Adversarial Attacks to Solar Power Forecast

Ningkai Tang, Shiwen Mao, and R. Mark Nelms
Dept. of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
Email: nzt0007@auburn.edu, smao@ieee.org, nelmsrm@auburn.edu

***Abstract*—With development of the photovoltaic industry, solar power generation forecasting using weather data has become an important problem. Various machine learning (ML) algorithms have been proposed to handle the random and massive weather data, with considerable recent interest on deep neural networks (DNN). Recent studies show that DNNs are vulnerable to adversarial examples, but most prior work has focused on their impact on the classification problem. In this paper, we investigate the problem of adversarial attacks on solar power generation forecasting, which is a regression problem. We examine the impact of adversarial attacks on both the DNN model and a LASSO-based statistical model proposed in our prior work. Both white-box attack and black-box attack are examined, along with the effect of adversarial training.**

***Index Terms*—Internet of Things (IoT), adversarial attack, LASSO, deep learning, solar intensity forecast.**

I. Introduction

The smart grid (SG), where communication technologies are used to connect sensors and actuators, renewable energy sources, microgrids, electrical vehicles, generators, distributors, and customers, is an important part of the Internet of Things (IoT). To fully harvest the potential of the SG, many advanced IoT technologies have been applied for environment-friendliness, efficiency, and resilience [1], [2]. As the main issue of SG, energy management aims to achieve both stability and efficiency [3]–[6]. The day-ahead strategy, as one of the widely adopted strategy, requires utility operators to predict day-ahead power demand and generation. At the hourly or minute timescale, predictions for demand and generation can be timely updated. Conventional forecasting methodologies usually require market information and knowledge of the power grid to avoid breaking the system stability. Because both power consumption and generation change rapidly over time, short-term prediction is critical for improving grid stability and energy efficiency [2], [7].

Wind and solar power are the two important renewable energy sources. Due to their random nature, precise short-term forecasts at a 15-minutes timescale (e.g., 2-day ahead) is critical to achieve efficient energy management. To predict solar power generation, the solar irradiance needs to be estimated, since solar intensity can be directly translated to solar power generation. Thus the problem is transformed into a solar intensity forecasting problem. Various weather data based forecasting models have been developed in the literature [8]–[10]. In [11], [12], statistical and machine learning models, such as autoregressive integrated moving-average (ARIMA) and artificial neural networks (ANN), were used to analyze temperature record data. While these models performed quite well for sunny days, the precision could drop sharply in cloudy, rainy, or other extreme weather conditions.

Machine learning models have been shown quite effective for capturing the relationship between solar intensity and various weather variables [1], [2]. Machine learning models such as support vector machine (SVM) [8], ANN [12], and long short-term memory (LSTM) [1], [2] have been applied and achieved remarkable accuracy. Due to the availability of data, computing power, and open-source platforms, deep learning has become the focus of machine learning in recent years. Deep neural networks (DNN) have been applied to solar power generation forecasting. Although with its unique advantages, the inherent black box feature of DNN could potentially lead to security problems, as shown in [13]. The DNN's image classification results could be very different when the data is disturbed by adversarial examples, which are created by small disturbances almost impossible for human eyes to notice. In [14], the authors implemented adversarial training to study the effect of adversarial examples over large-scale datasets and the relationship between model size and robustness. In [15], the authors noticed that there is a universal perturbation that could affect the classification of every image. In a more practical real-world scenario, the authors in [16] tested adversarial attacks even on real 3D printed items.

Although many works have been done in the computer vision area, it is still unknown if similar attacks could be replicated on solar power generation prediction. It is also worth mentioning that very few work has been done on statistical models such as LASSO and on the regression problem. In [17], the author examined adversarial examples on a regression problem. However, the Boston Housing dataset used in [17] is usually used for linear regression; so the effect on the non-linear regression problem has not been well studied yet.

In this paper, we aim to investigate the problem of adversarial attacks to the solar power generation forecasting problem. The main contributions of this work are as follows:

- We examine how adversarial attacks impact both the DNN model and our formerly proposed LASSO-based algorithm [9]. Through experiments, we study the effectiveness of adversarial attacks on such different models.
- We evaluate both white-box attack and black-box attack methods. Our study demonstrates the severity of adversarial attacks to existing solar intensity forecast schemes.
- We apply adversarial training to examine if it helps to alleviate the deterioration of performance in solar power generation forecast under adversarial attacks.

978-1-7281-8104-2/21/$31.00

The remainder of this paper is organized as follows. We introduce the preliminaries in Section II, and then formulate the problem and describe the evaluation methodology in Section III. Our experimental study is presented in Section IV. Finally, we conclude this paper in Section V.

II. Adversarial Attack Methodology

Adversarial attack has became a hot topic in the past few years due to its impact on deep learning. Most prior work has focused on image classification, where DNN has been shown vulnerable to adversarial attacks [18], [19]. In adversarial attacks, malicious adversarial examples are generated to deceive the trained machine learning model (e.g., a classifier). Given a supervised dataset $\{x, y_l\}$, a DNN with parameter set θ is trained to predict label y_l as $f_\theta(x)$. However, adversarial examples are designed to tamper x with a small perturbation to achieve a maximized change in the inference result.

According to the specific goal, adversarial attacks can be classified as targeted and non-targeted attacks [20]. Targeted attacks use adversarial examples to delude the machine learning model to achieve specifically targeted results. Meanwhile a non-targeted attack only aims to make the inference result incorrect. According to the design, adversarial attacks can be classified into single-step and iterative methods. In single-step attacks, e.g., the Fast Gradient Signed Method (FGSM), the loss gradient is calculated once to generate the perturbation for each example. Alternatively, an iterative attack, e.g., Projected Gradient Descent (PGD), calculates the current perturbation iteratively to achieve a maximized effect.

A. Fast Gradient Signed Method (FGSM)

Unlike non-linear models, the authors in [21] showed that linear models of high-dimension are more accurate and capable of generating adversarial examples. The author proposed FGSM to quickly produce adversarial examples in [21].

Define the original, or the "clean" samples as x, the perturbation applied to each x as ξ, and the supervised learning label as y_l. The perturbation ξ should keep its infinite norm smaller than δ, which is the magnitude constraint of the perturbation, as $||\xi||_\infty < \delta$. The adversarial examples are generated as $x_a = x + \xi$. The perturbation ξ can be computed as

$$\xi = \delta \cdot \text{Sign}(\nabla_x J_\theta(x, y_l)), \tag{1}$$

where J_θ represents the Jacobian matrix as a function of x and y_l, θ represents the model parameters, and $\text{Sign}(\cdot)$ ensures the maximized change caused by the perturbation.

Once the perturbation ξ is obtained, the weight augmented perturbation is given by $w^T x_a = w^T x + w^T \xi$. If the weight w has dimension p and mean m, we can see that the activation could be increased by δpm. In high dimensional problems, the small perturbations to each dimension could add up to make a large change in the output.

B. Projected Gradient Descent (PGD)

The iterative method Projected Gradient Descent (PGD) was presented in [22], which is a multi-step variant of FGSM. While PGD can generate adversarial examples, it also provides a possible method to defend against first-order adversarial attacks. When used as a defense method, it generates adversarial examples and uses them in the training process to increase the robustness of the DNN model.

The original idea of PGD is to solve the following saddle point problem:

$$\min_\theta R(\theta), \tag{2}$$

where $R(\theta) = \mathbb{E}_{(x,y_l)\sim D}\left[\max_{s\in S} \mathcal{L}(\theta, x+\xi, y_l)\right]$ is the population risk, which is also the objective function; D is the distribution of samples that defines the distribution of x and y_l; S is a nonempty compact topological space. The inner optimization aims to maximize the loss function $\mathcal{L}(\cdot)$ over S.

In the external part of Problem (2), PGD aims to find the model parameters to minimize the loss of adversarial attack, thus the most robust DNN network can be created. As in FGSM, the internal optimization aims to maximize the loss function $\mathcal{L}(\cdot)$. The samples will have a greater probability to be adversarial examples if they satisfy the maximization condition. With these two optimization parts, the saddle point problem offers an integration of both generating adversarial examples and improving the robustness of DNN against adversarial attacks.

In practical implementation, a K-step PGD attack is executed as follows.

1) First, initialize x^0 as $x^0 = x$;
2) Then iteratively calculate x^{k+1} as $x^{k+1} = \text{Clip}_{\{x,\xi\}}\left(x^k + \alpha \cdot \text{Sign}(\nabla_x J_\theta(x, y_l))\right)$, where α is the step size. In the tth iteration, the $\text{Clip}_{\{x,\xi\}}(x^t)$ function clips x^t to $[x^t - \delta, x^t + \delta]$, where δ is the overall perturbation limit;
3) After K iterations, we obtain the adversarial example x_a as $x_a = x^K$.

III. Problem Statement and Evaluation

A. Photovoltaic Power Generation Forecast

We use both historical and forecast weather data to train the DNN network. Each sample in the dataset includes values of several weather variables, a time stamp, and the solar intensity (i.e., the label). To fully utilize the DNN, we use the weather dataset for an entire year as the training set. In real world scenarios, the photovoltaic grid usually requires at least two-day ahead forecasts in order to schedule the future operations. Moreover, forecasts at 15-minute intervals are more useful. We use part of the weather data as the test set to predict the corresponding solar intensity at 15-minute intervals. More details can be found in our prior work [9].

B. Adversarial Attack Approaches

The weather data are time series datasets that change with season, day, and time [9]. By using certain basic weather variables as in [8], we create a DNN model to accurately forecast solar intensity. Due to the direct relationship with solar intensity, the photovoltaic power generation forecast is readily turned into a solar intensity forecast problem.

Although adversarial attacks have been studied and tested for DNN-based classification problems, there are few prior works related to the regression problem. It remains unknown whether adversarial attack will affect DNN-based regression models and how effective it will be. *We aim to study if adversarial attack is effective for solar intensity forecasting (i.e., an regression problem) and whether adversarial examples are effective to attack the DNN model and the LASSO based algorithm proposed in our prior work* [9].

In our experiments, we first train a DNN model for solar intensity forecasting. After training the DNN model, we use both FGSM and PGD to generate adversarial examples and test their effect on the trained DNN model. Depending on whether the attacker has acquired information about the targeted model, there are two different types of attacks: white box attack and black box attack. On one hand, white box attack is easier to execute and more effective; On the other hand, white box attack is not so practical since the attacker needs to gain knowledge of the target model. Black box attack is more realistic but usually is less effective than white box attack.

1) White Box Attack: In our white box attack experiments, we assume the target DNN model is exactly the same as the trained model [23], meaning that the model weights, target architecture, training method, activation function, and input format are all known information to the attacker. Therefore, generating adversarial examples from the trained model is equivalent to calculating gradients from the target model. In our simulations, FGSM and PGD are both used to launch white box attacks to the target DNN model.

2) Black Box Attack: Black box attack assumes the attacker is unable to access the target model but can only have the information about dataset. Several methods have been implemented to generate black box attacks to neural network models. For example, the zero-order optimization based attack was able to launch black box attacks without knowing gradients [24], where the estimated Hessian was used to approximate the correspondent target model parameters. Similar to this idea, other algorithms such as [25] were also developed. Researchers also tried to exploit the transferability of adversarial examples. The authors in [26] showed that adversarial examples were capable of fooling other neural networks with different architectures.

In our problem where DNN is used as a regression model, the impact of black box adversarial attack is yet to be studied. In addition, we use adversarial examples generated from the DNN model to test our LASSO based model [9]. Since the partially linear characteristics has been proved on certain datasets [8], it is interesting to see how well adversarial attacks work as a black box method on such statistical models.

3) Evaluation: To evaluate the effect of adversarial attacks, we use the root mean squared error (RMSE) between the forecasted and the observed solar intensity (i.e., the ground truth). Let n be the number of forecasted solar intensity values, and $\hat{y}_t$ and y_t the forecasted solar intensity and ground truth at time t, respectively. The RMSE is given by RMSE $= \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}$.

The forecasted solar intensity sets are divided into the original forecasted and adversarial attacked forecasted data. We evaluate the RMSE on both sets to find out the degradation from before-attack to after-attack. We also test the adversarial training capability provided by PGD to see if adversarial training could mitigate the loss caused by adversarial examples.

IV. Experimental Validation

In this section, we present our experimental validation of adversarial attacks on both DNN and LASSO-based models. Two different datasets gathered from the US and China, respectively, are used in our experiments.

A. Data Description

The China dataset was recorded from a photovoltaic station located in Zhuji, Zhejiang Province. Weather data was recorded every 15 minutes for a period of two years from January 2019 to December 2020, including temperature in Celsius degrees (^{o}C), pressure in Pascal (100Pa), humidity in percentage ($\%$), wind speed in meters per second (m/s), wind direction in degrees (^{o}), solar intensity in watt per square meter ($watt/m^2$), and timestamp. The datasets are well maintained, and no corrupted data sample is discovered.

The other dataset was collected by a weather station located in Amherst, MA, USA [27]. The weather data was recorded at 5-minute interval; the sensors of the weather station gathered data samples including temperature, wind chill, humidity, dew-point, wind speed, wind direction, rainfall, barometric pressure, sunlight, and Ultraviolet. The dataset was recorded through February 2006 to January 2013. Note that this dataset contains corrupted and missing samples, which are excluded in our experiments. In the dataset, temperature and dew-point are measured in Fahrenheitt (^oF), humidity in percentage ($\%$), windspeed in miles per hour (mph), rainfall in inches (inch), and solar intensity in watts per square meter (watt/m^2). Since the UMass dataset is good for day-average forecast, we calculate the average solar intensity by day.

B. White Box Attack with the Zhejiang Dataset

The Zhejiang dataset is first used to test the DNN model. During the training stage, weather data from Year 2019 are used. The structure of the DNN model is shown in Fig. 1. The DNN model contains an input layer, a $10 * 9 * 9$ dense layer to augment the input, three conv2d layers to compute high dimensional variables, one dropout layer with flatten layer to avoid overfitting, another $10 * 10$ dense layer, and an output layer. The Adam optimization from TensorFlow 2 with a learning rate of 0.001 is used to train the DNN network. The mean squared error between the forecasted solar intensity and the ground truth is used as loss function.

Our test data is the forecasted weather data (i.e., the future weather data is assumed unknown at the time of inference) and real solar intensity data (i.e., ground truth) in Year 2020. The first seven days are used for the results presented in this paper.

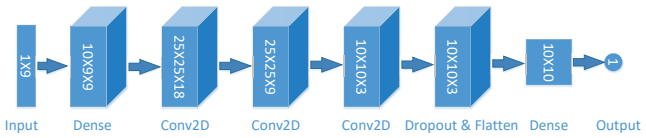

Fig. 1. Structure of the DNN model used in our experiments.

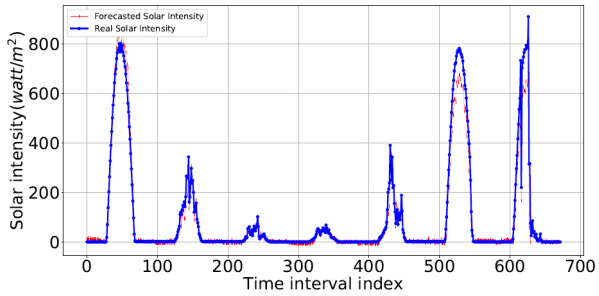

Fig. 2. DNN forecasted solar intensity vs. ground truth.

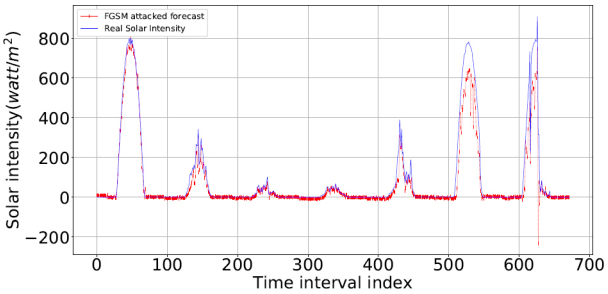

Fig. 3. DNN forecasted results using FGSM attacked data vs. ground truth.

After $10,000$ epochs of training, the forecast results using the DNN model are presented in Fig. 2. The RMSE of the DNN model for the first seven days of year 2020 is 21.0038 watt/m^2, while the RMSE for the entire year is 22.4122 watt/m^2. These results demonstrate the capability DNN, which can accurately predict solar intensity using historical weather data.

After validating the DNN model, we start to generate adversarial examples and test them on the trained DNN model. First, we apply FGSM to generate adversarial examples with perturbation limit $\delta = 0.01$. The comparison with real solar intensity is shown in Fig. 3. There are considerable differences between the DNN results using the original data and that using the adversarial attacked data. After the FGSM attack, the 7-day RMSE of the DNN model becomes 49.6529 watt/m^2, which is 2.3643 times of that with the original data using the same DNN model. Therefore, the answer is "yes" as to whether adversarial examples are effective to the regression problem.

After testing with FGSM, PGD is also examined in our experiments. The perturbation limit is set to $\delta = 0.01$, the attack iteration step size is $\alpha = 0.0001$, and the number of attack iterations is set to 40. In Fig. 4, we present the generated adversarial examples with a perturbation limit of $\delta = 0.01$, using the 0-1 normalized temperature data as an example.

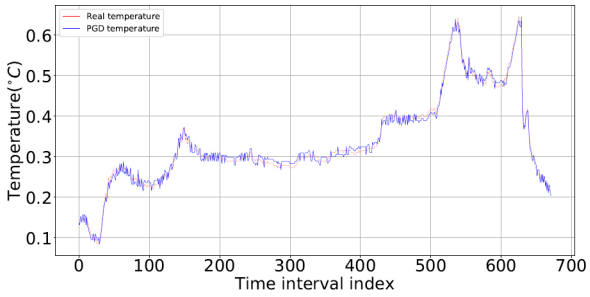

Fig. 4. Adversarial examples on 0-1 normalized temperature data.

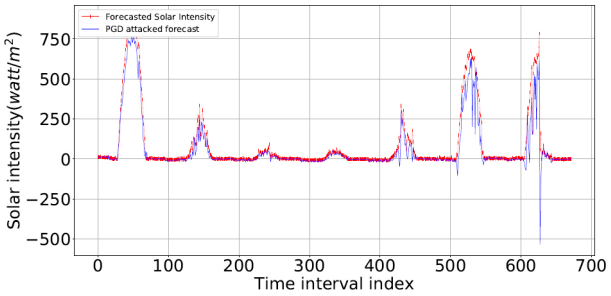

Fig. 5. DNN forecasted results using the original data vs. PGD attacked data.

It can be seen that the perturbations introduced by PGD are visually very small.

With PGD generated adversarial examples, forecasted results become even more distorted as shown in Figs. 5. The RMSE under PGD attacks is 94.1743 watt/m^2, which is a 97.5215% increase over the FGSM RMSE, and more than 4.4837 times of that with the untampered data. It is easy to see PGD has a stronger effect than FGSM. Solar intensity cannot be something away from "0" through midnight to the next morning. One fascinating characteristics of adversarial attacks is that they do not create suspicious values when the original samples are "0." With this feature, it is hard to identify the attack by just inspecting the curves.

Since PGD also provides us with the ability of adversarial training, we next examine if adversarial training can help to improve the resilience of DNN regression models. The 7-day results under PGD adversarial training are shown in Fig. 6. The RMSE for the 7-day period is 63.6610 watt/m^2, a 32.4010% reduction from that obtained without adversarial training, which demonstrates the benefit of PGD adversarial training. However, the overall RMSE of forecasting on untampered data with the adversarially trained DNN model has increased to 56.1179 watt/m^2, which is a 167.1797% increase from that obtained without adversarial training. Therefore, using adversarial examples can be seen as a trade-off between robustness and accuracy. Although adversarial training provides resilience against adversarial attacks, it also sacrifices the accuracy of forecasting with untampered data.

A comparison of the RMSE results achieved by the models under various attacks methods is summarized in Table I.

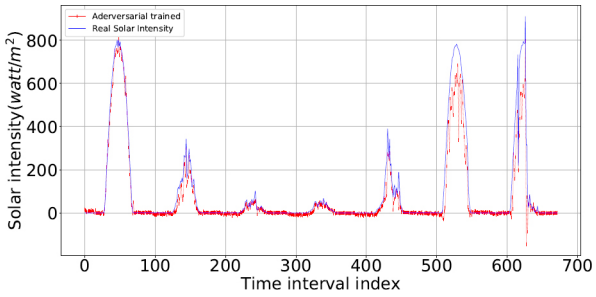

Fig. 6. Adversarial trained forecast results vs. ground truth.

TABLE I
RMSE COMPARISON OF WHITE BOX ATTACKS ON THE DNN MODEL WITH THE ZHEJIANG DATASET

	DNN	FGSM	PGD	Adv. Trained
RMSE (watts/m^2)	21.0038	49.6529	94.1743	63.6610
Increase over DNN	–	136.40%	348.37%	203.09%

C. Black Box Attack with the UMass Dataset

In [8], the authors applied the multi-linear regression technique to forecast solar intensity, where the data exhibits a strong linear correlation between weather parameters and solar intensity. In [2], several different models were used to capture the linear and non-linear relationship, respectively. Since the data structure is partially linearly correlated, we conjecture that a black box attack using adversarial examples generated from a well-trained DNN model would also be effective.

In this experiment, we use the PGD algorithm to generate adversarial examples. In [9], we proposed a LASSO-based algorithm to achieve a very decent accuracy with the UMass dataset [27], We use the dataset again here to demonstrate the impact of black box attacks on the statistical model. Since there are only five usable weather variables in the dataset and the data is averaged by day, the DNN model needs to be slightly modified to fit such data for generating adversarial examples. We use the forecasted temperature, dew-point, wind speed, precipitation, and humidity as input, and solar intensity is the output as before. In this scenario, our LASSO-based algorithm forecasts solar intensity by day average [9].

The results achieved by our LASSO-based algorithm on the PGD attacked data are presented in Fig. 7 along with the ground truth, where the perturbation limit is $\delta = 0.01$. The forecasting results by LASSO using the untampered UMass data are re-plotted in Fig. 8. Compared with the RMSE of 14.0262 watt/m^2 in Fig. 8, the RMSE of LASSO on PGD attacked data is increased to 91.1597 watt/m^2 (i.e., 5.4992 times higher). From Fig. 7, we find that the threshold feature of LASSO-based model [9] alleviates the loss caused by the adversarial attack. No negative values are generated for the evenings so the RMSE is reduced in the corresponding regions.

We also experiment with different perturbation levels, which are set to $\delta = 0.15$ and $\delta = 0.2$. The RMSEs for each of the black box attack scenarios on the LASSO-based model using the UMass Dataset are summarized in Table II.

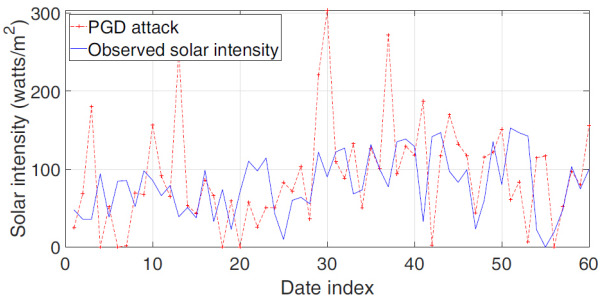

Fig. 7. Forecasting results achieved by LASSO using PGD attacked UMass data ($\delta = 0.01$) vs. ground truth.

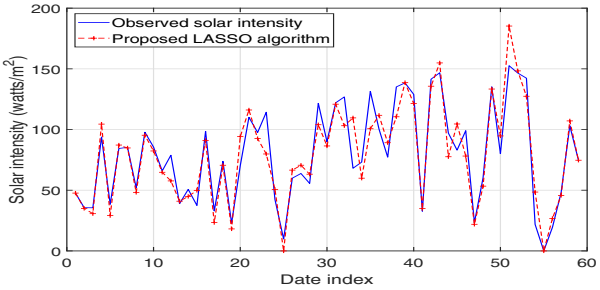

Fig. 8. Solar intensity prediction using the proposed LASSO-based method with the untampered UMass dataset ($\delta = 0$) vs. ground truth [9].

TABLE II
RMSE COMPARISON OF BLACK BOX ATTACKS ON THE LASSO-BASED MODEL WITH THE UMASS DATASET

Perturbation Limit	0.01	0.15	0.2
RMSE (watts/m^2)	91.1597	111.2547	156.9859

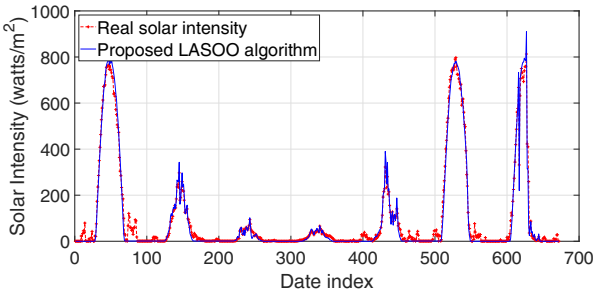

Fig. 9. Solar intensity prediction using the LASSO-based method with the untampered Zhejiang dataset ($\delta = 0$) vs. ground truth.

D. Black Box Attack with the Zhejiang Dataset

Finally, we test our LASSO-based model with the Zhejiang dataset. For comparison, we use the LASSO-based model to solve the same 15-minute interval problem, where time stamp is also used as variables. The experiment results of solar intensity forecasting achieved by the LASSO algorithm using the untampered Zhejiang dataset are presented in Fig. 9. The corresponding results using the PGD attacked Zhejiang dataset with perturbation limit $\delta = 0.01$ are presented in Fig. 10.

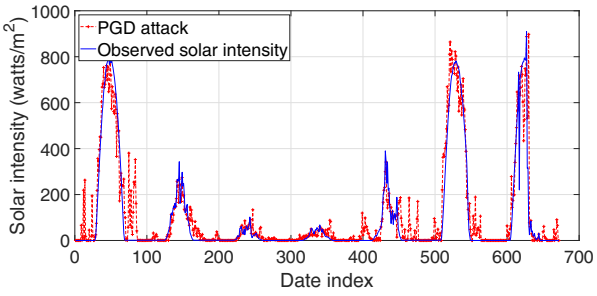

Fig. 10. Solar intensity prediction using the proposed LASSO-based method with the PGD attacked Zhejiang dataset ($\delta = 0.01$) vs. the real solar intensity.

Specifically, the LASSO-based model achieves an RMSE of 24.6907 watt/m^2 with the untampered Zhejiang dataset. The black box attack still affects the forecasting result, but the LASSO-based model performs a little better than the DNN model. It is more resilient to the PGD attacks with an RMSE of 73.6029 watt/m^2 (compared to the DNN model's RMSE of 94.1743 watt/m^2). As discussed, this is due to the LASSO-based algorithm's threshold feature. Thus, without any doubt, PGD adversarial examples generated by a DNN model can be used to launch black box attacks to our LASSO-based statistical model, or even other different models. This means, in addition to DNN, conventional solar generation forecast schemes are also vulnerable to adversarial attacks.

V. Conclusion

In this paper, we examined how adversarial attacks affect both DNN and our LASSO-based algorithm. We successfully used FGSM and PGD to generate white box attacks on the trained DNN model and found that PGD attacks were effective and PGD adversarial training only provided limited protection over regression problems. Furthermore, we tested PGD black box attacks on the LASSO-based algorithm. The results showed that adversarial attacks were capable of black box attacks and were likely to pose a deep threat to the regression problems with similar data structures. For future work, it would be interesting to develop defense mechanisms to make the DNN and statistical models more resilient.

Acknowledgments

This work is supported in part by the National Science Foundation under Grant CNS-2107190.

References

[1] Y. Wang, Y. Shen, S. Mao, G. Cao, and R. M. Nelms, "Adaptive learning hybrid model for solar intensity forecasting," *IEEE Transa. Ind. Informat.*, vol. 14, no. 4, pp. 1635–1645, Apr. 2018.

[2] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, "LASSO and LSTM integrated temporal model for short-term solar intensity forecasting," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 2933–2944, Apr. 2019.

[3] Y. Huang, S. Mao, and R. M. Nelms, "Adaptive electricity scheduling in microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 270–281, Jan. 2014.

[4] H. Zou, Y. Wang, S. Mao, F. Zhang, and X. Chen, "Distributed online energy management in interconnected microgrids," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2738–2750, Apr. 2020.

[5] H. Zou, S. Mao, Y. Wang, F. Zhang, X. Chen, and L. Cheng, "A survey of energy management in interconnected multi-microgrids," *IEEE Access Journal*, vol. 7, pp. 72 158–72 169, 2019.

[6] H. Zou, Y. Wang, S. Mao, F. Zhang, and X. Chen, "Online energy management in microgids considering reactive power," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2895–2906, Apr. 2019.

[7] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, Feb. 2001.

[8] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE SmartGridComm'11*, Brussels, Belgium, Oct. 2011, pp. 528–533.

[9] N. Tang, S. Mao, Y. Wang, and R. M. Nelms, "Solar power generation forecasting with a LASSO-based approach," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1090–1099, Apr. 2018.

[10] Y. Wang, G. Cao, S. Mao, and R. Nelms, "Analysis of solar generation and weather data in smart grid with simultaneous inference of nonlinear time series," in *Proc. IEEE INFOCOM WKSHPS'15*, Hong Kong, China, Apr./May 2015, pp. 600–605.

[11] J. Wu and C. Chan, "The prediction of monthly average solar radiation with TDNN and ARIMA," in *Proc. 11th Int. Conf. Machine Learning Applications*, Boca Raton, FL, Dec. 2012, pp. 469–474.

[12] H. Hejase and H. Assi, "Time-series regression model for prediction of mean daily global solar radiation at al-ain, uae," *ISRN Renewable Energy Energy*, vol. 2012, p. Article ID 412471, Apr. 2012.

[13] 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*, Feb. 2014. [Online]. Available: https://arxiv.org/abs/1312.6199

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, Feb. 2017. [Online]. Available: https://arxiv.org/abs/1611.01236

[15] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE CVPR'17*, Honolulu, HI, July 2017, pp. 1765–1773.

[16] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. 2018 International conference on machine learning (ICML)*, Stockholm, Sweden, July 2018, pp. 284–293.

[17] A. T. Nguyen and E. Raff, "Adversarial attacks, regression, and numerical stability regularization," *arXiv preprint arXiv:1812.02885*, Dec. 2018. [Online]. Available: https://arxiv.org/abs/1812.02885

[18] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proc. IEEE INFOCOM'20*, Toronto, Canada, July 2020, pp. 2469–2478.

[19] M. Patil, X. Wang, X. Wang, and S. Mao, "Adversarial attacks on deep learning-based floor classification and indoor localization," in *Proc. 2001 ACM Workshop on Wireless Security and Machine Learning (WiseML'21)*, Virtual Conference, June-July 2021, pp. 1–6.

[20] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sept. 2019.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR'15*, San Diego, CA, May 2015, pp. 1–11.

[22] ——, "Towards deep learning models resistant to adversarial attacks," in *Proc. ICLR'18*, Vancouver, Canada, Apr.-May 2018.

[23] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[24] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, Nov. 2017, pp. 15–26.

[25] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[26] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. 2016 IEEE European Symp. Security Privacy*, Saarbrücken, Germany, Mar. 2016, pp. 372–387.

[27] University of Massachusetts, "The UMass trace repository," [online] Available: http://traces.cs.umass.edu/index.php/Sensors/Sensors/.