is randomly selected from the 5 models to be the transfer model to allow the data to explore more gradient information and thus to improve the transferability. Then, the generated adversarial data from ensemble stage is further modified by adding perturbations obtained from the transfer model. For instance, the initial normal data is x_i and the final adversarial demand data x_k is obtained after K iterations. From this whole process, the gap of the gradient directions between black-box and white-box models is gradually narrowed. The adversarial data updating process can be any gradient-based AML method as described in previous sections. Therefore, this proposed ETAA is flexible and easily implemented. In the following sections, the BIM updating method is used as an example to explain the details of ETAA.

1) Ensemble Stage: The idea of ensemble stage is to explore more gradient information among different substitute models and therefore to allow the adversarial data to become more generalized. The adversarial data is updated in the computed gradient direction based on the ensemble logits of those models. In ETAA, we use 5 different widely-using DL models to compute the ensemble logits. The selected 5 models are listed in the Table I under Ensemble Model. Ideally, the more ensemble models would be better to achieve higher transfer attack rate by

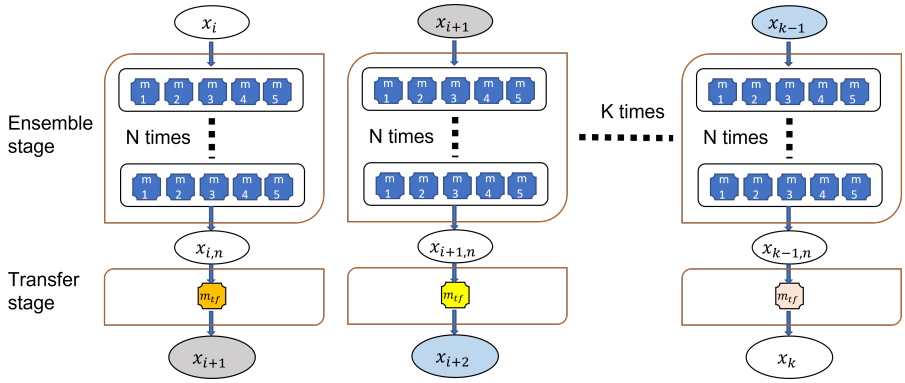

Fig. 2. Overview of Ensemble and Transfer Adversarial Attack framework.

exploring more gradient directions. In this paper, we use these 5 models as typical representative because they are the most common DL networks to deal with classification and regression problems [26].

To start the ensemble stage, the initial adversarial data is $x_{i,0}$. The ensemble logits $logits_{ens}$ are obtained from the 5 models:

$$logits_{ens}(x_{i,0}) = \sum_{m=1}^{5}(w_m \times logits_m(x_{i,0})). \quad (8)$$

The 5 models are denoted as m_1 to m_5. The $logits_m(x_{i,0})$ is the individual logits obtained from each model and the w_m is the weighting factor for each logit and $\sum_{m=1}^{5}(w_m) = 1$. In ETAA, the w_m have the equal value and the average logits of 5 models is obtained. Then, the cross entropy loss is calculated based on this average logits:

$$L_{ens}(x_{i,0}) = -Y \times \log(softmax(logits_{ens}(x_{i,0}))). \quad (9)$$

$L_{ens}(x_{i,0})$ is the loss to the current data $x_{i,0}$ and Y is corresponding label. As in the gradient-based AML attack, the adversarial data is updated along the direction which maximizes this loss:

$$x_{i,1} = x_{i,0} + \epsilon \times sign(\nabla_{x_{i,0}} L_{ens}(x_{i,0})). \quad (10)$$

Here $x_{i,1}$ is the adversarial data after 1 step of updating process. Equation (10) is repeated N times and $x_{i,n}$ is the final adversarial data obtained from the ensemble stage.

TABLE I
DL MODELS IN ETAA

No.	Ensemble Model
1	CNN
2	MLP
3	RBF
4	LSTM
5	GRU
No.	**Detection Black-box Models**
1	CNN-E
2	MLP-B
3	GRU-B

Note: Ensemble models are the models to be used in ETAA ensemble stage for the fused logits calculation which are further used for the adversarial data updating. The detection black-box models are used to evaluate the attack performance of the adversarial data generated. CNN-E is an existing detection model [17]. MLP-B and GRU-B are two black-box models that are different from the MLP and GRU model.

Algorithm 1 Ensemble and Transfer Adversarial Attack method

Require: x_0, N, K
$k = 1$
while $k \leq K$ **do**
  $n = 1$
  while $n \leq N$ **do**
    Compute the average logits of the 5 ensemble modes using Equation (8) and the cross entropy loss $L_{ens}(x_{0,n})$ using Equation (9).
    Obtain the perturbation δ using Equation (13).
    Obtain projection $\delta_{0-mean} = \delta - \frac{\delta^T n}{||n||^2} \times n$.
    Update $x_{0,n} = x_{0,n} + \delta_{0-mean}$.
    $n = n + 1$
  end while
  Compute the cross entropy loss of $x_{0,n}$ using the transfer model: $L_{tf}(x_{i,n})$ using Equation (11).
  Obtain the perturbation δ using Equation (13).
  Obtain projection $\delta_{0-mean} = \delta - \frac{\delta^T n}{||n||^2} \times n$.
  Update $x_k = x_{0,n} + \delta_{0-mean}$.
  $k = k + 1$
end while
Final adversarial data: x_k
return x_k

2) Transfer Stage: This is the stage to simulate a black-box attack and the adversarial data $x_{i,n}$ from ensemble stage is further modified. Here, one DL model is randomly selected from the 5 ensemble models to act as the black-box model. This model is called the transfer model and is denoted as m_{tf}. The logits $logits_{tf}(x_{i,n})$ are obtained from this transfer model with the input of $x_{i,n}$. Then the cross entropy loss is:

$$L_{tf}(x_{i,n}) = -Y \times \log(softmax(logits_{tf}(x_{i,n}))). \quad (11)$$

Therefore, the adversarial data is updated in the gradient direction that would maximize the loss function of $L_{tf}(x_{i,n})$:

$$x_{i+1} = x_{i,n} + \epsilon \times sign(\nabla_{x_{i,n}} L_{tf}(x_{i,n})). \quad (12)$$

This transfer stage is a single step to update the adversarial data and the transfer model is just a simulation of a black-box. The ensemble stage and the transfer stage together make one essential attack step. This attack step should be repeated K

times to perform the complete ETAA and the final adversarial data is denoted as x_k as shown in Figure 2. In this way, by iteratively going through ensemble models and transfer models, the gaps of the gradient directions between substitute model and black-box models are gradually narrowed. The final adversarial data should have better transferability when it attacks the other models.

3) Zero-mean Projection: In addition, to make the FDI attack to be stealthy, there is a limitation on the generated adversarial data. The perturbations added into the adversarial data during each update process should be considered. The sum of the generated x_k should be close to the original normal data, and this is same as Equation (3). That is to say, the perturbations applied to the data should have a zero mean such that when a value at any time slot has increased by a certain amount, there must be other time slots that has values decreased by the same amount. Thus, we use a method to project the perturbations onto a hyper-plane of zero-mean. The perturbation of data during the updating process is denoted as δ:

$$\delta = \epsilon \times sign(\nabla_{x_{i,n}} L(x_{i,n})). \tag{13}$$

where $L()$ is the corresponding loss function in ensemble or transfer stage. The zero-mean hyperplane consists all the elements of v whose mean is zero and $n^T v = 0$, and n is the vector of all ones with the same length as v. Then, the normal of the zero-mean hyperplane is n. Therefore, the projection of the perturbations δ is done by:

$$\delta_{0-mean} = \delta - \frac{\delta^T n}{||n||^2} \times n. \tag{14}$$

δ_{0-mean} is the modified perturbations that should be added to the adversarial data at each updating process when moving along the gradient directions that maximize the loss function. Equations (10) and (12) now becomes:

$$x_{i+1} = x_i + \delta_{0-mean}. \tag{15}$$

Algorithm 1 shows the overall procedure for the ETAA to generate adversarial data for the DR schemes. x_0 is the initial data and represents the normal demand values. After going through k iterations of attacks, the final adversarial data is x_k.

IV. Results

The simulations of the attacks are conducted using real-life data. The dataset used is synthesized from the Pecan street dataset and is the same as the one used in [11], [17]. To further examine the performance of the proposed ETAA method, 3 different black-box models are tested as FDI detectors. These models are listed in the Table I under Detection Black-box Models. The CNN-E is the same detector model in [17]. The MLP-B and GRU-B are trained separately and have different parameters and structures as compared to MLP and GRU used in ensemble stage. Overall, 10000 normal data are randomly sampled from the dataset and used as the input to the ETAA method to generate the final adversarial FDI data. The existing BIM and MIM methods are also implemented with various single DL models to compare the results with ETAA.

1) Metrics: The first metric is the Attack Success Rate (ASR). This is the percentage of generated attacked label data that can bypass the detection of DL models among all data. The formula is $ASR = \frac{FN}{n}$, where n=10000. A higher value ASR means the attack is more successful. Another metric is the difference of the sum of demand. Since we are using 10000 samples to generate the adversarial data, we use the average of the 10000 differences, that is SD_{avg}. This value should be as small as possible to ensure there are no significant changes to the total demand pattern and for the attack to be stealthy.

2) Attack Success Rate: The overall evaluation results are shown in Table II. Each individual ensemble model and the black-box models are used to make predictions on the generated adversarial data from BIM, MIM, ETAA with BIM, and ETAA with MIM. It is obvious to see that the ETAA method has improved the transferability of the adversarial attacks by comparing the average ASR under ensemble models and black-box models. The highest average ASR is 95.56% when using our ETAA with BIM method for ensemble models, which shows the generated adversarial data has reduced the gradient direction gap and can fool the other DL models with higher probabilities. The ETAA method not only performs well for the ensemble models, it also obtains high ASR for these black-box models. ETAA with BIM achieved the highest ASR of 82.69% for MLP-B and 91.06% for GRU-B. It is also noted that ETAA with MIM also performs well for GRU-B with a similar high ASR of 91.04%. Another point is that although the MIM method using RBF mode obtains the highest ASR for CNN-E, the average ASR for black-box is comparable to our ETAA method. These results prove that the proposed ETAA method is able to improve the attack transferability to the black-box attacks. The proposed ensemble stage and transfer stage narrow the differences of gradient directions between various DL models and thus the transferability of generated adversarial data can be improved.

3) Difference in Sum of Demand: The simulation results of the SD_{avg} are also listed in Table II. The smallest value is 11.84kW when using the ETAA with MIM, while ETAA method with BIM also can get a similar value of 12.13kW. Therefore, the proposed ETAA method can generate the adversarial data with the smallest changes to the total sum of demand and it can help the adversarial data to be stealthy. There are fewer changes to the total demand pattern. This result has shown that the proposed zero-mean projection of perturbations works successfully. It also shows that MIM method with RBF model has larger SD_{avg} than our ETAA method, even though it has similar ASR results as our ETAA in black-box attacks. Here we can see that the proposed ETAA method can generate adversarial demand data with higher transferability and at the same time maintains the data modifications to be as small as possible to ensure the stealthiness of the attack.

V. Conclusion

Existing works have analyzed and executed attacks to fool DL models in smart grids and specifically in DR applications. Most of the gradient-based AML attacks can generate adversarial data and use them directly to attack unseen black-box DL models based on their transferability, but with limited success. To

TABLE II
THE ATTACK SUCCESS RATE AND SD_{avg} UNDER DIFFERENT ATTACKS

	Attack Success Rate (%)											$SD_{avg}(kW)$
Method	**Substitute Model**	**Ensemble Models**						**Black-box Models**				
		CNN	MLP	RBF	LSTM	GRU	Avg	CNN-E	MLP-B	GRU-B	Avg	
BIM	CNN	69.72	19.33	16.97	35.45	34.00	35.09	19.44	25.38	27.85	24.22	43.33
	MLP	14.83	44.10	16.45	24.00	23.38	24.55	12.21	24.27	22.43	19.64	56.84
	RBF	52.21	66.88	87.49	72.53	76.3	71.08	57.32	71.08	72.81	67.07	63.67
	LSTM	22.12	15.8	11.94	64.86	31.76	29.30	7.32	18.99	30.41	25.01	234.17
	GRU	23.13	22.62	20.43	45.38	58.52	34.02	17.01	32.51	32.61	32.10	192.66
MIM	CNN	**99.39**	11.72	8.52	36.70	33.17	37.90	13.27	14.93	26.53	18.24	55.91
	MLP	11.53	58.32	14.05	25.52	22.92	26.47	9.15	19.68	22.42	17.08	35.09
	RBF	53.96	71.61	**96.26**	82.08	85.45	77.87	**60.09**	77.61	82.33	**73.54**	55.71
	LSTM	28.50	22.86	22.03	95.19	67.21	47.16	11.69	30.96	60.89	34.51	346.09
	GRU	29.07	23.35	22.07	79.16	**98.88**	50.51	12.02	35.73	60.41	36.05	353.33
ETAA with BIM	-	96.38	**93.35**	94.65	96.69	96.75	**95.56**	41.87	**82.69**	**91.06**	71.87	12.13
ETAA with MIM	-	95.67	91.15	91.90	**96.72**	96.54	94.39	46.84	82.28	**91.04**	73.38	**11.84**

Note: The highest ASR under each category like the different detection model and average value is made bold. ETAA method can have comparable ASR to existing methods and can outperform the existing methods in attack transferability. The smallest SD_{avg} is made bold. This value is smallest when using our proposed ETAA with MIM gradient attack.

further improve the transferability of adversarial data, this paper proposes the ETAA method. The proposed technique contains two stages, the ensemble stage and transfer stage, where the gradient information in different models is fetched and the gap of gradient direction is reduced. Furthermore, zero-mean projection is applied during the process to move data along the gradient direction so that the final adversarial data has minimal changes in the sum of demand data. Extensive evaluation has shown that the ETAA method can improve the transferability of adversarial data as compared to existing methods.

REFERENCES

[1] M. H. Albadi and E. F. El-Saadany, "Demand response in electricity markets: An overview," in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–5.

[2] 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.

[3] 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.

[4] Y. Huang, H. Li, K. A. Campbell, and Z. Han, "Defending false data injection attack on smart grid network using adaptive cusum test," in *2011 45th Annual Conference on Information Sciences and Systems*. IEEE, 2011, pp. 1–6.

[5] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[6] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019.

[7] M. Ashrafuzzaman, Y. Chakhchoukh, A. A. Jillepalli, P. T. Tosic, D. C. de Leon, F. T. Sheldon, and B. K. Johnson, "Detecting stealthy false data injection attacks in power grids using deep learning," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 219–225.

[8] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2017.

[9] X. Niu, J. Li, J. Sun, and K. Tomsovic, "Dynamic detection of false data injection attack in smart grid using deep learning," in *2019 IEEE ISGT*, 2019, pp. 1–6.

[10] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE S&P*. IEEE, 2017, pp. 39–57.

[11] 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.

[12] A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," in *IEEE ICMLA*. IEEE, 2017, pp. 511–516.

[13] L. Li, K. Ota, and M. Dong, "When weather matters: Iot-based electrical load forecasting for smart grid," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 46–51, 2017.

[14] N. Mohan, K. Soman, and R. Vinayakumar, "Deep power: Deep learning architectures for power quality disturbances classification," in *IEEE TAP Energy*. IEEE, 2017, pp. 1–6.

[15] C. Kaygusuz, L. Babun, H. Aksu, and A. S. Uluagac, "Detection of compromised smart grid devices with machine learning and convolution techniques," in *2018 IEEE ICC*. IEEE, 2018, pp. 1–6.

[16] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning," *IEEE Access*, vol. 6, pp. 27 518–27 529, 2018.

[17] T. Dayaratne, M. Salehi, C. Rudolph, and A. Liebman, "False data injection attack detection for secure distributed demand response in smart grids," in *European Symposium on Research in Computer Security*. Springer, 2021, pp. 1–9.

[18] 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.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[20] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *IEEE CVPR Workshops*, 2016, pp. 25–32.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *IEEE CVPR*, 2016, pp. 2574–2582.

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

[23] A. Sayghe, J. Zhao, and C. Konstantinou, "Evasion attacks with adversarial deep learning against power system state estimation," in *IEEE PESGM*. IEEE, 2020, pp. 1–5.

[24] 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.

[25] 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.

[26] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.