Lasso Based Schemes for Solar Energy Forecasting

by

Ningkai Tang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 1, 2021

Keywords: Smart Grid, LASSO, Machine learning, Solar Intensity Forecast, Adversarial Attack, Renewable Energy

Approved by

Shiwen Mao, Chair, Professor and Earle C. Williams Scholar Chair
Mark Nelms, Department Chair, Professor of Electrical and Computer Engineering
Xiaowen Gong, Assistant Professor of Electrical and Computer Engineering
Yin Sun, Assistant Professor of Electrical and Computer Engineering

Abstract

The smart grid (SG) has emerged as an important form of the Internet of Things (IoT). Despite the high promises of renewable energy in the SG, it brings about great challenges to the existing power grid due to its nature of intermittent and uncontrollable generation. In order to fully harvest the high potential of SG, accurate forecasting of renewable power generation is indispensable for effective power management. In this dissertation, we propose a least absolute shrinkage and selection operator (LASSO) based forecasting model and algorithm for solar power generation forecasting. We compare the proposed scheme with two representative schemes using three real world datasets. We find that the LASSO-based algorithm achieves a considerably higher accuracy comparing to the existing methods, using fewer training data, and being robust to anomaly data points in the training data. LASSO's variable selection capability also offers a convenient trade-off between computational complexity and accuracy. These advantages all make the proposed LASSO based approach a highly competitive solution to forecasting solar power generation.

With the development of the photovoltaic industry, solar power forecasting using weather data has become more and more important. Due to weather data's random and massive nature, many machine learning (ML) algorithms have been proposed. Among these, deep neural networks (DNN) is one of the most widely used ML algorithm. However, some recent studies show that certain algorithms are extremely vulnerable to adversarial examples, which are maliciously generated by cyber attackers. Such tampered examples can fool the DNN to produce some completely different result. In actual situation, the attacker can manipulate the weather data stored or to be transferred to the forecast model. The adversarial examples will greatly affect the original forecast values, which will cause power outage or even severe power grid disaster. In this dissertation, results point out that certain attacks are effective for both black box attack to DNN base models and white box attack to other algorithms. Through simulations, we will show that small perturbations introduced by adversarial examples could lead to distinct

outcomes, which will allow the attacker to cause a maximized loss while staying undetected. Moreover, we will use two types of adversarial attacks to show that the effect can be improved by iterative methods. Finally, we will implement the adversarial examples to our LASSO-based algorithm to demonstrate the effect of white box attacks.

Acknowledgments

In the beginning, I would like to express my deepest thanks to my committee chair and advisor Dr. Shiwen Mao, who has guided me over smart grid and wireless area throughout my graduate student life and inspired me of this idea. Without his support, I could have no chance to learn so much knowledge, not to mention to behave like a real researcher.

I also would like to thank all the other committee members of mine. Dr. Nelms has introduced a lot of power grid knowledge to me in his class; Dr. Xiaowen Gong has explained some important concepts in machine learning clearly to me which are very crucial to my dissertation; And Dr. Yin Sun has get me through with my algorithms. Sincerely thank you, for I could never have accomplished pursuing my Ph.D. degree here without you all.

Then, I need to thank all the other professors who has taught me in Auburn University: Dr. Prathima Agrawal, Dr. Vishwani Agrawal, Dr. John Hung, Dr. Jitendra Tugnait, Dr. Fa Dai, Dr. Bogdan Wilamowski and Dr. Chwan-Hwa Wu. Your knowledge has indeed broadened my view. Meanwhile, I also need to thank all staff in graduate school especially Ms. Sherry Ray, without their help I would probably lose the opportunity to defense in time.

In addition, I want to take this opportunity to appreciate the friendship and support from all my fellow colleagues in the Electrical and Computer Engineering at Auburn University through out these years: Chao Yang, Zhitao Yu, Xiangyu Wang, Lingxiao Wang, Runze Huang, Jing Ning, Dr. Yi Xu, Dr. Hui Zhou, Dr. Yu Wang, Dr. Zhifeng He, Dr. Zhefeng Jiang, Dr. Yu Wang (Same name), Dr. Xuyu Wang, Dr. Yingsong Huang and Dr.Mingjie Feng. Especially, I want to thank Ticao Zhang who kindly provided me with accommodation during the COVID-19 epidemic.

Finally, I would like to thank my dear parents, parents in law, my daughter and wife for their understand and support all these years, I really owe you a lot. You are my strongest backup and motivation all these years.

This work is supported in part by the US NSF under Grants DMS-1736470, ECCS-1923163, and CNS-1822055, and through the Wireless Engineering Research and Education Center (WEREC) at Auburn University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the foundation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The past decade has witnessed a rising of the Internet of Things (IoT), largely due to the fast development of wireless communications and mobile computing. As inter-connected devices and systems become "smarter," people's life style has been changing rapidly, and some traditional industries are undergoing fast changes. To utilize the benefits provided by the wireless technology, many intelligent ideas and systems have been proposed in the literature. Smart grid (SG), as an ongoing revolution for the power grid, becomes an indispensable part of the IoT.

SG is regarded as the next generation power grid. It is supposed to replace the current inefficient and vulnerable power grid. The advanced control techniques and communication systems allow SG to achieve higher power efficiency while maintaining its stability. Nowadays, SG is characterized by the two-way flow of both power and information, microgrid, and distributed renewable energy resources [3].

With the development of SG, renewable energy, such like photovoltaic and wind power, encounter an opportunity to replace conventional thermal power supplies. However, before we could harvest the advantages of the promised clean energy, there are still many challenging problems that need to be dealt with. For example, the fluctuation of solar power generation may cause unexpected problem toward macro grids, which must balance the generation and demand all the time [4].

For the reasons mentioned above, accurate solar generation forecasting remains one of the primary challenges in the area of renewable energy. To address the problem, many algorithms from statistics and machine learning have been proposed but a more accurate and less computational expensive model is always needed.

Nowadays, more and more industries start to use machine learning algorithms as a possible solution to solar generation forecast. Among them, deep neural network (DNN) is one of the most frequently used methodology because of its accuracy and versatility. However, recent studies show that DNN could be fooled by adding very tiny perturbation to the input samples. Adversarial attack, which is capable of fooling a well trained DNN model, has the potential to impact on results other than image classification. This problem also raises the concern about whether it will undermine solar power generation forecasting results.

Motivated by these problems, this dissertation aims to build a precise and fast algorithm for solar power generation forecasting and to evaluate the threat that the current forecasting schemes are facing from adversarial attacks.

The main contributions of this dissertation are summarized as follows:

- We propose a LASSO-based algorithm that accurately predict solar power generation with a small amount of historical data. While using fewer training data, the proposed algorithm can achieve a considerably higher accuracy compared to the existing methods, and is robust to anomaly data points in the training data. In addition, the variable selection capability of the proposed scheme offers a nice trade-off between computational complexity and accuracy, which makes it a highly competitive solution to forecasting of solar power generation.

- We examine how adversarial attack affects both the DNN model and our proposed LASSO-based algorithm for solar power generation forecasting. We use the Fast Gradient Signs Method (FGSM) method and the Projected Gradient Descent (PGD) method to generate white box attacks on a well trained DNN model and find that PGD adversarial training

only provides a limited protection over regression problems. We also show that adversarial attack is capable of black box attack to the LASSO model. It is likely to be a deep threat on the forecasting problem with similar data structure.

Chapter 2

Solar Power Generation Forecasting with a LASSO-based Approach

2.1 Introduction

Internet of things (IoT) is defined as uniquely identifiable objects that are organized in an Internet like structure. With technology developments and evolution of the power grid, the concept of smart grid (SG) has emerged. It is regarded as the next generation power grid [5] and becomes an important part of the IoT [5]. A smart grid is an electricity network that can intelligently integrate the interactions of all users connected to generators, consumers, and those that assume both roles, in order to efficiently deliver sustainable, economic and secure electricity supplies [5]. Such capabilities are enabled by the computation, communication, and control mechanisms that are incorporated with the power grid, where a large amount of interconnected wireless sensors (e.g., phasor measurement units (PMU)) are deployed to obtain real-time state information, while many actuators are deployed to enforce power scheduling, protection, and security decisions. Because of the IoT based technology, SG and IoT are naturally inseparable. Recently, numerous IoT technologies has been developed to fulfill the SG's potential, including energy distribution and management [6–9], load balancing [10], security and privacy [11], and the future smart building framework [12].

The smart grid is characterized with the two-way flow of power and information, microgrid, and distributed renewable energy resources (DRERs) [3]. In the meantime, the rise of new energy (e.g., photovoltaic power such as solar power) has brought new challenges to such

unconventional power networks. Although integrating the power charge from solar power generators could reinforce the macro grid, a large and uncertain amount of power generated by micro solar grids could lead to severe energy management problems [4].

In order to fully harvest the potential of DRERs, two key techniques, load forecasting (i.e., to predict the amount of power needed to achieve the demand and supply equilibrium) and power generation forecasting (i.e., to predict how much power will be generated at a future time), are indispensable. Load forecasting has been well studied in the literature [13–15], with different statistics and machine learning approaches, such as nonparametric functional time series analysis, state space models, and artificial neural networks. Similarly, generation forecasting has been investigated with various models and methods as well.

Since solar power generation is linked directly to solar intensity, the solar power forecasting problem naturally translates to a weather forecasting problem. In [16, 17], support vector machine (SVM) and nonlinear time series are used to predict solar intensity, respectively. Other prior works such as [18–20] also provided various effective solutions to the solar intensity prediction problem. Although the prior works have done a good job on achieving a low error rate, there is always room for improvement for more accurate forecasting. In addition, a deeper analysis will be helpful to gain a good understanding of the problem. For example, the SVM-based method [16] achieves a low error rate, but the selection of kernel is usually based on experience. For neural network based technologies [20, 21], the neural network structure needs to be pre-designed and a quite complicated structure is needed to achieve a good precision, which, however, leads to a high computational cost. In rainy or cloudy days, the time-series based method [22] are usually not effective to capture the high variations in data. In [17], we present a local linear model for nonlinear time series, which leads to an accurate approximation and an analysis on the relationship between the renewable power generation process and the weather variable processes. However, the importance of each variable is yet to be better identified.

In this chapter, we investigate the solar power generation forecasting problem, aiming to develop an effective method that not only achieve a high forecasting accuracy, but also helps to reveal the significance of weather variables. To this end, we propose a least absolute shrinkage and selection operator (LASSO) based method for solar power generation forecasting based on

historical weather data. Based on a single index model and LASSO, we develop an effective algorithm that maximizes Kendall's tau coefficient to estimate the prediction model coefficients. The goal of variable selection is achieved by the nature of LASSO, which automatically reduce the weights of less important variables and increase the sparsity of the overall coefficient vector. With the proposed algorithm, we can either maximize the prediction accuracy using all the weather data/variables, or achieve a trade-off between accuracy and complexity by using a limited number of variables. The proposed scheme is evaluated with the real dataset collected from a weather station [1], and comparison to two representative benchmark schemes. The proposed LASSO-based scheme outperforms both existing schemes with considerable reduction in prediction error.

The remainder of this chapter is organized as follows. We introduce LASSO and the forecasting problem in Section 2.3. Then we present our LASSO-based algorithm in Section 2.4. Meanwhile, we also introduce the time-series solution proposed in [17], which is used as a benchmark. We validate the performance of our solution and compare it with two benchmark schemes in Section 2.5. We conclude the chapter in Section 2.6.

2.2 Related Work

Power generation prediction is an essential issue in smart grid network, especially for system integrated with new energy such as wind or solar generation. In order to make accurate prediction, considerable work has been done. In addition to those discussed in the introduction section, we review several additional key related work here.

In [16], methods such as past-predicts-future (PPF) model, linear least square regression, and SVM regression (SVMR) are adopted to solve the solar generation prediction problem. According to the authors, linear regression and SVM outperform PPF due to the change of weather pattern. Also, as a conclusion, SVM shows a higher accuracy while linear least square achieves an interpretable model. Although the idea is novel, the accuracy of these models are yet to be improved.

Inspired by this paper, we propose the time-series based algorithm TLLE [17], which separates the historical data into small neighborhoods and estimates the coefficient of each

neighborhood with a linear solution. While the algorithm improves the accuracy comparing to MLR and SVM, the model also provides simultaneous confidence interval to making further interpretation for each covariate. However, the non-linear nature of the target problem could undermine the interpretation of the results.

Other than these statistical methods, neural network is also a widely used approach to address the problem. The authors in [19] propose several different neural network models as competitors. In this paper, the authors noticed the fact that weak stationarity and lack of continuity result in volatilities of solar data. Thus the proposed models use historical data similar to the target day to perform prediction. The accuracy for certain models are great but we should be aware of the case when there are no historical, similar days. Also, the heavy computation of neural networks is always a concern.

Not limited to algorithms, data preprocessing could become a useful means to reduce error. In [23] and [24], the authors propose to classify the data by weather conditions before using their similar-day based neural network algorithms. Specifically, in [23], the authors classify historical data by irradiance, total cloud, and cloud cover, while in [24], the authors take the weather feature such as sunny or cloudy for data classification. As certain tricks surely improve the performance of model, we should still notice that the availability and accuracy of these features are highly dependent on location of dedicated datasets, making the schemes less flexible.

2.3 Forecasting Model and Problem Statement

2.3.1 LASSO Preliminaries

In machine learning and statistics, LASSO has become a popular method for regression analysis, ever since it was firstly introduced by Robert Tibshirani in 1996 [25]. By applying LASSO to practical problems, we benefit from two main functions that LASSO has: regularization and variable selection. Due to the nature of LASSO, while a stronger l_1 penalty is used, LASSO is encouraged to shrink its coefficients to 0. In other words, it performs variable selection by dropping the corresponding variables from the model and achieves a sparse solution in

this case. On the other hand, while a weaker l_1 penalty is used, the algorithm tends to retain most variables and predict with better regularization. The level of l_1 penalty can be chosen by automatic techniques like cross-validation or by manually using the regularization path. In recent years, LASSO has been successfully applied to various SIMs [26–28] due to the above mentioned capability.

We propose to use LASSO for solar power generation with high accuracy. In addition, since weather data gathered from the local weather station can vary in different types of weather parameters to monitor, it is important to find out which variables are more important on solar power generation, especially when lacking of sufficient weather information, or when computation complexity is a concern. As discussed, linear regression, neural networks, and SVM based algorithms have already been applied to the solar power generation forecasting problem. To the best of our knowledge, this is the first application of LASSO to the problem, to achieve high prediction precision as well as variable selection.

2.3.2 System Model

The solar power forecasting problem is a good match for the single index model (SIM), which has the advantage of avoiding the so-called "curse of dimensionality" in fitting multivariate nonparametric regression functions by focusing on an index [29]. Specifically, we adopt the SIM as follows:

$$Y|\mathbf{X} \sim P(\cdot, f(\mathbf{X}^T\boldsymbol{\beta})), \tag{2.1}$$

where $Y \in \mathbb{R}$ is the response, $\mathbf{X} \in \mathbb{R}^p$ are the covariates, p is the dimension of variables, $P(\cdot, \theta)$ represents a stochastically increasing family of functions with parameter θ, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the coefficient of the covariate $\mathbf{X}$ and is unit normed, and $f(\cdot)$ is an unknown strictly smooth increasing link function.

To relate the model in (2.1) to our problem, Y is our desired estimation of solar intensity and $\mathbf{X}$ is the weather data collected from a weather station. In our forecasting algorithm, we

use a special case of the model as follows.

$$Y = f(\mathbf{X}^T\boldsymbol{\beta}) + \epsilon, \tag{2.2}$$

where ϵ is a zero mean variable with a finite variance representing error. Specifically, the weather data collected from a local weather station, $\mathbf{X}$, consists of five weather data variables, including temperature, humidity, dew point, wind speed, and precipitation, which compose a 5-dimensional dataset.

2.3.3 Kendall's tau Coefficient

With a set of i.i.d. data samples, the proposed algorithm is capable of simultaneous variable selection and forecasting through optimizing the relationship between Y and $\mathbf{X}^T\boldsymbol{\beta}$. Although it is not clear whether the problem is linear or not, the Multi-linear Regression (MLR) based approaches did not show a satisfying performance in [16] and [17], which could be an indicator that linear model is not suitable for the problem. Therefore, we propose to use Kendall's tau coefficient between Y and $\mathbf{X}^T\boldsymbol{\beta}$ instead of Pearson's correlation coefficient [30].

Kendall's tau coefficient is a statistic used to measure the rank correlation between two quantities [30]. Comparing to the widely used Pearson's correlation coefficient, which is a linear correlation measurement, Kendall's tau coefficient is more suitable for non-linear problems. Also, since the assumption of monotonicity, if we ignore the outliers caused by random errors, the increments of Y is highly possible to be synchronized with $\mathbf{X}^T\boldsymbol{\beta}$. Thus, we can precisely estimate $\boldsymbol{\beta}$ by maximizing the following Kendall's tau coefficient between Y and $\mathbf{X}^T\boldsymbol{\beta}$.

Assume there are n data units, $\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n\}$, and the corresponding response values are $\{Y_1, Y_2, ..., Y_n\}$, respectively. For discontinuous $\boldsymbol{\beta}$, Kendall's tau coefficient is expressed as

$$\tau_n(\boldsymbol{\beta}) = \frac{1}{n(n-1)} \sum_{1 \le i_1 \ne i_2 \le n} \text{sign}(Y_{i_2} - Y_{i_1}) \cdot \text{sign}\left(\mathbf{X}_{i_2}^T\boldsymbol{\beta} - \mathbf{X}_{i_1}^T\boldsymbol{\beta}\right), \tag{2.3}$$

where sign$(\cdot)$ is the signum function. For the continuous form of $\boldsymbol{\beta}$, Kendall's tau coefficient is defined as

$$\tau_n^*(\boldsymbol{\beta}) = \frac{1}{n(n-1)} \sum_{1 \le i_1 \ne i_2 \le n} \text{sign}(Y_{i_2} - Y_{i_1}) \tanh\left(\frac{\mathbf{X}_{i_2}^T\boldsymbol{\beta} - \mathbf{X}_{i_1}^T\boldsymbol{\beta}}{c}\right), \tag{2.4}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function and c is a small constant, which can be seen as a given value.

2.4 Proposed Algorithm

We present the proposed solution algorithm in this section. In particular, in Section 2.4.1, we introduce the proposed LASSO based algorithm. In Section 2.4.2, we discuss how to choose the parameters used in the algorithm. In Section 2.4.3, we show how to apply the proposed LASSO based algorithm for the solar intensity prediction problem.

2.4.1 Proposed Algorithm

With the definition of Kendall's tau coefficient, the proposed solution algorithm consists of two parts, i.e., coefficient estimation and link function estimation. The proposed algorithm consists of the following three steps.

Coefficient Estimation

First we need to find an index j that can maximize the following value ρ_j, $j = 1, 2, ..., p$, where p is the dimension of the variables.

$$\rho_j = \frac{1}{n(n-1)} \sum_{1 \le i_1 \ne i_2 \le n} \text{sign}(Y_{i_2} - Y_{i_1}) \cdot \text{sign}(X_{i_2 j} - X_{i_1 j}). \tag{2.5}$$

We call this index j_1, and set $\hat{\boldsymbol{\beta}}_{(1)} = \text{sign}(\rho_{j_1})\boldsymbol{e}_{j_1}$, where $\boldsymbol{e}_j = [0, ..., 1, ..., 0]^T$ is a $p \times 1$ vector with 1 at the jth position and 0 at all other positions.

Suppose we have $X_{j_1}, X_{j_2}, ..., X_{j_{k-1}}$ as the selected variables, and the currently optimized coefficient is $\hat{\boldsymbol{\beta}}_{(k-1)}$. For the remaining $j \notin \{j_1, j_2, ..., j_{k-1}\}$, we continue our procedure in

parallel, solving the following problem.

$$\hat{\beta}_j = \arg\max_{\beta_j} \left\{ \tau_n^*(\hat{\boldsymbol{\beta}}_{(k-1)} + \beta_j \boldsymbol{e}_j) - \lambda|\beta_j| \right\}, j \notin \{j_1, j_2, ..., j_{k-1}\}, \tag{2.6}$$

where λ is a system parameter (we will discuss its selection in detail in Section 2.4.2), and β_j is the jth element in $\hat{\boldsymbol{\beta}}_{(k-1)}$. We then set j_k as

$$j_k = \underset{j \notin \{j_1, j_2, ..., j_{k-1}\}}{\arg\max} \{\tau_n^*(\hat{\boldsymbol{\beta}}_{(k-1)} + \hat{\beta}_j \boldsymbol{e}_j)\}. \tag{2.7}$$

The algorithm will terminate if the following condition is satisfied, where ϵ is a small positive threshold value.

$$\tau_n^*(\hat{\boldsymbol{\beta}}_{(k-1)} + \hat{\beta}_{j_k} \boldsymbol{e}_{j_k}) - \tau_n^*(\hat{\boldsymbol{\beta}}_{(k-1)}) < \epsilon. \tag{2.8}$$

Otherwise, we set $\hat{\boldsymbol{\beta}}_{(k)}$ as

$$\hat{\beta}_{(k)} = \frac{\hat{\boldsymbol{\beta}}_{(k-1)} + \hat{\beta}_{j_k} \boldsymbol{e}_{j_k}}{||\hat{\boldsymbol{\beta}}_{(k-1)} + \hat{\beta}_{j_k} \boldsymbol{e}_{j_k}||_2}, \tag{2.9}$$

and repeat the above steps until the stop condition (2.8) is satisfied. Then we obtain the estimated coefficient vector $\hat{\boldsymbol{\beta}}$.

Link Function Estimation

Due to the monotone assumption and the Kendall's tau coefficient, we perform isotonic regression in our algorithm [31], which is usually applied for non-decreasing data. The goal is to estimate the link function $f(\cdot)$, which still remains unknown. First, we define

$$Z_i = \mathbf{X}_i^T \hat{\boldsymbol{\beta}}. \tag{2.10}$$

We then sort $\{Z_1, Z_2, ..., Z_n\}$ in ascending order, and denote the results as $\{Z_{(1)}, Z_{(2)}, ..., Z_{(n)}\}$. We also rearrange $\{Y_1, Y_2, ..., Y_n\}$ according to $\{Z_{(1)}, Z_{(2)}, ..., Z_{(n)}\}$. We next execute the pool-adjacent-violators algorithm (PAVA, as described in [32], which is a simple linear algorithm for isotonic regression) on the sorted Y's, and mark the results as $\{Y_{(1)}, Y_{(2)}, ..., Y_{(n)}\}$.

By choosing a symmetric and smooth kernel function $\text{Ker}(t)$, we estimate the link function $f(t)$ as

$$\hat{f}(t) = \frac{\sum_{j=1}^{n} \text{Ker}\left(\frac{1}{b}(t - Z_{(j)})\right) \times Y_{(j)}}{\sum_{j=1}^{n} \text{Ker}\left(\frac{1}{b}(t - Z_{(j)})\right)}, \tag{2.11}$$

where b is chosen by applying the cross validation technique. The kernal function can be any function that applies to the data; we use the Gaussian kernel in our simulations. The kernal function can be any function that applies to the dataset. We use the Gaussian kernel in this chapter, since it is widely used and outperforms several other kernels, such as Epanechnikov, sigmoid, and quartic in our simulations.

Solar Intensity Prediction

With the estimated coefficient vector $\boldsymbol{\beta}$, the estimated link function $f(t)$, and a new observation $\mathbf{X}'$, we can predict the solar intensity by

$$\hat{Y}' = \hat{f}(\mathbf{X}'^T \hat{\boldsymbol{\beta}}). \tag{2.12}$$

We manually set a threshold T for the minimum solar intensity. For example, we can set $T = 0$ by default since solar intensity cannot have a negative value. The final prediction result $\hat{Y}^*$ is computed as

$$\hat{Y}^* = \begin{cases} \hat{Y}', & \text{if } \hat{Y}' > T \\ T, & \text{otherwise.} \end{cases} \tag{2.13}$$

2.4.2 Selection of System Parameter λ

The system parameter λ in (2.6) is one of the most important parameters in the proposed algorithm. It is sensitive to various problems and should be carefully tuned. In this section, we present two basic methods on how to choose λ.

The first method is to use the cross validation technique. For initialization, we need to shuffle the dataset and randomly split the samples into five subsets (for a five-fold cross validation) with equal size. Then we pick a set of possible λ values and $\tau_n(\boldsymbol{\beta})$ is calculated on a one-fifth data subset by using the estimated $\boldsymbol{\beta}$ from the remaining data. After repeating the process until all five parts have been calculated (i.e., to avoid the possible unbalanced results caused by the randomness in data), we could choose the λ that maximizes the average estimation precision in the cross validation process. With the proposed algorithm, parallel computation can be employed, with which the processing speed will be greatly increased. The cross validation technique is used when we have abundant time and information, and the best estimation precision is preferred.

Alternatively, we can use the regularization path method to achieve a tradeoff between precision and speed. When we demand more on speed and an acceptable precision is specified, we could choose λ with the following process.

1. First, choose a set of possible λ values and sort them by increasing order;
2. Then execute the proposed algorithm for each λ and record their performance;
3. Plot the achieved precision performance versus the values of λ;
4. Choose an acceptable point on the curve to guarantee the performance while achieving the maximized estimation speed due to sparsity.

The regularization path method also has the potential to achieve high estimation accuracy even when information is lacking.

In Figs. 2.1, 2.2 and 2.3, we show an example of how to use the solution path method with a grid search. We first search by using a larger λ value ranging from 0.001 to 1. With the plotted curve and the related Root Mean Squared Error (RMSE), we zoom in the region that

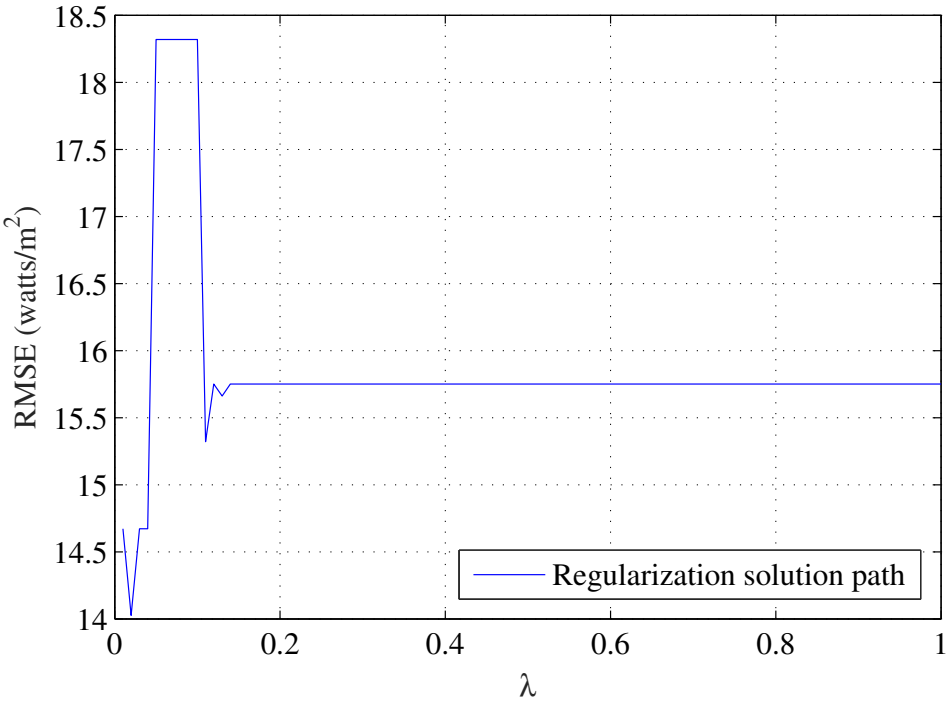

Figure 2.1: Grid search with λ from 0.01 to 1

has smaller RMSEs, to select a smaller step size 0.001 and narrower range from 0.001 to 0.1. Then repeat this procedure. For the last round we plot the curve with λ ranging from 0 to 0.03, since the result shows no merit to continue further, we stop here and choose the most accurate and stable λ value as 0.015. The corresponding solar power generation forecasting result is presented in Section 2.5.

2.4.3 Prediction Methodology and Performance Measures

It is noticed that both the observational and forecasted weather dataset are time-series datasets that changes over weather patterns and time. As the result shown in [16], solar intensity depends on multiple weather variables, which could help us to construct an accurate prediction model. The structure of the dataset and the possible relationship among the weather variables motivate our proposed LASSO-based method for developing solar intensity prediction models. To construct the model, we utilize historical weather data as input, which include several forecasting data parameters and the actual solar intensity, with totally six weather variables. The proposed algorithm establishes a function that computes solar intensity from the five forecasting weather variables. Thus we could use it as the prediction model for future solar

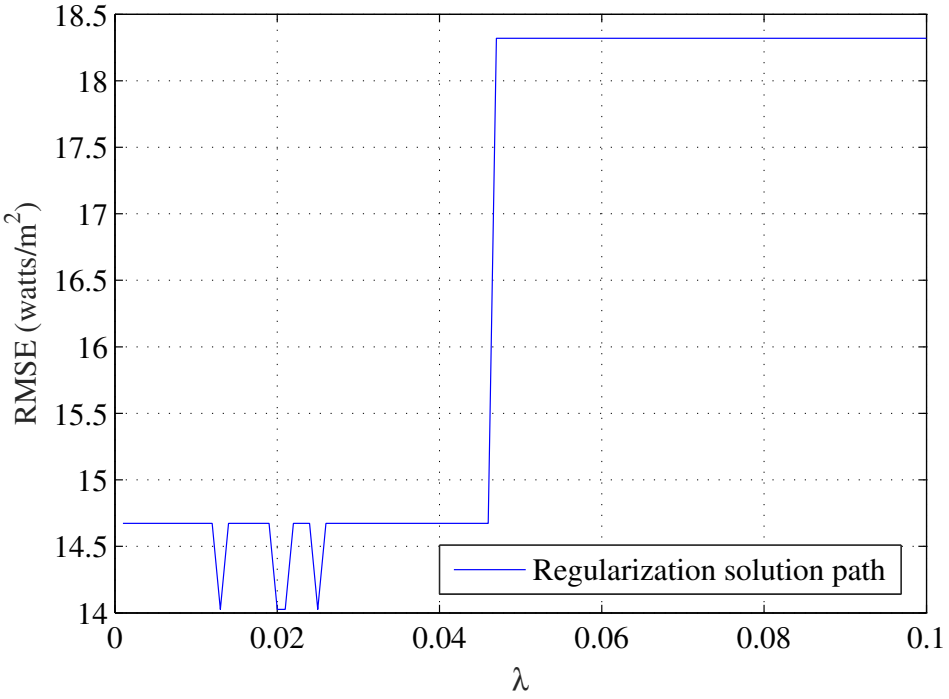

Figure 2.2: Grid search with λ from 0.001 to 0.1

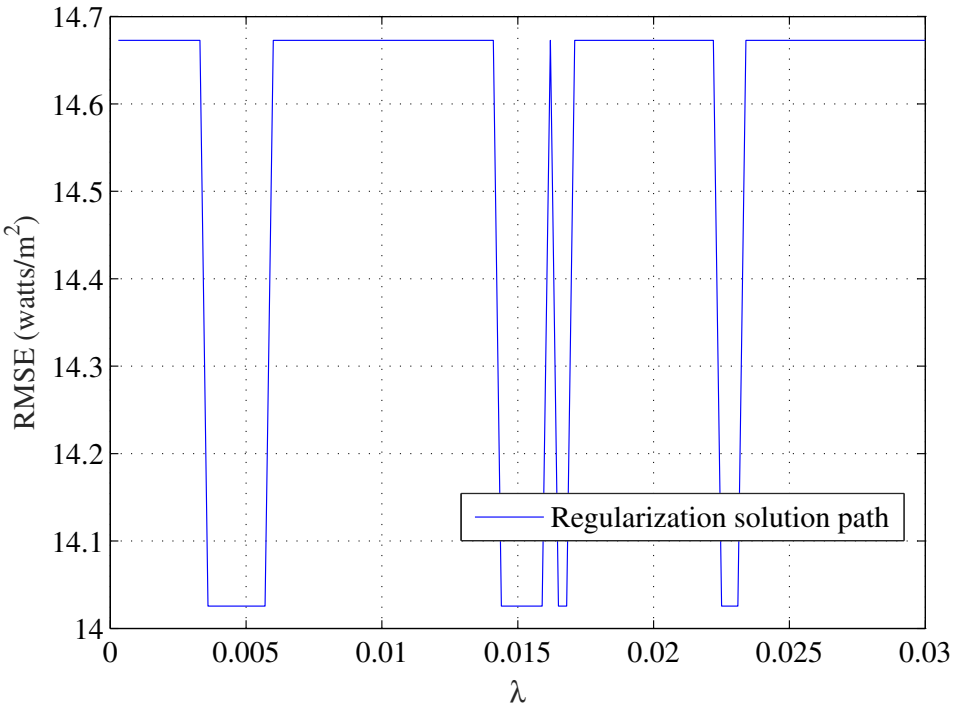

Figure 2.3: Grid search with λ from 0.01 to 0.03

power generation. We also use part of the remaining data to test the model's accuracy. One unique benefit of using our proposed technique is the relatively low requirement for data size.

In general, not too much data is needed, usually historical data over a $15 \sim 30$ day period will be sufficient.

We focus our study on short-term forecasting for the next few days. We develop a model that shows a relationship between solar intensity and forecasted weather data. For any time t, we build the LASSO model by using the historical data from the past 30 days as an input, i.e., the data from $(t-30)$ to $(t-1)$. Using the proposed model, we then predict the solar intensity for time t. In Section 2.5, we also compare the accuracy of our models with different popular and efficient models, including an SVM-based model [16] and a time-series based model [17].

Using the basic SIM presented in Section 2.3, the solar power general prediction model is

$$Y \sim P(\cdot, f(Temperature, DewPoint, WindSpeed, \quad Precipitation, Humidity)), \quad (2.14)$$

where $f(\cdot)$ is the link function that we determine using different prediction methods. The units of the parameters in the model are: $Temperature$ in degrees of Fahrenheit, $DewPoint$ in Fahrenheit, $WindSpeed$ in miles per hour, $Precipitation$ in inches, and $Humidity$ in percentage between 0% and 100%. However, to avoid potential scaling problems, before applying any selected algorithm, we normalize all feature data to have a zero mean and unit variance.

To quantify the accuracy of each model, we compute the RMSE and Mean Absolute Percentage Error (MAPE) between the predicted solar intensity and the actually observed solar intensity. RMSE and MAPE are well-known statistical measures of the accuracy of values predicted by models with respect to the observed values. RMSE and MAPE of zero indicate that the model exactly predicts solar intensity with no error (although this is impossible in reality). The closer the RMSE and MAPE values are to zero, the more accurate the model's prediction is. RMSE and MAPE are defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \quad (2.15)$$

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_t - y_t}{y_t}\right|, \quad (2.16)$$

where n represents the number of predicted data points, $\hat{y}_t$ stands for the prediction result for data point t, and y_t is the actual value of data point t.

2.5 Simulation Validation

In this section, we present our simulation validation of the proposed LASSO based scheme. We use three different datasets gathered in both US and UK to test the proposed scheme under a variety of environments. The datasets can be found in [1] and [2]. For comparison purpose, we use the SVM based method presented in [16] and the TLLE method [17] as benchmarks.

2.5.1 Dataset Description

The first dataset we use is gathered from a Davis Weather station located in Amherst, Massachusetts [1]. The weather data was collected every 5 minutes and the weather station is equipped with sensors to measure temperature, wind chill, humidity, dew-point, wind speed, wind direction, rainfall, barometric pressure, sunlight, and Ultraviolet (UV). The dataset is recorded for quite a long period from February 2006 to January 2013. However, the dataset contains errors, which are indicated by a value of -100000, as well as missing data for some periods. In the simulation study, we excluded such errors and missing data.

We plot the recorded daily solar intensity in Amherst, Massachusetts in Fig. 2.4, to clearly show how the data pattern varies over time. In accordance with our general knowledge, we can observe peaks in hot summer days and valleys in cold winter days. Also, we can see the strong correlation between consecutive days. Thus we try to use seasons and months as additional parameters and use historical data of the past 30 consecutive days as training samples.

The second dataset [2] is recorded in Harnhill and Diddington in UK. At each study location, two weather stations are installed (four in total), each record data every 30 minutes for rainfall, temperature, humidity, wind speed, wind direction, barometric pressure, and UV. The Harnhill dataset is from April 2011 to November 2012, while the Diddington dataset records weather information from August 2011 to December 2012. Missing data in both datasets are represented by NaN. By excluding all such invalid data, the amounts of valid samples are both for 397 days. The solar intensity recorded in Harnhill and Diddington, UK are plotted in

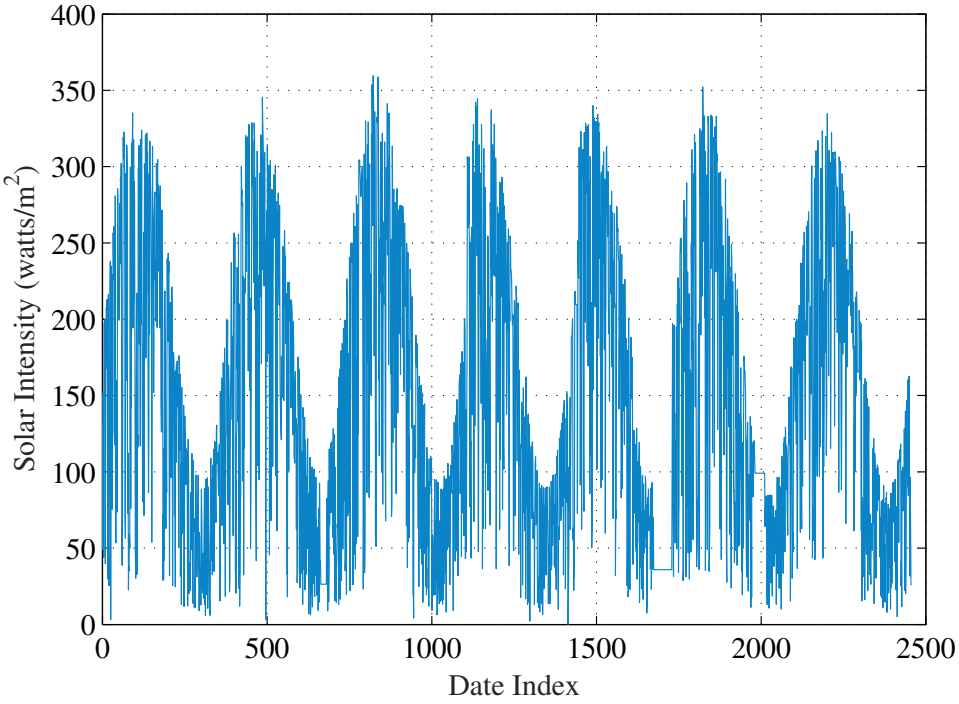

Figure 2.4: Solar intensity collected at the Davis weather station [1].

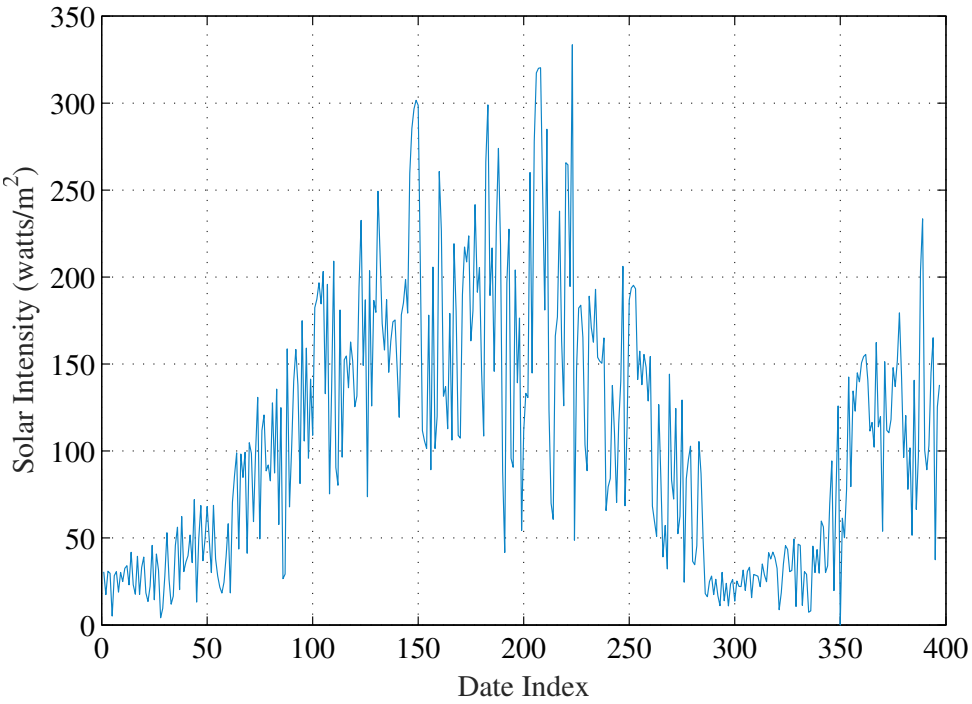

Figure 2.5: Solar intensity recorded in the Diddington dataset [2].

Figs. 2.5 and 2.6, respectively. It can be seen weather pattern varies with time just the same as mentioned before. The same strategy is used for both datasets, where 30 consecutive history days are selected as training samples.

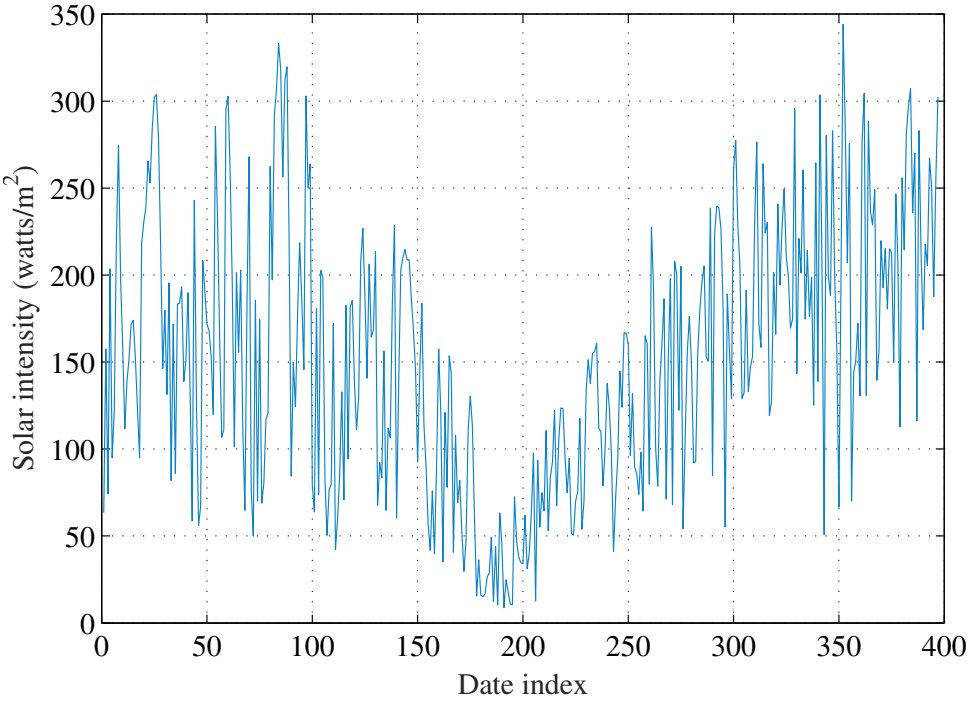

Figure 2.6: Solar intensity recorded in the Harnhill dataset [2].

2.5.2 Results with the UMass Dataset

SVM-based Method

With the UMass dataset [1], we first apply the SVM method since it is shown to be effective and widely used in prediction and classification [16]. Here we use historical weather data as training samples and aim to predict the solar intensity data through January 1st, 2013 to February 28th, 2013. In the simulations, we find that different sets of training data have considerable effects on estimation accuracy. Experimenting with all the data that is available, we achieve the optimal accuracy with the historical data from January 1st, 2012 to February 28th, 2012, which is exactly one year ahead of the target period for prediction. The predicted solar intensity is plotted along with the observed data in Fig. 2.7. The best RMSE achieved by the SVM-based method is 30.1524 watts/m^2. However the MAPE for the dataset is as high as 468.283, which is largely due to the large deviation of the 55th day data, which cloudy . If we exclude that day, the MAPE of the SVM-based method will be reduced to 39.2063.

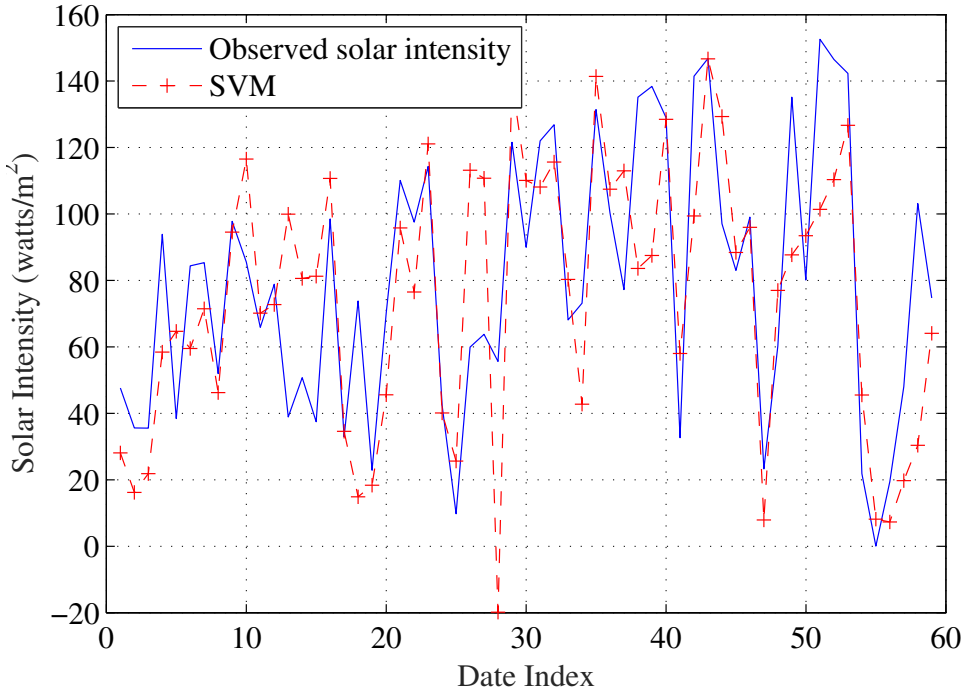

Figure 2.7: Solar power generation prediction using the SVM-based method with the UMass dataset.

TLLE Method

According to [17], TLLE has been proven to be a more accurate means when comparing to SVM and MLR based approaches. The same UMass Trace Repository data [1] is used as in [17]. The historical data from January 1st, 2012 to February 28th, 2012 is used to construct the TLLE model, and solar power generation is predicted for the period from January 1st, 2013 to February 28th, 2013.

We plot the predicted solar intensity along with the observed data in Fig. 2.8. The best RMSE we obtained with the TLLE-based method is 23.1464 watts/m^2, which is about the same as that reported in [17]. TLLE achieves a 23.2% reduction over the SVM-based method. This result validates the advantage of TLLE comparing to the SVM-based method. We also find a very high MAPE value in this simulation, also due to the anomaly data of the 55th day. The MAPE for the remaining data, after excluding the 55th day data, is reduced to 29.0174, which is a 26.0% reduction over the SVM-based method.

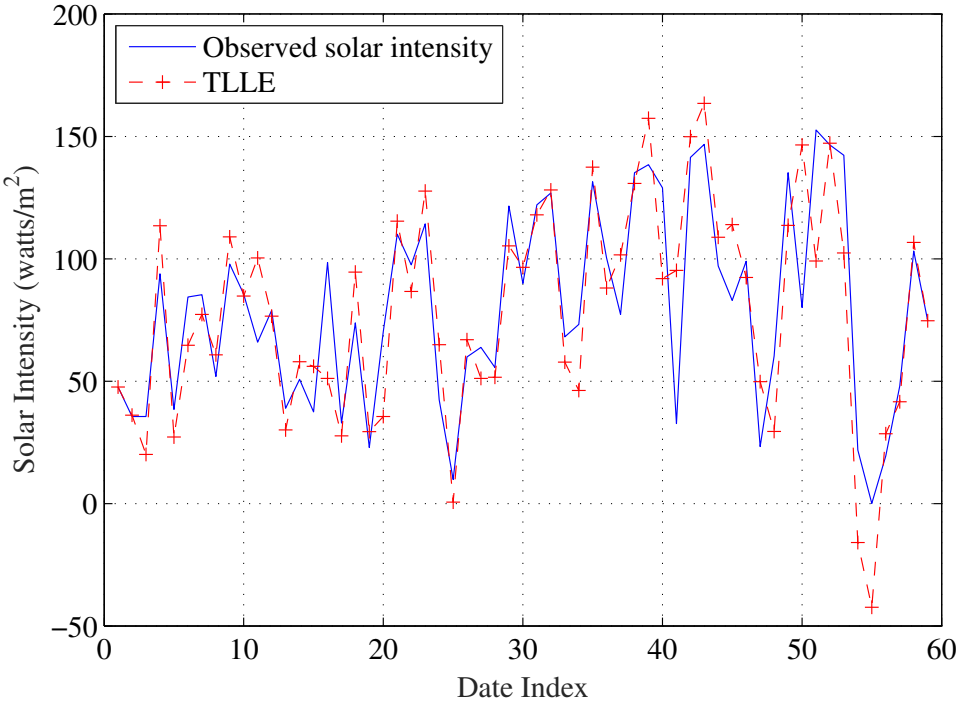

Figure 2.8: Solar power generation prediction using the TLLE-based method with the UMass dataset.

Proposed LASSO-based Algorithm

Now we apply the proposed LASSO-based method to predict solar intensity. In the simulation, we use a relatively smaller training sample size of 30, i.e., the training data here is gathered from the past 30 days of the target date. Applying the proposed LASSO-based method to the training data yields the prediction results that is plotted in Fig. 2.9.

For the LASSO-based prediction curve in Fig. 2.9, the RMSE is 14.0262 watts/m^2 and the MAPE is 17.817, representing further 39.4% and 60.1% reductions over the TLLE-based approach, respectively. More important, *these results are achieved with the entire* 30*-day original data, i.e., without excluding the* 55*th day anomaly data in the dataset*. Our method also achieves a very stable performance in MAPE. Furthermore, even if we reduce the number of training data for 30 days to 15 days, the proposed LASSO-based algorithm still achieves a reliable result, with an RMSE lower than 20 watts/m^2.

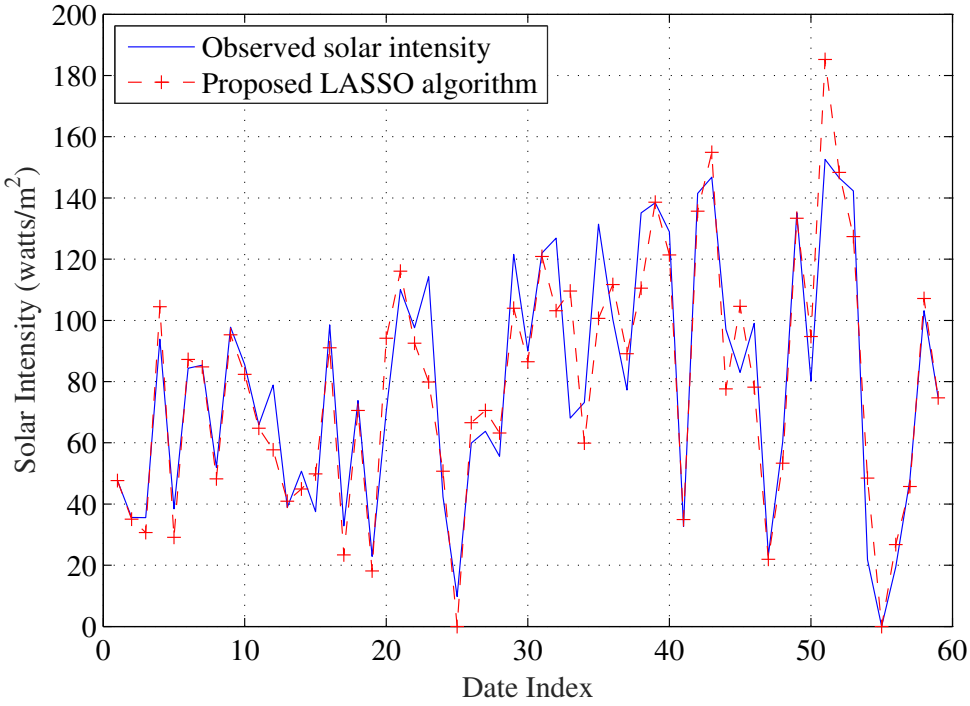

Figure 2.9: Solar power generation prediction using the proposed LASSO-based method with the UMass dataset.

2.5.3 Results with the UK Datasets

We also apply the three algorithms to the Diddington and Harnhill datasets described in Section 2.5.1 [2] for a more comprehensive evaluation. Unlike the weather in the US, the areas in Britain inevitably have less sunlight due to the much more rainy and cloudy days. This different data feature can be a practical test to our proposed method. For both datasets from [2], we predict the solar intensity for the period from the 365th to 394th day.

With the Diddington dataset, we find the SVM method have great difficulty with the small set of training samples, which forces us to increase the amount of training samples to 100. Here we use the first 100 days' weather data to train the SVM model and finally obtain an acceptable result as presented in Fig. 2.10. Meanwhile, the TLLE method still works better than SVM. We used the first 60 days' data as the model generating data to achieve the best performance, which is illustrated in Fig. 2.11. The prediction results with the proposed LASSO based approach is presented in Fig. 2.12, which is obtained with a much smaller 30 training data size than SVM and TLLE. The overall comparison of accuracy is presented in Table 2.1. For this dataset, the proposed LASSO based approach achieves reductions of 76.6763% and 66.1474% in RMSE

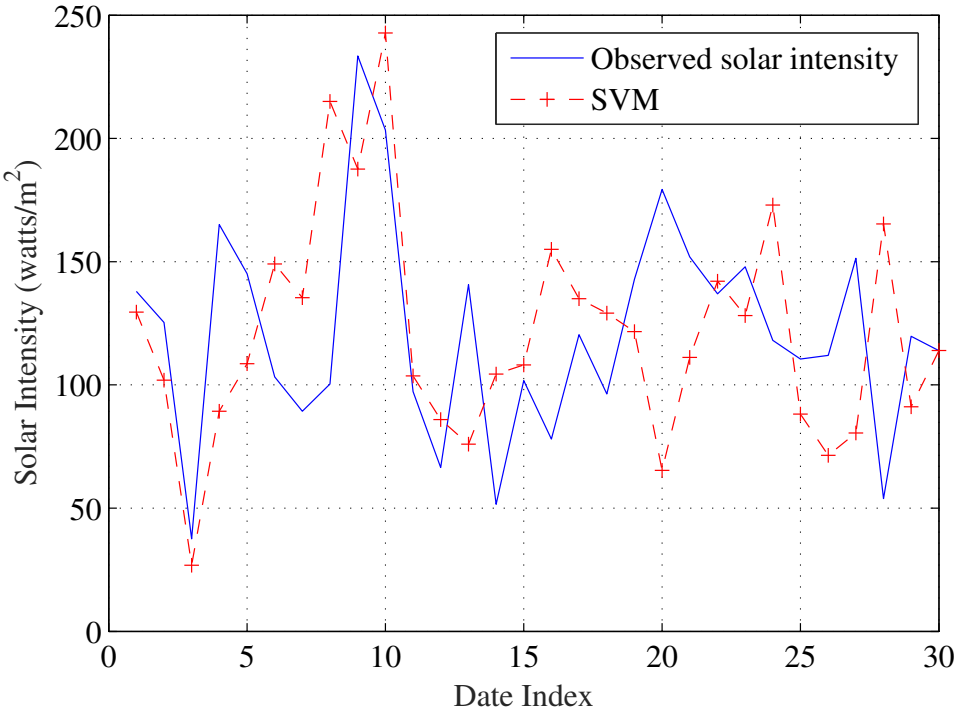

Figure 2.10: Solar power prediction using SVM with the Diddington dataset.

Table 2.1: Prediction Accuracy with the Diddington Dataset

	SVM	TLLE	LASSO
RMSE (watts/m^2)	52.4241	36.1189	12.2272
RMSE Reduction with LASSO	76.6763%	66.1474%	-
MAPE	38.3795	27.7550	5.5240
MAPE Reduction with LASSO	85.6069%	80.0973%	-

over SVM and TLLE, respectively, and reductions of 85.6069% and 80.0973% in MAPE over SVM and TLLE, respectively. Note that such considerable gains are achieved with a much smaller training data size.

The simulation results with the Harnhill dataset are presented in Figs. 2.13, 2.14, and 2.15 for the three schemes. The prediction accuracy results are summarized in Table 2.2. Due to the different weather pattern in UK, the level of solar intensity is considerably smaller than that in US. Although we could witness a much closer RMSE achieved with all the three methods, we should still notice their obvious difference in MAPE. For the dataset, the proposed LASSO based approach achieves reductions of 73.4156% and 66.5391% in RMSE over SVM and TLLE, respectively, and reductions of 82.4621% and 81.1672% in MAPE over SVM and

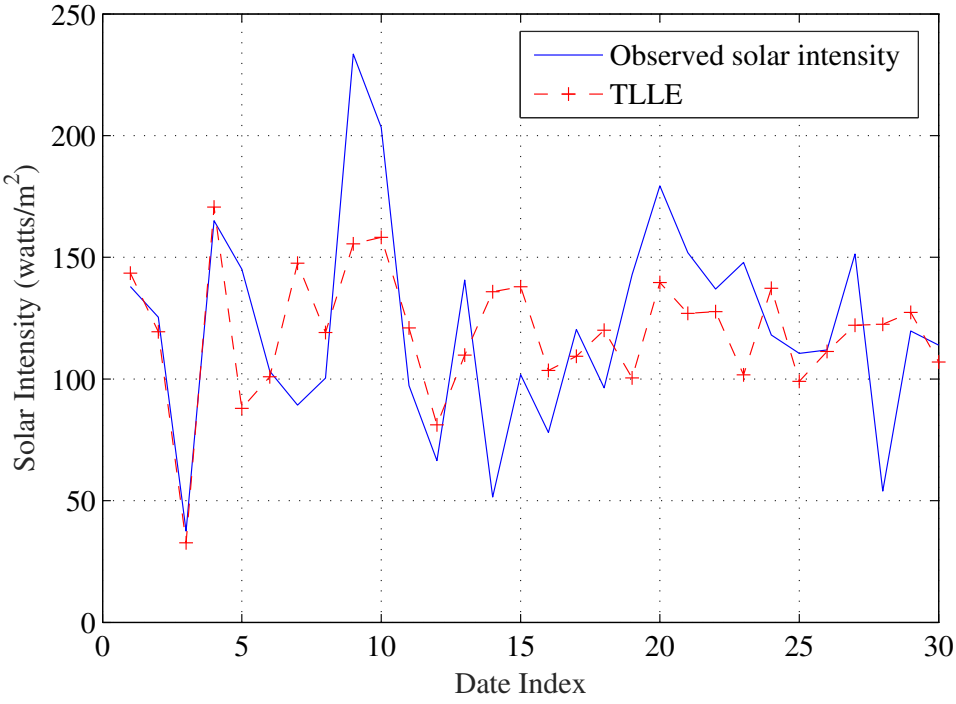

Figure 2.11: Solar power prediction using TLLE with the Diddington dataset.

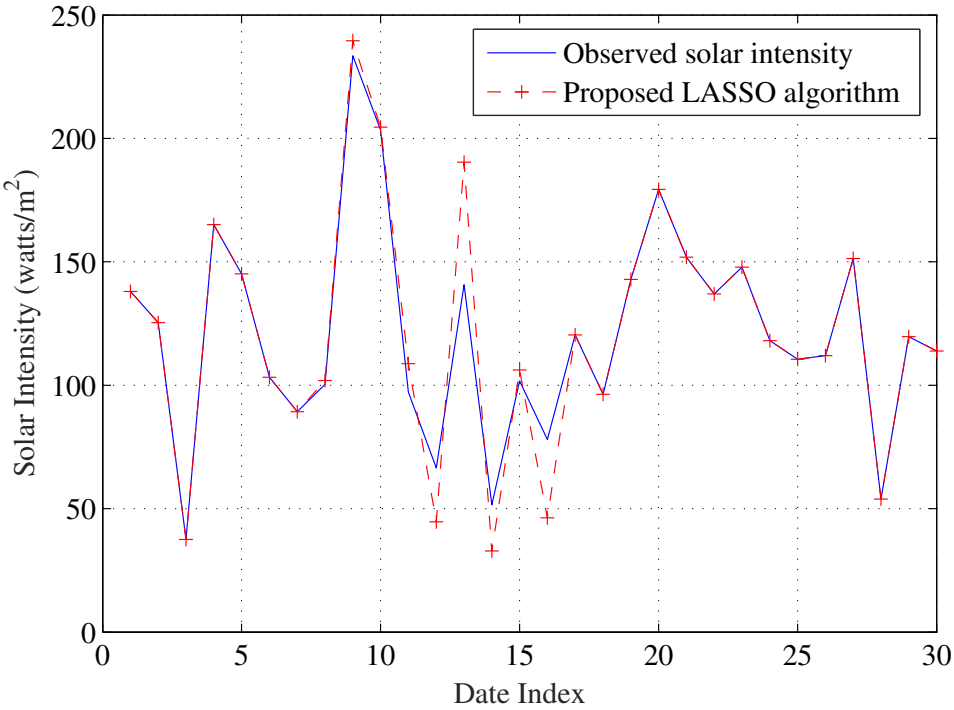

Figure 2.12: Solar power prediction using the proposed method with the Diddington dataset.

TLLE, respectively. Note that these performance gains are consistent with that observed with the Diddington dataset in Table 2.1.

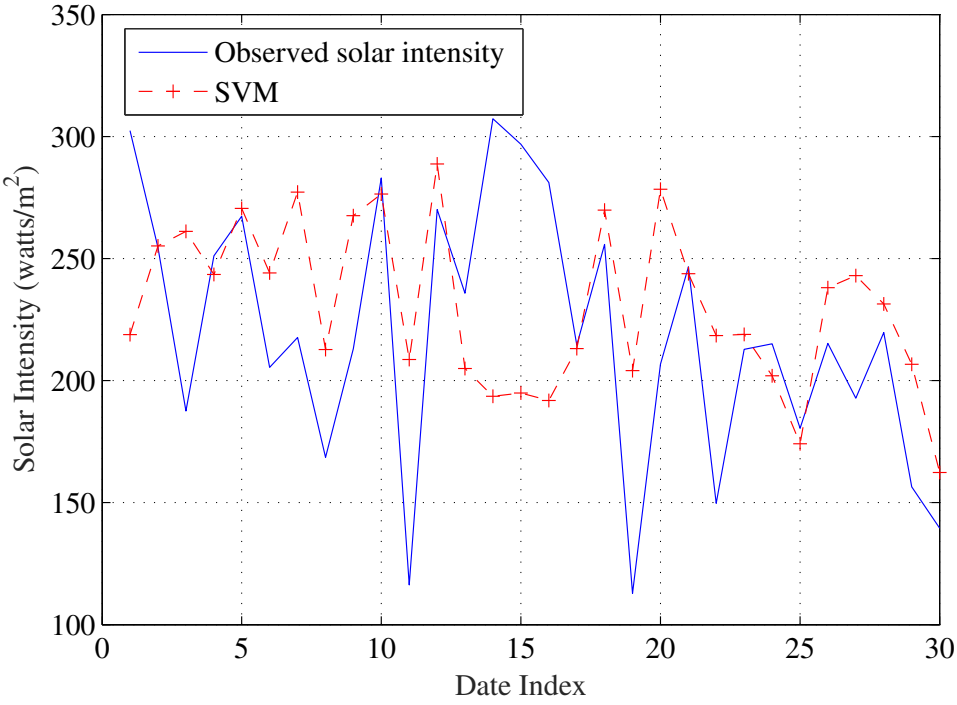

Figure 2.13: Solar power prediction with SVM of Harnhill dataset.

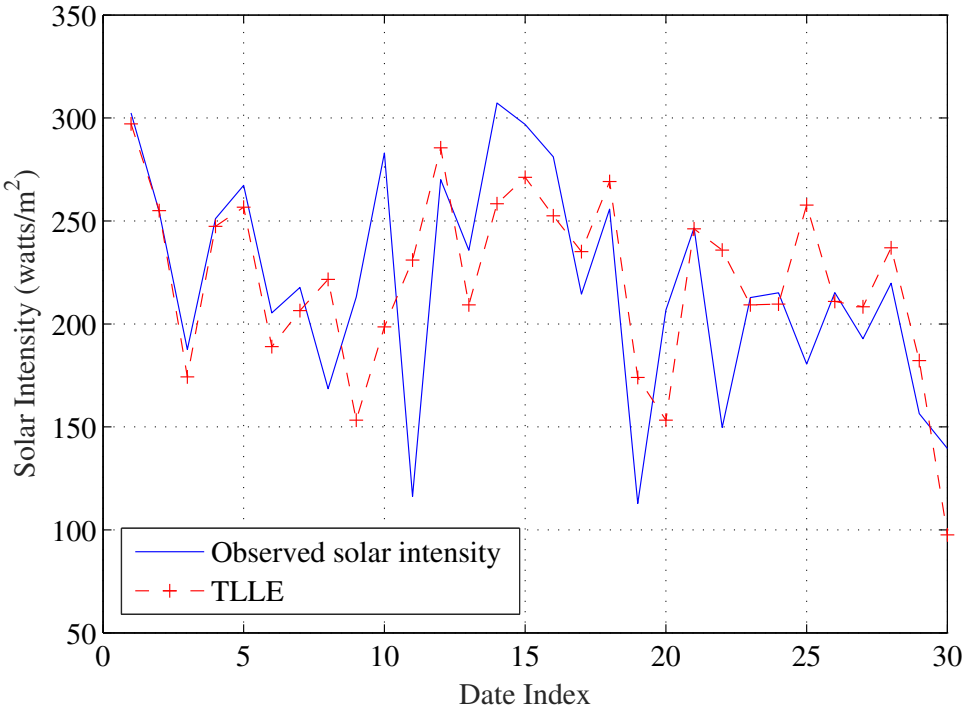

Figure 2.14: Solar power prediction with TLLE of Harnhill dataset.

Clearly, our LASSO-based algorithm has achieved considerably higher accuracy compared to the two existing methods. In addition, it requires fewer training data and is robust

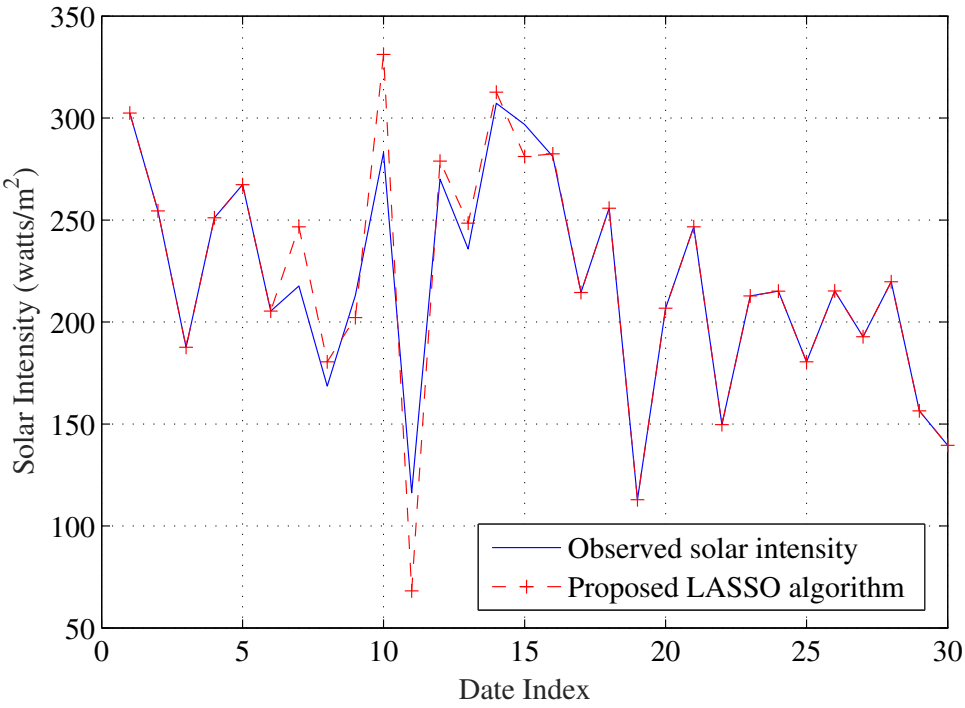

Figure 2.15: Solar power prediction with proposed method of Harnhill dataset.

Table 2.2: Prediction Accuracy with the Harnhill Dataset

	SVM	TLLE	LASSO
RMSE(watts/m^2)	54.2934	43.1357	14.4336
RMSE Reduction with LASSO	73.4156%	66.5391%	-
MAPE	19.0199	17.7122	3.3357
MAPE Reduction with LASSO	82.4621%	81.1672%	-

to anomaly data points in the training data, which make it a highly competitive solution to practical problems such as forecasting of solar power generation.

2.5.4 Variable Selection with the Proposed Scheme

A notable advantage of the LASSO-based algorithm is its ability of variable selection. By tuning the loss function with parameter λ, it allows to identify which variable(s) are more "important" to the prediction result. The process is to tune λ in condition of an acceptable prediction accuracy, until any β_j has reached 0. Then we can treat the corresponding variable X_j as least important. Repeating this procedure, we can identify the second least important variable, and so forth.

Table 2.3: Correlation Matrix of the UMass Dataset

	Temperature	*Humidity*	*DewPoint*	*WindSpeed*	*Precipitation*	*Solar*
Temperature	1.0000	0.2193	0.9604	−0.2819	0.0991	0.5943
Humidity	0.2193	1.0000	0.4733	−0.3918	0.4234	−0.4472
DewPoint	0.9604	0.4733	1.0000	−0.3512	0.2034	0.4027
WindSpeed	−0.2819	−0.3918	−0.3512	1.0000	0.0089	−0.0884
Precipitation	0.0991	0.4234	0.2034	0.0089	1.0000	−0.2771
Solar	0.5943	−0.4472	0.4027	−0.0884	0.2771	1.0000

Variable selection will at least provide us with two fascinating advantages: (i) reducing the computational complexity and (ii) simplifying the prediction model. Due to the structure of our proposed algorithm, historical data will also be used in the prediction stage. So when less parameters are used in the prediction model, the computational cost will be greatly reduced. In addition, a simplified model can provide us a clearer understanding of the relationship between solar power generation and the weather parameters. We can use the reduced model to estimate solar power generation when the dataset is incomplete, or to reduce the computation time when necessary (i.e., to tradeoff between complexity and accuracy).

As an example, we use the UMass dataset to illustrate the variable selection procedure. Table 2.3 shows the correlation matrix computed with the dataset. Although the problem cannot be simply defined as a linear one, we could still use the correlation matrix to obtain an intuitive observation. As the matrix shows, *Temperature* and *Humidity* are more closely correlated to Solar intensity, while *Precipitation* is quite independent with most other parameters. After adjusting the λ value to reduce the model to a 3-variable model, we have the optimized $\boldsymbol{\beta}$ values listed in Table 2.4.

From Table 2.4, we find that both *DewPoint* and *WindSpeed* are seen as less important variables. The result for *WindSpeed* coincides with the correlation matrix but *DewPoint* and *Precipitation* have a conflict. However, noticing from Table 2.3 that *DewPoint* is tightly correlated with *Temperature*, while *Precipitation* is quite independent to *Temperature*, the result for $\boldsymbol{\beta}$ becomes reasonable. Fig. 2.16 provides the prediction result with the 3-parameter model. The RMSE in this case is 21.2468 watts/m^2, which is still better than both SVM and TLLE, but the MAPE has increased to 68.5998 due to the inaccuracy on some small values. Such

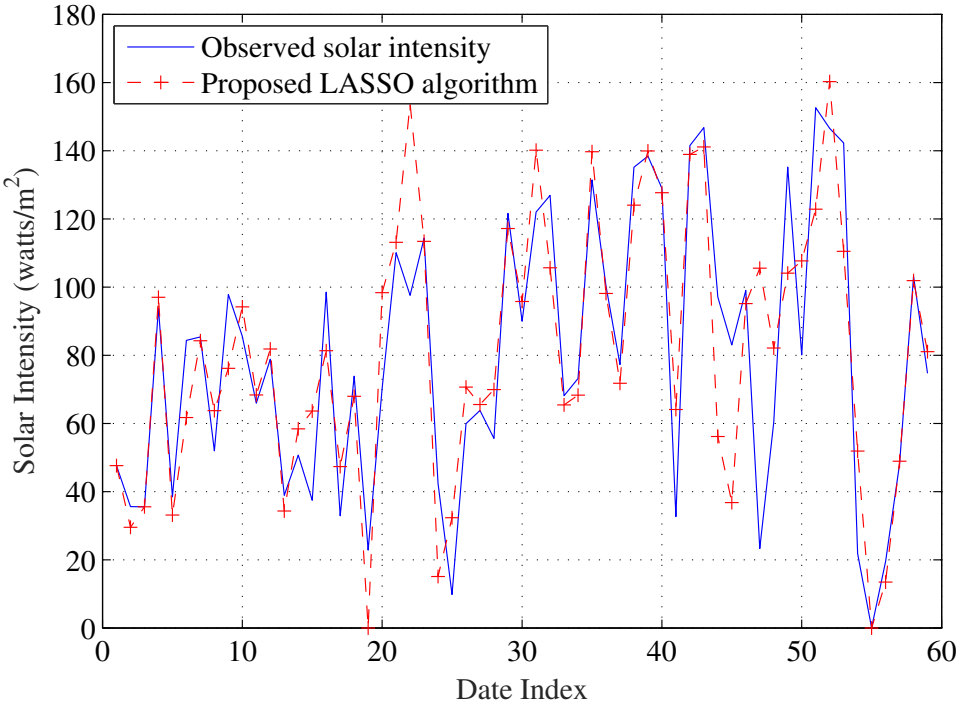

Figure 2.16: Solar power prediction using the three selected variables with the UMass dataset.

Table 2.4: Optimized $\boldsymbol{\beta}$ with Three Variables

	Temp.	*DewPoint*	*WindSpeed*	*Precipitation*	*Humidity*
β_j	0.4676	0	0	-0.7938	-0.3878

phenomenon can be explained by the inner characteristics of variance. When we use fewer variables, we actually lose a certain amount of variance and the prediction will become more unbiased to maintain accuracy. Thus, it is expected to have a lower accuracy and the absolute percentage error on certain small values could become high. The variable selection capability provides a useful trade-off between computational/model complexity and accuracy.

2.6 Conclusion

In this chapter, we proposed a LASSO-based algorithm that accurately predict solar power generation with a small amount of historical data. After presenting the detailed algorithm design, we compared the proposed scheme with two representative existing schemes using three datasets with different features. We found that the LASSO-based algorithm achieved considerably higher accuracy comparing to the existing methods, using fewer training data,

and being robust to anomaly data points in the training data. In addition, the variable selection capability offered a nice trade-off between complexity and accuracy. These features all made it a highly competitive solution to forecasting of solar power generation. For future work, it would be interesting to explore other advanced statistic tools, such as adaptive LASSO or group LASSO, to address the forecasting problem.

Chapter 3

Adversarial Attacks to Solar Power Generation Forecasting

3.1 Introduction

The Internet of Things (IoT) is defined as a network of items (or devices) that are integrated with communication technologies for purpose of exchanging data with each other over the Internet. Power, dataflow, sensors, electric vehicles, and communication technologies in the 5G wireless system [33–35] can all be part of the IoT. To achieve the goal of connecting "everything" intelligently, the IoT incorporates many state-of-the-art technologies and rejuvenates many traditional industries such as the power grid and its next generation smart grid (SG). For a traditional power grid, the power and data flows are almost one-way, while the trademark of the SG is its two-way transmission of power and information [8]. To fully harvest the potential of the SG, many advanced IoT technologies have been applied to build an environment-friendly, economical, efficient, and resilient SG [9, 21, 36–39].

As the main issue of SG, energy management aims to offer both stability and efficiency. Energy management techniques are focused on how to operate the smart grid under various practical constraints at different timescales. Day-ahead strategy, as one of the widely used strategy for power grid management, requires utility operator to predict day-ahead power demand and generation in the day-average form. On the other hand, at the hour or minute timescale, the prediction for demand and generation is updated every hour or per several minutes. The conventional prediction methodology usually requires market information and state knowledge to avoid breaking the power balance. Because both the power consumption and generation

change rapidly over time, after taking the fluctuation into consideration, short-term prediction is an important means to improve power network stability [15, 38].

For various kinds of renewable resources, wind and solar generation forecast are always regarded as key problems. Due to their natural characteristics of randomness, precise short-term forecasts at a 15-minutes timescale and 2-day ahead is more widely used and more practical in practice to achieve efficient energy usage.

To predict solar power generation, people estimate the solar irradiance falling on photovoltaic panels. Since solar intensity on these panels can be directly mapped to power generation, the problem is transformed into a solar intensity forecasting problem. Thus, the goal finally becomes to forecast the short-term solar intensity with 2-day ahead scheme. In [40, 41], statistical and machine learning models such as autoregressive integrated moving-average (ARIMA) and artificial neural networks (NN) were used to build temperature based time series models. While these models performed quite well in sunny days, the precision could drop sharply in cloudy, rainy, or other extreme weather conditions. Thus, various weather data based forecasting models have been developed in the literature [16, 17, 42, 43].

In the meantime, many machine learning techniques have also been proposed, which can map the relationship between solar intensity and various weather variables [21, 38]. In the prior studies, machine learning models like support vector machine (SVM) [16], ANN [41], and long short-term memory (LSTM) [21, 38] have been used and achieved quite remarkable accuracy.

Due to the availability of data, computing power, and open-source platforms (e.g., Tensorflow), deep learning has become the focus of machine learning in recent years. DNN, as the cornerstone of deep learning, is not only used in image recognition but also applied to solar power generation forecasting. Though deep learning has its unique advantages in solving various problems, the inherited black box feature of deep neural networks could potentially lead to security problems, as pointed out in [44]. When adding adversarial examples to input data, the classification results for images could be totally different. Here, the only difference between adversarial examples and the original samples is only a small perturbation, which is almost impossible for human eyes to notice. After this, many works have been done to evaluate the severity of adversarial attacks on various problems. In [45], 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 [46], the authors noticed that there is a universal perturbation that could affect the classification for every image. To a more practical real-world scenario, the authors in [47] tested adversarial attack even on real 3D printed items and found the attack remain effective.

Although we see many works have been done in the computer vision area, it is still unknown if the same type of attack could be reproduced on solar power generation prediction. It is also worth mentioning that very few work has been done on statistical models such as LASSO. In [48], the author examined adversarial examples on a regression problem. However, the Boston Housing dataset is usually used for linear regression, so the effect on the non-linear regression problem has not been tested yet.

Motivated by this, we aim to investigate the problem of adversarial attack to the solar power generation forecasting problem. The main contributions of this work are as follows:

- We examine how adversarial attack affects both the DNN model and our formerly proposed LASSO-based algorithm. By simulation, we study the effectiveness of adversarial attack on such different models.

- We evaluate both white-box attack and black-box attack methods. Our results demonstrate the severity of adversarial attacks to current solar power generation 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.

The remainder of this chapter is organized as follows. We introduce the adversarial attack methodology and background information in Section 3.2. Then we formulate our problem and describe our evaluation methodology in Section 3.3. We validate the performance of both white box and black box attacks and analyze the threat to the current solar power generation systems in Section 3.4. Finally, we conclude this chapter in Section 3.5.

3.2 Adversarial Attack Methodology

Attacks that could fool machine learning techniques have been studied since 2006 [49]. Adversarial attack has became a hot topic in the past few years due to its impact on the deep learning technology. Although most prior work has been focused on image processing, DNN has been proven to be vulnerable to adversarial attacks [50, 51].

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 normal DNN with parameter set θ will attempt to predict y_l as $f_\theta(x)$. However, adversarial examples will tamper x with a small perturbation to achieve maximized change in the loss function, thereby to obviously change the output result of machine learning model.

As for practical uses, adversarial attacks can be classified by their target and design. According to the goal of the attack, the adversarial attacks can be classified as targeted and non-targeted attacks [52]. Targeted attacks use adversarial examples to delude the machine learning model in order to get results toward specific targets. Meanwhile a non-targeted attack only tries to make the result incorrect. According to their design, adversarial attacks are usually classified into single-step and iterative methods. In single-step attacks, such as the Fast Gradient Signed Method (FGSM), the loss gradient is calculated once to generate the perturbation for each example. Unlike single-step attacks, an iterative attack, e.g., Projected Gradient Descent (PGD), calculates the current perturbation iteratively to achieve a maximized loss function.

3.2.1 Fast Gradient Signed Method (FGSM)

Unlike non-linear models, the authors in [53] pointed out that linear model of high-dimension is more accurate and capable of generating adversarial examples. In the paper, the author proposed the FGSM as an algorithm to quickly produce adversarial examples. The FGSM algorithm is described in the following [53].

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. Here, the perturbation ξ should keep its infinite norm

smaller than δ, which is the magnitude constraint of the perturbation, as

$$||\xi||_\infty < \delta. \tag{3.1}$$

The adversarial examples are generated as:

$$x_a = x + \xi. \tag{3.2}$$

To calculate perturbation ξ, we have

$$\xi = \delta \cdot \text{Sign}(\nabla_x J_\theta(x, y_l)), \tag{3.3}$$

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

After obtaining the gradient calculated during the back propagation stage, the perturbation is set as above. As we have acquired perturbation ξ, now we focus on the weight augmented perturbation, given by

$$w^T x_a = w^T x + w^T \xi. \tag{3.4}$$

If the weight w has dimension p and mean m, since the activation is now $w^T\xi$ larger, we can see that the activation will be increased by $\delta \cdot p \cdot m$. Thus in high dimensional problems, the small perturbations to each dimension could add up to make a large change in the final output, and here the high dimensional linear hypothesis given by the authors is proved.

3.2.2 Projected Gradient Descent (PGD)

In this section, we introduce another adversarial attack method: Projected Gradient Descent (PGD) presented in [54], which is a more powerful multi-step variant of FGSM. While PGD can generate adversarial examples to launch an attack, it also provides a possible method

to defend against first order adversarial attacks. When used as a defense method, it trains adversarial examples and uses them in the training process to increase the robustness of the trained DNN model.

The original idea of PGD is to solve the following optimization problem, which is known as a saddle point problem:

$$\min_{\theta} R(\theta), \tag{3.5}$$

where

$$R(\theta) = \mathbb{E}_{(x,y_l)\sim D}\left[\max_{s\in S} \mathcal{L}(\theta, x+\xi, y_l)\right]. \tag{3.6}$$

In the above equations, $R(\theta)$ is the population risk, which is also the objective function to be minimized; D is the distribution of samples, which defines the distribution of x and y_l; S is a nonempty compact topological space, while the inner optimization problem aims to maximize the loss function $\mathcal{L}(\cdot,\cdot,\cdot)$ over it.

In the external part of the optimization problem (3.5), PGD aims to find the model parameters to minimize the loss of adversarial attack, thus the most robust DNN network against adversarial attack can be created. Like the idea of FGSM, the internal optimization aims to maximize the loss function $\mathcal{L}$. As we could see, 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 example and improving robustness of the DNN model against adversarial attacks.

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

1. First, initialize x^0 as

$$x^0 = x. \tag{3.7}$$

2. Then iteratively calculate x^{k+1} as

$$x^{k+1} = \text{Clip}_{\{x,\delta\}}\left(x^k + \delta_{iter} \cdot \text{Sign}(\nabla_x J_\theta(x, y_l))\right). \tag{3.8}$$

δ_{iter} is the step size of perturbation level set by attacker. In the tth iteration, $\text{Clip}_{\{x,\delta\}}(x^t)$ function tries to clip x^t to be within $[x^t - \delta, x^t + \delta]$ where δ is the overall perturbation limit.

3. After all the iterations, we obtain the final adversarial example x_a as

$$x_a = x^K. \tag{3.9}$$

3.3 Problem Formulation and Evaluation

3.3.1 Photovoltaic Generation Forecast

We use both historical weather data and forecasted data to train the DNN network. Each sample in the dataset includes values for several weather variables, a time stamp, and the solar intensity. To fully utilize the big data processing capability of DNN, we use the weather dataset for an entire year as the training set. In real world scenarios, photovoltaic grid usually requires the forecast to be at least 2 days ahead in order to schedule the required future operations. Also, forecasts at 15-minute intervals are a more practical scenario. Therefore, we use part of the forecast weather data as the test set to predict the corresponding solar intensity with a 15-minute interval.

3.3.2 Adversarial Attack Schemes

As mentioned in our prior work [42, 43], both observational and forecasted weather data are time series datasets that change with season and time. By using certain basic weather variables as in [16], we can create a DNN model to accurately forecast solar intensity. Due to the direct relationship between solar intensity, the photovoltaic generation forecast is then turned into a solar intensity forecast problem.

Although adversarial attack has been studied and tested for DNN-based classification problems, there are few prior work related to the regression problem. Whether adversarial attack will affect DNN-based regression models and how effective it will be remain unknown. Motivated by this idea, we would like to study if the adversarial attack is effective for solar intensity forecasting and whether we can use the generated adversarial examples to launch a black box attack to our former LASSO based algorithm [42, 43].

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

White Box Attack

In our white box attack experiments, we assume the target DNN model is exactly the same as the trained model [55], which indicates 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 has the exact meaning of calculating gradients from the target model. In our simulations, FGSM and PGD are both used as white box attack schemes on the target DNN model.

Black Box Attack

Although beging more practical, black box attack assumes the attack is not able to access the target model but can only have the information about the input dataset and the label dataset.

On neural network models, several methodologies have been implemented to address black box attacks. For example, the zero-order optimization based attack, proposed in [56],

described how to launch black box attack without knowing gradients. During the attack, the attacker uses the Hessian estimation to approximate the correspondent target model parameters. Similar to this idea, other algorithms such as [57] were also developed. From another aspect, researchers also tried to migrate adversarial examples by exploiting their transferability. The authors in [58] showed that adversarial examples were also capable of fooling other neural networks with different architectures.

In our problem where the DNN is used as a regression model, the impact of black box adversarial attack is yet to be studied. To evaluate this scenario, we use adversarial examples generated from the DNN model, and feed them to our LASSO based model [42, 43]. Since the partially linear characteristics has been proved on certain data [16], it is interesting to see how well adversarial attacks work as a black box method on statistical models.

Evaluation

To evaluate the loss caused by the adversarial attacks, we use the root mean squared error between the forecasted solar intensity and the observed solar intensity (i.e., the ground truth). The calculation of RMSE is described as follows.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}, \tag{3.10}$$

where $\hat{y}_t$ represents the forecasted solar intensity, y_t is the ground truth at time t, and n is the number of forecasted solar intensity values.

In our prior research [42, 43], the mean average percentage error (MAPE) was also used as a performance metric. However, we were forecasting day-average solar intensity in [42, 43], which is different compared to the current 15-minute continuous forecasting. For the current issue, there are too many "0" values in the dataset from midnight to the next morning, and MAPE will not be feasible even if there are only very small errors on these points. Thus, RMSE is a better measurement in our evaluation in this work.

The forecasted solar intensity sets are divided into 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. Meanwhile, we also use the adversarial training capability provided by PGD to see if adversarial training could alleviate the loss caused by adversarial examples.

3.4 Simulation Validation

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

3.4.1 Data Description

The Chinese 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 from year to minute. Meanwhile, there is also a forecast dataset for the same period and in the same format, which is provided by the local weather station. The datasets are well maintained, and no corrupted data sample is discovered. The real solar intensity of the current dataset is plotted in Fig. 3.1 for 10000 time intervals starting from January 1st, 2020. We can see from the figure that there is a clear trend, which can be seen as an indication of seasonal change. Not all of the data are plotted in the figure, because if so, the samples in the figure would be hard to see.

The other dataset, which was also used in our former research, was collected by David weather station located in Amherst, Massachusetts [1]. The weather data was recorded with a 5 minutes interval and 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. Meanwhile, the dataset contains errors and missing samples, which are indicated by a negative value -100000. In our simulations, we exclude both the errors and missing samples.

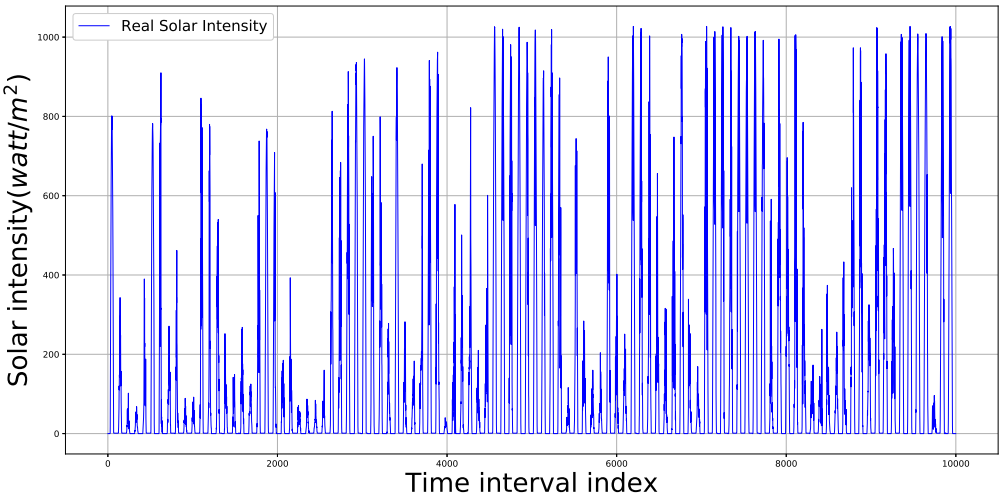

Figure 3.1: Real solar intensity in the Zhejiang dataset.

The rest usable variables are temperature, humidity, dew-point, wind speed, rainfall, and sunlight (solar intensity). 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 tested with our LASSO-based method, which is good at day-average forecast, we calculate the average solar intensity by day, and plot it in Fig. 2.4. We can easily see the time continuous feature and seasonal difference of solar intensity in the figure.

3.4.2 Data normalization

For image classification, the perturbation limit is never a problem since every pixel value is within the same range. However, the fact that perturbation limit is a global parameter certainly has some shortcomings when the limit is being implemented with certain kind of regression dataset. In Figs. 3.3−3.11, we present the generated adversarial examples over each dimension with a perturbation limit of $\delta = 0.3$. It can be seen from these figures the perturbations introduced by PGD are visually small. For example, focusing on temperature, pressure, humidity, and minute in Fig. 3.3, Fig. 3.4, Fig. 3.5, and Fig. 3.11, we can see that these data have a range around 30. Thus a 0.3 perturbation has relatively limited influences on these samples. The wind speed and month data have ranges only around 10. Especially for wind direction that

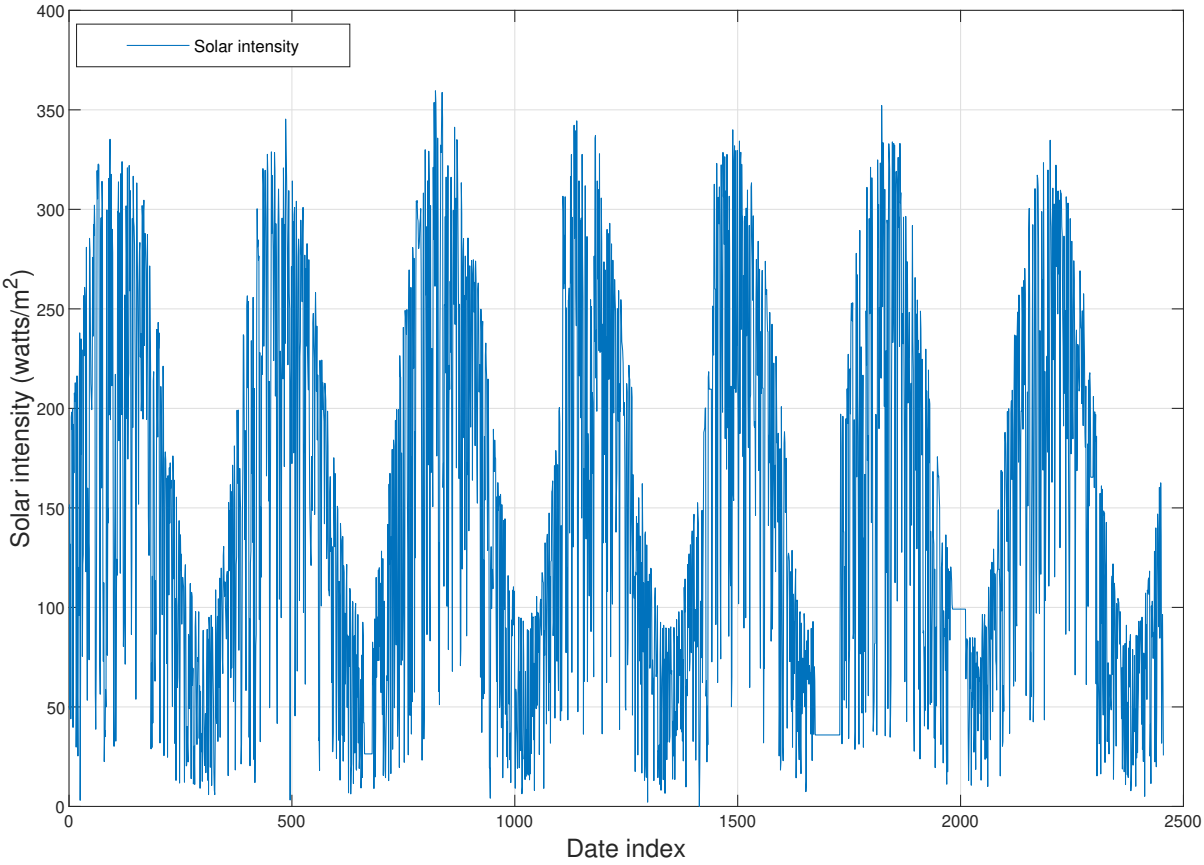

Figure 3.2: Real solar intensity from Davis weather station [1].

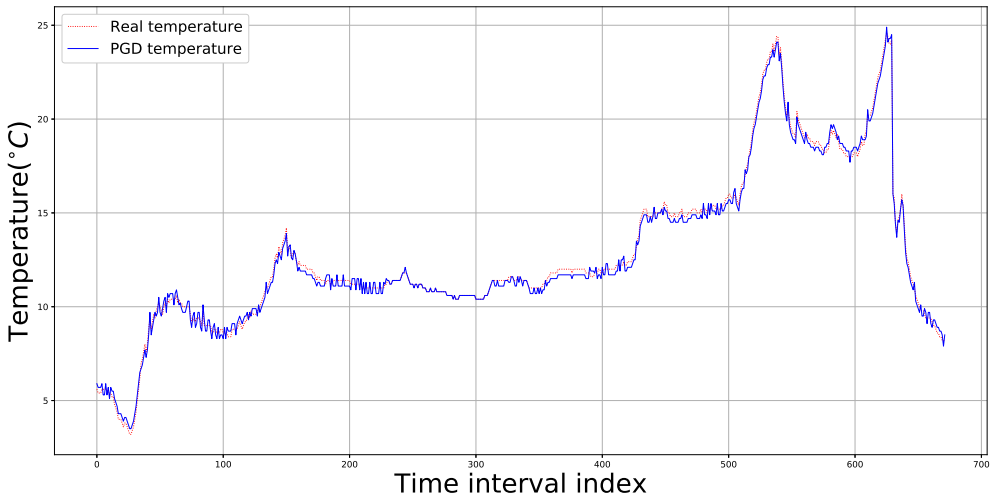

Figure 3.3: Adversarial examples on temperature.

varies rapidly from 0 to 359 in Fig. 3.7, the perturbation is almost meaningless. If the variables in the dataset have very different ranges, we should consider about the negative effect it could

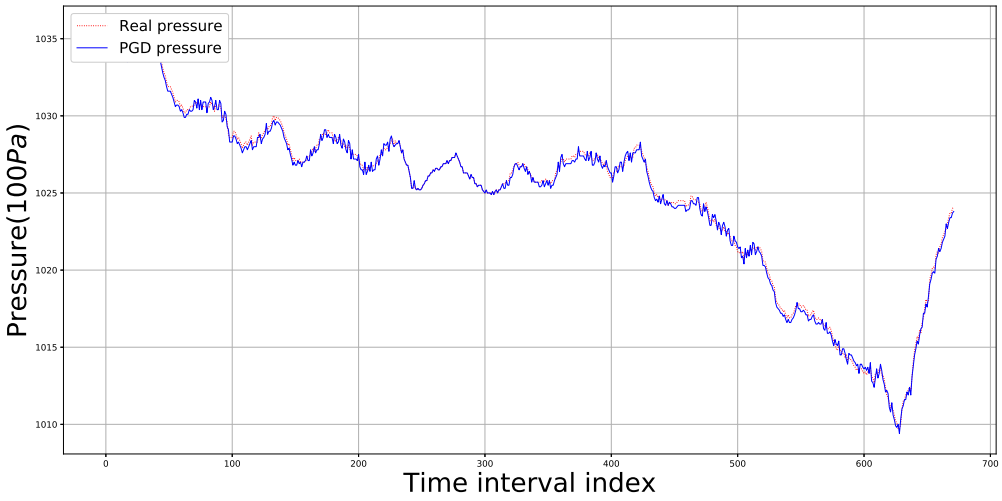

Figure 3.4: Adversarial examples on pressure.

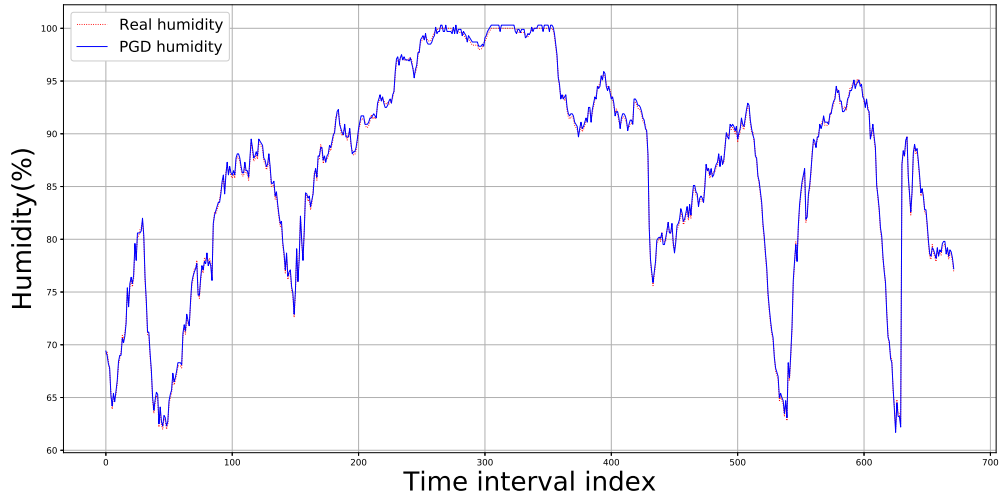

Figure 3.5: Adversarial examples on humidity.

bring in. During the experiments, we apply $0-1$ normalization to weather data first, so a fair evaluation over different weather variables can be provided.

After $0-1$ normalization, a perturbation limit at 0.01 is set as default in our simulation. Fig. 3.12, Fig. 3.13 and Fig. 3.14 are some examples. Since the range of all variables are now from 0 to 1, the perturbation limit has a fair effect over all weather features.

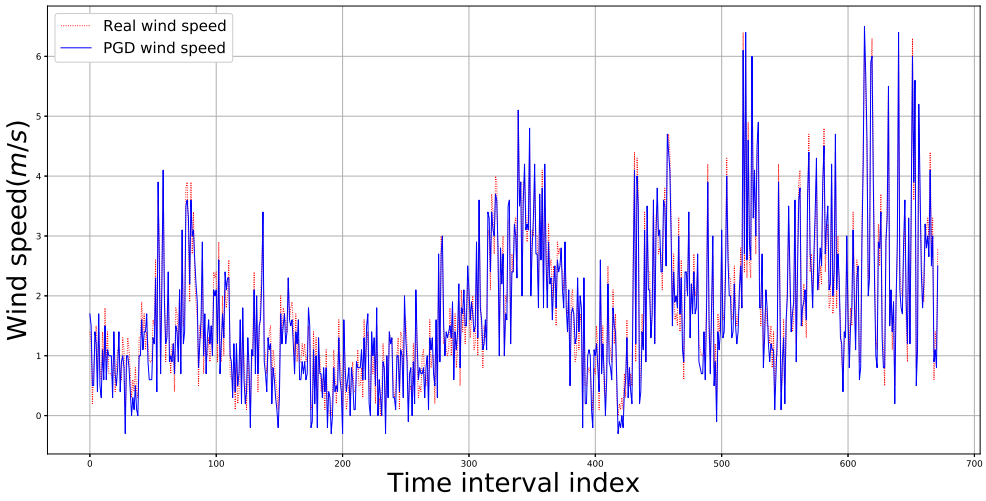

Figure 3.6: Adversarial examples on wind speed.

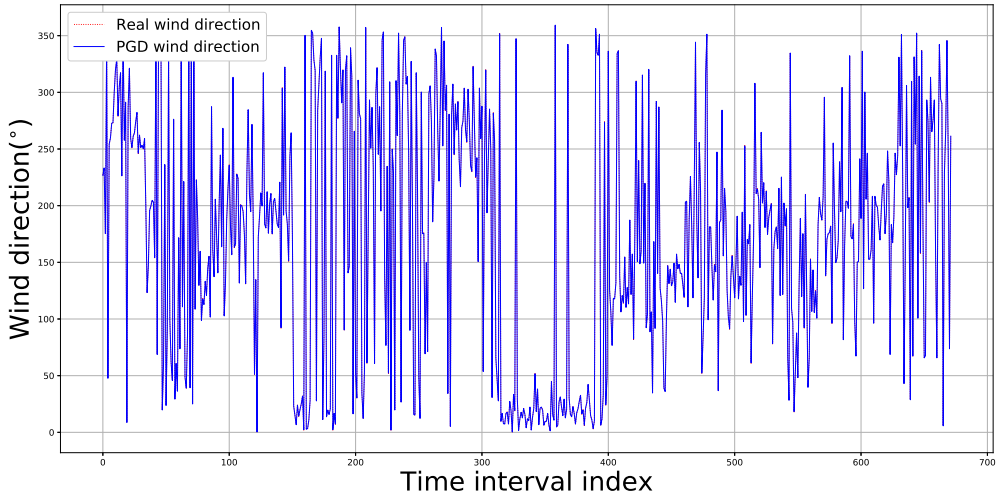

Figure 3.7: Adversarial examples on wind direction.

3.4.3 White Box Attack with Zhejiang Data

The Zhejiang dataset is used to test the adversarial attack performance on a DNN model. During the training stage, historical weather data together with forecasted data in year 2019 are used. The structure of the DNN model is shown in Fig. 3.15.

Other than the input and output layers, our DNN model contains a $10 * 9 * 9$ dense layer to augment the input, three conv2d layers to compute high dimensional variables, one dropout

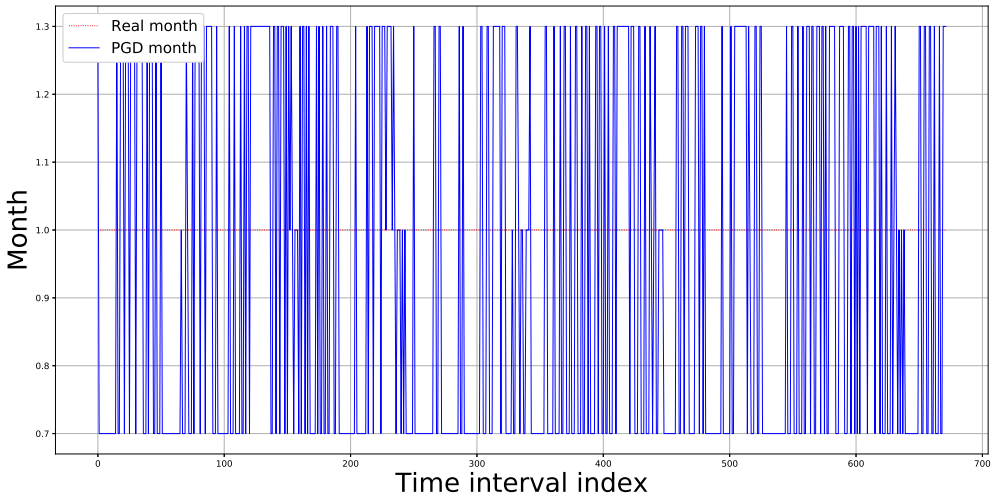

Figure 3.8: Adversarial examples on month.

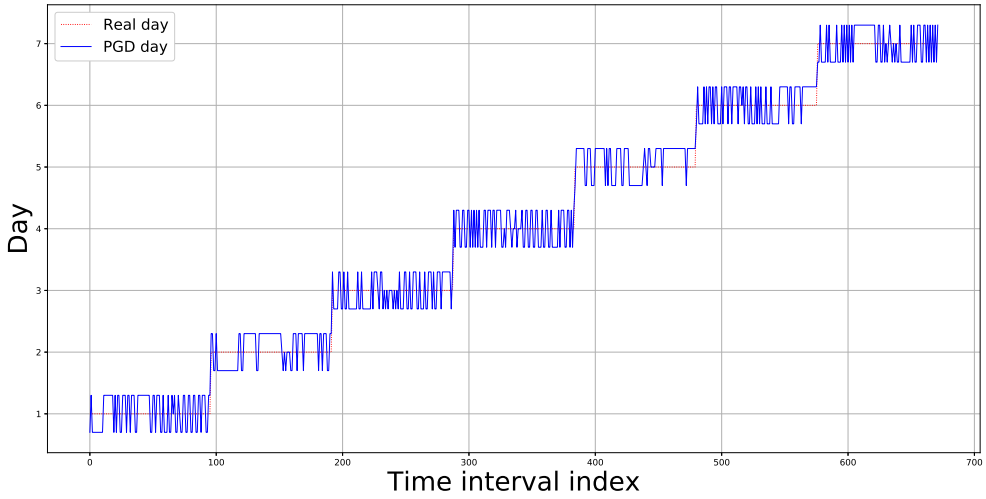

Figure 3.9: Adversarial examples on day.

layer with flatten layer to avoid overfitting, which connects to another $10 * 10$ dense layer. We use the Adam optimization from TensorFlow 2 with a learning rate of 0.001 to train our DNN network. Since we only evaluate RMSE, the mean squared error between the forecasted solar intensity and the real value is used as loss function.

Our test data is the forecasted weather data and real solar intensity in year 2020. The first seven days are used for presentation. To implement the DNN model with the datasets we have, we use weather data and solar intensity from both the real set and forecasted set in year

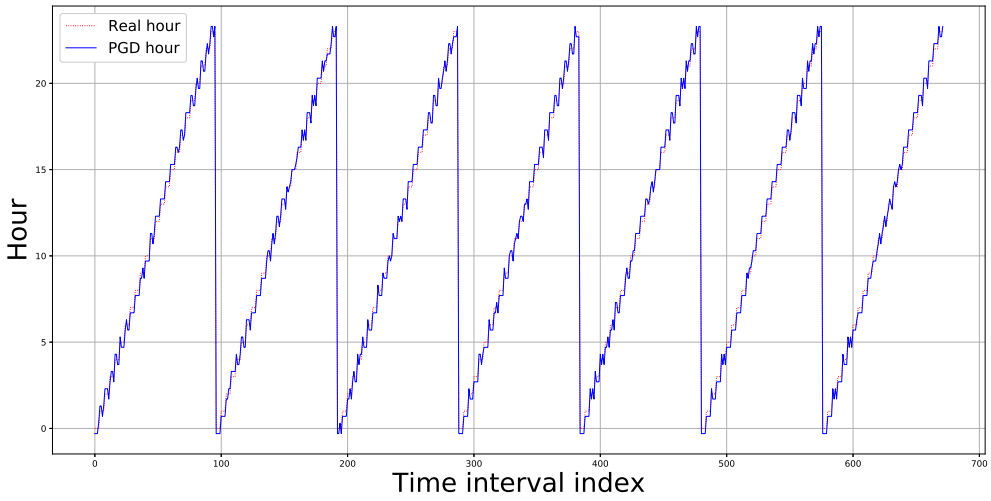

Figure 3.10: Adversarial examples on hour.

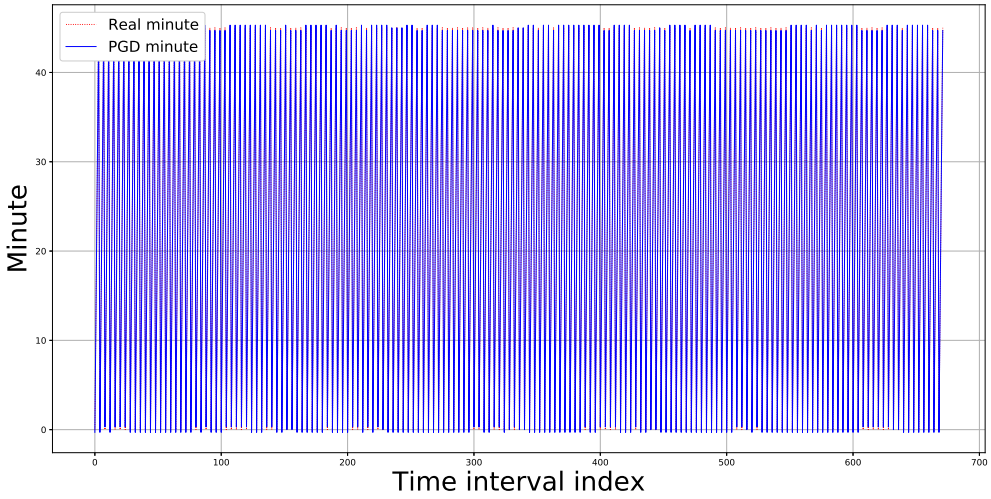

Figure 3.11: Adversarial examples on minute.

2019 to train our model. Solar intensity in year 2020 are being forecasted with the input of the corresponding forecasted weather data and time stamp. After 10000 epochs of training, the forecast results using the DNN model is presented in Fig. 3.16. 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 the DNN model has to accurately predict solar intensity using historical weather data.

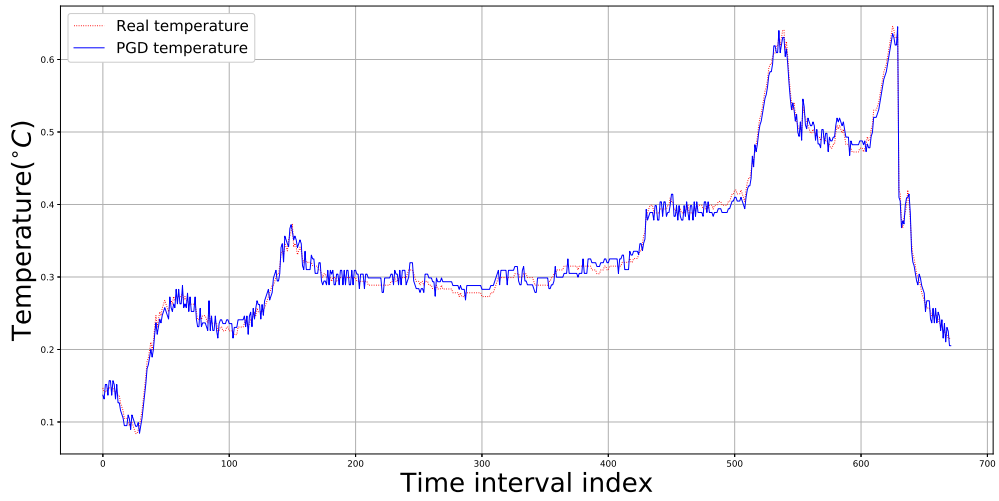

Figure 3.12: Adversarial examples on normalized temperature.

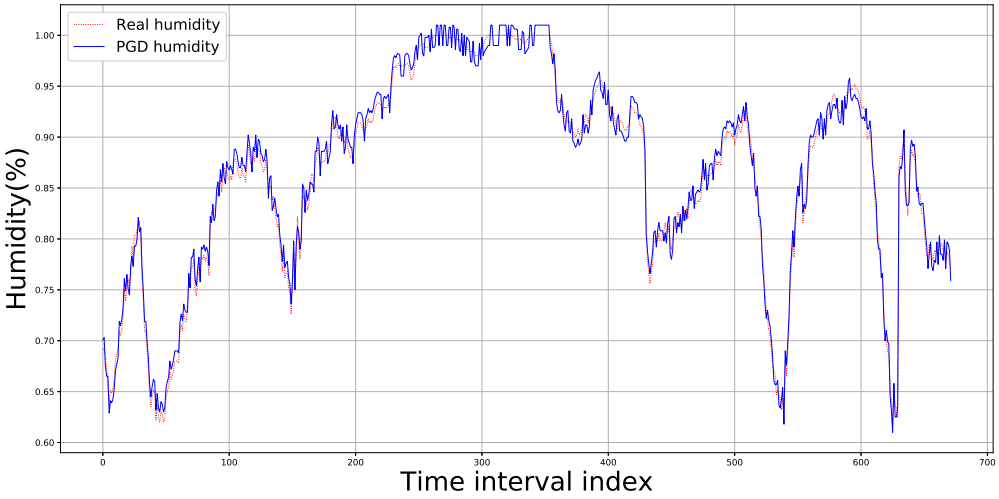

Figure 3.13: Adversarial examples on normalized humidity.

After validation of the forecast accuracy of the DNN model, we start to generate adversarial examples to see if it undermines the performance of the trained DNN model. First, we apply FGSM to generate adversarial examples. The result comparing with both real solar intensity and DNN forecast are shown in Fig. 3.17 and Fig. 3.18, respectively. As shown in Fig. 3.18, there is a considerable difference between the the DNN results using the original data and that using the adversarial attacked data. After the FGSM attack, the RMSE of the DNN model becomes 49.6529 watt/m^2, which is 1.5 times higher than that with the original data using

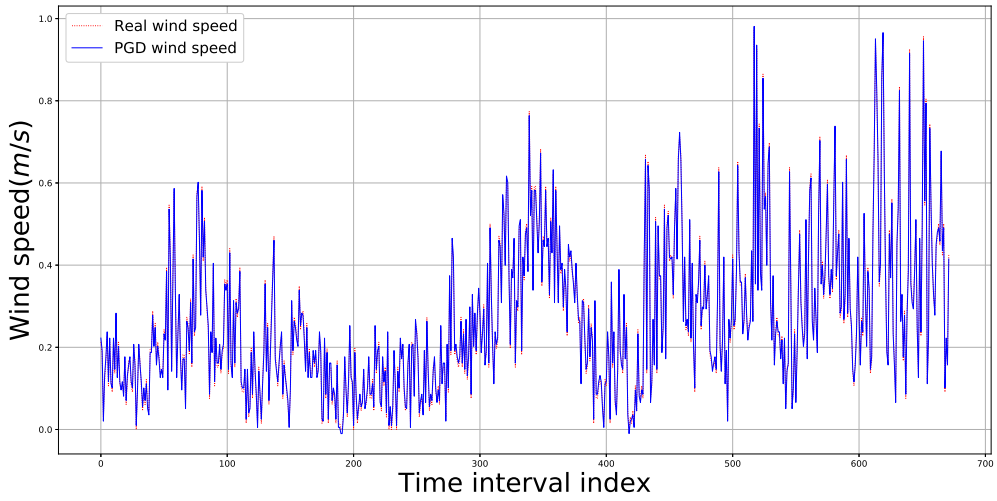

Figure 3.14: Adversarial examples on normalized wind speed.

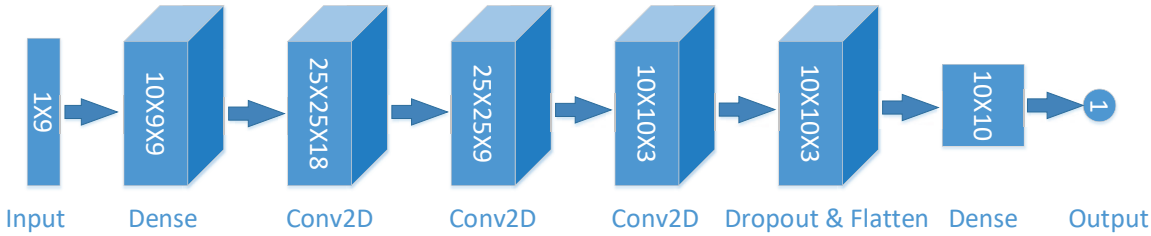

Figure 3.15: Structure of the DNN model used in our experiments.

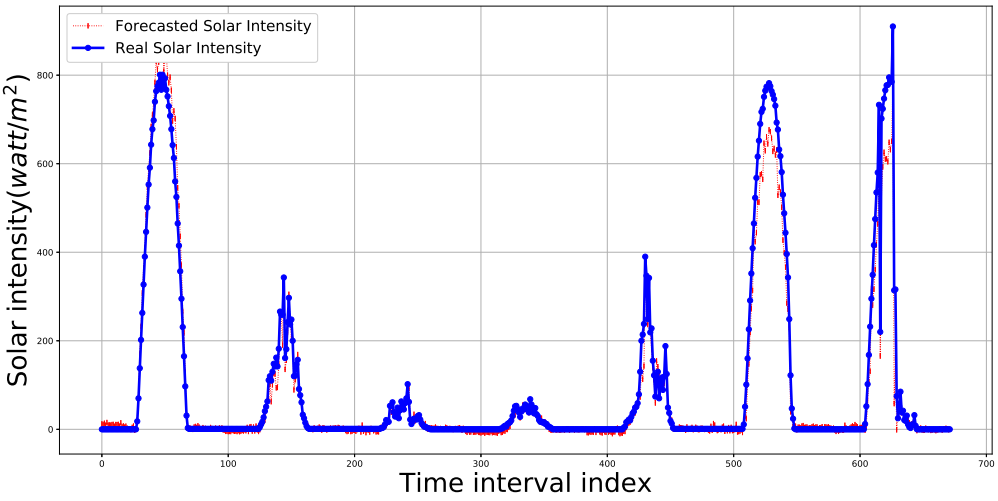

Figure 3.16: DNN forecasted solar intensity vs. ground truth solar intensity.

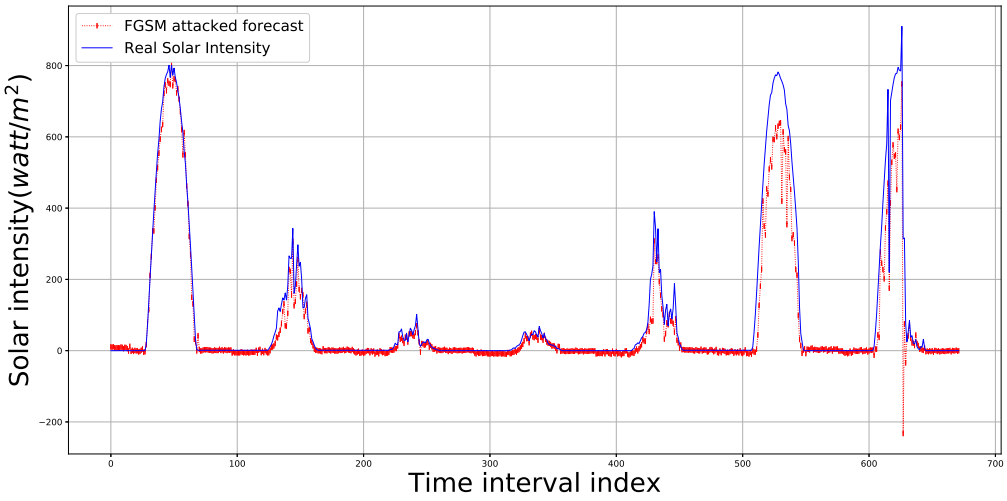

Figure 3.17: DNN forecasted results using FGSM attacked data vs. real solar intensity.

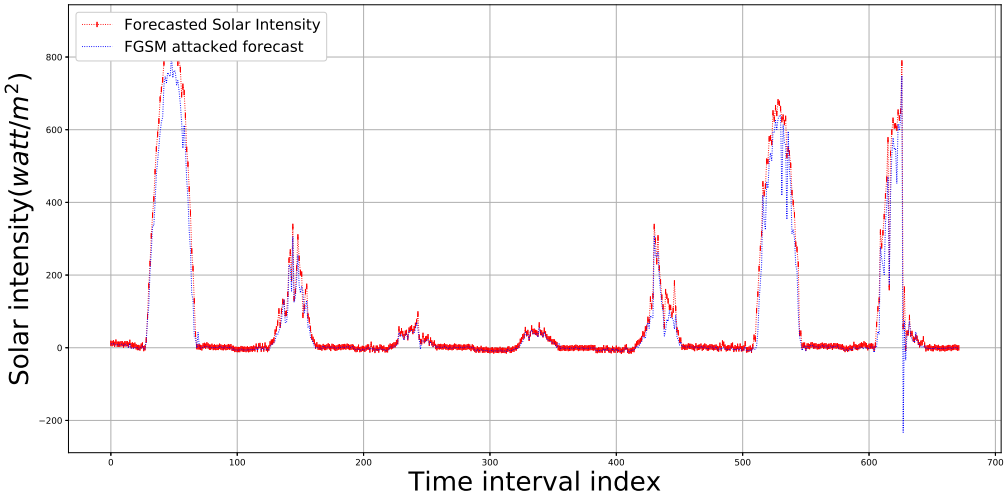

Figure 3.18: DNN forecasted results using the original data vs using FGSM attacked data.

the same DNN model. Therefore, the answer to whether adversarial examples are effective to regression problem is yes.

After testing over FGSM, PGD is also examined in our experiments. The perturbation limit is set to $\delta = 0.01$ at first like previous FGSM attack, the attack iteration step size is 0.0001, and the number of attack iterations is set to 40.

With the PGD adversarial examples, forecasted results become even more distorted as shown in Figs. 3.19, 3.20, and 3.21. To evaluate the difference between FGSM and PGD, we

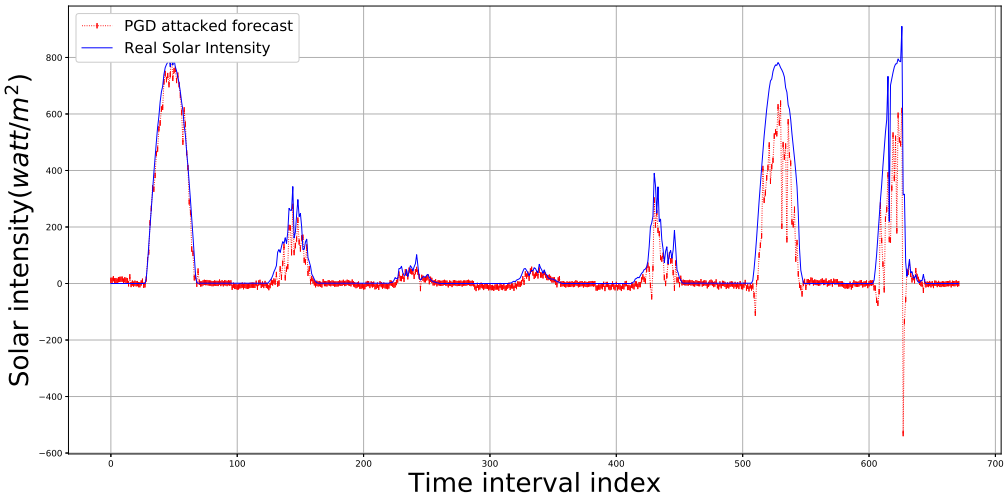

Figure 3.19: DNN forecasted results using PGD attacked data vs. real solar intensity.

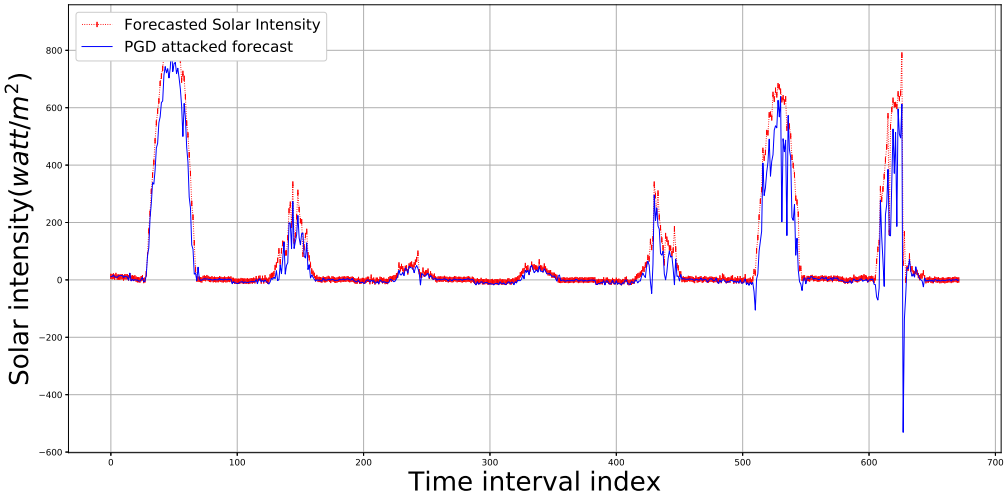

Figure 3.20: DNN forecasted results using the original data vs. using PGD attacked data.

find that the RMSE of PGD attacks is 94.1743, which is another 90% increase compared to FGSM, and more than 4 times of that of the original DNN forecast. It is easy to see PGD has a stronger effect than FGSM on attack efficiency.

One fascinating characteristics of adversarial attacks is that they do not create suspicious value on the original "0"s. That is, as we can observe from every result, solar intensity cannot be something away from "0" through midnight to the next morning. With this feature, it is very hard for a person to identify the problem by just inspecting the curves.

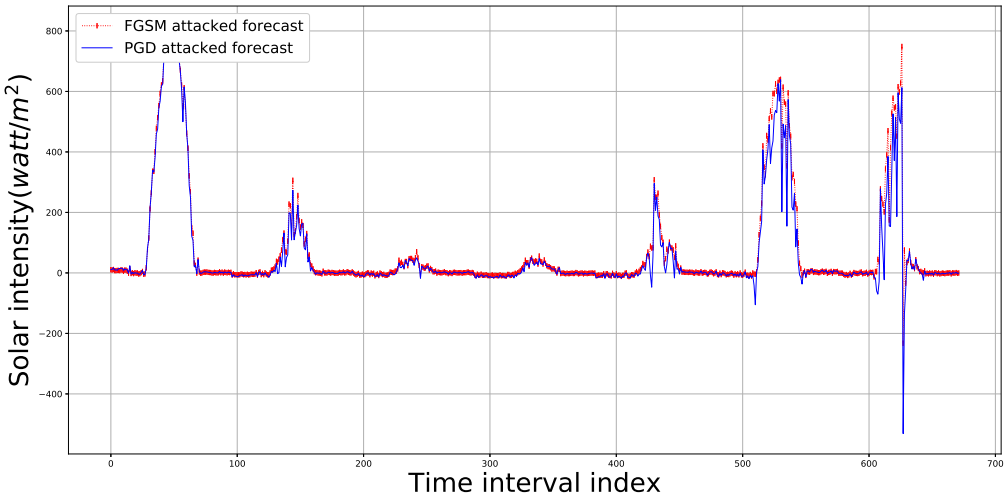

Figure 3.21: DNN forecasted results using FGSM attacked data vs. PGD attacked data.

Table 3.1: RMSE Comparison of White Box Attacks

	DNN	FGSM	PGD	Adversarially Trained
RMSE (watts/m^2)	21.0038	49.6529	94.1743	63.6610
Increment over DNN	–	136.3996%	348.3679%	203.0928%

Since PGD also provide us with the ability of adversarial training, we are also interested to see if adversarial training can help to improve the resilience of DNN regression models. The 7-day results are shown in Fig. 3.22. If we compare the results with the previously attacked results, we can see the resistance provided by the current DNN model to adversarial examples. The RMSE for the 7-day period is 63.6610 watt/m^2, a 32.4010% reduction from 94.1743 watt/m^2 without adversarial training. However, the overall RMSE of forecasting using non-attacked data using the adversarially trained DNN model has increased to 56.1179 watt/ms^2, which is a 150.3900% increase over 22.4122 watt/m^2 without adversarial training. Our explanation is, using adversarial examples can be seen as a trade-off between robustness and accuracy. Although adversarial training provides resilience against malicious data, it also sacrifices the accuracy of forecasting with the untampered data.

A comparison of the RMSE values achieved by the models and attacks methods is summarized in Table 3.1.

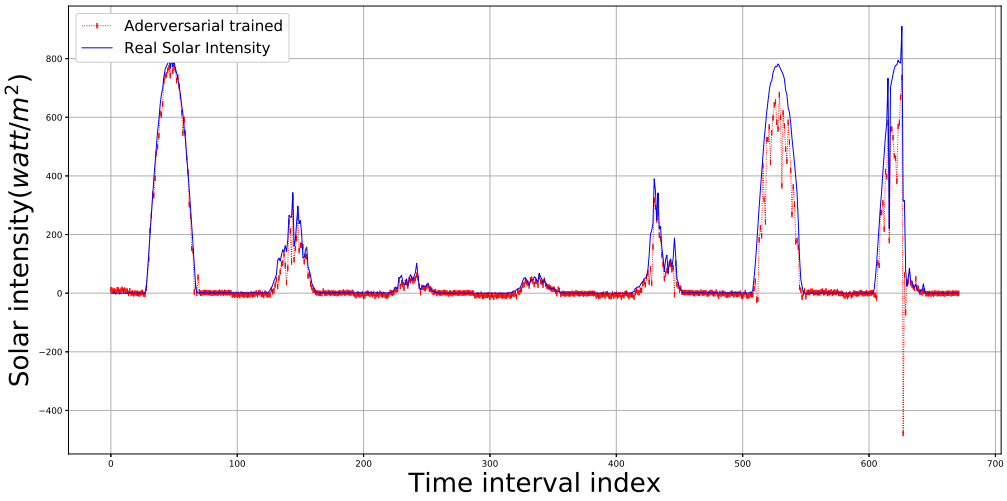

Figure 3.22: Adversarial trained forecast results vs. real solar intensity.

3.4.4 Black Box Attack with the UMass Dataset

In [16], the authors applied the multi-linear regression technique to forecast solar intensity. Though not highly accurate, the data itself still reflects quite a strong linear correlation between weather parameters and solar intensity. In [38], several different models are used to capture the linear and non-linear relationship, respectively, in the dataset for solar intensity forecasting. Since the data structure is partially linear correlated, we conjecture that a black box attack using adversarial data, which are directly generated from a well-trained DNN model, would also be effective.

In this experiment, we use the PGD algorithm to generate adversarial examples. Because we have used our LASSO-based algorithm to achieve a very decent accuracy with the UMass [1] dataset, we now use the dataset again to demonstrate the impact of black box attack 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 have some minor modification to fit such data for generating adversarial examples, but the basic idea stays the same. Here, we use forecasted temperature, dew-point, wind speed, precipitation, and humidity as input, while solar intensity as output as before. In this scenario, our proposed Lasso-based algorithm will forecast solar intensity by day average.

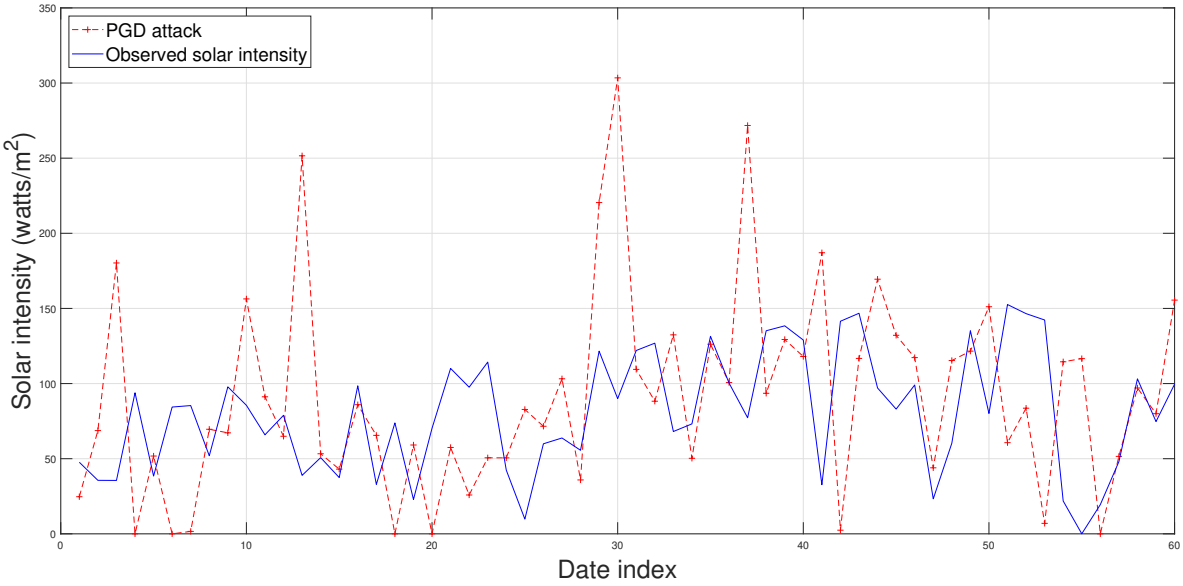

Figure 3.23: Forecasting results achieved by LASSO using PGD attacked data vs. real solar intensity ($\delta = 0.01$).

The results by applying the LASSO algorithm to the PGD attacked data are presented in Fig. 3.23 for perturbation limit $\delta = 0.01$, compared to the ground truth. The forecasting results by LASSO using the untampered UMass data are presented in Fig. 3.24. Compared with the RMSE of $14.0262 watt/m^2$ in Fig. 3.24, the RMSE of LASSO on PGD attacked data is increased to $91.1597 watt/m^2$ (i.e., 5.4992 times higher). From Fig. 3.24, we can also see that the threshold feature of LASSO-based model (see (2.13)) alleviates the loss caused by the adversarial attack. There is no minus value generated so the RMSE is reduced in the corresponding regions.

We also try different perturbation levels, which are set to $\delta = 0.015$ and $\delta = 0.02$. The results are presented in Fig. 3.25 and Fig. 3.26, resepectively. In the result above, corresponding RMSEs increase with the rise of perturbation limit as expected. We also need to notice that the difference between attacked forecast and real value on every point increases in general, which means the adversarial example might be able to find the maximal loss even being applied to other models.

The RMSEs for each of the black box attack scenarios on the LASSO-based Model with the UMass Dataset are summarized in Table 3.2.

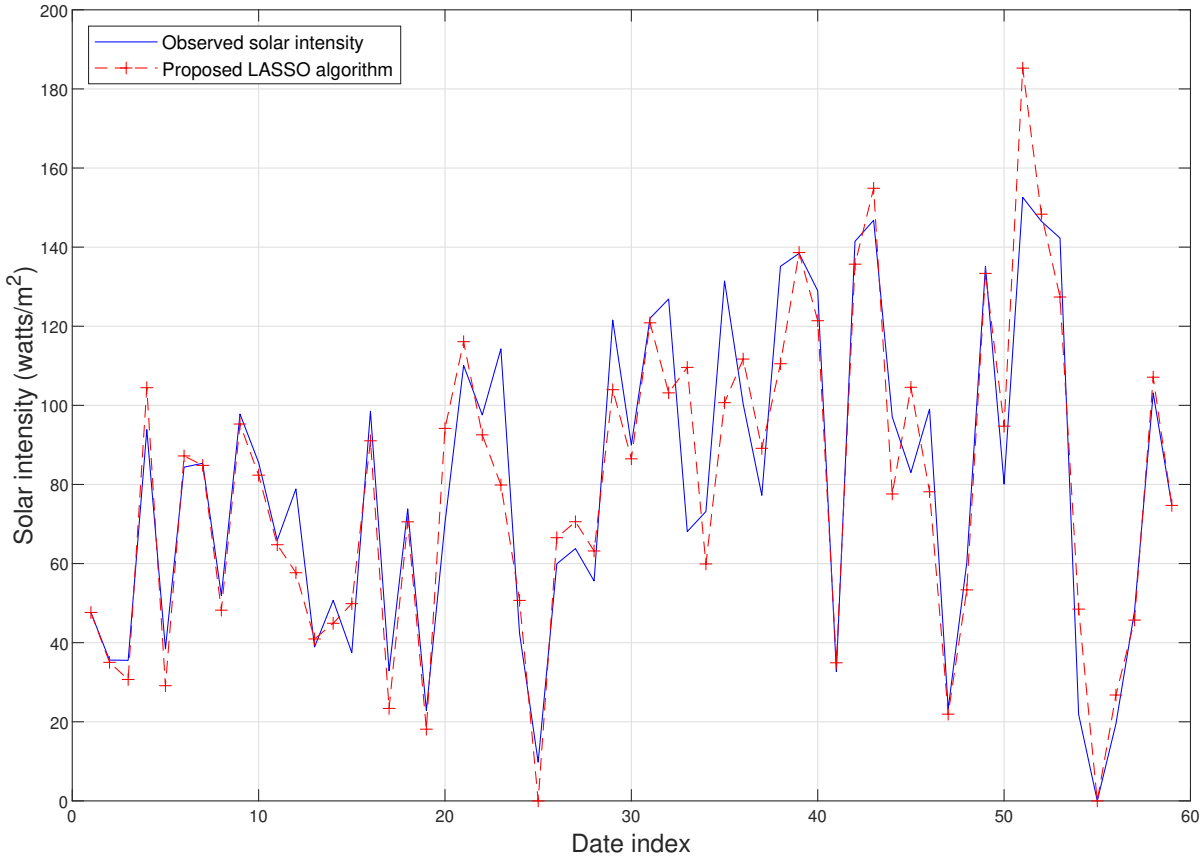

Figure 3.24: Solar intensity prediction using the proposed LASSO-based method with the untampered UMass dataset ($\delta = 0$).

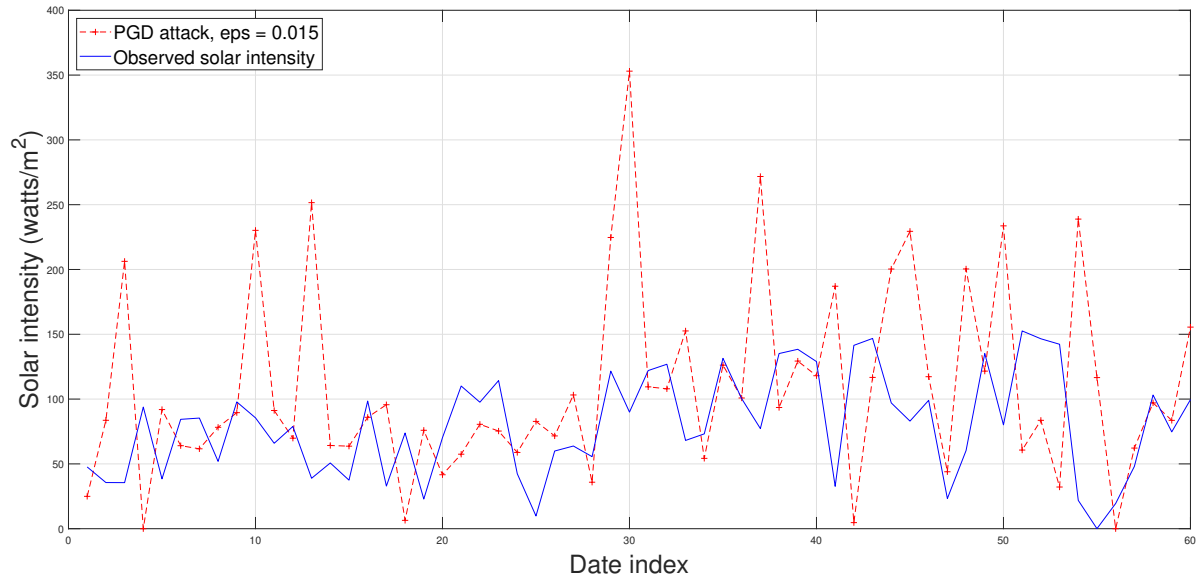

Figure 3.25: Forecasting results achieved by LASSO using PGD attacked data vs. real solar intensity ($\delta = 0.015$).

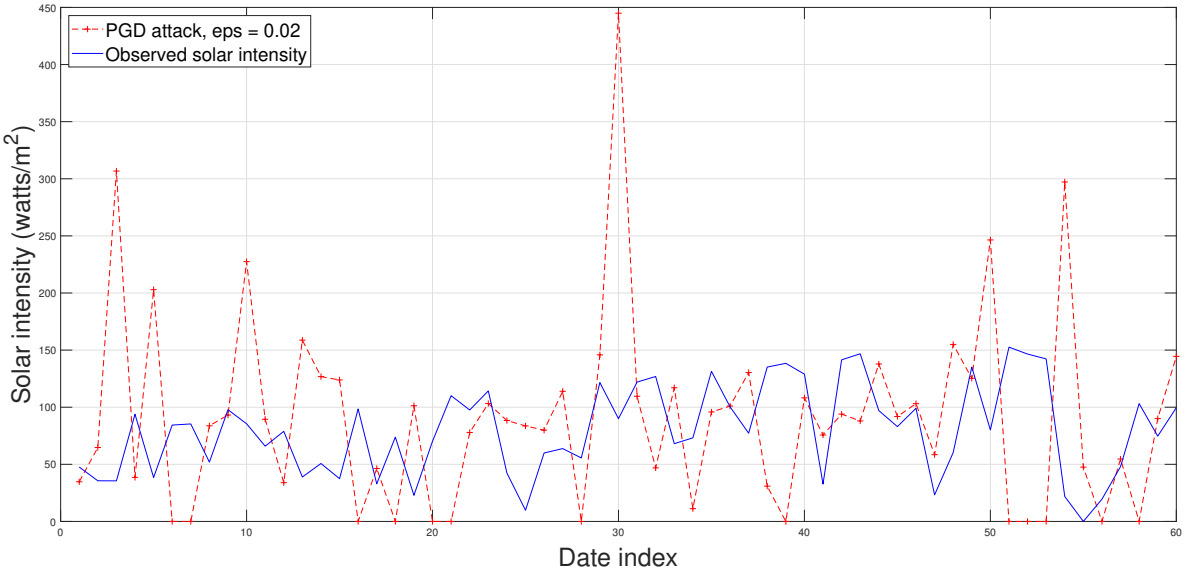

Figure 3.26: Forecasting results achieved by LASSO using PGD attacked data vs. real solar intensity ($\delta = 0.02$).

Table 3.2: RMSE Comparison for Black Box Attacks on the LASSO-based Model with the UMass Dataset

Pertubation Limit δ	0.01	0.015	0.02
RMSE (watts/m^2)	91.1597	111.2547	156.9859

3.4.5 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. 3.27. The corresponding results using the PGD attacked Zhejiang dataset with perturbation limit $\delta = 0.01$ are presented in Fig. 3.28.

Specifically, the LASSO-based model achieves an RMSE of 24.6907 watt/m^2 with the untampered Zhejiang dataset. Although black box attack still affects the forecast result, the LASSO-based model performs a little better than the DNN model, i.e., more resilient to the PGD attacks, with an RMSE of 73.6029 watt/m^2 (comparing 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

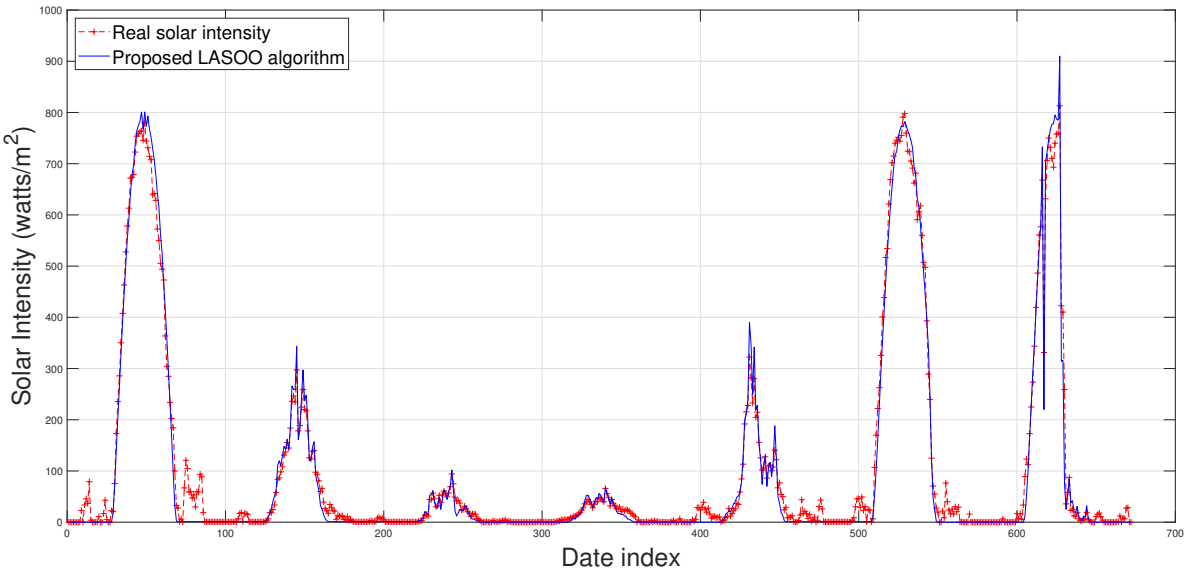

Figure 3.27: Solar power generation prediction using the proposed LASSO-based method with the untampered Zhejiang dataset vs. the real solar intensity.

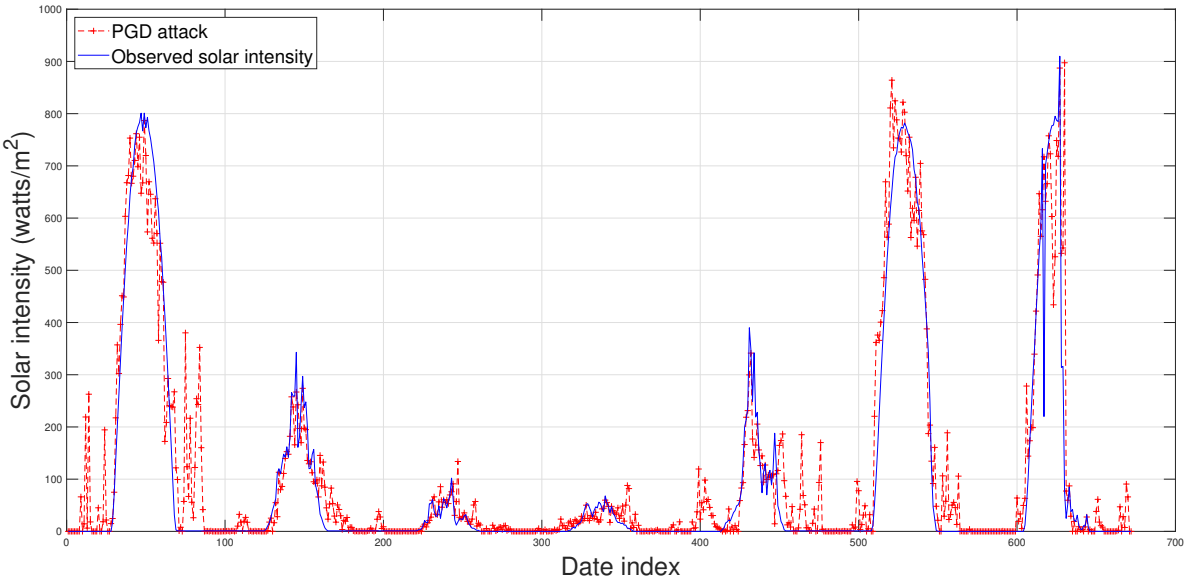

Figure 3.28: Solar power generation prediction using the proposed LASSO-based method with the PGD attacked Zhejiang dataset ($\delta = 0.01$) vs. the real solar intensity.

to launch a black box attack on our LASSO-based statistical model, or even other different models. This means, conventional solar generation forecast schemes which depends on other weather information may also be vulnerable to adversarial attacks.

3.5 Conclusion

In this chapter, we examined how adversarial attacks affect both DNN model and our proposed LASSO-based algorithm presented in Chapter 2. We first introduced the preliminaries of adversarial attacks. Then we designed both white box and black box attack to the DNN and LASSO-based models, correspondingly. We successfully used FGSM and PGD to generate white box attacks on the trained DNN model and found that PGD adversarial training only provides a 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 is very likely to pose a deep threat on the forecasting problems with similar data structures.

For future work, it would be interesting to transform the current LASSO-based algorithm to a short interval prediction version for practical use. Furthermore, due to the potential threat over conventional solar intensity forecasting problem, it is time to build a defense mechanism or update the current forecast techniques to make them more resilient to such attacks.

Chapter 4

Conclusions

In this dissertation work, we proposed a LASSO-based algorithm to accurately forecast day-ahead photovoltaic power generation. Furthermore, we examine the potential threat that adversarial examples can cause to the current solar forecast models.

In Chapter 2, we investigated the relationship between a series of weather data and solar intensity. A LASSO-based algorithm was proposed to maximize the day-average forecast accuracy. To validate its advantage, we compared our proposed algorithm with several other widely used regression algorithms, including SVM and TLLE. Through simulation results, we found that the LASSO-based algorithm outperformed the several baseline schemes in accuracy even when a smaller training dataset was used. Meanwhile, we also investigated its feature of variable selection. With a trade-off between accuracy and model complexity, we could have a reduced model while maintaining an acceptable prediction precision.

In Chapter 3, we studied the impact of adversarial attacks to practical solar forecast schemes. We first applied white box attacks on a DNN model and used both FGSM and PGD to test their effectiveness. We also tried adversarial training to see if it helped to build robust regression models. The result showed that FGSM and PGD were both able to degrade the accuracy of the two forecasting scheme, while PGD had a more serious impact on the models. Adversarial training provided some resilience to adversarial attacks, but it also degraded the overall accuracy of the forecasting models on untampered data. Furthermore, we also generated black box attacks to our LASSO-based statistical model. The result showed that adversarial examples were capable to attack such statistical models as well.

Overall, our work advanced the state-of-the-art of solar power generation forecasting with more effective, accurate forecasting schemes. Our study of potential threats that undermine the performance of representative forecasting models shed insights on such problems and would be useful for developing effective defense mechanisms for resilient renewable energy utilization in the future.

Chapter 5

Future Work

The development of renewable energy is accelerated by the progress of IoT and smart grid technologies. While this dissertation is focused on solar power generation forecasting, we plan to explore other smart grid related problems in practical focused areas, include but not limited to load forecasting, demand response, and the power market.

Nowadays, machine learning has already become an indispensable methodology for optimization, prediction, and system control. While the conventional power grid relies on clear information collected from the system and environment, there has always been issues about stability and security. When merged with machine learning techniques, the original planning problem of system control and resource allocation can often be transferred to a learning problem, which could lead to lower cost and better automation. For example, reinforcement learning like DDPG, PPO and SAC emphasis on the action and reward circle, which does not require the operator to have a specific knowledge of the grid parameters. From another aspect, the core idea of smart grid is to involve more interactive behaviors. However, the common resident would like to enjoy the convenience brought about by new technology, but does not have sufficient professional knowledge to fully utilize certain tools. Therefore, compared to planned grid operation and user defined electric usage, machine learning based methodology seems to be a promising way, which could provide better performance and user experience. According to the reasons mentioned above, we plan to use machine learning as a major tool for future research.

Several future research directions will be explored, which are given in the following:

- **Robust Load Prediction based on Group Behavior:** Traditional load prediction usually use history data and weather condition as input. However, the design completely ignored

the impact of human behavior. For example, in a long national holiday, residential load could be low in the beginning but high in the middle of the day due to different group activities. Such group activities and their corresponding causes could be considered as an important factor. I plan to use XGBoost or LSTM to model group behavior to achieve a much better performance than current load prediction scheme.

- **User Experience Oriented Demand Response:** In prior demand response studies, people usually focuse on how to achieve the minimum cost. Although the economic condition can be satisfied, user experience may not even be considered. Comfort index should be a primary goal for the experience of residents, so we should take it as part of the reward function in a reinforcement learning model. How to build a reinforcement learning model with a trade-off between the overall minimum cost and user satisfaction will be the objective of my future research.

- **Power Market Optimization with Distributed Renewable Energy Resource Groups:** With the development of smart grid, techniques of integrated renewable energy resource are always seen as a promising means for both reducing local energy cost and making additional profit. It would be practical to take power market decisions into consideration since the optimization for trading in a group of local renewable energy sources is yet to be studied. In my future work, I will study on how to achieve optimal profit while avoiding power failure in such a setting.

References

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

[2] The Detection of Archaeological Residues using Remote-sensing Techniques (DART) Project, "DART weather data," [online] Available: https://dartportal.leeds.ac.uk/dataset/dart_monitoring_weather_data/.

[3] Y. Wang, S. Mao, and R. Nelms, "On hierarchical power scheduling for the macrogrid and cooperative microgrids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1574–1584, Dec. 2015.

[4] M. Alam, K. Muttaqi, and D. Sutanto, "A comprehensive assessment tool for solar PV impacts on low voltage three phase distribution networks," in *Proc. IEEE ICDRET'12*, Dhaka, Bangladesh, Jan. 2012, pp. 1–5.

[5] Y. Wang, S. Mao, and R. Nelms, *Online Algorithms for Optimal Energy Distribution in Microgrids*, 1st ed. New York, NY: Springer, 2015.

[6] Y. Wang, S. Mao, and R. M. Nelms, "Distributed online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Internet of Things J.*, vol. 1, no. 1, pp. 70–80, 2014.

[7] L. Yu, T. Jiang, and Y. Zou, "Distributed online energy management for data centers and electric vehicles in smart grid," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 1373–1384, 2016.

[8] Y. Wang, S. Mao, and R. M. Nelms, "An online algorithm for optimal real-time energy distribution in smart grid," *IEEE Trans. Emerging Topics Comput.*, vol. 1, no. 1, pp. 10–21, July 2013.

[9] 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.

[10] M. Li, P. He, and L. Zhao, "Dynamic load balancing applying water-filling approach in smart grid systems," *IEEE Internet of Things J.*, vol. 4, no. 1, pp. 247–257, 2017.

[11] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid," *IEEE Internet of Things J.*, 2017.

[12] J. Pan, R. Jain, S. Paul, T. Vu, A. Saifullah, and M. Sha, "An internet of things framework for smart energy in buildings: designs, prototype, and experiments," *IEEE Internet of Things J.*, vol. 2, no. 6, pp. 527–537, 2015.

[13] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 411–419, Jan. 2014.

[14] V. Dordonnat, S. J. Koopman, and M. Ooms, "Dynamic factors in periodic time-varying regressions with an application to hourly electricity load modelling," *Elsevier Comput. Stat. & Data Analy.*, vol. 56, no. 11, pp. 3134–3152, Nov. 2012.

[15] 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.

[16] 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.

[17] 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.

[18] R. J. Bessa, A. Trindade, and V. Miranda, "Spatial-temporal solar power forecasting for smart grids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 232–241, Feb. 2015.

[19] Y. Zhang, M. Beaudin, R. Taheri, H. Zareipour, and D. Wood, "Day-ahead power output forecasting for small-scale solar photovoltaic electricity generators," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2253–2262, Sept. 2015.

[20] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," *CSEE J. Power Energy Sys.*, vol. 1, no. 4, pp. 38–46, Dec. 2015.

[21] 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.

[22] L. Martín, L. F. Zarzalejo, J. Polo, A. Navarro, R. Marchante, and M. Cony, "Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning," *Elsevier Solar Energy*, vol. 84, no. 10, pp. 1772–1781, Oct. 2010.

[23] C. Chen, S. Duan, T. Cai, and B. Liu, "Online 24-h solar power forecasting based on weather type classification using artificial neural network," *Elsevier Solar Energy*, vol. 85, no. 11, pp. 2856–2870, Nov. 2011.

[24] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Trans. Ind. App.*, vol. 48, no. 3, pp. 1064–1069, May/June 2012.

[25] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Royal Stat. Soc., Series B (Methodological)*, vol. 73, no. 3, pp. 267–288, June 1996.

[26] P. Zeng, T. He, and Y. Zhu, "A Lasso-type approach for estimation and variable selection in single index models," *J. Comput. Graph. Stat.*, vol. 21, no. 1, pp. 92–109, Apr. 2012.

[27] W. Y. Hwang, H. H. Zhang, and S. Ghosal, "FIRST: Combining forward iterative selection and shrinkage in high dimensional sparse linear regression," *Statistics and its Interface*, vol. 2, no. 3, pp. 341–348, June 2009.

[28] S. Luo and S. Ghosal, "Forward selection and estimation in high dimensional single index models," *Statistical Methodology*, vol. 33, pp. 172–179, Dec. 2016.

[29] R. Zhang, Z. Huang, and Y. Lv, "Statistical inference for the index parameter in single-index models," *Elsevier J. Multivariate Analysis*, vol. 101, no. 4, pp. 1026–1041, Apr. 2010.

[30] M. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–89, June 1938.

[31] R. E. Barlow, D. J. Bartholomew, J. Bremner, and H. D. Brunk, *Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression*. Wiley New York, 1972.

[32] O. Burdakov, A. Grimvall, and M. Hussian, "A generalised PAV algorithm for monotonic regression in several variables," in *Proc. COMPSTAT'04*, Prague, Czech Republic, 2004, pp. 761–767.

[33] M. Feng and S. Mao, "Harvest the potential of massive mimo with multi-layer techniques," *IEEE Network*, vol. 30, no. 5, pp. 40–45, 2016.

[34] M. Feng, S. Mao, and T. Jiang, "Base station on-off switching in 5g wireless networks: Approaches and challenges," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 46–54, 2017.

[35] ——, "Boost: Base station on-off switching strategy for green massive mimo hetnets," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7319–7332, 2017.

[36] 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.

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

[38] 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 Journal*, vol. 6, no. 2, pp. 2933–2944, Apr. 2019.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] N. Tang, S. Mao, Y. Wang, and R. Nelms, "LASSO-based single index model for solar power generation forecasting," in *Proc. IEEE GLOBECOM'17*, Singapore, Dec. 2017, pp. 1–6.

[44] 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

[45] 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

[46] 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.

[47] 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.

[48] 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

[49] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. 2006 ACM Symp. Inform., Computer and Commun. Security*, Taipei, Taiwan, Mar. 2006, pp. 16–25.

[50] 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.

[51] 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.

[52] 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.

[53] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 2015 Int. Conf. Learn. Representations (ICLR'15)*, San Diego, CA, May 2015, pp. 1–11.

[54] ——, "Towards deep learning models resistant to adversarial attacks," in *Proc. 2018 Int. Conf. Learn Representations (ICLR'18)*, Vancouver, Canada, Apr.-May 2018.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

Appendix A

Publications

A.1 Conference Publications

1. **Ningkai Tang**, Shiwen Mao, Yu Wang and Mark Nelms, "LASSO-Based Single Index Model for Solar Power Generation Forecasting," in *Proc. IEEE GLOBECOM 2017*, Singapore, Dec. 2017, pp. 1–6 .

2. **Ningkai Tang**, Shiwen Mao and Sastry Kompella, "On power control in full duplex underlay cognitive radio networks," in *Proc. IEEE MILCOM 2014*, Baltimore, MD, Oct. 2014, pp. 949–954.

A.2 Journal Publications

1. **Ningkai Tang**, Shiwen Mao, Yu Wang and Mark Nelms, "Solar Power Generation Forecasting With a LASSO-Based Approach," in *IEEE Internet of Things Journal*, Journal, vol. 5, no. 2, pp. 1090–1099, April 2018.

2. **Ningkai Tang**, Shiwen Mao and Sastry Kompella, "On power control in full duplex underlay cognitive radio networks," in *Elsevier Ad Hoc Networks*, vol. 37, no. 2, pp. 183–194, 2016.