Design of Load Forecast Systems Resilient Against Cyber-Attacks

Carlos Barreto(✉) and Xenofon Koutsoukos

Vanderbilt University, Nashville, USA
{Carlos.A.Barreto,Xenofon.Koutsoukos}@vanderbilt.edu

Abstract. Load forecast systems play a fundamental role the operation in power systems, because they reduce uncertainties about the system's future operation. An increasing demand for precise forecasts motivates the design of complex models that use information from different sources, such as smart appliances. However, untrusted sources can introduce vulnerabilities in the system. For example, an adversary may compromise the sensor measurements to induce errors in the forecast. In this work, we assess the vulnerabilities of load forecast systems based on neural networks and propose a defense mechanism to construct resilient forecasters.

We model the strategic interaction between a defender and an attacker as a Stackelberg game, where the defender decides first the prediction scheme and the attacker chooses afterwards its attack strategy. Here, the defender selects randomly the sensor measurements to use in the forecast, while the adversary calculates a bias to inject in some sensors. We find an approximate equilibrium of the game and implement the defense mechanism using an ensemble of predictors, which introduces uncertainties that mitigate the attack's impact. We evaluate our defense approach training forecasters using data from an electric distribution system simulated in GridLAB-D.

Keywords: Security · Machine learning · Power systems · Load forecast · Game theory

1 Introduction

Load forecast systems play a fundamental role in the operation of power systems, because utilities and generators need estimations of the future loads to plan their operation. For example, the utilities procure (or sell) energy in electricity markets based on estimations of the future demand. The relevance of forecast systems will increase due to uncertainties coming from new technologies (e.g., renewable generation, electric vehicles, and smart appliances); however, these technologies also introduce vulnerabilities.

Some works have demonstrated that *false data injection* (FDI) attacks, which manipulate sensor readings, can induce errors in state estimation systems of

T. Alpcan et al. (Eds.): GameSec 2019, LNCS 11836, pp. 1–20, 2019.
https://doi.org/10.1007/978-3-030-32430-8_1

power grids, affecting the system's operation [14,29]. An adversary can design attacks to damage the system or to change the electricity market prices. Likewise, an adversary can manipulate the forecast system exploiting vulnerabilities of artificial intelligence models [1,4].

In this work, we assess the vulnerabilities that forecast systems introduce in electricity markets. We focus on forecast models based on *artificial neural networks* (NNs) that accept as inputs the historical measurements from some sensors (e.g., power sensors and thermometers). We consider an adverse generator who injects a bias in some measurements to induce errors in the forecast. Unlike other works, this adversary must choose its strategy taking into account that the attack will affect future predictions that use the biased measurements.

We model the strategic interaction between the defender and the attacker as a Stackelberg game, in which the defender decides first the prediction scheme and the attacker chooses afterwards its attack strategy [9]. In this case, the defender chooses randomly the sensor measurements to use in the forecast. A near optimal defense strategy consists in selecting each sensor's measurements with the same probability. With this strategy the defender reduces the number of compromised sensors used in the prediction.

We find some practical limitations implementing the proposed defense strategy, due to the large strategy space. In particular, since the defender selects randomly the sensor measurements, the number of possible models grows exponentially with respect to the number of sensors. For practical reasons, we propose an approximate implementation of the defense mechanism using a ensemble of prediction models.

The defense strategy can fail if the ensemble becomes more sensitive to the attacks than the original model. This can happen because each model in the ensemble makes predictions using less sensors; therefore, an attack with fewer resources can still create a significant deviation in the predictions. We find that the ensemble becomes more resilient when its models predict a fraction of the total load.

We evaluate our defense approach training forecasters to predict the future load of an electric distribution system simulated in GridLAB-D. Our simulation includes both residential and commercial loads, which have appliances, such as heating, ventilation, and air conditioning (HVAC) systems, water heaters, pool pumps, among others.

The paper is structured as follows. In Sect. 2 we introduce a model of the electricity market and explain how an adverse generator can manipulate the sensor readings to profit. Section 3 presents our methodology to design resilient forecasters. In this section we introduce the game between the adversary and the defender and find an approximated equilibrium. Section 4 presents a way to implement the optimal defense policy using a small number of models, without losing its efficacy. In Sect. 5 we validate our approach with some experiments. Finally, in Sect. 6 we comment on related work and we conclude the paper in Sect. 7.

2 System Model

In this section we introduce an electricity market model and show how load forecasts affect the profit of both utilities and generators. Then we explain how an adverse generator can manipulate the sensor readings to profit. Also, we quantify the consequences of an attack, that is, the costs for the utility and the benefits for the adversary.

2.1 Electricity Markets

In general, electricity systems use markets as mechanisms to allocate resources efficiently. The electricity markets, unlike other markets, need an operator who guarantees that the system's equilibrium (allocation of resources) satisfies the system's physical constraints.[1] Some power systems use two markets, namely the *day-ahead market* (DAM) and the *real-time market* (RTM) [22].

The DAM accepts bids for the next day and produces commitments for demand and generation. The commitments reduce uncertainties of demand, which allows the system's operator to schedule generators with anticipation. However, unexpected events may change the production capacity or the needs for energy. The RTM complements the DAM correcting periodically imbalances between demand and generation, preventing frequency deviations that may damage components with thigh operational limits. These adjustments translate into trades settled at the price of the RTM [16].

The market participants must fulfill the agreements from both the day-ahead and the real-time markets. For example, buyers must pay sellers the price agreed in the DAM; however, if a generator fails to supply energy, then the system operator has to buy energy in the RTM. Likewise, if a customer uses less resources, then the system operator sells the excess in the RTM.

Why Load Forecasting Is Important?. Customers usually do not perceive changes in prices because they pay a flat tariff to retailers, who serve as their intermediaries in the markets. Hence, the retailers deal with the risk of uncertain market prices, that is, they buy energy at variable prices in the market and sell it at a fixed price to their customers. For this reason, the retailers try to reduce uncertainties by forecasting the demand of their customers [25].

2.2 Load Forecasting

Utilities and generators use *short-term load forecasting* (STLF), which ranges from hours to weeks, to adjust their bids in electricity markets. Shorter forecast horizons help to control the power flow, while long-term forecasts help to plan the operation and the system expansion [10].

[1] These constraints prevent damage to the equipment and the environment. For example, generators may have operational limitations to prevent emissions or to regulate the use of water (for hydroelectric plants) [16].

Here we consider a forecaster that uses past sensor measurements of loads and the weather to predict the total demand during a future time period. In particular, the forecaster uses the measurements available a time t to estimate the demand at time $t+\tau$, where τ is the forecast horizon.

Let us denote with $\mathcal{M} = \{1, \ldots, m\}$ the set of sensors, which have measurements $l_k(t)$ at time $t = 1, 2, \ldots, T$, with $k \in \mathcal{M}$. Each measurement $l_k(t)$ corresponds to average values during a period of one hour. Furthermore, we denote with $y(t)$ the total demand at time t. Thus, the prediction problem consists in finding a function $f(\cdot)$ that uses sensor measurements available at time t to estimate the future demand $y(t+\tau)$.

We denote with the vector $x_k(t) = [l_k(t-1), \ldots, l_k(t-H-1)]$ the historical measurements of the k^{th} sensor available at time t. In this case we use H past samples to estimate the future load. Moreover, we denote with the vector $X(t) = [x_k(t-\tau)]_{k\in\mathcal{M}}$ the whole historical data at time t.

Remark 1. In this case we build a forecaster using load and temperature measurements; however, we assume that the adversary manipulates only the load measurements.

Here we use a NN to estimate $y(t)$ as a function of the historical data $X(t)$. The estimated demand is

$$\hat{y}(t) = f(X(t), w^*) = f(X(t)),$$

where the vector w^* represents the weights of the NN that minimizes an error metric (loss function) $l(\cdot)$, that is,

$$w^* \in \arg\min_w l(y, f(X, w)).$$

Hence, the prediction error at time t is

$$\varepsilon(t) = y(t) - \hat{y}(t).$$

In general, nonlinear distance metrics are more sensitive to outliers, since large errors in individual samples have a larger impact. Hence, the *mean squared error* (MSE) is more sensitive to outliers than the *mean absolute error* (MAE) [12] (we illustrate this in Sect. 5). For this reason, we choose MAE as loss function, that is,

$$l(y, \hat{y}) = \frac{1}{T} \sum\nolimits_{t=1}^{T} |y(t) - \hat{y}(t)|. \tag{1}$$

Forecast's Economic Impact. Recall that the utility uses load forecasts to choose its bids, which in turn create commitments in the electricity market. In our model the utility purchases the estimated load $\hat{y}$ in the DAM and trades the demand imbalance (estimation error) ε in the RTM. Hence, the utility pays

$$\Omega_u(y, \hat{y}) = \sum\nolimits_{t=1}^{T} \left\{ \hat{y}(t) p^{DA}(t) + \varepsilon(t) p^{RT}(t) \right\}, \tag{2}$$

where p^{DA} and p^{RT} represent the price in the DAM and RTM, respectively. On the other had, we model the profit of generators (revenues minus generation costs) as

$$\Omega_g(y, \hat{y}) = \Omega_u(y, \hat{y}) - C(y), \tag{3}$$

where $C(\cdot)$ represents the generation cost. For simplicity, we formulate the generation cost as a function of the total energy produced y. However, in practice the trades in each market can affect the generation costs.

2.3 Adversary Model

According to Eq. (3), the generators can profit from estimation errors that increase the utility's cost Ω_u. In particular, we consider a cyber-attack that injects false data in the sensor measurements and transforms them as

$$l_k^a(t) = l_k(t) + b_k(t),$$

where the $b_k(t)$ represents the bias in the k^{th} sensor at time t. Likewise, the historical data of the k^{th} sensor becomes $x_k^a(t) = x_k(t) + \varphi_k(t)$, where

$$\varphi_k(t) = [b_k(t-1), \ldots, b_k(t-H-1)]. \tag{4}$$

We denote the total historical data manipulated by an adversary as $X_a(t) = X(t) + B_a(t)$, where the vector

$$B_a(t) = [\varphi_k(t-\tau)]_{k \in \mathcal{M}} \tag{5}$$

represents the bias observed by the forecast model. An attack with bias $B_a(t)$ transforms the load forecast as

$$\hat{y}_a(t) = f(X(t) + B_a(t)) \approx \hat{y}(t) - \delta(B_a, t),$$

where $\delta(B_a, t)$ denotes the impact of the attack (the deviation from the original prediction), which satisfies $\delta(0, t) = 0$. Therefore, the net prediction error becomes

$$y(t) - \hat{y}_a(t) \approx y(t) - \hat{y}(t) + \delta(B_a, t) = \varepsilon(t) + \delta(B_a, t). \tag{6}$$

Impact of the Attack. From Eqs. (2) and (6), the utility's cost with an attack is

$$\Omega_u(y, \hat{y}_a) \approx \sum\nolimits_{t=1}^{T} \left\{ \hat{y}_a(t) p^{DA}(t) + (\varepsilon(t) + \delta(B_a, t))\, p^{RT}(t) \right\}.$$

Hence, the benefit of generator is

$$\Omega_g(y, \hat{y}_a) - \Omega_g(y, \hat{y}) \approx -\sum\nolimits_{t=1}^{T} \delta(B_a, t)(p^{DA}(t) - p^{RT}(t)). \tag{7}$$

The precise goal of the attack depends on the price difference between the DAM and the RTM. The next result shows some conditions in which the adversary benefits by either increasing or decreasing the forecasts.

Lemma 1. *Assume that* $\delta(B_a)$ *and* $p^{DA} - p^{RT}$ *are independent random variables. If either* $\mathrm{E}[\delta(B_a)] \leq 0$ *and* $\mathrm{E}[p^{DA} - p^{RT}] \geq 0$ *or* $\mathrm{E}[\delta(B_a)] \geq 0$ *and* $\mathrm{E}[p^{DA} - p^{RT}] \leq 0$*, then the adversary profits from the attack.*

In the remainder of the paper we assume that $\frac{1}{T}\sum_t p^{DA}(t) \leq \frac{1}{T}\sum_t p^{RT}(t)$; hence, the adversary seeks to induce under-estimations of the future load ($\frac{1}{T}\sum_t \delta(B_a, t) \geq 0$). For this reason, we formulate the adversary's objective as

$$\begin{aligned} \underset{[b_k]_{k=1}^m}{\text{maximize}} \quad & \frac{1}{T}\sum_{t=1}^{T} |\hat{y}(t) - \hat{y}_a(t)| = \frac{1}{T}\sum_{t=1}^{T} |\varepsilon(t) + \delta(B_a, t)| \\ \text{subject to:} \quad & Eq.\,(4) \\ & Eq.\,(5) \\ & \sum_{t=1}^{T} \delta(B_a, t) \geq 0 \\ & b_k(t) = 0 \text{ if } k \notin \mathcal{M}^a \end{aligned} \tag{8}$$

We measure the impact of the attack for the defender as the damage in the forecast's accuracy. The defender's loss function (see Eq. (1)) with an attack becomes

$$l(y, \hat{y}_a) = \frac{1}{T}\sum_{t=1}^{T} |y(t) - \hat{y}_a(t)| \approx \frac{1}{T}\sum_{t=1}^{T} |\varepsilon(t) + \delta(B_a, t)|. \tag{9}$$

Thus, the defender and the attacker pursue opposite goals.

Remark 2. A FDI attack may have broader consequences, since other forecasters can use the same historical data with different purposes. For example, the system operator may calculate reserves based on load predictions. Thus, under-estimations in the future load may expose the system to both failures and other attacks.

2.4 Attack Capabilities and Restrictions

We make the following assumptions about the attack:

1. The adversary knows both the forecast system (or estimates it [4,8,28]) and samples of the historical measurements. With this information the adversary can find an attack that solves Eq. (8).
2. The attack does not depend on the current state of the system. This can occur if the adversary cannot read the sensor measurements in real time or if it is unable to use such information to compute the bias.
3. The prices' distribution do not change with the attack; hence, the adversary's goal does not change after the attack.
4. The adversary compromises a subset of sensors $\mathcal{M}^a \subseteq \mathcal{M}$, with $m^a = |\mathcal{M}^a|$. Hence, $b_k(t) = 0$ for the sensors $k \notin \mathcal{M}^a$.
5. The adversary injects the same bias in all the compromised sensors. Hence, $b_k(t) = b_j(t) = b(t)$ for all $k, j \in \mathcal{M}^a$.

6. The number of sensors compromised is the main variable that determines the impact of an attack.
7. The impact of the attack $\delta(\cdot)$ is concave increasing with respect to the number of sensors compromised. Intuitively, the attacker may experience diminishing returns in its attacks, that is, the impact increases with the number of sensors compromised, but the growth rate decreases with each additional sensor. Likewise, we assume that forecasters that use the same number of sensors have the same impact function.

The adversary must design its attack considering its future effects, because the utility uses the biased measurements during H periods. In this case, the adversary can leverage the periodicity of the load to design a successful attack. In particular, the loads follow a 24 h period determined by the daily habits of the consumers. Likewise, the forecaster also has some periodicity, because it uses H samples in its estimation. Thus, the adversary can manipulate the sensors to report periodically the same bias, that is, $b_k(t) = b_k(t + H)$.

3 Resilient Forecasting

The defense problem consists in designing a forecast system using data from untrusted sources. Here we consider the possibility of mitigating the impact of attacks by introducing randomness in the system, which in turn creates uncertainties for the attacker. In particular, we analyze the efficacy of building forecast models using randomly selected sensor measurements. Intuitively, uncertainties in the system's model can reduce the success of the adversary, who has to design the attack considering possible contingencies.

3.1 Game Formulation

We model strategic interaction between the defender and the attacker as a Stackelberg game, where the defender decides first the prediction scheme and the attacker chooses afterwards its attack strategy [9].

Strategies. In this case, the defender chooses the probability of using the k^{th} sensor $\rho_k^d \in [0, 1]$, for $k \in \mathcal{M}$, which satisfy $\sum_{k\in\mathcal{M}} \rho_k^d = m^d$. The above condition implies that the forecaster uses in average m^d sensors. We denote the defense strategy with the vector $\rho^d = [\rho_k^d]_{k\in\mathcal{M}}$. Likewise, we represent the strategy of the adversary with the vector $\rho^a = [\rho_k^a]_{k\in\mathcal{M}}$, where $\rho_k^a \in [0, 1]$ denotes the probability of attacking the k^{th} sensor. In this case, the adversary compromises at most m^a sensors in average; hence, $\sum_{k\in\mathcal{M}} \rho_k^a \leq m^a$. Let us denote with $\mathcal{M}^d$ and $\mathcal{M}^a$ the sets of sensors selected by the defender and the attacker, respectively. Thus, the set $\mathcal{M}^c = \mathcal{M}^d \bigcap \mathcal{M}^a$ contains the compromised sensors that the defender uses in the prediction.

Let W_k be a Bernoulli random variable with success probability $\rho_k = \rho_k^d \rho_k^a$. In other words, W_k describes whether both the defender and the adversary select

the k^{th} sensor. Hence, the number of compromised sensors (attacked sensors used in the forecast) is

$$S_m = \sum\nolimits_k W_k = |\mathcal{M}^c|,$$

where S_m follows the *m-generalized binomial distribution*. Hence, the expected number of compromised sensors is

$$\mathrm{E}[S_m|\rho^d, \rho^a] = \lambda(\rho^d, \rho^a) = \sum\nolimits_{k\in\mathcal{M}} \rho_k^d \rho_k^a.$$

Player's Payoff. Let us express the impact of the attack as a function of the sensors selected by the players ($\mathcal{M}^d$ and $\mathcal{M}^a$) and the bias b

$$y(t) - \hat{y}_a(t) \approx \delta(\mathcal{M}^d, \mathcal{M}^a, b, t).$$

Since we assume that the impact depends only on the number of sensors compromised, then two attacks on the sets $\mathcal{M}^{a_1}$ and $\mathcal{M}^{a_2}$ that satisfy $|\mathcal{M}^d \bigcap \mathcal{M}^{a_1}| = |\mathcal{M}^d \bigcap \mathcal{M}^{a_2}|$ have approximately the same impact,[2] that is,

$$\delta(\mathcal{M}^d, \mathcal{M}^{a_1}, b, t) \approx \delta(\mathcal{M}^d, \mathcal{M}^{a_2}, b, t). \tag{10}$$

Henceforth we denote the impact function as

$$\delta(S_m, t) = \delta(\mathcal{M}^d, \mathcal{M}^a, b^*, t),$$

where b^* represents the optimal attack schedule that solves Eq. (8).

According to Eqs. (8) and (9), the defender's objective consist in reducing the expected impact of the attack, while the adversary attempts to create an error in the prediction. For this reason, we define the payoff of the adversary as

$$\Pi^a(\rho^d, \rho^a) = \mathrm{E}[\delta(S_m)|\rho^d, \rho^a].$$

On the other hand, we define the payoff of the defender as

$$\Pi^d(\rho^d, \rho^a) = -\Pi^a(\rho^d, \rho^a).$$

3.2 Game's Approximate Equilibrium

The equilibrium of the game is the solution to

$$\min_{\rho^d} \max_{\rho^a} \Pi^a(\rho^d, \rho^a). \tag{11}$$

The concavity of the impact with respect to the number of sensors compromised implies

$$\mathrm{E}[\delta(S_m)|\rho^d, \rho^a] \leq \delta\left(\mathrm{E}[S_m|\rho^d, \rho^a]\right) = \delta(\lambda(\rho^d, \rho^a)). \tag{12}$$

[2] Although the impact depends on the particular model, that is, the set $\mathcal{M}^d$, we assume that models with the same number of sensors m^d have the same impact function.

The next result shows that an approximate equilibrium to the game in Eq. (11) comes from the solution to

$$\min_{\rho^d} \max_{\rho^a} \delta(\lambda(\rho^d, \rho^a)). \tag{13}$$

Proposition 1. *Let* (ρ^d, ρ^a) *be the solution Eq.* (13). *Then* (ρ^d, ρ^a) *is a* $\xi-$*equilibrium of the game in Eq.* (11), *that is,*

$$\Pi^d(\rho^d, \rho^a) \geq \Pi^d(\tilde{\rho}^d, \rho^a) - \xi$$

and

$$\Pi^a(\rho^d, \rho^a) \geq \Pi^a(\rho^d, \tilde{\rho}^a) - \xi$$

for some strategies $\tilde{\rho}^d$ *and* $\tilde{\rho}^a$. *and* $\xi \geq 0$.

This means that the players cannot get benefits superior to ξ by adopting another strategy. Moreover, the next result shows that the optimal defense strategy ρ^d for the game in Eq. (13) consists in selecting the sensors with the same probability.

Proposition 2. *The defense strategy* ρ^d *in the equilibrium of Eq.* (13) *satisfies* $\rho^d_k = \frac{m^d}{m}$, *for all* $k \in \mathcal{M}$.

Remark 3. The adversary's optimal strategy consists in targeting the sensors with the highest selection probability. However, when the defender chooses all the sensors with the same probability, then the adversary doesn't have any preference for the sensors.

Properties of the Defense Mechanism. If the defender selects m^d sensors with an uniform distribution, then the expected number of compromised sensors is

$$\lambda(\rho^d, \rho^a) = \frac{m^d}{m} m^a.$$

Thus, the proportion of compromised sensors is $\lambda(\rho^d, \rho^a)/m^d = m^a/m$, which doesn't depend on m^d. In other words, by selecting sensors randomly we reduce the number, but not proportion, of compromised sensors.

We improve the resiliency of the system if the ensemble has a lower impact than the original model; hence, from Eq. (12) we need

$$\delta(\mathcal{M}, \mathcal{M}^a, \tilde{b}, t) \geq \mathrm{E}[\delta(S_m)|\rho^d, \rho^a], \tag{14}$$

where $\tilde{b}$ represent the optimal bias when the forecast model uses all the sensors. The above condition can fail if the ensemble becomes more sensitive to the attacks than the original model.

Besides selecting randomly the sensor measurements, the defender may adjust the training of models to guarantee Eq. (14). In particular, the defender may implement some form of regularization to make the models less sensitive. For example, [20] makes NNs robust against attacks implementing an algorithm equivalent to Lipschitz regularization. In Sect. 5 we explore how the target in the training phase affects the sensitivity of the models.

4 Implementation of the Forecast

From the game formulation, the defender selects randomly m^d sensors and constructs a forecaster. Thus, the defender must either train a new model for each prediction task or train and store the models with anticipation. However, such approaches may require a prohibitively large amount of time and resources, because the defender can build $\binom{m}{m^d}$ different forecasters. For this reason, we approximate the defense strategy constructing an ensemble with n models guaranteeing that they use each sensor's data with probability m^d/m (the desired defense strategy (see Proposition 2)).

Let us partition the set of sensors $\mathcal{M}$ in n sets $\mathcal{P}_i$ of size m/n, where $\bigcup_{i=1}^{n} \mathcal{P}_i = \mathcal{M}$ and $\mathcal{P}_i \bigcap \mathcal{P}_j = \emptyset$ for all $i \neq j$, $1 \leq i, j \leq n$. If $m^d/m \leq 0.5$, then we construct an ensemble of $n = m/m^d$ models, where each model uses the set $\mathcal{M}_i = \mathcal{P}_i$ for its training. In this way, each model uses each sensor with probability m^d/m.

On the other hand, if $m^d/m \geq 0.5$, then we construct n models that use all except one of the subsets. In this case, the i^{th} model uses $\mathcal{M}_i = \bigcup_{j \neq i} \mathcal{P}_j$ sensors. Thus, we select each sensor's data with probability $(n-1)/n$, which must satisfy $\frac{n-1}{n} = \frac{m^d}{m}$ (that is, $n = \frac{m}{m-m^d}$). This means that each partition has size $|\mathcal{P}_i| = m - m^d$ and each model uses $|\mathcal{M}_i| = m^d$ sensors.

When n is not integer, we can still achieve the desired selection probability merging two ensembles. Let us construct two partitions $\{\mathcal{P}_i^a\}_{i=1}^{n_1}$ and $\{\mathcal{P}_i^b\}_{i=1}^{n_2}$ with $n_1 = \lfloor n \rfloor$ and $n_2 = \lceil n \rceil$. With these partitions we can build two ensembles that select each sensor with probability $\gamma_k = \frac{n_k - 1}{n_k}$, for $k = 1, 2$. We can merge the ensembles selecting them with probability $\beta_k \in [0, 1]$ to satisfy $\sum_k \gamma_k \beta_k = \frac{m^d}{m}$, for $k = 1, 2$. In this way, we can construct an ensemble that guarantees that the prediction uses each sensor with probability m^d/m.

Since the ensemble uses models trained beforehand, the adversary my target the sensors of particular forecasters to improve its profit, rather than selecting them randomly. The next result shows that the adversary's optimal strategy consists in allocating its resources equally to all the partitions.

Proposition 3. *Consider an ensemble constructed from a partition* $\{\mathcal{P}_i^a\}_{i=1}^{n}$ *and let* σ_i *be the proportion of resources allocated to the set* $\mathcal{P}_i$*. Then the adversary maximizes the impact selecting* $\sigma_i = \frac{1}{n}$*, which leads to an expected impact*

$$\delta\left(\frac{m^d}{m} m^a\right) = \delta(\lambda(\rho^d, \rho^a)).$$

Remark 4. According to the previous result, our mechanism to implement the ensemble has the same expected impact than the ensemble proposed in Sect. 3. Hence, combining multiple ensembles doesn't improve the forecaster's resiliency, because individually they have the same expected impact.

Remark 5. The previous results hold if the models of the ensemble have the same impact as a function of the number of sensors compromised.

5 Evaluation

In this subsection we examine how some parameters of the forecasters affect their sensitivity to attacks. Based on the results from these experiments we design the ensemble and show its robustness against attacks.

5.1 Experimental Setup

Power System. We make a detailed simulation of an electric distribution system using GridLAB-D and the prototypical distribution feeder models provided by the Pacific Northwest National Laboratory (PNNL) [24]. The distribution models capture fundamental characteristics of distribution utilities from the U.S. In this case, we use the prototypical feeder *R1-12.47-3* that represents a moderately populated area with 109 commercial and residential loads composed by appliances such as heating, ventilation, and air conditioning (HVAC) systems, water heaters, and pool pumps, among others. We simulate the distribution system during summer time (June to August) to build a dataset with measurements of the power consumed by each load and the outdoor temperature.

Forecast Models. We implement each forecaster in Keras [6] using NNs composed by five layers (three layers with 150 Long Short-Term Memory (LSTM) units [11] and two layers with 200 and 100 rectified linear units (ReLU), respectively). We train the NN using Adadelta as optimizer, which adapts the learning rate based on a moving window of gradient updates.

We use as input data X the last $H = 24$ measurements from 110 sensors (109 power sensors and 1 temperature sensor). We train the NNs to estimate the load during the next hour ($\tau = 1$), and we make the predictions every hour. In the experiments we use 80% of the samples to train the forecasters, 10% to determine the attack policy, and 10% to evaluate the impact of the attacks. Figure 1 shows an example of the prediction made with the forecaster.

Design of Attacks. We find the attack schedule solving Eq. (8) using the L-BFGS-B algorithm from [15]. We use the gradient of the forecaster (e.g., the expected gradient of the ensemble) and part of the samples (10%) to find the optimal attack schedule. In other words, the adversary uses the available information about the forecaster and the loads's behavior to design the attack.

Moreover, we make Monte Carlo simulations to assess the impact of the strategy of each player. In particular, we train 20 forecasters reflecting the defense strategy ρ^d. Likewise, for each attack we choose randomly a forecaster and find an attack selecting randomly m^a sensors (we repeat this random selection 20 times).

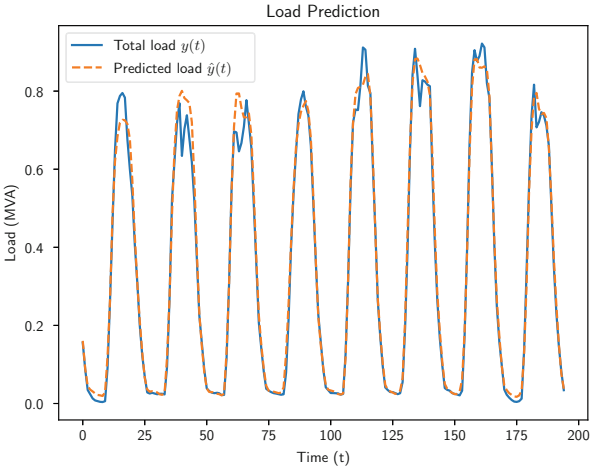


Fig. 1. Example of the load forecast.

5.2 Sensitivity of Forecast Models

Loss Function. Figure 2 shows the expected attack's impact $\delta(B_a, t)$ on models trained with different loss functions, namely the MSE and the MAE. The experiment confirms that the NN trained with MSE suffers a higher impact with the attack. Moreover, the impact is approximately concave with respect to the number of sensors compromised m^a for both MSE and MAE. In the remainder we use models trained with MAE.

Ensemble Training. Now we examine how to design the models of an ensemble guaranteeing that it has a lower impact than a single model (see Eq. (14)). In particular, we experiment with different targets of the models (the value that they learn).

We construct four forecasters $\{f^j\}_{j=1}^4$ with different characteristics. The first forecaster f^1 estimates the total load y using data from m sensors (this is the nominal case). Each one of the remainder forecasters has an ensemble of two models, f_1^j and f_2^j for $j = 2, 3, 4$, trained using half of the sensors ($m^d = 0.5m$) to predict a values y_1^j and y_2^j, respectively. We build the second forecaster with models that estimate the total load; hence, $y_1^2 = y_2^2 = y$ and $f^2 = (f_1^2 + f_2^2)/2$. On the other hand, the models of the third forecaster estimate a fraction of the load with $y_1^3 = y_2^3 = 0.5y$ and $f^3 = f_1^3 + f_2^3$. We define the last forecaster as $f^4 = f_1^4 + f_2^4$, where the models f_1^4 and f_2^4 estimate the total load of their sensors, that is, $y_i^4 = \sum_{k \in \mathcal{M}_i} l_k$.

Figure 3 shows that the model's target affects the sensitivity of the ensembles. In this case, the forecasters f^2 and f^3 suffer a larger impact than the original model f^1, while f^4 succeeds in reducing the impact of attacks (but has a larger prediction error (0.075) than the other forecasters). For this reason, in

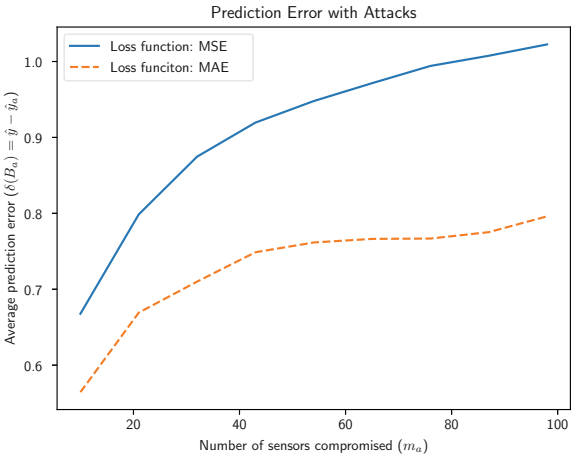


Fig. 2. Attack's impact on models trained with MSE and MAE. The model trained with MAE is more resilient to attacks.

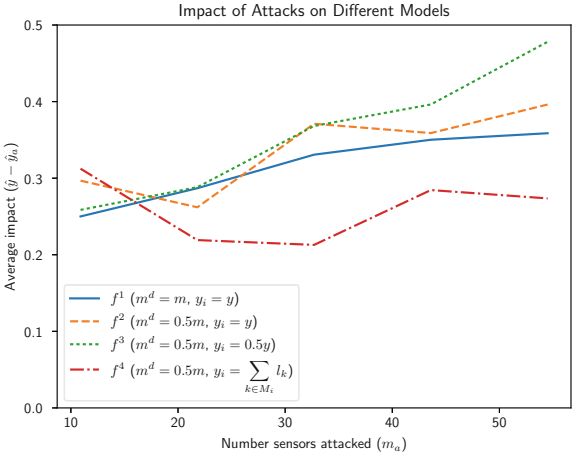
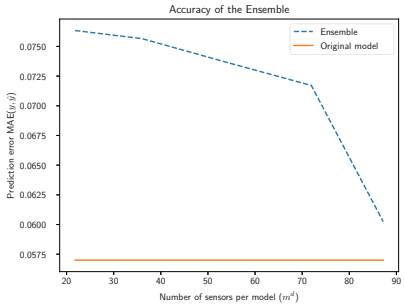


Fig. 3. Impact of an attack on four different forecasters. The training's target y_i affects sensitivity of the ensemble.

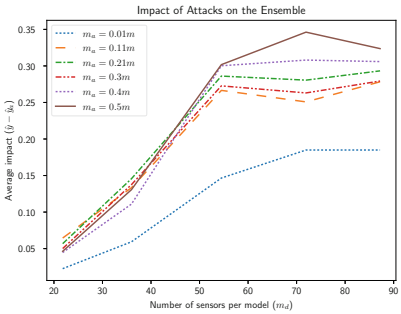
the remainder of the text we train ensembles as f^4, dividing the prediction task among the ensemble's individual models.

5.3 Selection of the Best Ensemble

Now we test the robustness of ensembles using models with different values of m^d. Figure 4a shows that the ensemble's prediction error increases as we reduce the number of sensors used in each model m^d, but it does not increase significantly.



(a) The ensemble has a larger prediction error, which increases as we reduce the number of sensors used in each model m^d.



(b) The impact of the attack is approximately convex with respect to the number of sensors used by each model m^d.

Fig. 4. Prediction error and impact of ensembles with different parameters m^d.

This may happen because as m^d decreases, the forecast tends to estimate the demand of fewer loads, giving the ensemble a greater detail of them.[3] On the other hand, Fig. 4b shows that the impact increases with respect to m^d.

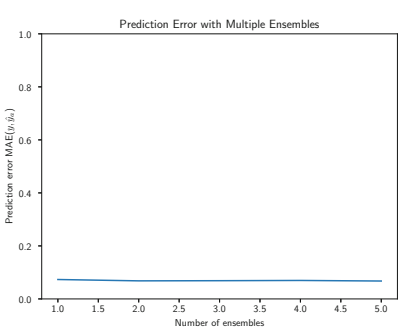
Since both the prediction error and the attack's impact have concave shapes, the value of m^d that minimizes the cost of the attack (see Eq. (9)) falls in one of the extremes, that is, $m^d \in \{1, m\}$. In our particular scenario, $m^d = 1$ attains the lowest cost of the attack (the ensemble that predicts each load individually).

Ensemble Size. Figure 5 shows the impact of attacks as a function of combined ensembles. We train the models selecting randomly $m^d = 0.5m$ models and consider random attacks on $m^a = 0.5m$ sensors. This experiment shows that the number of ensembles (or models) doesn't affect significantly the impact, confirming Remark 4.

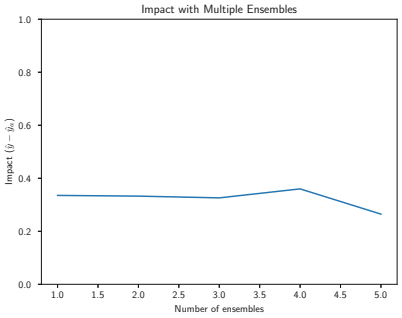
6 Related Work

Previous works have analyzed the vulnerability of CPS against *false data injection* (FDI) attacks, which modify sensor measurements to manipulate the system's operation. The seminal work by Liu et al. [21] considers attacks on sensors that induce errors in the state estimation of power grids. Such errors can affect the system's operation, in particular, the electricity prices. An adversary that manipulates the electricity prices can profit and/or cause damage to the system. This attack requires historical data and real time measurements to calculate the attack.

[3] Other forecast models make predictions using less information (e.g., the aggregate loads); hence, their accuracy decrease significantly with less loads [25].



(a) Average prediction error ϵ.



(b) Impact of an attack with $m^a = 0.5m$.

Fig. 5. Combining multiple ensembles does not improve its accuracy nor its resiliency.

Other works have considered FDI attacks that modify information about the congestion patterns (the rate of the transmission lines) [14,29] and the topology of the power system [5]. Also, the attacks can also target information that consumers use to make decisions, misleading them to take actions that benefit the adversary [2,3]. In most cases, the adversary calculates the attack based on the system's state.

Some works have recognized the vulnerabilities of cyber attacks on forecasting systems. For example, [19] analyzes attacks that manipulate load forecasts to manipulate the *economic dispatch*, which determines the production of each generator based on the estimations of future demand. However, the paper focuses on the consequences of the attack (e.g., how an adversary profits), rather than its precise implementation.

Chen et al. [4] show how an adversary can manipulate the temperature measurements to increase or decrease load forecasts. In particular, the adversary does not need knowledge about the forecasting model (e.g., the precise structure of the NN) or the power system. This and other works (for example [8,28]) show that the adversary can estimate the system's model through either historical data or queries from the forecast system. Our work is closely related to [4]; however, our attack targets load, rather than temperature measurements. Nonetheless, our defense approach can be applied the attack presented in [4].

The research of FDI attacks against NNs analyzes how an adversary can design *adversarial examples* to induce errors in the system's task (e.g., misclassify images) [27]. In particular, the attacks are *transferable* among models. In other words, two models trained independently (even with different data) to perform the same task can suffer from the same attacks [23].

Ilyas et al. [13] explain the existence of adversarial examples due to non-robust-features (patterns in the data that are highly predictive). In other words, models may become sensitive to well-generalizing features of the data. Hence,

attacks that target such features, regardless of the model, can induce errors in the outcome (this also explains why we have transferability of the attacks).

Some papers design robust NNs in image classification applications introducing randomness in the system. However, these approaches differ from ours in the way they introduce the uncertainties. For example, [18] proposes a robust NN borrowing ideas from *differential privacy* (DP). DP randomize computations on databases such that a small change in the data set has a bounded change in the distribution over the outputs. This property guarantees that bounded changes in the input of NN will induce bounded changes in the output, preventing the misclassification of images. On the other hand, [20] prevents gradient based attacks adding noise in the layers of the NN. In this way, a single NN acts as multiple models, which combined conform an ensemble of models. Also, [7] proposes stochastic activation pruning, which removes a subset of the activations (nodes) in each layer to protect pre-trained NN against adversarial examples. This approach resembles *dropout* [26], but the selection is based on the magnitude of the activation. [7] also formulates the interaction between defender and attacker as a zero-sum game, but it does not present the equilibrium of the game.

Some literature on statistics consider the problem of designing robust predictors or estimators [17]. In this case, a robust predictor has a small sensitivity to outliers (e.g., random failures). In other words, it has the capacity to handle disturbances for a wide type of distributions. Thus, the design decisions focus on selecting the loss function, rather than manipulating the data.

In general, nonlinear distance metrics are more sensitive to outliers, since large errors in individual samples have larger impact. Hence, the MSE is more sensitive to outliers than the MAE [12]. Nonetheless, robust models have a cost in terms of efficiency, that is, they may have a larger variance (confidence interval).

7 Conclusions

In this work, we show that an adverse generator can profit by inducing errors in load forecasts of utilities. The adversary with knowledge about the forecast model and historical samples from the sensors can succeed injecting a bias in the sensor measurements.

We model the interaction among defender and attacker using game theory and find a defense strategy that can mitigate the attack's impact. In this case, building forecast models using each sensor's measurements with a fixed probability reduces the number of compromised sensors. However, this strategy may fail if forecasters that use less information become more sensitive to attacks.

Due to the large strategy space, we approximate the defense strategy with an ensemble of predictors. Beside its practical benefits, the ensemble allows us to divide the forecast task, improving the resiliency against attacks. In this case, the ensemble becomes more resilient as its models use less measurements, that is, as it estimates fewer loads. In this way, with a careful selection of the training data we can incorporate uncertainties in regular NNs that help to mitigate the impact of attacks.

Other protection schemes may complement the proposed approach. For example, regularization during the training also can mitigate the attack's impact, because it makes the models less sensitive to deviations in the data.

A Appendix

Proof (Lemma 1). Let $\delta(B_a)$ and $p^{DA} - p^{RT}$ be independent random variables; hence, we can approximate their expected value using a Monte Carlo integration with T terms, that is,

$$\mathrm{E}[\delta(B_a)] = \frac{1}{T}\sum\nolimits_{t=1}^{T} \delta(B_a, t),$$

$$\mathrm{E}[p^{DA} - p^{RT}] = \frac{1}{T}\sum\nolimits_{t=1}^{T} \left\{p^{DA}(t) - p^{RT}(t)\right\}.$$

Now, since two independent random variables X and Y satisfy $\mathrm{E}[XY] = \mathrm{E}[X]\mathrm{E}[Y]$, we can approximate their expected product $\mathrm{E}[\delta(B_a)(p^{DA} - p^{RT})]$ as

$$\mathrm{E}[\delta(B_a)(p^{DA} - p^{RT})] = \frac{1}{T}\sum\nolimits_{t=1}^{T} \delta(B_a, t)\frac{1}{T}\sum\nolimits_{t=1}^{T} \left\{p^{DA}(t) - p^{RT}(t)\right\}.$$

Thus, if either $\sum_t p^{DA}(t) - p^{RT}(t) \geq 0$ and $\sum_t \delta(B_a(t)) \leq 0$ or $\sum_t p^{DA}(t) - p^{RT}(t) \leq 0$ and $\sum_t \delta(B_a(t)) \geq 0$, then the attacker has positive profit (see Eq. (7)).

Proof (Proposition 1). Let us consider the following bounds on the difference between expected impact and its approximation from Eq. (12)

$$\underline{\xi} \leq \delta(\lambda(\rho^d, \rho^a), t) - \Pi^a(\rho^d, \rho^a) \leq \overline{\xi}.$$

Since $\Pi^d(\rho^d, \rho^a) = -\Pi^a(\rho^d, \rho^a)$, then the previous expression implies

$$\Pi^d(\rho^d, \rho^a) \geq -\delta(\lambda(\rho^d, \rho^a), t) + \underline{\xi} \tag{15}$$

and

$$-\delta(\lambda(\tilde{\rho}^d, \rho^a), t) \geq \Pi^d(\rho^d, \rho^a) - \overline{\xi}. \tag{16}$$

Moreover, the solution to Eq. (13), denoted (ρ^d, ρ^a), satisfies the following properties

$$\begin{aligned} \delta(\lambda(\rho^d, \rho^a), t) &\geq \delta(\lambda(\rho^d, \tilde{\rho}^a), t), \\ \delta(\lambda(\rho^d, \rho^a), t) &\leq \delta(\lambda(\tilde{\rho}^d, \rho^a), t), \end{aligned} \tag{17}$$

for some strategies $\tilde{\rho}^d$ and $\tilde{\rho}^a$. Thus, from Eqs. (15) and (17) we have

$$\Pi^d(\rho^d, \rho^a) \geq -\delta(\lambda(\rho^d, \rho^a), t) + \underline{\xi} \geq -\delta(\lambda(\tilde{\rho}^d, \rho^a), t) + \underline{\xi}.$$

Now, using the previous expression with Eq. (16) we obtain

$$\Pi^d(\rho^d, \rho^a) \geq \Pi^d(\tilde{\rho}^d, \rho^a) - \xi.$$

where $\xi = \overline{\xi} - \underline{\xi} \geq 0$. With a similar approach we can show that

$$\Pi^a(\rho^d, \rho^a) \geq \Pi^a(\rho^d, \tilde{\rho}^a) - \xi.$$

Proof (Proposition 2). Since $\delta(\cdot)$ is increasing with respect to the number of sensors compromised, the following holds

$$\max_x \delta(x) = \delta(\max_x x).$$

The previous property can be applied also to minimization problems; hence, we can express the game's equilibrium of Eq. (12) as

$$\min_{\rho^d} \max_{\rho^a} \delta(\lambda(\rho^d, \rho^a)) = \delta\left(\min_{\rho^d} \max_{\rho^a} \lambda(\rho^d, \rho^a)\right)$$

This means that the adversary designs its strategy to maximize the number of compromised sensors, while the defender pursues the opposite goal.

The adversary's optimal strategy consists in attacking the sensors with highest selection probability. Without loss of generality, let $\rho_1^d \geq \rho_2^d \geq \ldots \geq \rho_m^d$. Then, the attack strategy $\rho_i^a = 1$ and $\rho_j^a = 0$ for $1 \leq i \leq m^a$ and $j > m^a$ leads to the following expected number of compromised sensors

$$\lambda(\rho^d, \rho^a) = \sum\nolimits_{i=1}^{m^a} \rho_i^d.$$

Since a different attack strategy cannot increase the number of compromised sensors, this attack strategy is *weakly dominant*.

Given the previous attack strategy, the defender's optimal strategy consists in selecting all the sensors with the same probability

$$\rho_k^d = \frac{m^d}{m}.$$

Observe that any deviation from this strategy increases the number of sensors compromised.

Proof (proposition 3). Here we consider that the adversary compromises m^a sensors. Let σ_i be the proportion of resources allocated to the set $\mathcal{P}_i$. According to Sect. 4, we create a partition of sensors $\{\mathcal{P}_i\}_{i=1}^n$. First, let us consider ensembles trained with sensors in $\mathcal{M}_i = \cup_{j \neq i} \mathcal{P}_j$, for $i = 1, \ldots, n$, where $\frac{n-1}{n} = fracm^d m$. Thus, the total number of compromised sensors used by the i^{th} model amount to $m^a \sum_{j \neq i} \sigma_j = m^a(1 - \sigma_i)$.

Due to the concavity of the impact function, the expected impact on the ensemble satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \delta(m^a(1-\sigma_i)) \leq \delta\left(\frac{1}{n}\sum_{i=1}^{n} m^a(1-\sigma_i)\right) = \delta\left(\frac{n-1}{n} m^a\right) = \delta\left(\frac{m^d}{m} m^a\right).$$

Thus, the allocation that maximizes the impact attains the previous upper bound satisfying $\sigma_i = \frac{1}{n}$, for all $i = 1, \ldots, n$. In other words, the adversary's best strategy consists in allocating its resources uniformly in the partition's sets.

Now, if $\mathcal{M}_i = \mathcal{P}_i$, for $i = 1, \ldots, n$, with $n = \frac{m}{m^a}$, then the expected impact on the ensemble becomes

$$\frac{1}{n}\sum_{i=1}^{n} \delta(m^a \sigma_i) \leq \delta\left(\frac{1}{n}\sum_{i=1}^{n} m^a \sigma_i\right) = \delta\left(\frac{m^a}{n}\right) = \delta\left(\frac{m^d}{m} m^a\right).$$

In this case, the attack strategy that attains the upper bound satisfies $\sigma_i = \frac{1}{n}$. Therefore, the adversary allocates its resources equally in all the sensors in the partition.

In practice, the adversary can compromise at most m^d sensors form each partition. Hence, the optimal attack policy must satisfy $\sigma_i = \min\{1/n, m^d/m^a\}$. When $1/n > m^d/m^a$ the adversary cannot implement its ideal strategy.

References

1. Alfeld, S., Zhu, X., Barford, P.: Data poisoning attacks against autoregressive models. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
2. Amini, S., Pasqualetti, F., Mohsenian-Rad, H.: Dynamic load altering attacks against power system stability: attack models and protection schemes. IEEE Trans. Smart Grid **9**(4), 2862–2872 (2016)
3. Barreto, C., Cardenas, A.: Impact of the market infrastructure on the security of smart grids. IEEE Trans. Ind. Inform. 1 (2018)
4. Chen, Y., Tan, Y., Zhang, B.: Exploiting vulnerabilities of load forecasting through adversarial attacks. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy 2019, pp. 1–11 (2019)
5. Choi, D.H., Xie, L.: Economic impact assessment of topology data attacks with virtual bids. IEEE Trans. Smart Grid **9**(2), 512–520 (2016)
6. Chollet, F., et al.: Keras (2015). https://keras.io
7. Dhillon, G.S., et al.: Stochastic activation pruning for robust adversarial defense. arXiv preprint arXiv:1803.01442 (2018)
8. Esmalifalak, M., Nguyen, H., Zheng, R., Xie, L., Song, L., Han, Z.: A stealthy attack against electricity market using independent component analysis. IEEE Syst. J. **12**(1), 297–307 (2015)
9. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press, Cambridge (1991)
10. Hernandez, L., et al.: A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. IEEE Commun. Surv. Tutor. **16**(3), 1460–1495 (2014)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. Int. J. Forecast. **22**(4), 679–688 (2006)
13. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175 (2019)
14. Jia, L., Thomas, R.J., Tong, L.: Malicious data attack on real-time electricity market. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5952–5955 (2011)
15. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: open source scientific tools for Python (2001). http://www.scipy.org/

16. Kirschen, D.S., Strbac, G.: Fundamentals of Power System Economics. Wiley, Hoboeken (2004)
17. Klebanov, L.B., Rachev, S.T., Fabozzi, F.J.: Robust and Non-robust Models in Statistics. Nova Science Publishers, Hauppauge (2009)
18. Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. arXiv preprint arXiv:1802.03471 (2018)
19. Liu, C., Zhou, M., Wu, J., Long, C., Kundur, D.: Financially motivated FDI on SCED in real-time electricity markets: attacks and mitigation. IEEE Trans. Smart Grid **10**(2), 1949–1959 (2019)
20. Liu, X., Cheng, M., Zhang, H., Hsieh, C.-J.: Towards robust neural networks via random self-ensemble. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 381–397. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_23
21. Liu, Y., Ning, P., Reiter, M.K.: False data injection attacks against state estimation in electric power grids. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 21–32 (2009)
22. Nudell, T.R., Annaswamy, A.M., Lian, J., Kalsi, K., D'Achiardi, D.: Electricity markets in the United States: a brief history, current operations, and trends. In: Stoustrup, J., Annaswamy, A., Chakrabortty, A., Qu, Z. (eds.) Smart Grid Control. PEPS, pp. 3–27. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-98310-3_1
23. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016)
24. Schneider, K.P., Chen, Y., Chassin, D.P., Pratt, R.G., Engel, D.W., Thompson, S.E.: Modern grid initiative distribution taxonomy final report. Technical report, Pacific Northwest National Laboratory (2008)
25. Sevlian, R., Rajagopal, R.: A scaling law for short term load forecasting on varying levels of aggregation. Int. J. Electr. Power Energy Syst. **98**, 350–361 (2018)
26. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
27. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
28. Tan, S., Song, W.Z., Stewart, M., Yang, J., Tong, L.: Online data integrity attacks against real-time electrical market in smart grid. IEEE Trans. Smart Grid **9**(1), 313–322 (2016)
29. Xie, L., Mo, Y., Sinopoli, B.: Integrity data attacks in power market operations. IEEE Trans. Smart Grid **2**(4), 659–666 (2011)