Exploiting Vulnerabilities of Load Forecasting Through Adversarial Attacks

Yize Chen
Electrical and Computer Engineering,
University of Washington
Seattle, WA
yizechen@uw.edu

Yushi Tan
Electrical and Computer Engineering,
University of Washington
Seattle, WA
ystan@uw.edu

Baosen Zhang
Electrical and Computer Engineering,
University of Washington
Seattle, WA
zhangbao@uw.edu

ABSTRACT

Load forecasting plays a critical role in the operation and planning of power systems. By using input features such as historical loads and weather forecasts, system operators and utilities build forecast models to guide decision making in commitment and dispatch. As the forecasting techniques becomes more sophisticated, however, they also become more vulnerable to cybersecurity threats. In this paper, we study the vulnerability of a class of load forecasting algorithms and analyze the potential impact on the power system operations, such as load shedding and increased dispatch costs. Specifically, we propose data injection attack algorithms that require minimal assumptions on the ability of the adversary. The attacker does not need to have knowledge about the load forecasting model or the underlying power system. Surprisingly, our results indicate that standard load forecasting algorithms are quite vulnerable to the designed black-box attacks. By only injecting malicious data in temperature from online weather forecast APIs, an attacker could manipulate load forecasts in arbitrary directions and cause significant and targeted damages to system operations.

CCS CONCEPTS

• **Security and privacy** → *Systems security*; • **Theory of computation** → *Machine learning theory*;

KEYWORDS

Adversarial Attacks, Cyber-Physical Security, Load Forecast, Machine Learning, Unit Commitment, Economic Dispatch

ACM Reference Format:
Yize Chen, Yushi Tan, and Baosen Zhang. 2019. Exploiting Vulnerabilities of Load Forecasting Through Adversarial Attacks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19), June 25–28, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3307772.3328314

1 INTRODUCTION

Load forecasting is a fundamental step in power system planning and operations. It is used to inform system operators the future load profiles, and serves as the basis of decision-making problems such as unit commitment, reserve management, economic dispatch and maintenance scheduling [14]. Consequently, the accuracy of forecasted loads directly impact the cost and reliability of system operations [16]. With a growing penetration of new technologies into the demand side, the utilities and system operators need to place more importance on both accurate and robust forecasts.

For years, the holy grail in short-term load forecasting has been to *improve the forecast accuracy*, which has been vigorously pursued by the research community. The variations in load are driven by many different factors, including temperature, weather, temporal and seasonal effects (e.g., weekday vs. weekend) and other socioeconomic factors. All of these factors influence the load in nonlinear and complex ways. Over the past decades, a myriad of load forecasting algorithms have been proposed and adopted in practice. See, for example, [7, 13, 14] and the references within.

For simplicity, in this paper, we restrict the inputs of the algorithms to be the historical load data, time indicators and temperature information. These algorithms can be thought of as finding a mapping between the (high dimensional) input features to the forecasted time series of load values. Statistical and machine learning techniques, such as support vector regression [6], ARIMA [12] and neural networks [8, 15] have been applied to short term load forecasting and are well adopted in practice. The recent advances in deep learning opened the door to using more input features and deeper model architectures to further improve load forecasting accuracy and provided some of the best performances to date [20, 32, 38].

As the forecasting methods become more complex and accurate, they are also more susceptible to cybersecurity threats. In this paper, we look into the data vulnerabilities of such methods, where an attacker adversarially injects false data into the input features of forecasting algorithms. Specifically, we investigate false data injection attacks of the temperature data. It is an important input to load forecasting algorithms and is mostly obtained from external services/APIs, therefore providing an easier avenue for data perturbations and attack injections. The potential damage of these types of attacks can be significant, leading to increases in system operation costs and maybe even more catastrophic events such as load shedding. In Figure 1, we show the schematic of threats and proposed attacks to systems.

In this paper, we take the perspective of an attacker and develop attack strategies on load forecasting algorithms, and conduct damage analysis of the proposed attacks. We take a restrictive setting of both the attacker's "knowledge" and "capabilities", where the

e-Energy '19, June 25–28, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6671-7/19/06…$15.00
https://doi.org/10.1145/3307772.3328314

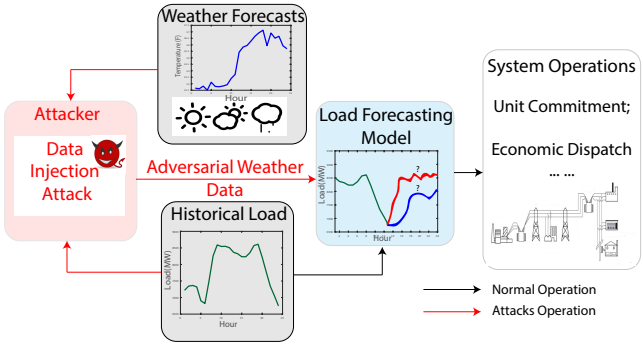

Figure 1: The schematic of our proposed attacks on load forecasting algorithms along with the threats over power system operations. Without knowledge about the forecast model's parameters, the attacker injects designed small, undetectable data perturbations into weather forecasts to induce abnormal system operations.

attacker does not know any parameter of the targeted load forecasting algorithms, and could only inject perturbations into input temperatures under constraints to avoid detection.

Under this setup, we develop two simple data-driven attack strategies for finding the injected perturbations onto the original temperature data. Surprisingly, we find the proposed attacks significantly degrade the performance of a class of (accurate) load forecasting algorithms. With only few degrees of perturbations injected into input temperatures, the load forecasting algorithm's output deviates drastically from original values. We also assess the damages brought by such model vulnerabilities in power system operations. Simulations based on real-world load datasets show that by changing only few degrees of temperature, adversarial forecasts not only increase the operation cost of power systems, but can also lead to load shedding and infeasible generator schedules.

This study illustrates the need to look at other properties of load forecasting techniques in addition to *forecast accuracy*. We demonstrate that accuracy may not mean robustness, and since a wrong forecast of load potentially leads to costly operation decisions or system damages, we call for a more comprehensive analysis when developing and applying load forecasting techniques. Specifically, we make the following contributions in this work:

- To the best of our knowledge, this is the first to evaluate the security issues of load forecasting procedures in power system operations. Data vulnerabilities of current forecasting methods are discussed and formulated.
- Two data-driven, black-box attack algorithms, namely `learn and attack` and `gradient estimation`, are proposed to generate hard-to-detect, adversarial input data for load forecasting algorithms.
- Case studies on power system operations demonstrate potential damages via proposed attacks. We show that the strategically designed adversarial injections could lead to either increased system operating costs or load shedding.

We make our code open source on load forecasting model development, attack implementations and market operation evaluation, and make it as a package for evaluating load forecasting robustness and security[1].

The rest of the paper is organized as follows. A literature review is presented in Section 2; we then briefly summarize a general load forecasting model, and formulate the objective and constraints of attackers in Section 3; in Section 4, we detail the algorithms for implementing the attack; to illustrate the attack's threats to the power system operations, we describe the market setup and a toy example in Section 5; through simulations based on real-world load data in Section 6, we demonstrate the threats posed by the proposed attacks; furthur discussion on model/data security and conclusion are drawn in Section 7.

2 RELATED WORK

In this section, we give brief literature review on both the load forecasting methods and cyber-security of power systems. Our work is different from most related work in two aspects: most of the studies in forecasts do not consider security and robustness, while most of the studies in power system security evaluate attacks with almost no knowledge about the targeted system or constrained capabilities.

Our work is related to the large body of work on forecasting in power networks, such as renewables forecasting [31] and load forecasting [14, 30]. Since the costs of making erroneous forecasts are so high, even reducing forecast error in a few percent points are important [13]. Various methods have been applied and evaluated in load forecasting problems, including using nonparametric regression [7], support vector regression [6], ARIMA [12] and neural networks [9, 15]. Among these forecast models, neural network has become increasingly more popular, as it provides highly accurate results due to the ability of representing the complex relations between high-dimensional features and outputs.

The recent progress in deep learning and data science also promotes the use of deep neural networks and more complicated feature representations in forecast models [8, 10, 20]. Many works focus on feature selection and feature engineering by considering the uncertainties coming from both electrical loads [39] and exogenous variables such as weather [17, 38], customer behaviors [32] etc. However, most research doesn't look into the robustness issues, and model performances under adversarial environements are rarely discussed [11, 23].

Our work is also under the scope of cyber-physcial system security, especially the cyber-security of power systems [25]. Many studies focused on compromising the communication, sensing or monitoring process in modern smart grids [26, 34]. For instance, *denial of service attacks* and *deception attacks* are aimed at compromising either communication channel or communication packets [2]; false data injections on state estimation have been widely discussed [21, 22], where the attackers introduce estimation errors on state variables, e.g., phase angles and voltage magnitudes. Such attacks strategically manipulate meter measuresments to bypass conventional bad data detection. In [36, 40], the authors analyzed how maliciously changed system states could affect the market operations during dispatch process. Most of the previous attacks assume full knowledge of system configuration. It is also assumed

[1] https://github.com/chennnnnyize/load_forecasts_attack

that attackers possess strong capabilities to implement attacks, e.g., to compromise communication channel or to modify meter data arbitrarily.

In this paper, we focus on the previously overlooked vulnerabilities in the load forecasting process. For instance, forecasting model inputs can be exposed to adversarial modification and the model performance may be impacted by such malicious changes. Recently, there has been a hot debate on the security of machine learning models [35] following the deep learning's state-of-the-art achievements on a bunch of benchmark tasks. In computer vision, researchers found a small, adversarially designed noises injected to clean image would deceive a well-trained image classifier [3, 29]. We are interested in whether such attacks could also impact the performance of load forecasting models and if so to what extent. The proposed class of data injection attacks do not assume the forecasting model itself is known to the attackers. In addition, successful distortion on load forecasting also impacts the reliable operations of power systems, so it is important to investigate the data vulnerabilities in existing load forecasting methods.

3 FORMULATION: FORECASTERS AND ATTACKERS

In this section, we formally describe the forecasting and attacking models. To set up realistic vulnerability analyses, we also describe the set of restrictions on the knowledge and capability of the attacker.

3.1 Load Forecasting Formulation

The schematic of general load forecasting model is depicted in Figure 1. We consider the setup for a family of load forecasting algorithms with different architectures. The input features of these algorithms include historical records of load, weather forecasts including temperature, weather indicators (e.g., sunny, rainy or cloudy) and seasonal indicator variables such as weekdays/weekends and hour of the day. Mathematically, the system operator would be able to collect a training dataset $\mathcal{D}_{tr} = \{(\mathbf{X}_{t-H}, ..., \mathbf{X}_t); L_{t+k}\}_{t=1}^{T}$ based on available historical data. Here $L_{t+k} \in [0, 1]$ are scalars representing scaled load values (or *response variables*) [14]. k is the model's forecast horizon, typically ranging from one hour to one day in short-term load forecasts. $\mathbf{X}_{t-i} \in [0, 1]^d,\ i = 0, ..., H$ are scaled, d-dimensional input feature verctors (or *numerical predictor variables*). Let's denote $\mathbf{X}_t := \{L_t, \mathbf{X}_t^{temp}, \mathbf{X}_t^{index}\}$, where L_t are the load history records; $\mathbf{X}_t^{temp}$ are the temperature value vectors, which could be acquired from either system historical records or weather forecast API; $\mathbf{X}_t^{index}$ are a collection of indicators, indicating the weather characteristics, seasonal factors and time factors. H determines how much history of training data the operators want to take into consideration for forecasting. Longer history would provide more information to the forecast model, yet brings more difficulty in model training and fitting.

In the task of load forecasting, one is interested to find a function parameterized by θ: $f_\theta(\mathbf{X}_{t-H}, ..., \mathbf{X}_t) = \hat{L}_{t+k}$, which learns the mapping from $(\mathbf{X}_{t-H}, ..., \mathbf{X}_t)$ to future loads $\hat{L}_{t+k}$. The mean absolute error (MAE) is widely used to measure the performance of forecasting algorithm, which is defined by the average L_1 norm of difference on forecasted loads. Estimation of θ is given by minimizing the L_1-norm of the difference between model predictions and ground truth values:

$$\min_{\theta} \quad \frac{1}{T}\sum_{t=1}^{T} ||f_\theta(\mathbf{X}_{t-H}, ..., \mathbf{X}_t) - L_{t+k}||_1 \tag{1a}$$

$$s.t. \quad \theta \in \Theta \tag{1b}$$

During training, ground truth of historical records on $\mathbf{X}_t$ and L_{t+k} are used; during testing and real-world system implementations, we are using $\mathbf{X}_t$ which are coming from weather forecasts to forecast future loads. Once the model is learned, it can be applied in a rolling-horizon fashion to make use of forecasted $\hat{L}$ along with $\mathbf{X}_t^{temp}$ and $\mathbf{X}_t^{index}$ to forecast for furthur into the future.

3.2 Specific Forecasting Models

We describe the model setup for several representative load forecasting algorithms which have achieved good performances and have been widely adopted [15]. In Appendix A we detaild the model parameter settings and training approaches. We note that the vulnerabilitiy analysis conducted by this paper is not constrained to the following forecasting algorithms. As long as the model output is sensitive with respect to input features, our proposed attack methods would be able to alter the load patterns maliciously.

3.2.1 Feed-Forward Neural Networks. A multi-layered, feed-forward neural networks (NN) has been widely used to represent the nonlinearities between input features and output forecasts [15]. For the input layer of neural networks, each neuron represents one feature of training input, and all features of past H steps $(\mathbf{X}_{t-H}, ..., \mathbf{X}_t)$ are stacked as the inputs. For each intermediate layer, NN could have a tunable number of hidden units, which represent the input feature combinations. Recent advances in deep learning also allow for deeper and more comlicated network design [8].

3.2.2 Recurrent Neural Networks. A recurrent neural networks (RNN) is a class of neural networks that are specially designed for sequential modeling [37]. Instead of stacking all time steps' features together as in the feed-forward neural networks, RNN feeds each step's input $\mathbf{X}_t$ sequentially, and outputs a hidden unit to represent the feature combination of current input and historical features. The last neuron outputs the forecasted load values.

3.2.3 Long Short-Term Memory. Long Short-Term Memory network (LSTM) is designed to deal with the vanishing gradient problem existing in the RNN with long-time dependencies [20]. The major improvements over RNN are the design of "forget" gates to model the temporal dependencies and capture long time dependencies in load patterns more accurately.

3.3 Objective of Attacker

The attacker's goal is to distort the forecasted load as much as possible in a certain direction, e.g., to either increase or decrease forecasted values. In order to distort the output forecast values, the attacker actually has two choices of inserting attacks: *to attack* $\boldsymbol{X_t}$ or *to attack* θ. While the trained model itself is often safely kept by the operators, it has to use external data such as weather forecasts

$\mathbf{X}_t^{temp}$ as input features. Then the attacker's goal is to inject perturbations into the weather forecasts coming from external services to generate adversarial input data $\tilde{\mathbf{X}}_t^{temp}$ for $f_\theta(\cdot)$, so that predictions are modified. We use $\gamma = \{-1, 1\}$ to denote the chosen attack direction by attackers. If $\gamma = 1$, the attacker tries to find $\tilde{\mathbf{X}}$ to decrease the load forecast values; when $\gamma = -1$, the attacker tries to find $\tilde{\mathbf{X}}$ to increase load forecasts values. Since load values are always positive, the attacker's goal is to find $\tilde{\mathbf{X}}$ that minimizes the value of $\gamma f_\theta(\tilde{\mathbf{X}}_{t-H}, ..., \tilde{\mathbf{X}}_t)$.

3.4 Attacker's Knowledge

We consider two attack scenarios, *white box* and *black-box* attacks. In the *white-box* settings, the attacker is assumed to know exactly the model parameters θ. This is a strong assumption in the sense that load forecast model $f_\theta(\cdot)$ is fully exposed to the attacker. On the contrary, in the *black-box* setting, the attacker only knows which family of load forecasting model has been applied (e.g., NN or RNN), but is blind to the forecasting algorithms and has no knowledge of any parameters of f_θ. We consider two possible avenues of attacks. In the first case, the attacker possesses a *substitute training dataset* $\mathcal{D}'_{tr}$ which may or may not be the same as $\mathcal{D}_{tr}$. Such dataset also represents the ground truth of historical load and features. In the second case, the attacker cannot acquire such dataset due to lack of access to the historical load records. We assume the attacker has *query* access to the load forecasting model[2]. That is, the attacker could query the implemented load forecasting model by using different values of input features for a limited number of times, and then try to get insights on how f_θ works.

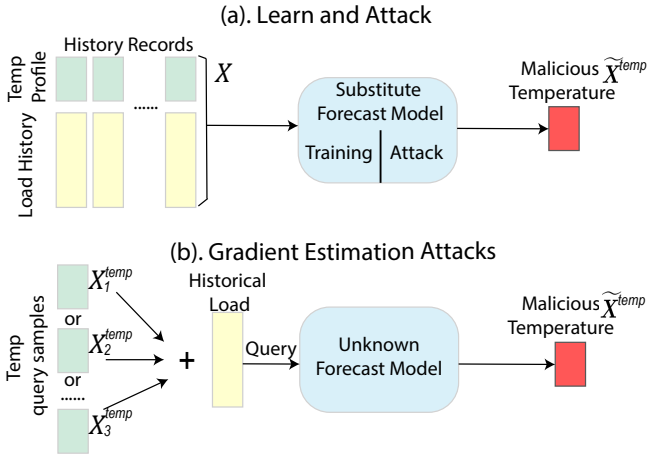

Figure 2: Schematics for two proposed attacks on load forecasting models by changing input temperature vectors: (a). the `learn-and-attack` algorithm and (b). the `gradient estimation` algorithm.

[2]Such query access assumption is possible in many *forecast-as-a-Service* businesses, e.g., SAS energy forecasting and Itron forecasting.

3.5 Attacker's Capability

As an attacker, it is important to avoid being detected by the bad data detection algorithms used by system operators. The attacker's capability could be upper bounded by the allowed number of perturbed entries in the input data; it could be bounded by the average deviations on all features; or it could be also bounded by the largest deviation from the original value. Mathematically, the attacker wants to keep $||\tilde{\mathbf{X}}_t^{temp} - \mathbf{X}_t^{temp}||_p$ bounded, where p can take different values such as $0, 1, \infty$ to express certain norm constraints corresponding to different detection algorithms.

In summary, we formulate the model of attackers as the following optimization problem:

$$\min_{\tilde{\mathbf{X}}_{t-H}^{temp}, ..., \tilde{\mathbf{X}}_t^{temp}} \quad \gamma f_\theta(\tilde{\mathbf{X}}_{t-H}, ..., \tilde{\mathbf{X}}_t) \tag{2a}$$

$$s.t. \quad ||\mathbf{X}_{t-i}^{temp} - \tilde{\mathbf{X}}_{t-i}^{temp}||_p \le \epsilon, \ i = 0, ..., H \tag{2b}$$

Note that there is a parallel between the forecast problem (1) and attack problem (2), where the objective's optimization directions and optimization variables are exactly in the opposite directions: forecasting model works on model parameters to minimize forecast errors, while attacker works on model inputs to maximize the errors to targeted directions. However, due to lack of model knowledge in the black box setting, it is a challenging task for attackers to find efficient attack input $\tilde{\mathbf{X}}^{temp}$ via (2). In the next section, we will show two attack methods generally working with attacker's knowledge coming from *substitute training dataset* and *query access* respectively.

4 BLIND ATTACK ON LOAD FORECASTING

In this section, we first describe attacks under the *white-box* setting, where an attacker possesses full knowledge of load forecasting model parameters. This serves as a benchmark for evaluation of the success of attackers. We then focus on two more realistic settings where the attacker does not know the model parameters. We describe how data injection attacks can be implemented when either the historical data is known or the attacker has limited query access to the load forecasting model.

4.1 White-Box Attack

Under the *white-box* setting, since the model parameters are known to the attacker, it is possible to find the attack input via solving (2). For the convenience of notations, we omit the superscript on $\mathbf{X}$ in some of the following paragraphs, and introduce the generalizable attack methods not only suitable for attacking temperature forecasts, but also suitable for injecting false data into other features.

Since most state-of-the-art load forecasting algorithms use complex models such as neural networks, the attacker's problem (2) is nonconvex and furthurmore, there is no closed-form solution for $\tilde{\mathbf{X}}_{t-H}, ..., \tilde{\mathbf{X}}_t$. Nevertheless, an attacker can still find some attack vectors iteratively by taking gradients with respect to each time step's temperature values. Even though this may not find the optimal solution to (2), because of the highly nonconvex nature of the forecasting model, a slight (suboptimal) perturbation of the input features would drastically change the forecast output.

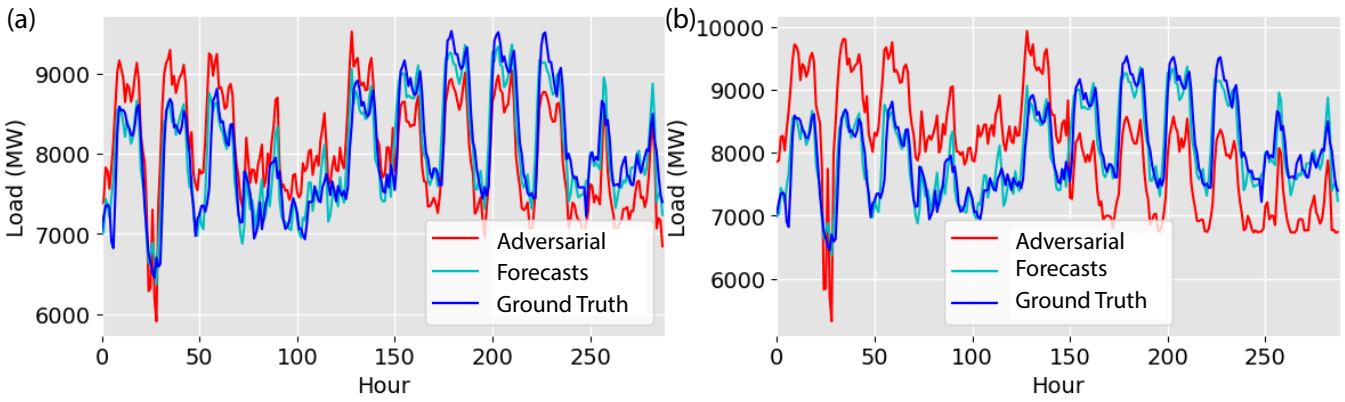

Figure 3: We show 300 hours forecasts based on original and adversarial temperature data for the aggregated load of Switzerland. The load forecasting algorithn is an recurrent neural networks with inputs composed of past load, regional temperature forecast values and weather indicators. The attack perturbations are generated by using the `learn and attack` method, and it implements load maximization strategy in the first 150 hours and load minimization strategy in the latter 150 hours. (a). Load forecasting results with temperature attack constraint of (maximum perturbations) $1F$; (b). load forecasting results with temperature attack constraint of $5F$.

Based on (2), we define a loss function $\mathcal{L}$ with respect to each time step's feature $\tilde{\mathbf{X}}_{t-i},\ i = 0, ..., H$. Then the attacker iteratively takes gradients of $\mathcal{L}$ to find the adversarial input $\tilde{\mathbf{X}}_{t-i}$. The constraints in (2b) is included in the loss function using a log-barrier:

$$\mathcal{L}(\tilde{\mathbf{X}}_{t-i}) = \gamma f_\theta(\tilde{\mathbf{X}}_{t-H}, ..., \tilde{\mathbf{X}}_t) - \beta \log(\epsilon - ||\mathbf{X}^{temp}_{t-i} - \tilde{\mathbf{X}}^{temp}_{t-i}||_p) \quad (3)$$

where β is the weight of the barrier term. Since there are a large number of parameters and input features in many load forecasting algorithms, it can be computationally expensive to compute the exact gradient values for each input feature. We follow a simpler method in [35] to only update the feature values based on the sign of the gradient at each iteration j:

$$\tilde{\mathbf{X}}^{(j+1)}_{t-i} = \tilde{\mathbf{X}}^{(j)}_{t-i} - \alpha \cdot \text{sign}(\nabla_{\tilde{\mathbf{X}}^{(j)}_{t-i}}(\mathcal{L}(\tilde{\mathbf{X}}^{(j)}_{t-i}))) \quad (4)$$

where α controls the step size for updating adversarial temperature values. The resulting adversarial temperature vector is obtained by applying (4) a number of times.

4.2 Learn and Attack

In the `learn and attack` setting, we assume the attacker does not have access to the model parameters, and there is no query access to the model. The only knowledge the attacker has is a historical dataset $\tilde{\mathcal{D}}_{tr}$, which includes same features as data set $\mathcal{D}_{tr}$ used to train the load forecasting model[3]. The proposed attack algorithm consists of a *training phase* and an *attack phase* as shown in Fig. 2(a). In the training phase, the attacker trains substitute model $f_{\tilde{\theta}}$ based on $\tilde{\mathcal{D}}_{tr}$ to minimize the training loss. In the attack phase, the attacker pretends that the substitute model is the true load forecast model and performs white-box attacks on it to find the attack vectors. This strategy is based on the assumption that the substitute model behaves similarly to the true model not only for the training data $\mathbf{X}$, but also for the attack vector $\tilde{\mathbf{X}}$. Then by injecting $\tilde{\mathbf{X}}$ into the true load forecasting model, the forecast values go to attacker's desired directions.

It is useful to evaluate the *transferability* of proposed attacks across different set of models f_θ and $f_{\tilde{\theta}}$. The phenomenon of transferability in adversarial attacks for machine learning models have been discussed in [18, 28], where adversarial instance generated using $f_{\tilde{\theta}}$ can be also treated as an adversarial instance by f_θ with high probability. The theoretical understanding of why attacks transfer remains an open question and is out of scope for this paper. In Fig. 3 we show such *transferability* also exists in the load forecasting model. The temperature inputs are generated by implementing the iterative gradient update (4) based on a substitute model under L_∞-norm of attack perturbations, yet such adversarial temperature values also mislead the (unknown) true load forecasting model to be wildly inaccurate.

4.3 Gradient Estimation Attack

To implement `learn and attack` on load forecasting algorithms, the attacker needs get a version of the training data to learn a substitute load forecasting model. In the case there is no available historical data records, if the attacker is able to query the load forecasting algorithm for a limited number of times, it is still possible to construct adversarial temperature inputs by using queries to estimate the gradients. In Figure 2 (b) we show the schematic on generating adversarial temperature instances via querying.

For k-th dimension of the input feature at time stamp $t - i$, $\tilde{\mathbf{X}}^{j+1}_{k,t-i}$, the attacker needs to query the load forecasting system on each feature dimension to calculate the two-sided estimation of the gradient of f_θ:

$$\nabla_{\tilde{\mathbf{X}}_{k,t-i}} f_\theta(\tilde{\mathbf{X}}) \approx \frac{f_\theta(\tilde{\mathbf{X}} + \delta \mathbf{e}_k) - f_\theta(\tilde{\mathbf{X}} - \delta \mathbf{e}_k)}{2\delta} \quad (5)$$

where $\mathbf{e}_k$ is a d-dimensional vector with all zero except 1 at k-th component, and δ takes a small value for gradient estimation.

[3]In Learn and Attack setting, we make assumption that the attacker know the family of targeted load forecasting model, e.g., a feedforward neural networks or a Recurrent Neural Networks.

Once the gradient is estimated for each dimension of temperature features, we can follow the same method of (4) to iteratively build the adversarial features using the estimated gradient vectors:

$$\tilde{\mathbf{X}}_{t-i}^{(j+1)} = \tilde{\mathbf{X}}_{t-i}^{(j)} - \alpha\gamma \cdot \text{sign}(\nabla_{\tilde{\mathbf{X}}_{t-i}} f_\theta(\tilde{\mathbf{X}}^{(j)})) \tag{6}$$

To satisfy the norm constraints on the allowed perturbation of $\tilde{\mathbf{X}}$, the attacker projects the adversarial data back into the predefined norms after finishing the iterative attack constructions. In [3] techniques on reducing number of queries are also discussed for attacking an image classifier, which could also help improve the query efficiency of load forecasting attacks.

5 ATTACKS ON SYSTEM OPERATIONS

In this section, we first illustrate a power system operation case consisting of a day-ahead planning stage and a real-time operational stage, which is simple yet close to real-world market operations. We then describe two simple temporal attack strategies that pose threats to such system operations via injecting perturbations into load forecasting inputs.

5.1 Power System Operations Model

(1) A commitment schedule based on the day-ahead load forecasts is created by a unit commitment (UC) model based on the day-ahead load forecast:

$$\min_{\mathbf{u},\mathbf{p}} \; C(\mathbf{p}) + S(\mathbf{u}) \tag{7a}$$

$$\text{s.t.} \sum_{g\in G} p_g^t = \hat{L}_t, \;\; \forall t \in T \tag{7b}$$

$$u_i^t p_g^{\min} \le p_g^t \le u_g^t p_i^{\max}, \;\; \forall g \in G, \;\; \forall t \in T \tag{7c}$$

$$u_g^t - u_g^{t-1} = z_g^t - y_g^t, \;\; \forall g \in G, \;\; t \in T \tag{7d}$$

$$\sum_{\tau=t-t_g^{\text{up}}+1}^{t} z_g^\tau \le x_g^t, \;\; \forall g \in G, \forall t \in T \tag{7e}$$

$$\sum_{\tau=t-t_g^{\text{dn}}+1}^{t} z_g^\tau \le 1 - x_g^t, \;\; \forall g \in G, \forall t \in T \tag{7f}$$

$$-R_g^{\text{dn}} \le p_g^{t+1} - p_g^t \le R_g^{\text{up}}, \;\; \forall g \in G \tag{7g}$$

$$u_g^t, z_g^t, y_g^t \in \{0,1\}, \;\; \forall g \in G, \;\; t \in T \tag{7h}$$

where u_g^t is the binary decision variable of the commitment status of generator g at time t, with 1 indicating g is online; p_g^t is the real power output of generator g at time t; all the u_g^t's and p_g^t's are collected together into vectors $\mathbf{u}$ and $\mathbf{p}$; $C(\mathbf{p})$ and $S(\mathbf{u})$ represent the dispatch costs and startup and shutdown costs, respectively, of all the generators in all periods; the constraints are system-wide power balance constraint (7b), generation limits constraints (7c), generator logical constraint (7d), minimum up time constraint (7e), minimum down time constraint (7f) and ramping constraints (7g). Once solved, the operator gets the schedule for the set of online generators G_t at each time t.

(2) For each time stage t of each day, the dispatch of the scheduled units and the actual dispatch cost are calculated according to a basic Economic Dispatch (ED) model [19] based on the actual load and generation schedule G_t:

$$\min_{\mathbf{p}_t} \; C(\mathbf{p}_t) \tag{8a}$$

$$\text{s.t.} \sum_{g\in G_t} p_g^t = L_t, \tag{8b}$$

$$p_g^{\min} \le p_g^t \le p_g^{\max}, \;\; \forall g \in G_t \tag{8c}$$

where it aims to find the real power dispatch at time t, $\mathbf{p}_t$, that minimizes the dispatch costs at time t, $C(\mathbf{p}_t)$, considering system-wide power balance constraint (8b) and generation limits constraints (8c). The daily operation cost is obtained by summing the 24-hour dispatch costs and the startup and shutdown costs. When the ED based on the day-ahead commitment does not have a feasible solution, a load is shed to maintain the balance between supply and demand.

5.2 Attack Strategies

Under normal operating conditions, the load forecasting algorithms provide accurate forecasts on day-ahead load for system operators to solve (7). During an attack, adversarial temperature forecasts are injected into the day-ahead planning stage to cause deviation from the normal operations, e.g., increased system costs, load shedding, no feasible generation dispatch or violation of ramping constraints. We assume the attacker does not know the parameters of underlying system such as each generator's capacity and ramp constraints.

We propose two intuitive attack strategies that move the load forecasts as far away as possible to stress the system. Simple as it is, the toy example in Section 5.3 and case studies in Section 6 using real-world load data reveal the potential vulnerabilities brought by these types of load forecasting attacks.

5.2.1 Load Maximization. Under this strategy, the attacker increases the load forecasts as much as possible. Then with an overestimation of the system loads at each time step in the day-ahead stage, the operator tends to turn on more than necessary generation units, which will increases the system operation costs.

5.2.2 Load Minimization. Under this strategy, the attacker decreases the load forecasts as much as possible. Then in day-ahead planning stage, the system operator underestimates the future load, and fewer generators are scheduled than needed. If the real load is not too much higher than the adversarial load, the system can still use spinning reserve to satisfy the underestimated loads, but could cause expensive dispatch. If the real load exceeds the available capacity, load shedding could take place.

5.3 Toy Example

To illustrate why such simple attack strategies would cause an increase of system costs and occurrences of operations anomalies, we show an toy example here with 2 generators of same capacity serving an aggregate load. We demonstrate four possible unexpected cases in Figure 4. In the simplifying case, we consider a 4-step forecast and unit commitment. For ease of illustration, we are still assuming there exist ramp constraints and capacity constraints in the toy example, but no minimum up and down time constraints. In Figure 4(a) and 4(d), the attacker drives the forecasts lower than the real loads, and we observe either the actual load exceeds the

scheduled generator's generation capacity, or actual ramp exceeds the scheduled generator's ramping capacity. In Figure 4(b) and 4(c), the attacker either increases the peak load forecasts, or keeps the forecast larger than actual load. Both cases cost the system to keep one more generator online for some time, and dispatch the load in an uneconomical way.

There are other possible attack strategies, such as changing forecasts to random directions, shifting the peak load, cutting the forecast peak load or decreasing the forecasted ramp magnitudes. All these attacks could bring economic and operational damages to the system, but should need more specific design based on the specific load profile, temporal patterns and may require more knowledge of the system.

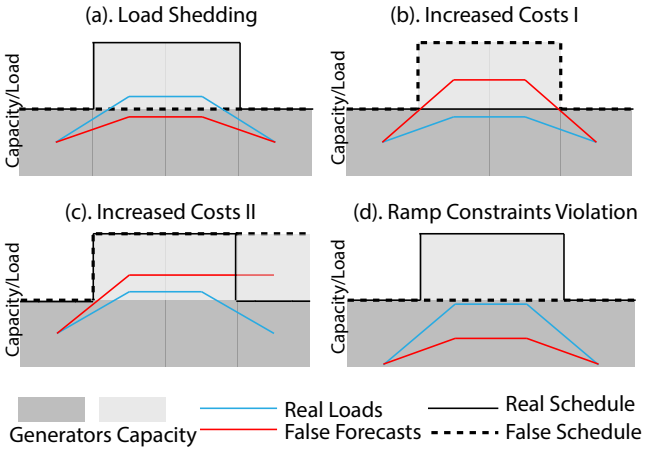

Figure 4: Four cases illustrating consequences by using falsified load forecasts data in a toy example of power system operations. Two generators with same capacity are scheduled for 4 time steps' operations.

5.4 Key Insights

For the general power system operations including a planning stage (unit commitment) and a real-time operational stage (economic dispatch), we observe the following characteristics of impacts by adversarial load forecasts:

- Increasing the load maliciously will normally incur extra system costs, such as starting to operate redundant generators, using more expensive generation combinations and etc;
- By decreasing the peak load maliciously, system operators would ignore the real peaks of future loads, and schedule fewer generators. This would potentially cause load shedding or failing to follow the severe ramps in the actual load patterns;
- We assume an attacker with constrained capability on modifying the input features for load forecast models, and with no knowledge about the system parameters such as generator schedule and load forecasting model parameters. The proposed attack could be even more detrimental if the attacker possesses extra knowledge of the system and implement targeted attack during certain time periods.

6 CASE STUDIES

In this section, we show a detailed simulation on real-world Swiss load data, and show the threats posed by our data injection attacks in several ways. In particular, we first illustrate the proposed attacks could degrade a set of accurate load forecasting algorithms dramatically; we then quantitatively evaluate the damages brought to the system operations, and compare the results with the case using clean data for load forecasting. We demonstrate that attackers with little efforts and knowledge are able to cause load shedding or infeasible dispatch.

6.1 Experimental Setup Description

Dataset Description: We collected and queried hourly actual load data from European Network of Transmission System Operators for Electricity(ENTSO-E)'s API[4] ranging from Jan 1st, 2015 to May 16th, 2017, and we followed [24] to collect day-ahead historical weather forecasts coming from major cities in Switzerland such as Zurich, Basel, Lucerne and etc. All the weather data were queried from Dark Sky API [5]. We also collect other indicator features $\mathbf{X}^{index}$, such as one-hot vectors of hour of day, day of week (weekend or weekday), and season of year. We split 80% of data as our training sets, and use the remaining 20% of data on validating and evaluating the load forecasting prediction accuracy, attack performance and case studies on market operations. Note that even though we collected offline data to train and validate both of our load forecasting and attack models, these data collection procedures could be applied in an online fashion so that attacker could inject real-time adversarial attacks into certain load forecasting models.

Power Systems Setup: The system has 1 aggregated load for Switzerland based on the ENTSO-E data. The nominal load values are in the range of $[6,500MW,\ 9,500MW]$. We take a simplified power system model of using 7 generators with total capacity of $11,900MW$, and omit the network constraints. We adopt the generator parameter settings of ramp capacity, generation costs and minimum on/off time based on [19]. We set the spinning reserve requirement as 3% of the total forecasted demand based from [33]. During the run of day-ahead unit commitment, either normal day-ahead forecasts or adversarial forecasts are used for generation scheduling; during the run of economic dispatch, the real loads are used for generation dispatch. The models of UC and ED are implemented in Python using PyPSA [5], and these two modules are directly interfaced with the load forecastling and attack algorithms. Note that even in our simiulated system, it ignores the line constraints, while the attacker does not know any information about the system operation, we already observe a set of damages posed by load forecasting attack. We expect more severe effects of attacks with either more generation constraints or less attack constraints.

Model Training and Attack Implementation: We set up three load forecasting models, NN, RNN and LSTM respectively, and use standard stochastic gradient descent methods for model training [4]. For detailed model setup and training, we refer to Appendix.A. All three forecast methods could get similar converged validation errors, and as shown in the first column of Table 1, the errors in mean absolute percentage error (MAPE) are comparable to the errors

[4] https://transparency.entsoe.eu/
[5] https://darksky.net/forecast/47.3769,8.5414/us12/en

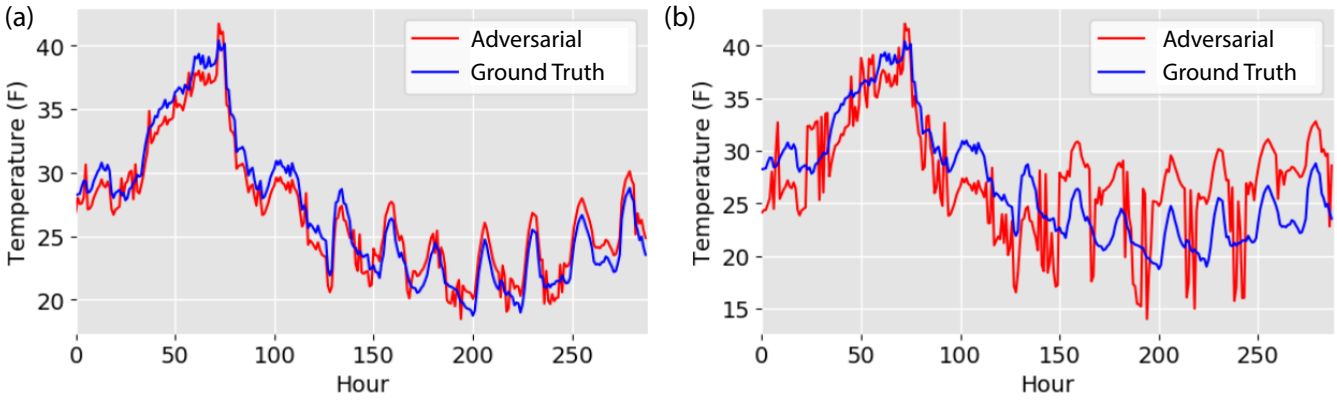

Figure 5: We show 300 hours of original temperature forecasts and adversarially perturbed temperature using same data as shown in Figure 3. (a)Temperature with maximum attack constraint of $1F$; (b). temperature with maximum attack constraint of $5F$.

Forecasts Error (MAPE)	Clean Data	Learn and Attack	Gradient Estimation
NN	1.68%	12.72%	13.09%
RNN	1.58%	9.82%	11.68%
LSTM	1.51%	9.04%	11.87%

Table 1: Forecasts errors evaluated on clean test data and adversarial data for 3 different forecast models. Allowed maximum perturbations are $4F$.

reported in several recent studies on load forecasting [8, 20]. We save the model parameters and keep them away from black-box attackers. For the substitute model training of `learn and attack` method, we keep the training set $\tilde{\mathcal{D}}$ same as the load forecasting model training set $\mathcal{D}$. Decreasing the size of $\tilde{\mathcal{D}}$ or using different substitute dataset could decrease the performance of `learn and attack`. We use L_∞ constraints on the attacker's capability (2b), such that the attacker is constrained by the maximum deviation of perturbed temperature values. We validate trained model's performance under attacks with varying constrained values. For details of training techniques, training accuracies, training and attack implementation time, we refer to Appendix A and Appendix B.

6.2 Load Forecasting Performance

We calibrate and compare the load forecasting model performance with and without adversarial attacks on test datasets. Though all three models exhibit good performances on clean test data, we inject different level of perturbations generated by `learn and attack` and `gradient estimation` methods respectively, and found the forecasting performance decrease drastically as the adversarial perturbations become larger (Table 1). In Figure 3 we show the RNN's load forecasting results for 300 hours using `learn and attack` algorithm with maximum perturbation on temperature of $1F$ and $5F$. The attacker tries to increase the load in the first 150 hours, and decrease the load in the latter hours. We observe that the algorithm

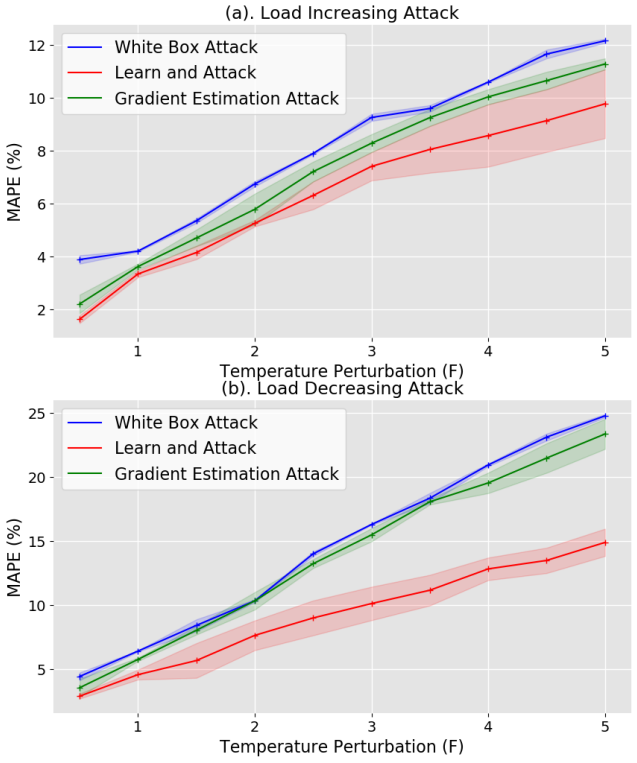

Figure 6: The forecast MAPE under (a). attacks to increase the load; and (b). attacks to decrease the load. Simulations are run for three times with different random seeds, and shaded area denotes the variance.

finds the correct attack direction to either increase or decrease the load. What's more, with only $1F$ deviation on temperatures, the load forecasts changes over 500MW at some time steps. When the attacker increases the perturbation to $5F$, larger forecasts error over 1, 200MW are observed. The temperature profile before and

after attack still looks similar, which could avoid system operators' security inspection (Figure 5). Table 1 compares all three load forecasting models' performance using clean and adversarial data. For both `learn and attack` and `gradient estimation` algorithms, they distort all three load forecasting models' output and increase model's forecast error. `Gradient estimation` attack works generally better for all three models, and this is due to estimating the gradients via querying f_θ directly is more accurate than calculating it from the substitute model and transferring to f_θ.

In Figure 6, we evaluate RNN's load forecasting performance under two attack strategies: load maximization or load minimization. We observe `gradient estimation` attack causes similar MAPE compared to white box attack. The load decreasing attack is normally more successful than load increasing attack in terms of MAPE. Load minimization attack is more harmful results than load increasing ones, since increased forecasts only let system operators start up more generations, while adversarially decreasing the forecasted load leads to wrong generation decisions that fails to meet the larger real load.

6.3 Impact of Attacks on Operation Costs

As mentioned earlier, we are interested in the possible consequences caused by wrong forecasts. We first analyze the increased costs caused by adversarial forecasts. We implement `learn and attack` algorithm on 3 weeks' random selected test data to increase the forecasted load at each time step. Under such circumstances, the system operator sets day-ahead generator schedule based on adversarial loads larger than actual loads. In Figure 7 we show the bar plot of increased costs versus varying perturbation on temperature forecasts. When the temperature perturbations are small, increased costs are limited, and such increments are mostly due to extra start-up costs. When perturbation becomes larger, system operators sometimes derive totally different unit commiment schedule to accomodate higher loads and larger ramps, so in some days we observe larger increase in system costs, of which values are $4-5$ times of nomial hourly operating costs.

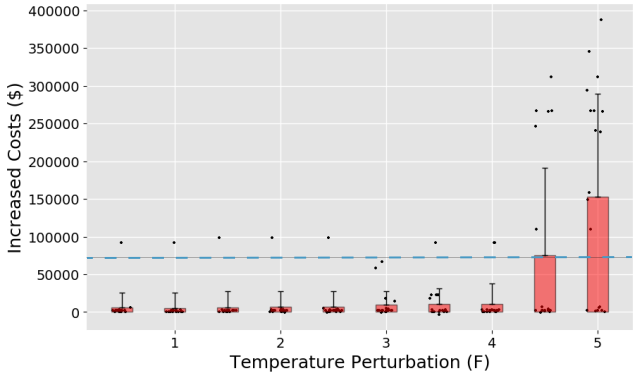

Figure 7: With different level of malicious perturbations injected into temperature features, wrong forecasts increase system operation costs. The nomial hourly operating costs is around $72,000 (blue dashed line).

Occurrences (Number of Days)	Learn and Attack	Gradient Estimation
Load Shedding	27.14%	31.43%
Ramp Constraints Violation	90.0%	85.71%

Table 2: The occurrence frequencies of load shedding and ramp constratins violation in 10 weeks' system operation test. Both attackers are allowed to inject $4F$ perturbations on temperature features.

6.4 Impacts of Attacks on Feasibility

In addition to increasing system costs, adversarial attacks on load forecasting could even lead to worse situations. We illustrate a load minimization strategy that leads to infeasible solutions (e.g., load shedding, ramp constraint violations) to the economic dispatch problem. We implement both `learn and attack` and `gradient estimation` algorithms with maximum perturbation of $4F$, and test the results on 10 weeks' load data. In Table 2, we note the occurrence frequencies of both load shedding and ramp constraints violation. Since $4F$ change in temperature forecasts can lead to over $1,000$MW decreasing on load forecasts, system operators tend to keep fewer generators on. This leads to many days' generation capacity fall short of the load, and the scheduled generators can not fulfill the large ramps in real load profiles. In Figure 8 we show two examples on this two kind of failures respectively. In Figure 8(a), during peak hours, the adversarial load forecasts let the system operator schedule one $1,500MW$ generator off compared to the case of correct forecasts. Even taking the spinning reverse during the day-ahead unit commitment, the actual load at the mid of the day exceeds the adversarial load by over $1,000MW$ and the total load exceeds the generator capacity. In Figure 8(b), the actual loads are increasing rapidly at hour 5 and 6, yet the adversarials load profile flattens such ramps, and cause the scheduled generators incapable of meeting the large ramp. We expect more frequent violations of ramp constraints if the attacker specifically design attack strategies based on the load patterns.

7 DISCUSSION AND CONCLUSION

In this paper, we studied the potential vulnerabilities generally existing in many load forecasting algorithms. Such vulnerabilities have been overlooked by the development of many forecasting techniques. We design two attack algorithms which do not require much knowledge about the forecast algorithms, but lead to large increase in forecast errors with adversarial data injections in load forecasting input features. The proposed attack adversarilly manipulate the load forecasting values either to increase or decrease, and thus provide system operators wrong information on future demands. Experiments on real-world load datasets demonstrate such threats over power system operations. Such threats model along with damage analysis indicate that there need more security evaluations in the design and implementation of load forecasting algorithms. In order to mitigate the damages brought by such false data injection attacks, countermeasures in building robust load forecasting algorithms are strongly recommended, which may include

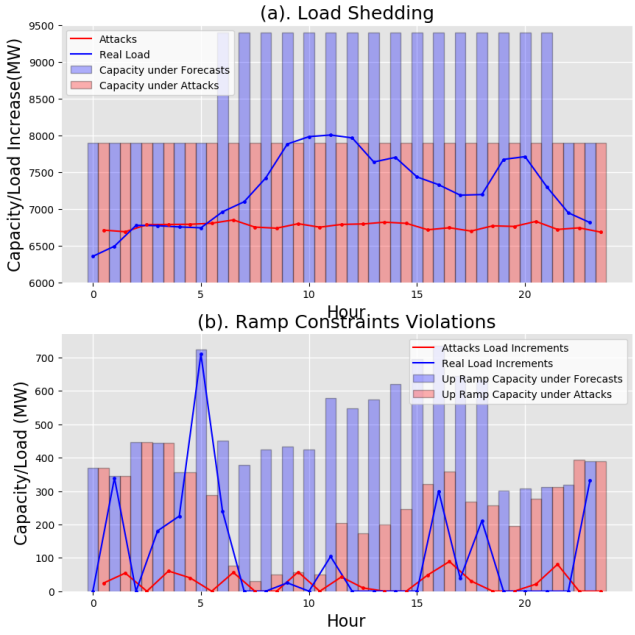

Figure 8: (a). An example showing that forecasts under attack would cause load shedding when real loads exceed total generation capacity; bars indicate generators' total capacity. (b). An example showing that forecasts under attack would cause violation on ramp constraints during economic dispatch; bars indicate generators' available total up-ramp capabilities. Maximum allowed perturbations are $4F$.

anomaly detection techniques considering input data distribution as well as other robust statistics.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

[2] Saurabh Amin, Alvaro A Cárdenas, and S Shankar Sastry. 2009. Safe and secure networked control systems under denial-of-service attacks. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 31–45.

[3] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical Black-box Attacks on Deep Neural Networks using Efficient Query Mechanisms. In *European Conference on Computer Vision*. Springer, Cham, 158–174.

[4] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[5] T. Brown, J. Hörsch, and D. Schlachtberger. 2018. PyPSA: Python for Power System Analysis. *Journal of Open Research Software* 6, 4 (2018). Issue 1. https://doi.org/10.5334/jors.188 arXiv:1707.09913

[6] Ervin Ceperic, Vladimir Ceperic, Adrijan Baric, et al. 2013. A strategy for short-term load forecasting by support vector regression machines. *IEEE Transactions on Power Systems* 28, 4 (2013), 4356–4364.

[7] W Charytoniuk, MS Chen, and P Van Olinda. 1998. Nonparametric regression based short-term load forecasting. *IEEE transactions on Power Systems* 13, 3 (1998), 725–730.

[8] Kunjin Chen, Kunlong Chen, Qin Wang, Ziyu He, Jun Hu, and Jinliang He. 2018. Short-term Load Forecasting with Deep Residual Networks. *IEEE Transactions on Smart Grid* (2018).

[9] Ying Chen, Peter B Luh, Che Guan, Yige Zhao, Laurent D Michel, Matthew A Coolbeth, Peter B Friedland, and Stephen J Rourke. 2010. Short-term load forecasting: similar day-based wavelet neural networks. *IEEE Transactions on Power Systems* 25, 1 (2010), 322–330.

[10] Yize Chen, Yuanyuan Shi, and Baosen Zhang. 2017. Modeling and optimization of complex building energy systems with deep neural networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1368–1373.

[11] Yize Chen, Yushi Tan, and Deepjyoti Deka. 2018. Is Machine Learning in Power Systems Vulnerable?. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 1–6.

[12] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. 2003. ARIMA models to predict next-day electricity prices. *IEEE transactions on power systems* 18, 3 (2003), 1014–1020.

[13] Jan G De Gooijer and Rob J Hyndman. 2006. 25 years of time series forecasting. *International journal of forecasting* 22, 3 (2006), 443–473.

[14] George Gross and Francisco D Galiana. 1987. Short-term load forecasting. *Proc. IEEE* 75, 12 (1987), 1558–1573.

[15] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. 2001. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems* 16, 1 (2001), 44–55.

[16] Benjamin F Hobbs, Suradet Jitprapaikulsarn, Sreenivas Konda, Vira Chankong, Kenneth A Loparo, and Dominic J Maratukulam. 1999. Analysis of the value for unit commitment of improved load forecasts. *IEEE Transactions on Power Systems* 14, 4 (1999), 1342–1348.

[17] Tao Hong, Pu Wang, Anil Pahwa, Min Gui, and Simon M Hsiang. 2010. Cost of temperature history data uncertainties in short term electric load forecasting. In *Probabilistic Methods Applied to Power Systems (PMAPS), 2010 IEEE 11th International Conference on*. IEEE, 212–217.

[18] Hossein Hosseini, Yize Chen, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Blocking transferability of adversarial examples in black-box learning systems. *arXiv preprint arXiv:1703.04318* (2017).

[19] Daniel S Kirschen and Goran Strbac. 2018. *Fundamentals of power system economics*. John Wiley & Sons.

[20] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. 2017. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid* (2017).

[21] Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. 2010. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 220–225.

[22] Yao Liu, Peng Ning, and Michael K Reiter. 2011. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)* 14, 1 (2011), 13.

[23] Jian Luo, Tao Hong, and Shu-Cherng Fang. 2018. Benchmarking robustness of load forecasting models under data integrity attacks. *International Journal of Forecasting* 34, 1 (2018), 89–104.

[24] Daniel L Marino, Kasun Amarasinghe, and Milos Manic. 2016. Building energy load forecasting using deep neural networks. In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 7046–7051.

[25] Patrick McDaniel and Stephen McLaughlin. 2009. Security and privacy challenges in the smart grid. *IEEE Security & Privacy* 3 (2009), 75–77.

[26] Yilin Mo and Bruno Sinopoli. 2009. Secure control against replay attacks. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 911–918.

[27] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[28] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 372–387.

[30] Dong C Park, MA El-Sharkawi, RJ Marks, LE Atlas, and MJ Damborg. 1991. Electric load forecasting using an artificial neural network. *IEEE transactions on Power Systems* 6, 2 (1991), 442–449.

[31] Pierre Pinson, Christophe Chevallier, and George N Kariniotakis. 2007. Trading wind generation from short-term probabilistic forecasts of wind power. *IEEE Transactions on Power Systems* 22, 3 (2007), 1148–1156.

[32] Franklin L Quilumba, Wei-Jen Lee, Heng Huang, David Yanshi Wang, and Robert L Szabados. 2015. Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities. *IEEE Trans. Smart Grid* 6, 2 (2015), 911–918.

[33] Yann Rebours and Daniel Kirschen. 2005. What is spinning reserve. *The University of Manchester* 174 (2005), 175.

[34] Siddharth Sridhar, Adam Hahn, Manimaran Govindarasu, et al. 2012. Cyber-Physical System Security for the Electric Power Grid. *Proc. IEEE* 100, 1 (2012), 210–224.

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks.

arXiv preprint arXiv:1312.6199 (2013).
[36] Song Tan, Wen-Zhan Song, Michael Stewart, Junjie Yang, and Lang Tong. 2018. Online data integrity attacks against real-time electrical market in smart grid. *IEEE Transactions on Smart Grid* 9, 1 (2018), 313–322.
[37] J Vermaak and EC Botha. 1998. Recurrent neural networks for short-term load forecasting. *IEEE Transactions on Power Systems* 13, 1 (1998), 126–132.
[38] Pu Wang, Bidong Liu, and Tao Hong. 2016. Electric load forecasting with recency effect: A big data approach. *International Journal of Forecasting* 32, 3 (2016), 585–597.
[39] Yi Wang, Ning Zhang, Qixin Chen, Daniel S Kirschen, Pan Li, and Qing Xia. 2018. Data-driven probabilistic net load forecasting with high penetration of behind-the-meter PV. *IEEE Transactions on Power Systems* 33, 3 (2018), 3255–3264.
[40] Le Xie, Yilin Mo, and Bruno Sinopoli. 2010. False data injection attacks in electricity markets. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 226–231.

A DETAILS ON LOAD FORECASTING ALGORITHMS

We set up all load forecasting models using Tensorflow [1] package in Python. Standard model architectures such as Dropout layers and nonlinear activation functions (e.g., ReLU or Sigmoid functions) are adopted in the deep learning models [27]. Since all three networks are set up to solve the load forecasting regression problem, we set the first layer having most neurons, and decrease the number of units in subsequent layers.

Forecasts Models	NN	RNN	LSTM
Number of Layers	4	3	3
Training Epochs	20	30	30
Hidden Units in First Layer	512	64	64

Table 3: Model architectures and training configurations for load forecasting algorithms used in the simulations.

As shown in Figure 9, all three model's loss are converged during training, and we use the trained model in the subsequent planning and operation problem as well as the testbed for attack algorithms. Plots are showing the mean and variance during 3 runs with different random seeds.

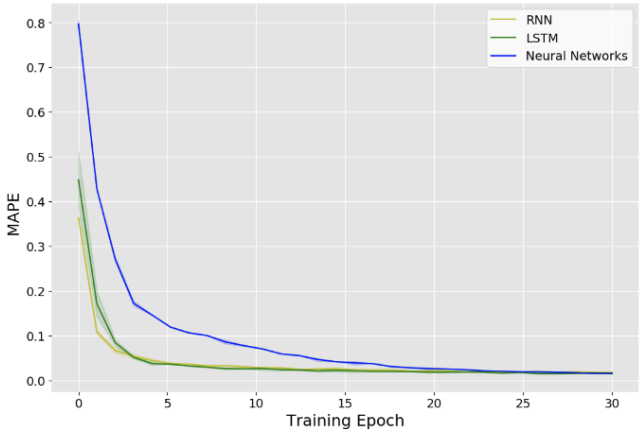

Figure 9: All three forecasting models, show convergence of forecast error on validation data as training evolves. Shaded areas show the variance of MAPE.

B COMPUTATION TIME

We recorded the computation time for neural network training and the implementation time for two proposed attack algorithms. All time are recorded on a laptop with Intel 2.3GHz Core i5-8259U 4 Cores CPU and 8 GB RAM. The training time for NN, RNN and LSTM are calculated for 20, 30 and 30 epochs respectively. The implementation time for the attacks are averaged over all test instances. We could observe that `learn and attack` approach takes longer time than `gradient estimation` due to the longer time taken to calculate gradient signs over the whole neural networks; and as LSTM includes more complicated model architectures, it takes longer time to find the adversarial instance. Yet compared to the long model training time and application scenarios of day-ahead forecasts, the attacker is still efficient enough to find the adversarial perturbations.

Forecasts Models	NN	RNN	LSTM
Training Time	12.988	47.998	143.830
`Learn and Attack`	0.133	0.157	0.579
`Gradient Estimation Attack`	0.082	0.119	0.253

Table 4: Computation time (in seconds) for load forecasting model training and implementation time for attacks.