Structural Deep Learning for Guaranteed Attacks with Zero System Knowledge

by

Napoleon Costilla-Enriquez

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2023 by the
Graduate Supervisory Committee:

Yang Weng, Chair
Baosen Zhang
Gautam Dasarathy
Miguel A. Ortega-Vazquez

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

The present outlook in power systems is rapidly changing due to the introduction of (1) new active devices into the grid, such as photovoltaic (PV) panels, wind generators, and energy storage devices, and (2) new data from sensing and control devices. While this abundant data improves situational awareness and enhances control schemes, it can make the power grid more vulnerable than ever to cyber-attacks with dire consequences. Cyberattack withdraws much attention due to its potential impact, its financial losses, and its implications for national security. To understand the risks, this work looks into the operation of the electric grids, e.g., how to solve power flow equations. Specifically, this work investigates the good and the bad parts of existing methods and proposes to have a stochastic solution for power flow analysis for robustness. The finding is that no matter how the solution method is improved, system information is crucial to securely analyzing the grid. This gives utilities a false sense of security by hiding such information. For example, in a false data injection attack (FDIA), an attacker must know system information and measurements. If system information is hidden, the grid seems impossible to attack successfully, e.g., passing the Chi-square test based on system information. This dissertation shows that a carefully designed system can not only attack successfully but also with a strong performance guarantee.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

The rapid integration of variable energy sources (VRES) into power grids increases the variability and uncertainty of the net demand, challenging the power system planning, control, and operation. To solve many of the operation and planning problems in the electric grid, the power flow (PF) problem is an indispensable tool that has been studied for the last half-century. Currently, popular algorithms require second-order methods, which may lead to poor performance when the initialization points are poor or when the system is stressed. Chapter 2 presents a hybrid first-order and second-order method that improves the convergence of the PF problem when the initialization points are poor or when the system is stressed.

To provide a robust grid with new but diversified components (e.g., VRES), modern power grids are on the road to integrate unprecedented real-time and offline data for monitoring, control, and protection. However, this new data-driven outlook makes the power grid more vulnerable than ever to cyber-attacks with dire consequences. False data injection attacks (FDIAs) are a real and latent threat in modern power systems networks due to this unprecedented integration of data acquisition systems. It is of utmost importance to understand attacking mechanisms to design countermeasures. Chapter 3 studies such power grid vulnerability against a FDIAs under the current data-driven outlook. Specifically, this Chapter shows that it is possible to deploy an attack without confidential information (e.g., power system parameters or topology) by constructing an implicit model with only intercepted sensor measurements. Chapters 4 and 5 analyze the theoretical guarantees of the model-free FDIA. Altogether, this work focuses on improving modern power systems' planning,

operation, and security. Finally, Chapter 6 presents the conclusions of this work.

Chapter 2

COMBINING NEWTON-RAPHSON AND STOCHASTIC GRADIENT DESCENT FOR POWER FLOW ANALYSIS

The power flow problem is an indispensable tool to solve many of the operation and planning problems in the electric grid and has been studied for the last half-century. Currently, popular algorithms require second-order methods, which may lead to poor performance when the initialization points are poor or when the system is stressed. These conditions are becoming more common as both the generation and load profiles changes in the grid. In this Chapter, we present a hybrid first-order and second-order method that effectively escapes local minima that may trap existing algorithms. We demonstrate the performance of our algorithm on standard IEEE benchmarks.

2.1 Introduction

Power flow (PF) analysis is one of the most important and well-studied problems in the power system community. It is commonly formulated as finding the solution to a system of nonlinear algebraic equations, and a host of algorithms have been proposed to solve this system of equations. Therefore, in order to obtain a solution, it is necessary to use iterative procedures such as the Gauss-Seidel or Newton-Rapshon (NR) methods Stott (1974), Tinney and Hart (1967). The most common among these is the Newton-Raphson (NR) method, where the inverse of the Jacobian is used to update the solutions iteratively Stott (1974); Tinney and Hart (1967). The popularity of the NR method (and its variants) is partially because it has a fast convergence speed. However, convergence is not guaranteed, especially if the initial guess is not close enough to the final solution or the Jacobian matrix becomes ill-conditioned in

the iteration process Milano (2009).

It is well known that the NR method (1) may not converge if the initial guess is not close enough to the final solution, (2) has a high computational cost of the computation and inversion of the Jacobian matrix, and (3) may diverge if the Jacobian matrix is singular or close to singular.

Several studies have been done in the literature to address the above issues. Specifically, we can classify these studies for improving the NR method in two ways: robustness and computational efficiency. To overcome the above-mentioned issues, the authors in Bacher and Tinney (1989); Semlyen (1996); Luo and Semlyen (1990); da Costa *et al.* (1999) proposed different formulations and variants of the NR to solve the PF problem. The authors in Expósito and Ramos (2002) presented a power flow solution using an augmented system with rectangular coordinates. The bus current injections are introduced as additional variables in this augmented system. Similarly, the authors in Da Costa and Rosa (2008) compared the convergence of polar, rectangular, and current injection Newton-Raphson formulations on well-behaved and ill-conditioned systems. Another way to improve the convergence of the NR method is to have good starting points. In this regard, Stott (1971) analyzes selecting effective starting points. The authors in Sasson *et al.* (1971); Braz *et al.* (2000) also analyze the convergence, considering different factors, e.g., step size at each iteration. Similarly, in Milano (2009), a power flow formulation to address ill-conditioned power flow cases is presented; this formulation is based on the vector continuous Newton's method. Other works have proposed formulations and methods to improve computational efficiency Stott and Alsac (1974); Chen and Shen (2006); Abhyankar *et al.* (2014). Finally, the authors from Pirnia *et al.* (2013) take a different approach. They reformulate the PF problem as an optimization one. That work uses complementary conditions to solve the switching bus type problem and solves the optimization

problem using the generalized reduced gradient method (GRG).

This Chapter presents an algorithm that combines gradient descent (GD) methods and the NR methods to overcome some of the standard computational challenges in PF problems. By formulating the PF problem as an optimization one, gradient descent steps can be taken without inverting the Jacobian matrix. In addition, we use stochastic gradient descent (SGD) to escape from local optima and saddle-points that would have trapped deterministic algorithms. Once the iterations are close enough to the global optimal solution, we can then utilize NR-type methods to accelerate the convergence.

The rest of the Chapter is organized as follows: Section 2.2 provides the problem formulation, Section 2.3 presents the algorithm, Section 2.4 shows numerical simulations to validate our theory and makes a comparison between existing methods and our proposed algorithm. Section 2.5 concludes the Chapter.

2.2 Problem Formulation

2.2.1 Conventional Power Flow Formulation

In order to make the derivation of our formulation, we have to revisit the conventional PF formulation. The power flow problem solution is a steady-state operating point given a set of bus loads and specified voltage magnitudes. The conventional power flow equations in polar coordinates for the active and reactive power flow through lines are given by Wood and Wollenberg (2012)

$$\begin{aligned} P_{km} =& g_{km} |v_k|^2 \\ & - |v_k||v_m| \left[g_{km} \cos\left(\theta_{km}\right) + b_{km} \sin\left(\theta_{km}\right) \right], \end{aligned} \tag{2.1a}$$

$$
\begin{aligned}
Q_{km} = & - b_{km} \left|v_k\right|^2 \\
& - \left|v_k\right|\left|v_m\right| \left[g_{km} \sin\left(\theta_{km}\right) - b_{km} \cos\left(\theta_{km}\right)\right],
\end{aligned}
\tag{2.1b}
$$

where $\theta_{km} \triangleq \theta_k - \theta_m$, v_k and v_m are the complex phasors at buses k and m, θ_k and θ_m are the bus voltage angles at buses k and m, $|v_k|$ and $|v_m|$ are the voltage magnitudes at buses k and m, g_{km}, and b_{km} are the conductance and susceptace of line from node k to node m, respectively. With Eq. (2.1) we can write the nodal equations of the active and reactive power injection at each bus Lourenco *et al.* (2010); Arrillaga and Harker (1983)

$$
\begin{aligned}
p_k &= \sum_{m \in \mathbf{\Omega}_k} P_{km} = \mathrm{Re}\left(v_k I_k^*\right) = \mathrm{Re}\left(v_k \sum\nolimits_{j=1}^{n} \left(y_{kj} v_j\right)^*\right), \\
&= \left|v_k\right| \sum_{m=1}^{N} \left|v_m\right| \left(G_{km} \cos\theta_{km} + B_{km} \sin\theta_{km}\right),
\end{aligned}
\tag{2.2a}
$$

$$
\begin{aligned}
q_k &= -b_k^{sh} \left|v_k\right|^2 + \sum_{m \in \mathbf{\Omega}_k} Q_{km} = \mathrm{Im}\left(v_k I_k^*\right) = \mathrm{Im}\left(v_k \sum\nolimits_{j=1}^{n} \left(y_{kj} v_j\right)^*\right), \\
&= \left|v_k\right| \sum_{m=1}^{N} \left|v_m\right| \left(G_{km} \sin\theta_{km} - B_{km} \cos\theta_{km}\right),
\end{aligned}
\tag{2.2b}
$$

where $\mathbf{\Omega}_k$ is the set of adjacent buses to bus k, I_k is the injected current at bus k, b_k^{sh} is the shunt susceptance of bus k, G_{km} and B_{km} are the real and imaginary part of the (k, m) element in the bus admittance matrix, respectively, and N in the number of buses in the system.

We can write in compact form the set of non-linear algebraic equations, which represent the PF problem as follows

$$
f\left(\mathbf{v}\right) = \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{spec} - \mathbf{p} \\ \mathbf{q}^{spec} - \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{2.3}
$$

where $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ are the active and reactive vector of power mismatches, respectively, $\mathbf{p}^{spec}$ and $\mathbf{q}^{spec}$ are the vectors of specified values of active and reactive power

injection, respectively, and $\mathbf{p}$ and $\mathbf{q}$ are the vectors of non-linear algebraic equations of active and reactive power injections, whose entries are given by Eq. (2.2).

The standard procedure is to apply the Newton-Rapshon method to Eq. (2.3) Stott (1974); Tinney and Hart (1967); Bacher and Tinney (1989); Semlyen (1996); Luo and Semlyen (1990); da Costa *et al.* (1999), which generates the following linear system

$$\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix}^{(t)} = -\mathbf{J}\left(\mathbf{v}^{(t)}\right) \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta|\mathbf{v}| \end{bmatrix}^{(t)}, \tag{2.4}$$

where $\boldsymbol{\theta}$ is the vector of voltage angles, $|\mathbf{v}|$ is the vector of voltage magnitudes, $\mathbf{J}$ is the Jacobian matrix, and t represents the iteration number.

The power flow solution can be iteratively obtained by solving the linear system of equations in Eq. (2.4). The variables are updated as $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \Delta\boldsymbol{\theta}^{(t)}$ and $|\mathbf{v}|^{(t+1)} = |\mathbf{v}|^{(t)} + \Delta|\mathbf{v}|^{(t)}$. This process can run into some issues, for instance, if the Jacobian matrix ($\mathbf{J}$) is singular (or close to singular) the NR will diverge.

2.2.2 Power Flow Reformulation

Consider a power system with n buses. For bus k, we denote its complex voltage, active power and reactive power as v_k, p_k and p_k, respectively. We use bold fonts $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{C}^n$, $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$ and $\mathbf{q} = (q_1, \ldots, q_n) \in \mathbb{R}^n$ to denote the vector version of the quantities. Let $\mathbf{Y} \in \mathbb{C}^{n\times n}$ denote the admittance bus matrix. Then the power flow equation can be written in a compact form as

$$f(\mathbf{v}) = \mathbf{p} + j\mathbf{q} = \mathrm{diag}(\mathbf{v}\mathbf{v}^{\mathbf{H}}\mathbf{Y}^{\mathbf{H}}), \tag{2.5}$$

where $(\cdot)^H$ denotes the Hermitian transpose Zhang and Tse (2013).

Proof. Let's expand $\mathbf{v}\mathbf{v}^H\mathbf{Y}^H$

$$\left(\mathbf{v}\mathbf{v}^H\right)\mathbf{Y}^H = \begin{bmatrix} v_1 v_1^* & \cdots & v_1 v_n^* \\ \vdots & \ddots & \vdots \\ v_n v_1^* & \cdots & v_n v_n^* \end{bmatrix} \begin{bmatrix} y_{1,1}^* & \cdots & y_{n,1}^* \\ \vdots & \ddots & \vdots \\ y_{1,n}^* & \cdots & y_{n,n}^* \end{bmatrix}. \tag{2.6}$$

If we compute the k-th diagonal element in Eq. (2.6), we get $v_k \sum_{j=1}^{n} \left(y_{kj} v_j\right)^*$, which is the expression in Eq. (2.2). ■

Given a complex load vector $\mathbf{s}$, PF solves for the complex voltage vector $\mathbf{v}$ such that $f(\mathbf{v}) = \mathbf{s}$. Instead of directly solving this nonlinear equation, we consider the following optimization problem:

$$\min_{\mathbf{v}} \frac{1}{2} ||f(\mathbf{v}) - \mathbf{s}||_2^2 = \min_{\mathbf{v}} \frac{1}{2} \sum_{i=1}^{n} \left(f_i\left(\mathbf{v}\right) - s_i\right)^2, \tag{2.7}$$

where if the PF problem is feasible, then the optimal value of the objective is 0, and there is a $\mathbf{v}^*$ such that $f(\mathbf{v}^*) = \mathbf{s}$. Given that the optimization problem is unconstrained with a smooth objective function, it is natural to use gradient descent to solve it.

2.3 Combining Newton-Raphson and SGD for Power Flow Analysis

For notational simplicity, let $\mathcal{L}$ denote the objective function in Eq. (2.7). Its gradient with respect to $\mathbf{v}$ is given by the chain rule:

$$\nabla_{\mathbf{v}} \mathcal{L} = \mathbf{J}^T (f(\mathbf{v}) - \mathbf{s}), \tag{2.8}$$

where $\mathbf{J}$ is the power flow Jacobian. The standard GD algorithm is given by

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \eta \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_t), \tag{2.9}$$

where t denotes the iteration number, and η is the step size or learning rate, which may be constant or adaptive. Let $\mathcal{I}_{ref}, \mathcal{I}_{PV}$, and $\mathcal{I}_{PQ}$ denote the sets of bus indices

of the reference bus, PV buses, and PQ buses, respectively. Then, in Eq. (2.9), only the voltage angles $\delta_{\{i\}}$ for $i \notin \mathcal{I}_{ref}$ and the voltage magnitudes $v_{\{m\}}$ for $i \in \mathcal{I}_{PQ}$ will be updated. Note that by controlling which variables are updated at each iteration, we can also set specified voltage magnitudes on PV buses. From Eq. (2.8) and Eq. (2.9), the GD algorithm would stop under two conditions:

i) The global optimal is reached and $f(\mathbf{v}) - \mathbf{s} = 0$,

ii) $\mathbf{J}$ loses rank and $f(\mathbf{v}) - \mathbf{s}$ is in the null space of $\mathbf{J}^{\mathrm{T}}$.

The latter case means that the iterates $\mathbf{v}_t$ is trapped in a local minimum or in a saddle-point. To escape this minimum, it needs to stop following the gradient (since it is zero) and move in another direction. Of course, the direction it moves in should not be random. In this work, we advocate for a type of stochastic gradient (SGD) approach Bertsekas (1997). Instead of computing the exact gradient, each SGD iteration performs a parameter update for a single term of the objective function in Eq. (2.7). Specifically, a randomly index i is picked at an iteration, and the gradient with respect to $\frac{1}{2}\left(f_i\left(\mathbf{v}\right) - s_i\right)^2$. We denote this gradient as $\nabla_{\mathbf{v}}\mathcal{L}_i\mathbf{v}_t$. The algorithm is shown in Algorithm 1.

Even when the Jacobian $\mathbf{J}$ loses rank, it is typically close to full rank. Therefore, the SGD gradients $\nabla_{\mathbf{v}}\mathcal{L}_i\mathbf{v}_t$ would not all be in the null space of $\mathbf{J}$, and some of them would still provide useful directions for updating the voltage vector. Since the PF problem is non-convex, it typically has many local minima. The SGD algorithm allows the updates to "jump" out of local minima by the fluctuations induced by the randomness in the bus index selection process. Note if a global minimum is reached, then $\frac{1}{2}\left(f_i\left(\mathbf{v}\right) - s_i\right)^2$ is zero for all i, so the SGD algorithm terminates as well. However, SGD can also lead to slow convergence. Hence, we propose a hybrid method which consists in starting solving the PF problem with NR. If it stalls (when

Algorithm 1: SGD algorithm for the PF problem.

Inputs : An initial vector $\mathbf{v}_0$, the number of maximum iterations $n_{iter,max}$, for $t = 0$.

Output: $\mathbf{v}_t$

while $f(\mathbf{v}) - \mathbf{s} \leq \epsilon$ *or* $n_{iter} \leq n_{iter,max}$ **do**

  Pick a random bus i from $\{1, \ldots, n\}$

  Update the voltage vector: $\mathbf{v}_{t+1} = \mathbf{v}_t - \eta \cdot \nabla_{\mathbf{v}} \mathcal{L}_i(\mathbf{v}_t)$

end

Get vector $\mathbf{v}_t$.

the condition number of the Jacobian degrades), we switch to the SGD method. After we escape from the local minimum, then the method switches back to the NR to reach the final solution, which is depicted in Fig. 2.1. The computational complexity for k iterations for the Newton-Raphson's method is $\mathcal{O}(k \times n^3)$ Battiti (1992). For the GD-based methods the computational complexity for k iterations is $\mathcal{O}(k \times n)$.

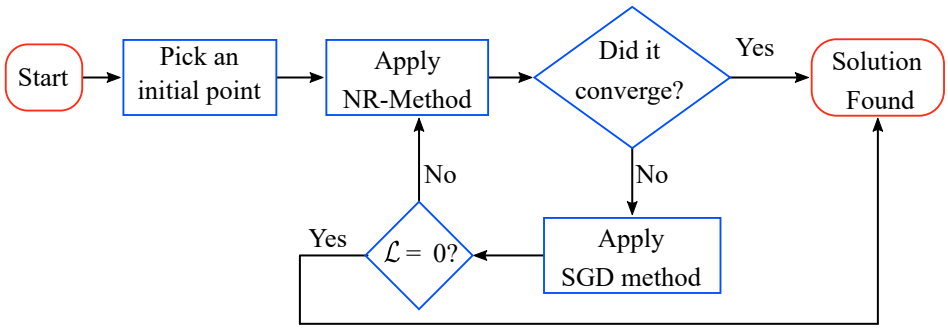
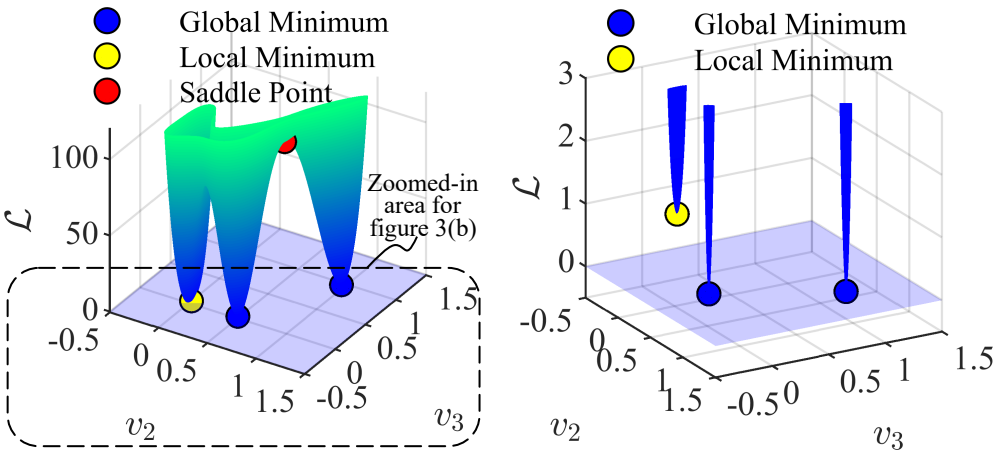


Figure 2.1: Hybrid Algorithm to Solve Pf Problem.

2.4 Numerical Results

2.4.1 Escaping Local Minima

We illustrate the behavior of the SGD algorithm using a 3-bus resistive network that is arranged in a line. We choose this example since we can explicitly plot the local minima and the saddle-point of the PF problem, as shown in Fig. 2.2a. To compare the behavior of the SGD and the standard NR algorithms, we initialize a NR solver at some point and track the error through the iterations. Of course, because of the local minimum and the saddle-point, a NR solver can get trapped at a suboptimal solution. At this point, a vanilla gradient algorithm also gets trapped, since the Jacobian loses rank. According to the algorithm in Algorithm 1, we apply the SGD algorithm. As shown in Fig. 2.3, the SGD escapes this point and is able to converge to one of the global optimal solutions.



(a) Saddle point and minima.

(b) Zoomed-in view.

Figure 2.2: Global and Local Minima of the 3-bus Network.

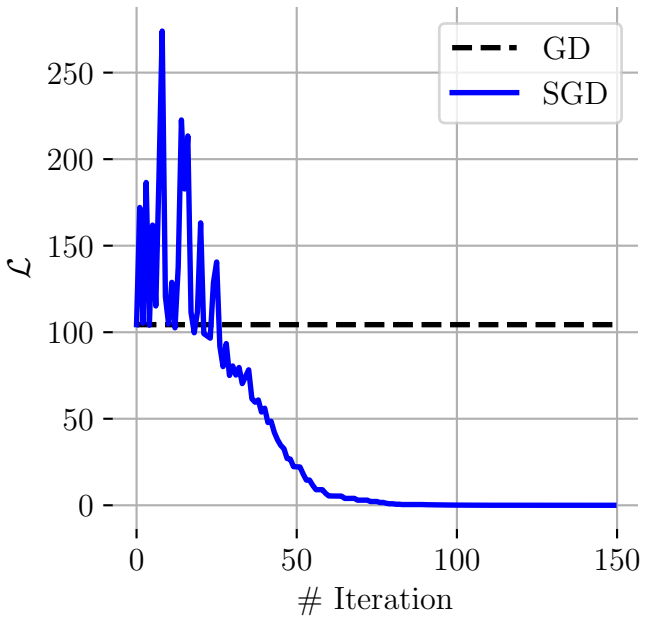


Figure 2.3: Objective Function Values of the 3-bus Network.

2.4.2 Stable and unstable PF solutions

One important aspect of the PF solution problem is the stability. In this context, there are two types of stability perspectives: (1) the PF solution's stability, and (2) the stability of an operating point. In the context of the stability of an operating point in the power system, it is defined as: *Power system stability is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact* Kundur *et al.* (2004).

In this manuscript, we do not refer to the former stability notion. When we have unstable saddle points in the objective function, we are talking about *a point on the surface of the graph of the objective function where the gradient values are all zero (equilibrium point), but which is not a local or global minimum point* Wainwright

et al. (2005). The proposed formulation cannot distinguish between a stable and unstable solution. Since we are trying to solve the power flow problem, any solution that solves the equations would be a global minimum, which corresponds to a valid mathematical solution to the PF problem.

Detecting unstable and stable operating points in the power system dynamics context is important. We believe that other metrics and methods that could be potentially embedded or used with our approach Ajjarapu and Christy (1992); Milano (2008), and is a direction for future work. We can test the principle of this idea. To do so, we carried out the following numerical experiment. Take any power test case and add a bus to the slack one, as shown in Fig. 2.4. If the line reactance on the branch and the active load power on the bus is large, the voltage magnitude will be low. In order to raise the voltage magnitude to acceptable levels, we need to compensate for reactive power on the *added bus*. Once we inject reactive power on this bus, we will have an ill-condition power system. In specific, for our added bus and added line, we have the reactance $x = 0.43$ p.u.; for the injected power on the added bus, we have $s_{\text{added}} = 2 - j1.2$ p.u.

Under these conditions, we plot the objective function in Eq. (2.7) associated with the slack bus and the added bus, which is shown in Fig. 2.5. In this Figure, we can see that we have two global minimum points and one saddle point (yellow). The two global minimum points correspond to PF solutions; one solution is stable ($v_{\text{added}}^{\text{stable}} = 1.08\angle{-52.3^\circ}$, green point) and the other is unstable ($v_{added}^{unstable} = 0.92\angle{-68.5^\circ}$, red point) from the voltage stability perspective Van Cutsem and Vournas (2007). We note that both solutions are within operating limits, which means that they are close to each other. In addition, there is a saddle point near these PF solutions, which, as we have discussed, can cause convergence issues.

Under these conditions, if we solve the PF problem with the NR method (initial-

izing at flat start) we will get the unstable PF solution. On the other hand, if we use our proposed hybrid method, we can obtain both stable and unstable PF solutions. And this is precisely our contribution in this work: To find a path to global minimum points by jumping around using the inherent randomness of the stochastic gradient descent (SGD) method.

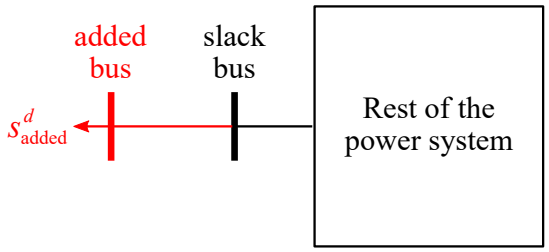


Figure 2.4: Adding a Bus to a Power System.

We can also solve the PF problem associated with the slack bus and the added bus, in Fig. 2.4, with the continuation PF algorithm Milano (2008). The result is shown in Fig. 2.6, where we can see that the continuation PF finds the two same stable and unstable solutions. However, we cannot directly compare the continuation PF and our proposed hybrid algorithm because they are conceptually different. The continuation PF method solves many conventional PF problems with the NR method; in specific, one PF problem by each point on the blue line in Fig. 2.6. In contrast, our method only solves one PF problem to find either the stable or unstable solution in Fig. 2.5. We could, in principle, embed our proposed hybrid algorithm to each iteration of the continuation PF.

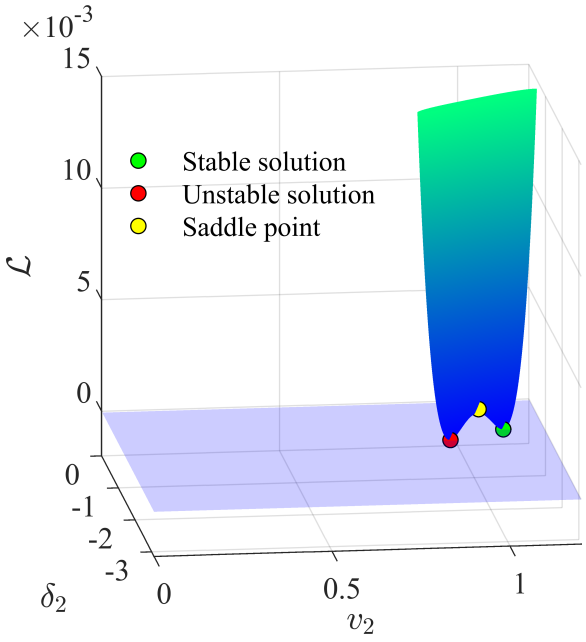


Figure 2.5: Pf Solutions from the Objective Function.

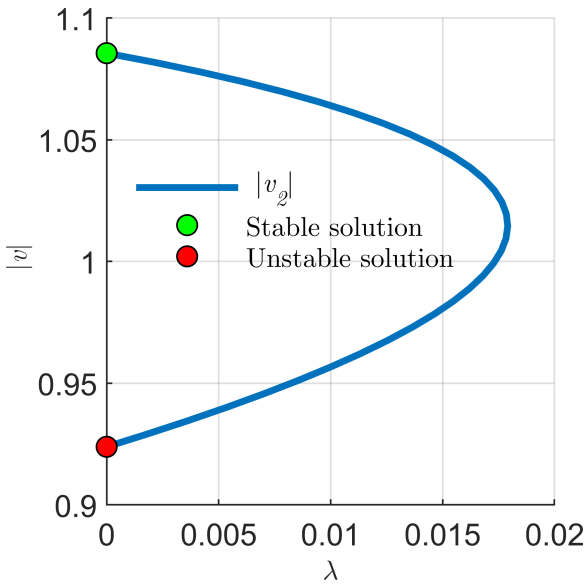


Figure 2.6: Pf Solutions with the Continuation Pf.

2.4.3 Voltage stability

We can also give an insight into the voltage stability problem with our framework. Let's take the example of a two-bus test case. We have only two variables, the magnitude voltage v_2, and the angle voltage δ_2. This means we can visualize our objective function and the PV curve (from Van Cutsem and Vournas (2007), with resistance $r = 0$), as shown in Fig. 2.7. We can see that when we have a normal load, we have two solutions: (1) one corresponds to a stable equilibrium point and (2) the other to an unstable one. Note that those solutions are far from each other. When we increase the load, we can see in Fig. 2.7 that both stable and unstable equilibrium points get closer to each other. If we keep increasing the load to the maximal power transfer point, we observe that we only have one solution to the PF problem. If we increase the load even further, we get no solution to the PF problem, and we get only a local minimum solution. In this context, no solution exists for the PF problem. Still, the objective function term associated with that node will give us insight into problematic buses that hinder the convergence of the PF problem. This is an exciting topic for future research, as we highlight in the conclusion.

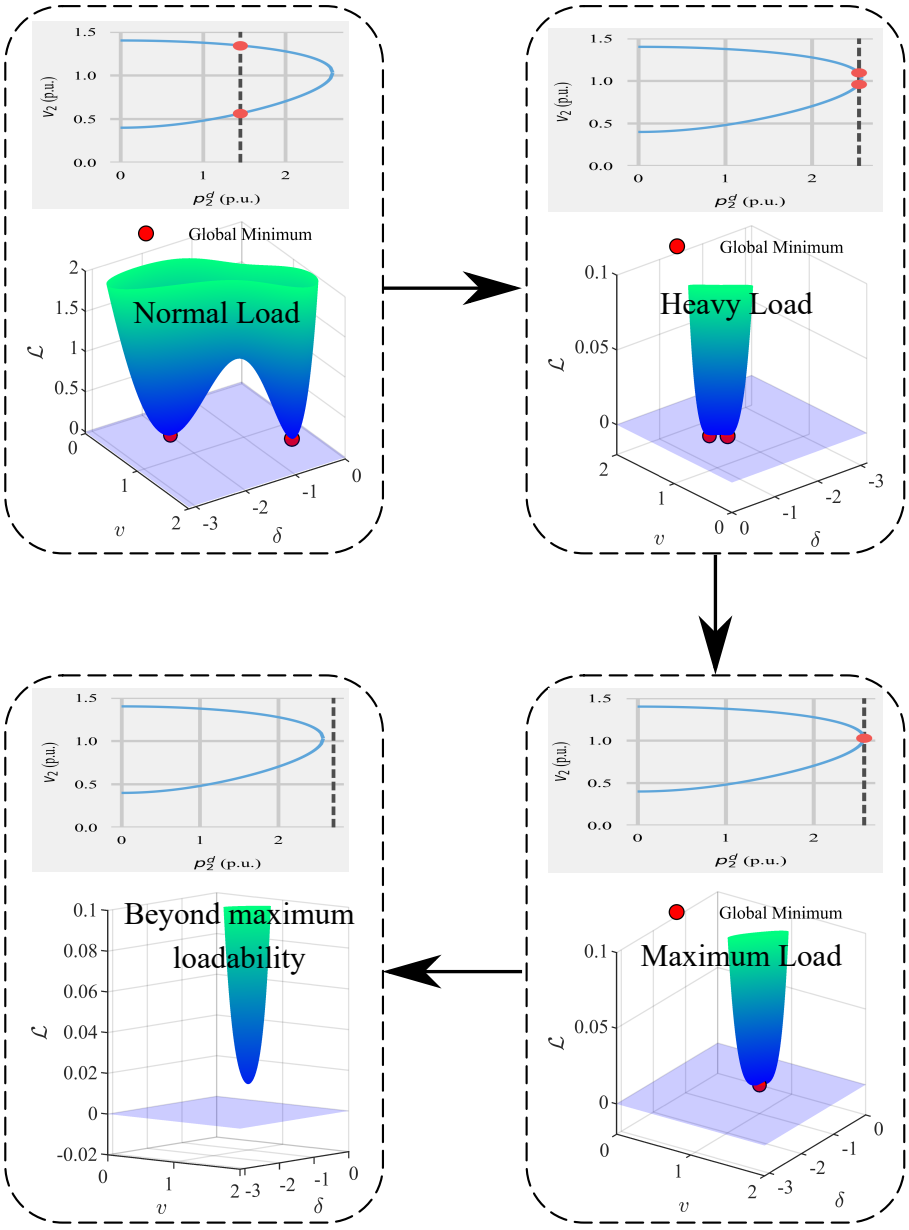


Figure 2.7: Stable and Unstable Solutions of a Pf Problem.

2.4.4 Larger Systems

In this section, we illustrate the behavior and benefit of using the hybrid method in Fig. 2.1 to solve the PF problem for standard IEEE test systems. In the hybrid method, we start with the NR algorithms, and if we detect divergence (when the condition number of the Jacobian deteriorates), then we switch over to the SGD algorithm. After running a few SGD steps, we then again switch over to the NR iterates and repeat the process until an optimal solution is found.

The reason we switch back and forth between the NR and SGD updates is to utilize the NR algorithm as much as possible. Because if NR is able to converge, it will converge much faster (quadratic in the iterations) than when SGD is used. For a system where the operating points do not change appreciably, the NR algorithm can usually converge in a few iterations from a good starting point. However, when the operating conditions vary considerably, for example, when the penetration of renewable resources is significant, then finding a good start point becomes challenging Weng *et al.* (2019); Molzahn *et al.* (2013). Therefore, the role of SGD is to "correct" the actions of the NR algorithm by escaping from suboptimal solutions and saddle-points when a bad starting point is used.

In order to show the usefulness of this hybrid approach, we performed a set of simulations. We compare the NR against our hybrid approach. For both PF methods, we model transformers and phase shifters with specified tap ratios and phase shift angles that are kept constant throughout the simulation. For all the simulations, we set specified voltage magnitudes and generator reactive power limits on PV buses. We enforce reactive power limits by using the conventional procedure Kothari and Nagrath (1989). That is, if any generator has a violated reactive power limit, its reactive injection is fixed at the limit, and the corresponding bus is converted to a

PQ bus. For the SGD simulations, we used an adaptive learning rate. Specifically, we used the Adam adaptive learning algorithm Kingma and Ba (2014) with stepsize $\eta = 0.01$, and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set $n_{\text{iter,max}} = 100$. First, as a baseline, we perform simulations under normal load conditions with a flat start with the NR method and our proposed one. We can see the results in Table 2.1, which shows that both methods are successful. However, when the conditions are changed, the NR method will struggle to find a solution to the PF problem, as we will show in the next simulations.

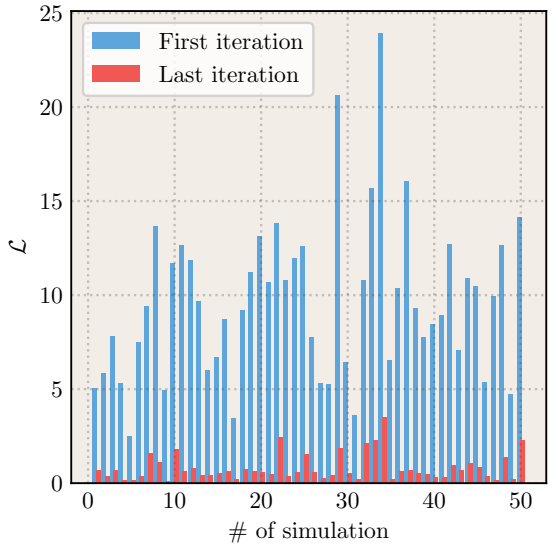


Figure 2.8: Values of the First and Last Iteration.

Now, we will change the starting points for the simulations. This means that the initial guesses will be further away from a solution. The experiment design is as follows. We first randomly pick a voltage solution vector $\mathbf{v}^*$ and compute the associated active and reactive power. Subsequently, we randomly pick initial starting points from a uniform distribution as follows: $|\mathbf{v}| \sim \mathcal{U}(\min(|\mathbf{v}|^*), \max(|\mathbf{v}|^*))$ and $\boldsymbol{\theta} \sim \mathcal{U}(\min(\boldsymbol{\theta}^*), \max(\boldsymbol{\theta}^*))$ and test whether the algorithms can reach $\mathbf{v}^*$ from the starting points. We use this random initialization to make more challenging the

Table 2.1: Convergence Results with a Flat Start.

Case	NR	Hybrid
14-bus	✓	✓
39-bus	✓	✓
57-bus	✓	✓
118-bus	✓	✓
300-bus	✓	✓

convergence when a PF solution exits. We use the 14-bus system as an illustration to explore the convergence of our hybrid method. The experiment design is as follows. We randomly initialize voltage angles and magnitudes (as we described before). Then, we use the SGD method for 50 iterations. Fig. 2.8 shows the result of 50 simulations, where the value of the objective function at the first iteration (in blue) is quite large, but the value of the last iteration (in red) is minimal. Then, we initialize the NR method with the starting points associated with the first value iteration and the points associated with the last iteration. The result of doing this experiment is shown in Table 2.2 (under normal load), where the convergence rate is 10% with the NR method. On the other hand, we have a 100% convergence rate with our hybrid approach. We carry out the same experiments for the 39-, 57-, 118-, and 300-bus cases, in which we obtained better results than the conventional NR approach, as shown in Table 2.2.

For the proposed method under worse conditions, we make simulations with a higher load level. We increase the power system load by multiplying the active load by a factor α that will produce an ill-conditioned test case. Table 2.2 shows the result (under heavy load), where the convergence rate of the hybrid method is better than the NR method.

Finally, we explore the convergence of both PF methods around local minimum and saddle points. For this numerical experiment, we perform two simulations by test case. In one simulation, we choose the starting point to be a local minimum one, and in the other simulation, we choose the starting point to be a saddle-point. The results are shown in Table 2.2. We can see that the NR method does not converge, whereas we achieve convergence in both simulations with our method. This result is expected due to the NR Jacobian is singular at the first iteration.

Table 2.2: Convergence of Different Test Cases for the Nr and Hybrid Methods with Original and Increased Loads.

	Convergence rate (%)				**Convergence Result**	
	Normal Load		**Heavy Load**		**Local minimums and saddle points**	
Case	**NR**	**Hybrid**	**NR**	**Hybrid**	**NR**	**Hybrid**
14-bus	10	100	10	96	✗	✓
39-bus	10	100	10	84	✗	✓
57-bus	10	90	0	82	✗	✓
118-bus	5	80	5	66	✗	✓
300-bus	5	75	0	61	✗	✓

2.5 Conclusion

To provide a robust solution for the power flow problem for the current challenging power system operation and control conditions, a novel hybrid method for the power flow problem was proposed in this Chapter. This method combines the Newton-Raphson and stochastic gradient algorithms to achieve fast convergence speed as well as the ability to escape local minima and saddle points. Numerical tests on power

systems of various sizes and topologies demonstrated the effectiveness and efficiency of the proposed approach in solving fast and reliable the PF problem under different load conditions and initial starting points.

Modern power grids are evolving not only with the integration of new active devices, making it challenging to control and operate but also with the unprecedented incorporation of new sensing devices that poses security vulnerabilities into the grid. The next Chapter studies how these abundant data from sensing devices could be exploited to explore such menace.

Chapter 3

ATTACK POWER SYSTEM STATE ESTIMATION BY IMPLICITLY LEARNING THE UNDERLYING MODELS

The last Chapter introduced a hybrid algorithm to solve the power flow problem with the ongoing stressed grid operating conditions resulting from the inclusion of new active devices. Additionally, power systems simultaneously integrate prodigious sensing devices that could endanger the grid to cyber menaces, such as false data injection attacks (FDIAs). Thus, to design effective schemes to protect the grid against such attacks, it is of utmost importance to understand how attackers could exploit these new massive data from sensing devices. To successfully deploy a FDIA, most past FDIA strategies need privileged power system information, which is carefully held by the power system operator. Newer approaches circumvent this issue by solely relying on intercepted measurement data, but they lack mathematical warranties of succeeding. This Chapter exposes power systems' vulnerability by showing that it is possible to deploy an attack without confidential information and, at the same time, to have a high probability of being successful. We present a scheme that learns (1) the implicit power system measurement distribution and (2) a surrogate of the unknown state estimator model. The proposed framework utilizes a Wasserstein generative adversarial network to learn the measurement distribution and an autoencoder to learn the unknown state estimator model. Additionally, we present a convergence proof that ensures that the proposed framework converges to the power system measurement distribution. The proposed method is demonstrated to be successful via extensive simulation on IEEE 9-, 14-, 57-, 118-, and 300-bus test cases.

3.1 Introduction

Data revolution takes place worldwide in different disciplines, including power systems. To provide a robust grid with new but diversified components, modern power grids are on the road to integrate unprecedented real-time and offline data for monitoring, control, and protection. However, this new data-driven outlook makes the power grid more vulnerable than ever to cyber-attacks with dire consequences. For instance, the power system operator may take wrong corrective actions that can cause a blackout; wrong actions can also cause inaccurate energy prices in a real-time electricity market Xie *et al.* (2010); Xie *et al.* (2011).

To better protect the system, it is essential to understand potential attack mechanisms. Among various attack categories Li *et al.* (2012); Zhou *et al.* (2019); Costilla-Enriquez and Weng (2021), False Data Injection Attacks (FDIA) gained the attention of the power system community after the work in Liu *et al.* (2011), which showed that unobservable attacks against DC State Estimators (SE) are possible. In this type of attack, the attacker modifies measurement data such that the estimated states are different from the real ones Mohammadpourfard *et al.* (2018, 2019). These first works have the following assumptions, which may be impractical:

(i) The attacker has access to the entire network information (e.g., line parameters, grid topology, state estimator model, and estimated states) Hug and Giampapa (2012); Wang *et al.* (2020). It is impractical to think that an attacker can gather all this data without an insider in the Independent System Operator (ISO). Since this information is guarded by the power system operator it is difficult for an attacker to have this knowledge.

(ii) These first studies rely upon the DC power flow model when power system operators use the AC power flow model in real-world settings. The reason, AC-

based FDIAs are harder to design and deploy due to the inherent complexity of the nonlinear power flow equations Costilla-Enriquez *et al.* (2020); Jin *et al.* (2019).

Subsequent work relaxed the first assumption. Specifically, Liu *et al.* (2011); Yuan *et al.* (2011); Xie *et al.* (2011); Valenzuela *et al.* (2013); Mo *et al.* (2012); Kosut *et al.* (2010); Zhang *et al.* (2015) propose various frameworks to design FDIAs with only partial network information, but they still rely on a DC-based model. To relax the DC model's assumption, a few studies have focused on FDIA with an AC-based model Hug and Giampapa (2012); Liu and Li (2017); Jia *et al.* (2012). However, all the aforementioned approaches construct an attack vector relying upon the power system underlying information; we can call these techniques model-based FDIAs.

Later works showed that it is also possible to deploy FDIA without knowing privileged power system information such as power system parameters and topology or the state estimator model. The only needed information to deploy a FDIA are the power system measurements, and we classify these kinds of attacks as model-free FDIAs. In modern power system networks, the information is sent via remote terminal units that are designed avoid system intrusion Teixeira *et al.* (2015); Wang and Lu (2013). However, conventional approaches such as security software and firewalls could be insufficient to protect the system against breaches and cyber threats Jin *et al.* (2019). For example, in 2015, a cyber-attack was successfully deployed on Ukraine's electricity infrastructure. Around one year before the attack, the attackers gained access to multiple industrial networks by using the malware tool *BlackEnergy 3* (BE), Styczynski and Beach-Westmoreland (2019). This malware enables unauthorized network access with valid (stolen) user credentials to move laterally across internal utilities' system. In this incident, the attackers gained access to targeted networks using weaponized Microsoft Office files by embedding BE in Visual Basic macro scripts. This latent

risk has been recognized by the National Academies of Sciences, Engineering, and Medicine National Academies of Sciences, Engineering, and Medicine (2017). In the same work, they conclude that the United States' power system network is vulnerable to cyber-attacks. Thus, for an attacker, it would be feasible to collect sensor measurements by exploiting the protection schemes Jin *et al.* (2019).

The authors in Yu and Chin (2015) showed that it is possible to deploy a stealthy FDIA by using principal component analysis (PCA). The extension of this work in Chin *et al.* (2017) proposed a geometric approach to carry out a FDIA based only on power system measurements. The authors in Kim *et al.* (2014) proposed a data-driven attack that learns the system operation subspace from measurements around a linearized nominal state. The work in Zhang *et al.* (2021) presented a zero-parameter information attack that only requires power system's topology information. The works in Ahmadian *et al.* (2018); Mohammadpourfard *et al.* (2020) employed machine learning techniques to carry out a FDIA. Specifically, they trained a generative adversarial network (GAN) to generate tampered power system measurements that will be stealthy with high probability. While the works in Ahmadian *et al.* (2018); Mohammadpourfard *et al.* (2020) and our work use generative adversarial networks (GANs) to carry out a FDIA, our approach has some important differences. Both works in Ahmadian *et al.* (2018); Mohammadpourfard *et al.* (2020) use the DC linear power flow model. In contrast, our proposed approach uses the AC non-linear power flow model. Whereas the work in Mohammadpourfard *et al.* (2020) requires normal and tampered measurements to train a conditional adversarial network (cGAN), our approach only requires normal measurements, which is a more reasonable assumption. This means that our attack is more appealing at the level of the information needed to train our model.

The difficulty with the model-free FDIAs is that it is hard to ensure that the

model-free approach truly captures some properties of the power system model to bypass tests, such as the Chi-squared test to obtain the trust from energy management systems. To show the power system vulnerability under this setting, we introduce a data-driven approach that generates tampered measurements with the desired properties to deploy a FDIA, and at the same time, to have mathematical guarantees about the model accuracy. We achieve this goal by (1) implicitly learning the power system measurement distribution from data; and (2) learning a proxy model for the unknown state estimator.

Specifically, we aim to design a flexible model that captures the complex underlying interactions in the power system to learn the measurement distribution from data. Nonparametric methods are flexible since they build models from data making as few assumptions as possible, which usually means utilizing statistical models that are infinite-dimensional Wasserman (2006). While these type of models are flexible by keeping the underlying assumptions as weak as possible, they are computationally demanding due to the required increment of number of parameters Ferraty and Vieu (2006); Hollander and Sethuraman (2001). For example, the work in Ji *et al.* (2017) shows that their nonparametric model grows in complexity as additional data is used to train the model. As real power systems could have thousands of buses and data measurements from many years, the number of parameters needed in non-parametric models are computationally intractable Ferraty and Vieu (2006). Therefore, we choose parametric models, which can be designed with a fixed number of parameters that depend upon the specific problem. In recent years, these parametric models have had tremendous success in the ML community because they can learn complex high-dimensional distributions (for example, images in high resolution). In power systems, for example, the work in Lakshminarayana *et al.* (2022) physics-informed parametrized neural networks (PINN) to learn the underlying power grid's

parameters. In the parametric models, we introduce a framework utilizing generative adversarial networks (GANs) to learn the power system measurement distribution to create spurious measurements to deploy a FDIA, as GAN's loss function is fully specified. As a comparison, variational autoencoders (VAEs)'s loss function is only the evidence lower bound (ELBO), which is hard to be embedded into other learning. Even more importantly, we can present mathematical proof to show that the GAN reliably learns the power system measurement distribution. In specific, we use the Wasserstein Generative Adversarial Network (WGAN), which is guaranteed to converge under mild assumptions to the actual observed distribution Zhang *et al.* (2018).

In addition, to mimic the data distribution, one knowledge we do have is the form of residual error test. Therefore, we propose to boost our attack capability by learning the state estimator model for the residual error test. However, learning the state estimator model directly is difficult because neither the power system nor the state estimator is known. To circumvent this issue, we use a surrogate model to mimic the state estimator. The residual error test and an autoencoder (AE) share the same mathematical structure. Thus, an AE can be trained as a proxy to mimic the state estimator. We leverage this similarity and employ an AE as a proxy for the residual test error. Specifically, in our proposed scheme, we include this proxy as a regularization term, which helps to improve the quality of the created tampered measurements. Finally, a second regularization term is added to maximize the impact of the attack. Whereas the model-based attacks need the complete network information (e.g., line parameters, grid topology, state estimator model, and estimated states), our proposed model-free approach needs a dataset of the measurements of the considered network to work. And such a data set does not need all the measurements to be included, which is another advantage.

The performance of the proposed model-free FDIA is verified by simulations on the standard IEEE 9-, 14-, 57-, 118-, and 300-bus test networks. Also, to contrast the differences and advantages between our approach and the existing ones in the literature, we carry out comparisons between our proposed FDIA and three other successful methods reported in Hug and Giampapa (2012); Chin *et al.* (2017); Liu and Li (2017). These results show that our proposed model-free is successful. Specifically, our proposed model-free attack tampers measurements in a way that can fool the power system operator with high probability.

3.2 Problem Formulation

To show the proposed model-free FDIA attack, we first review the model-based approaches based on AC state estimation.

3.2.1 State Estimation with AC Power Flow Model

State estimation (SE) infers the state variables (i.e., voltage angles and voltage magnitudes) $\mathbf{x} = (x_1, \ldots, x_n)$ from a set of measurements $\mathbf{z} = (z_1, \ldots, z_m)$ Wood *et al.* (2013), where n is the number of buses or nodes in the grid, and m is the number of measurements. Mathematically, we can describe the problem as $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$, where $\mathbf{h}(\cdot)$ is the physical (non-linear) relationship between state variables and measurements, and $\mathbf{e}$ is a vector that represents white noise from the collected measurements (e.g., SCADA or PMU). In practice, measurements are collected and sent to the power system operator, which obtains the estimated states $\hat{\mathbf{x}}$ by solving Tarali and Abur (2012); Weng *et al.* (2017):

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\mathbf{z} - \mathbf{h}(\mathbf{x})\right)^T \mathbf{W}^{-1} \left(\mathbf{z} - \mathbf{h}(\mathbf{x})\right) = \mathrm{SE}(\mathbf{z}), \tag{3.1}$$

where, for compactness, we define the state estimator operator $\mathrm{SE}(\cdot)$. The input of this operation is a vector of measurements and its output are the estimated states. However, the vector of measurements $\mathbf{z}$ may contain bad or wrong data due to telecommunication failures, meter errors, or even FDIAs Abur and Exposito (2004); Wang *et al.* (2020). To estimate the states with confidence, the SE possesses a Bad Data Detector (BDD) module to detect and filter suspicious data.

Bad Data Detector (BDD)

The measurement errors are assumed to follow a Gaussian distribution $e_i \sim \mathcal{N}(0, \sigma_i)$ Abur and Exposito (2004) (where σ_i is the standard deviation of the i-th measurement). Therefore, the squared measurement residual error $\mathbf{r} = \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$ follows a Chi-square distribution χ_k, where k represents the number of independent variables in the power system, and $\hat{\mathbf{z}} = \mathbf{h}(\hat{\mathbf{x}})$ is the vector of estimated measurements. Then, the presence of errors in the measurements can be detected with the Chi-square test (or residual error test) Abur and Exposito (2004); Weng *et al.* (2016). This test works as follows:

(i) Select the detection confidence probability p (e.g., 0.95), and compute its associated threshold value $\tau = \chi^2_{k,p}$ with $p = \Pr\left(\mathrm{J}(\hat{\mathbf{x}}) \leq \chi^2_{\mathrm{k,p}}\right)$.

(ii) Compute the normalized measurement error $J(\hat{\mathbf{x}}) = \sum_{i=1}^{m} \left(z_i - h_i(\hat{x}_i)\right)^2 / \sigma_i^2$.

(iii) If the inequality in equation 3.2 holds, bad data will be suspected, or else the measurements are assumed to be free of bad data.

$$J(\hat{\mathbf{x}}) \geq \tau \tag{3.2}$$

3.2.2 Model-based FDIA in AC State Estimation

A FDIA modifies the estimated states $\hat{\mathbf{x}}$ or measurements $\hat{\mathbf{z}}$ by changing the original SCADA and PMU measurements $\mathbf{z}$ with a maliciously tampered measurement vector, that is, $\mathbf{z}_a = \mathbf{z}+\mathbf{a}$, where $\mathbf{a}$ is an attack vector. The attacker designs this attack vector to compromise the system's reliability by creating a wrong state estimate. For a FDIA to be successful, it must circumvent the bad data detector equation 3.2 He *et al.* (2017). The assumptions in the literature for a model-based FDIA about the attacker's knowledge are the following Liu *et al.* (2011); Hug and Giampapa (2012); Zhang and Sankar (2016): (1) the attackers can intercept and alter the power system measurements that are used to obtain the estimated states in the grid; (2) the attacker has access to the power system model, which includes transmission line parameters and topology information; and (3) the attacker possess the SE model or can obtain the estimated states of the network. Under these strong assumptions, the attacker would be able to launch a perfect FDIA Wang *et al.* (2020). In this perfect FDIA, the attacker can define the attack vector as $\mathbf{a} = \mathbf{h}(\hat{\mathbf{x}} + \mathbf{c}) - \mathbf{h}(\hat{\mathbf{x}})$, where $\mathbf{c}$ is the vector of changes in the estimated states. In this scenario, if the original measurements $\mathbf{z}$ can pass the residual-based bad data detector test in equation 3.2, the corrupted measurements $\mathbf{z}_a$ will also pass this test Hug and Giampapa (2012).

The work in Hug and Giampapa (2012) proposed an FDIA needing only partial power system information. In this context, there are two types of variables. (1) Measurements and state variables that are not altered, which are denoted with subscript 1, $\hat{\mathbf{x}}_1$ and $\mathbf{z}_1 = \mathbf{h}_1(\cdot)$. (2) Measurements and state variables that are maliciously altered, which are denoted with subscript 2, $\hat{\mathbf{x}}_2$ and $\mathbf{z}_2 = \mathbf{h}_2(\cdot)$. If an attacker constructs the attack vector as

$$\mathbf{a}_2 = \mathbf{h}_2(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 + \mathbf{c}) - \mathbf{h}_2(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2), \tag{3.3}$$

the tampered measurements will have the same residual error as the real ones. Note that to create the attack vector in equation 3.3, the attacker must know the estimated values of the state variables appearing in $\mathbf{h}_2$, which is still a strong assumption. There are other types of FDIA. For example, if $\mathbf{a} \neq \mathbf{h}(\hat{\mathbf{x}} + \mathbf{c}) - \mathbf{h}(\hat{\mathbf{x}})$ but equation 3.2 holds, then the attack is called a generalized FDIA Liang *et al.* (2017).

3.3 Proposed Model-Free FDIA

Contrary to the model-based FDIAs, the model-free models only make one assumption Yu and Chin (2015); Chin *et al.* (2017); Ahmadian *et al.* (2018); Mohammadpourfard *et al.* (2020): The attackers can intercept and alter the power system measurements that are used to obtain the estimated states in the grid. So, in this section, we show a theoretically sound method to deploy a FDIA by only using the power system measurements. If we want to deploy an attack without any underlying power system knowledge, we have to learn an implicit model through observations, that is, from power system measurements (SCADA and PMU). This implicit model should capture the inherent non-linearity relationships between different measurements based on residual error tests. Also, this model should be able to create new tampered measurements such that they are overlooked by the power system operator but change the estimated states and measurements. To summarize, we present a data-driven approach based on a WGAN with two regularization terms. First, the measurement distribution is learned with the WGAN, $\mathbf{z} + \mathbf{e}$. Second, to pass the residual error test, a proxy of the unknown SE model is embedded into the WGAN as a regularization term, $\mathbf{h}(\mathbf{z})$. Finally, a regularization term is added to maximize the attack impact.

3.3.1 Learning the Measurement Distribution

Goodfellow *et al.* (2014) introduced the idea of generative adversarial networks, which revolutionized the machine learning (ML) field. GAN is a framework to teach a Deep Learning (DL) model the implicit training data distribution so that we can sample from it and generate new data from that same distribution; in our case, the power system measurement distribution. Specifically, rather than sampling directly from an (assumed) parametric distribution, the target random variable is generated as a deterministic transformation of a simple, independent noise source, for instance, a Gaussian distribution. GANs are made of two distinct models, a generator and a discriminator. Formally, the minimax objective of the GAN is

$$\min_{G} \max_{D} \quad \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_r}\mathbb{E}_{\boldsymbol{\lambda}\sim\mathbb{P}_\lambda}\left[\log D\left(\mathbf{z}\right) + \log\left(1 - D\left(G(\boldsymbol{\lambda})\right)\right)\right], \tag{3.4}$$

where D is a discriminative network, G is a generative network, $\mathbb{P}_r$ is the real data distribution, and $\boldsymbol{\lambda}$ is the latent space, which it is sampled from an independent distribution $\mathbb{P}_\lambda$; that is, $\boldsymbol{\lambda} \sim \mathbb{P}_\lambda$ (usually a Gaussian distribution).

However, GANs have some issues, such as vanishing gradient and the lack of guarantee to convergence. The work in Arjovsky *et al.* (2017) presented the Wasserstein GAN (WGAN) that solves these issues. Also, WGANs possess stronger mathematical guarantees. For example, the authors in Zhang *et al.* (2018) proved that (under mild assumptions) the generator in the WGAN will converge to the true data distribution $\mathbb{P}_r$. Therefore, in this work, we will use this type of WGAN. These models are made of two distinct neural networks, a generator G and a discriminator D (or critic). The minimax objective of the WGAN is

$$\min_{G} \max_{D\in\mathcal{D}} \quad \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_r}\mathbb{E}_{\boldsymbol{\lambda}\sim\mathbb{P}_\lambda}\left[D\left(\mathbf{z}\right) - D\left(G(\boldsymbol{\lambda})\right)\right], \tag{3.5}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions Arjovsky *et al.* (2017); $\mathbb{P}_r$ is the real data

distribution; $\boldsymbol{\lambda}$ is known as the latent space, and it is sampled from an independent distribution $\mathbb{P}_{\lambda}$. The generator G learns the real distribution $\mathbb{P}_r$, which, in our context, this real distribution is the set of historical observed measurements $\mathcal{Z} = \{\mathbf{z}_i \in \mathbb{R}^m\}_{i=1}^{L}$ (where L is the number of elements in the dataset), where $\mathbf{z}_i = \mathbf{h}(\mathbf{x}_i) + \mathbf{e}_i$. In other words, G implicitly learns to generate samples from the underlying model $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$.

3.3.2 Learning the State Estimator Model

To gain trust from the power system operator, the created tampered measurements, $\tilde{\mathbf{z}} = G(\boldsymbol{\lambda})$, must pass the residual error test in equation 3.2. This residual error for the tampered measurements is given as

$$\tilde{\mathbf{r}} = \left\| \tilde{\mathbf{z}} - \hat{\tilde{\mathbf{z}}} \right\|^2, \tag{3.6}$$

where $\hat{\tilde{\mathbf{z}}} = \mathbf{h}\left(\hat{\tilde{\mathbf{x}}}\right)$ is the vector of estimated tampered or fake measurements, and $\hat{\tilde{\mathbf{x}}} = \mathrm{SE}(\tilde{\mathbf{z}})$ is the vector of estimated states from tampered measurements. As equation 3.2 suggests, the smaller the residual error $\tilde{\mathbf{r}}$, the bigger the probability of passing the test for a given tampered measurement, $\tilde{\mathbf{z}}$. In other words, a given vector of tampered measurements, $\tilde{\mathbf{z}}$, should produce a similar estimated vector, $\hat{\tilde{\mathbf{z}}} = \mathbf{h}\left(\hat{\tilde{\mathbf{x}}}\right)$. However, in this model-free approach, we do not have access to the state estimator model $\mathbf{h}(\cdot)$. This non-linear function $\mathbf{h}(\cdot)$ can be thought of as a mapping from the measurement space to the estimated measurement space. For a vector of real measurements, the estimated measurements will be similar so that the residual error is low. This state estimator function $\mathbf{h}(\cdot)$ is unknown. Still, given its properties, it is possible to learn it from data and create a proxy to impose the same behavior in the tampered measurements.

The residual error expression in equation 3.6 resembles the loss function from an

autoencoder (AE). Thus an AE model is a natural option to learn a proxy model of the unknown state estimator function $\mathbf{h}(\cdot)$. An autoencoder is a neural network that aims to produce or replicate its input to its output Goodfellow *et al.* (2016). To do this, the autoencoder is trained to learn an encoding for a particular distribution and then with such encoding, learn to reconstruct the input distribution. To learn a meaningful encoding, the model's architecture prioritizes which traits from the input should be learned. By this process the autoencoder learns to ignore superfluous data, which could be noise. We will see how this autoencoder property improves the generation of fake measurements in Section 3.5.5. Mathematically, the autoencoder is represented as a function, that is, $\mathrm{AE}(\cdot)$, and it is trained with the squared loss function:

$$\|\mathbf{z} - \mathrm{AE}(\mathbf{z})\|^2. \tag{3.7}$$

A trained AE with real measurements with equation 3.7 will learn the unknown function $\mathbf{h}(\cdot)$ that will minimize the residual error in equation 3.6. Once the autoencoder is trained (denoted as AE^*), the loss function in equation 3.7 can be embedded into equation 3.5 to incentivize the generation of tampered measurements that will produce similar estimated measurements, and thus lower the residual error. This can be done by adding the regularization term $\|\tilde{\mathbf{z}} - \mathrm{AE}^*(\tilde{\mathbf{z}})\|_2^2$ in equation 3.5:

$$\min_{G}\max_{D\in\mathcal{D}} \quad \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_r}\mathbb{E}_{\boldsymbol{\lambda}\sim\mathbb{P}_\lambda}\Big[D(\mathbf{z}) - D(\tilde{\mathbf{z}}) + \|\tilde{\mathbf{z}} - \mathrm{AE}(\tilde{\mathbf{z}})\|^2\Big], \tag{3.8}$$

where $\tilde{\mathbf{z}} = G(\boldsymbol{\lambda})$.

3.3.3 Maximize the FDIA Impact

The WGAN in equation 3.8 implicitly learns the underlying model that generates the observed data Goodfellow *et al.* (2014); Arjovsky *et al.* (2017). To train a WGAN

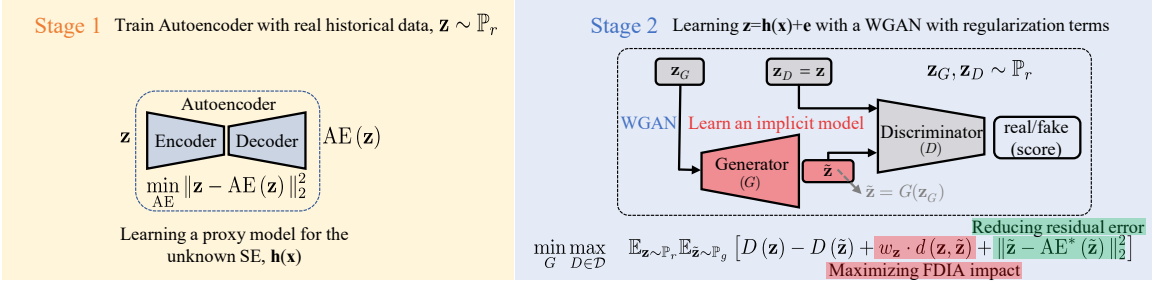


Figure 3.1: Proposed Model-free Architecture with a Wgan and Two Regularization Terms to Deploy an FDIA.

with equation 3.8, we need to sample $\mathbf{z}$ from the true data distribution $\mathbb{P}_r$. However, the generator in equation 3.8 conventionally takes a random signal as input and maps it to the true data distribution space; that is, $\boldsymbol{\lambda} \sim \mathbb{P}_\lambda$, where $\mathbb{P}_\lambda$ is usually a Gaussian distribution. This means that we do not have any control over the created fake measurements. To successfully attack a power system, we want these fake measurements, produced by our WGAN, to create different states from the actual ones. The attacker can only see and modify observed measurements. Thus, the attacker can attempt to markedly change the unobservable states by stealthy and sizeably manipulating the intercepted measurements to perform a successful FDIA. To accomplish this, we need to generate tampered measurements from the observed ones.

If we want to generate tampered measurements from the observed ones, rather than using a random distribution $\mathbb{P}_\lambda$ as latent space to feed our generator, we use the power system measurements as input to the generator, that is, $\mathbb{P}_\lambda = \mathbb{P}_r$. The result is that the generator's latent space is not fed with an arbitrary random distribution: it is fed with the power system measurement distributionF. Specifically, we are conditioning the WGAN with respect to the actual measurement vector $\mathbf{z}$, as depicted in Fig. 3.1. This is desirable because in this way, rather than creating

tampered measurements from an arbitrary distribution, they are constructed based on the observed ones. F Furthermore, the created tampered measurements will differ from those received as input due to a regularization term that we include in our model, as we explain below.

F To successfully deploy an FDIA, we want to incentivize the generator to construct measurements that will produce different measurements from those received as input. This will provoke the SE with high a likelihood to produce erroneous estimated states, the main objective in a FDIA. To accomplish this, we can incentivize the model to generate such fake measurements with the regularization term $w_{\mathbf{z}} \cdot d\left(\mathbf{z}, \tilde{\mathbf{z}}\right)$ in equation 3.9 (the first regularization term in Fig. 3.1 in red), where $\tilde{\mathbf{z}} = G\left(\mathbf{z}\right)$, $d\left(\cdot\right)$ is a distance function (e.g., mean squared or mean absolute distance), $d\left(\mathbf{z}, \tilde{\mathbf{z}}\right)$ represents the distance between the original measurement and the generated one, and $w_{\mathbf{z}}$ is a hyper-parameter that represents the weight of this distance. This regularization term incentivizes the WGAN to produce a tampered measurement vector $\tilde{\mathbf{z}}$ that will generate completely wrong estimated measurements. Finally, we can explicitly induce sparsity in the attack vector. This sparsity property is desirable and essential because the attacker has to alter fewer measurements to successfully deploy a FDIA, Chen *et al.* (2018). We can add it into the model in equation 3.9 with the regularization term, $w_{\text{sparse}} \cdot \left\|\mathbf{z} - \tilde{\mathbf{z}}\right\|_1$, where w_{sparse} is the weight of the sparsity regularization term. This leads to the following loss function

$$\min_{G} \max_{D \in \mathcal{D}} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_g} \Big[D\left(\mathbf{z}\right) - D\left(\tilde{\mathbf{z}}\right) + \left\|\tilde{\mathbf{z}} - \text{AE}\left(\tilde{\mathbf{z}}\right)\right\|_2^2 + w_{\mathbf{z}} \cdot d\left(\mathbf{z}, \tilde{\mathbf{z}}\right) + w_{\text{sparse}} \cdot \left\|\mathbf{z} - \tilde{\mathbf{z}}\right\|_1 \Big]. \quad (3.9)$$

Training the WGAN with regularization terms adds complexity to the training process. If the regularization term becomes too large with respect to the original WGAN loss, the generator will struggle to learn the correct distribution. If the

regularization term is too small, it will not have any effect on the training process. Thus, the regularization term will not fulfill its purpose. To solve this issue, a dynamic weight is introduced to control the size of $d(\mathbf{z}, \tilde{\mathbf{z}})$ throughout the training phase. This weight must maintain a balance between the generator loss term $D(\tilde{\mathbf{z}})$ and the regularization term $d(\mathbf{z}, \tilde{\mathbf{z}})$, so that the WGAN learns the desired distribution, and at the same time, the regularization term accomplishes its purpose. We can achieve this balance by setting the regularization term to be half of the generator loss term. We express this as $\frac{1}{2}|D(\tilde{\mathbf{z}})| = w_{\mathbf{z}} \cdot d(\mathbf{z}, \tilde{\mathbf{z}})$. Then, the result of such dynamic weight $w_{\mathbf{z}}$ is described in equation 3.10 where $t > 1$ is the iteration number in the training phase. This dynamic weight adapts during training, controlling the impact of the regularization term.

$$w_{\mathbf{z}}^{(t)} = \frac{1}{2} \cdot \left| \frac{D\left(\tilde{\mathbf{z}}^{(t-1)}\right)}{d\left(\mathbf{z}^{(t-1)}, \tilde{\mathbf{z}}^{(t-1)}\right)} \right|. \tag{3.10}$$

To summarize, our proposed architecture is shown in Fig. 3.1 with two stages. First, an autoencoder is trained with historical SCADA and PMU measurement data. Second, the WGAN is trained with the same data and the two regularization terms: (1) one incentivizes the WGAN to produce measurements that will pass the residual error test and (2) another to maximize the impact of the attack. More important features are described below, and the complete algorithm for our proposed FDIA is in Algorithm 2.

(i) The inputs for the generative network are actual power system measurements instead of random noise. This gives us control over the created measurements.

(ii) The generator is incentivized to generate measurements that will be different than the ones as input, causing an incorrect estimation of state variables and measurements.

(iii) The generated tampered measurements will have a small residual error, thus

passing the residual error test with high probability.

Note that our proposed approach can be easily formulated to deploy an attack on a specific area in the power system, as proposed in Liu and Li (2017). Specifically, a FDIA can be launched in a specific area by tampering the measurements within the area under attack and not modifying the sensor measurements at boundary buses. In this way, the attacker only has to get the sensor measurements in the specific area under attack, which would reduce the amount of collected data. For conciseness and sake of clarity, we will analyze our proposed FDIA in the complete power grid.

3.4 WGAN Guarantee

The last section presented a framework to create fake power system measurements to deploy a FDIA. However, to successfully deploy a FDIA without relying upon the underlying power system model, we need to be confident that our learned model will produce measurements that look legit so that the residual error test does not detect them. To show that our proposed framework converges to the underlying measurement distribution, we present mathematical proof that certifies the WGAN convergence to the measurement distribution, thus creating fake measurements that look real. The only requirement for this proof to work is to have data to train the WGAN.

Generative adversarial networks can be understood as minimizing a moment matching loss defined by a set of discriminator functions Zhang *et al.* (2018), mathematically

$$\min_{\nu \in \mathcal{G}} \left\{ \begin{array}{l} d_{\mathcal{F}}\left(\hat{\mu}_m, \nu\right) := \\ \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim \hat{\mu}_m} \mathbb{E}_{\tilde{x} \sim \nu} f\left(x\right) - f\left(\tilde{x}\right) + w_{\mathbf{z}} \cdot d\left(x, \tilde{x}\right) \end{array} \right\}, \tag{3.11}$$

where $\hat{\mu}_m$ is the empirical measure of the observed data (in this case the power system measurements), and $\mathcal{F}$ and $\mathcal{G}$ are the sets of discriminators and generators,

respectively. The practical WGANs take $\mathcal{F}$ as a parametric function class, that is, $\mathcal{F}_{nn} = \{f_\theta(x) : \theta \in \Theta\}$ where $f_\theta(x)$ is a neural network indexed by parameters θ that take values in $\Theta \subset \mathbb{R}^p$.

Notation and definitions. X denotes a subset of $\mathbb{R}^d$. For each continuous function $f : X \to \mathbb{R}$, we define the maximum norm as $\|f\|_\infty = \sup_{x \in X} |f(x)|$, and the Lipschitz norm $\|f\|_{Lip} = \sup\{|f(x) - f(y)| / \|x - y\| : x, y \in X, x \neq y\}$, and the bounded Lipschitz (BL) norm $\|f\|_{BL} = \max\left\{\|f\|_{Lip}, \|f\|_\infty\right\}$. The set of continuous functions on X is denoted by $C(X)$, and the Banach space of bounded continuous functions is $C_b(X) = \{f \in C(X) : \|f\|_\infty < \infty\}$.

Weak convergence. If $\mathcal{F}$ is discriminative, then $d_\mathcal{F}(\mu, \nu) = 0$ implies $\mu = \nu$. This means that the learned distribution is the same as the observed one. In reality, we cannot strictly get $d_\mathcal{F}(\mu, \nu) = 0$. Rather, we have $d_\mathcal{F}(\mu, \nu) \to 0$ for a sequence of ν_n and want to establish the weak convergence $\nu \rightharpoonup \mu$.

Theorem 1. *Let (X, d_X) be any metric space. If span$\mathcal{F}$ is dense in $C_b(X)$, we have $\lim_{n\to\infty} d_\mathcal{F}(\mu, \nu_n) = 0$ implies that the learned distribution ν_n weakly converges to the real observed distribution μ.*

In our context, the observed distribution μ corresponds to the set of observed power system measurements. Fig. 3.2 gives the intuition for the convergence proof. The learned distribution ν_n (in red) converges to the real one μ (in blue) as $n \to \infty$. In other words, the WGAN is learning to create samples that look as taken from the true observed distribution μ.

Proof. Given a function $g \in C_b(X)$, we say that g is approximated by $\mathcal{F}$ with error decay function $\epsilon(r)$ if for any $r \geq 0$, there exists $f_r \in span\mathcal{F}$ with $\|f_r\|_{\mathcal{F},1} \leq r$ such that $\|f - f_r\|_\infty \leq \epsilon(r)$. We note that $\epsilon(r)$ is a non-increasing function with respect to r. We know that the closure of $span\mathcal{F}$ is equal to the space of

bounded continuous functions $C_b(X)$, that is, $cl(span\mathcal{F}) = C_b(X)$, then we have $\lim_{r\to\infty}\epsilon(r) = 0$. Now denote $r_n := d_F(\mu,\nu_n)^{-\frac{1}{2}}$, $f_n := f_{r_n}$ and $w_{\mathbf{z}} = 1/r_n$. We have $|\mathbb{E}_\mu g - \mathbb{E}_{\nu_n} g| + w_{\mathbf{z}} \cdot d(x,\tilde{x}) \leq |\mathbb{E}_\mu g - \mathbb{E}_\mu f_n| + |\mathbb{E}_\nu g - \mathbb{E}_\nu f_n| + |\mathbb{E}_\mu f_n - E_{\nu_n} f_n| + w_{\mathbf{z}} \cdot d(x,\tilde{x}) \leq 2\epsilon(r_n) + r_n d_{\mathcal{F}}(\mu,\nu_n) + w_{\mathbf{z}} \cdot d(x,\tilde{x}) = 2\epsilon(r_n) + 1/r_n + w_{\mathbf{z}} \cdot d(x,\tilde{x})$. If $\lim_{r\to\infty} d_F(\mu,\nu_n) = 0$, we have $\lim_{r\to\infty} r_n = \infty$. Given that $\lim_{r\to\infty}\epsilon(r) = 0$, we prove that $\lim_{n\to\infty} |\mathbb{E}_\mu g - \mathbb{E}_{\nu_n} g| + w_{\mathbf{z}} \cdot d(x,\tilde{x}) = 0$. Since this holds true for any $g \in C_b(X)$, we conclude that ν_n weakly converges to μ. If $\mathcal{F} \subseteq BL_C(X)$ for some $C > 0$, we have $d_{\mathcal{F}}(\mu,\nu) \leq C d_{BL}(\mu,\nu)$ for any μ, ν. Because the bounded Lipschitz distance metrizes the weak convergence, we obtain that $\nu_n \to \mu$ implies $d_{BL}(\mu,\nu_n) \to 0$, and $d_{\mathcal{F}}(\mu,\nu_n) \rightharpoonup 0$. ■

Theorem 1 guarantees us that the learned distribution ν by the WGAN will converge to the observed one μ. This idea is depicted in Fig. 3.2. The blue points represent the real measurements, and the red ones represent the fake measurements. At the beginning, the red points are random because the WGAN is not trained ($n = 1$). However, as training progresses, the WGAN produces samples (red points) that look more similar to the blue ones. Ideally, the fake samples will be indistinguishable from the real ones. In other words, our model will create fake measurements that look like real ones. This means that the WGAN captures the underlying power system's interactions that produce the observed measurements.

3.5 Experiments

This section will show how we deploy FDIAs on power grids with our proposed WGAN framework without knowing their mathematical or physical model. To show the contributions and generality of our approach, we carried out extensive experiments on different power networks.

First, we train a WGAN with historical SCADA and PMU measurements to

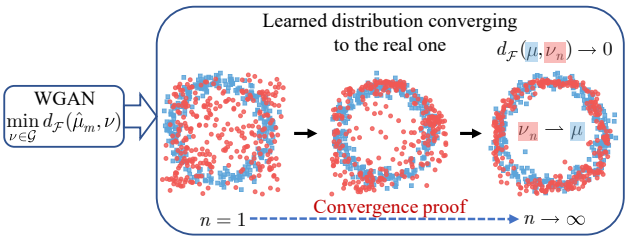


Figure 3.2: Intuition for the Wgan Convergence Proof to the True Observed Distribution.

demonstrate that the output of the WGAN converges to the true distribution of observed power system measurements, $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$. Note that the sampling rate of PMU measurements is faster than the sampling rate of SCADA measurements. We use PMU measurements alongside with SCADA measurements when the SCADA measurements are available. We will also show that the fake measurements will pass the residual error test, corroborating the aforementioned convergence theorem. Second, we show that the trained WGAN creates different measurements (and therefore states) from the actual ones. This will show that the regularization term works, and it is maximizing the FDIA impact. Next, we show that our proposed framework is more reliable than the model-based ones by showing that our WGAN produces more realistic measurements. This implies that our model is capturing the underlying power system model. Finally, an ablation study is carried out to show that embedding a surrogate state estimator model, $\mathbf{h}(\mathbf{x})$, improves the proposed framework to create tampered measurements that pass the residual error test. We carried out the aforementioned experiments in various test cases with similar results. Specifically, we use the small IEEE 9-bus test case to illustrate how our framework works. Then, we perform the same simulations in the IEEE 14-, 57-, 118-, and 300-bus networks to demonstrate that our proposed method scales well with larger power system networks.

3.5.1 Data Generation and Model Architecture

Data Generation

For both the 9- and 118-bus test cases, we consider all the active and reactive power flow measurements through transmission lines and transformers as SCADA measurements, and voltage magnitudes and angles as PMU measurements. The 9-bus network has 9 branches, which gives us 36 SCADA measurements and 18 PMU measurements. The measurements are arranged as follows: 1–9 correspond to the sent active power through branches, 10–18 correspond to the sent reactive power, 19–27 are the received active power measurements, 28–36 are the received reactive power on branches, 37–45 are the voltage magnitudes, and 46–54 are the voltage angles. The IEEE 118-bus network has 186 branches; thus, 980 measurements arranged as follows: 1–186 sent active power, 187–372 sent reactive power, 373–558 received active power, 559–744 received reactive power, 745–862 are the voltage magnitudes, and 863–980 are the voltage angles.

We obtain the power systems' measurements by solving L times the AC power flow under different load conditions using MATPOWER Zimmerman *et al.* (2011). To simulate the 24-hour fluctuation, we use the real yearly load data from the Electric Reliability Council of Texas (ERCOT) for 2021 Electric Reliability Council of Texas, (ERCOT) (2022). ERCOT reports 8 weather zones: *COAST*, *EAST*, *FWEST*, *NORTH*, *NCENT*, *SOUTH*, *SCENT*, and *WEST*. Fig. 3.3 depicts the load profiles of these zones for 2 days in 2021. For our simulations, we multiply each busload with the normalized loading parameter associated with a randomly selected area, γ, obtained from these realistic profiles. Similarly, we also adjust generation by scaling the generation profiles by multiplying them by the same loading parameter, γ, Ajjarapu and Christy (1992); Milano (2008). To make it more realistic, we

add white noise to all measurements according to the standard deviation associated with the measurement devices. That is, active power flow: 0.02 p.u., reactive power flow: 0.04 p.u., active power injection: 0.02 p.u., reactive power injection: 0.04 p.u., PMU voltage magnitude: 0.0001 p.u., and PMU voltage angle: 0.006 rad, according with Shahriar *et al.* (2018). Finally, if we do not find an AC power flow solution, we do not include it in the dataset. This data generation approach will give us rich data variety with the power system under different load conditions. The same procedure is used to generate data for the IEEE 14-, 57-, and 300-bus test cases.

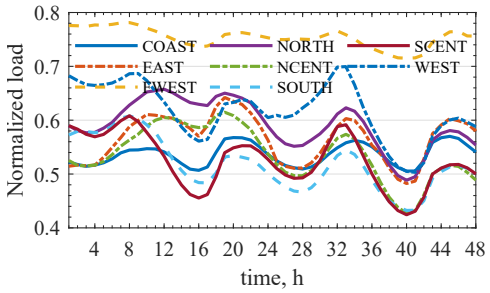


Figure 3.3: ERCOT Hourly Normalized Load Data for 2021.

Model Architecture

The architecture of our proposed WGAN model is inspired by the architecture of the DCGAN Radford *et al.* (2015) with the following modifications to adapt it to our power system data. Since the sensor measurement vectors are one-dimensional, we use fully connected layers instead of convolutional layers. The generator, G, consists of 5 layers with ReLU activation function for all layers except for the output, which uses tanh. The discriminator, D, is composed of 5 layers with LeakyReLU activations with the slope of the leak set to 0.2.

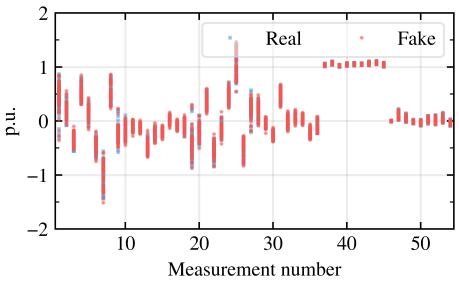
3.5.2 Learning the Implicit Power System Measurement Model

This section tests if the learned distribution by the WGAN converges to the true underlying power system measurement distribution, $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$. We train the WGAN according to Algorithm 2 with a dissimilarity weight $w_{\mathbf{z}} = 0.5$. We use the hyper-parameters from Arjovsky *et al.* (2017): $n_{critic} = 5$, learning rates $\alpha = 0.00005$ (for autoencoder, generator, and discriminator), clipping parameter $c = 0.01$, batch size $b = 64$, and Adam adaptive learning algorithm Kingma and Ba (2014). Also, we train the AE and the WGAN models for all test cases for 10 and 100 epochs, respectively. The normalized load from the Electric Reliability Council of Texas (ERCOT) for 2021 Electric Reliability Council of Texas, (ERCOT) (2022) contains hourly data for one year, which means that there are 8,760 load samples. From these 8,760 samples, we split the set into a training and a test dataset with 7,760 and 1,000 randomly chosen samples, respectively. This yearly data contains seasonal variation, so it captures the behavior of a real power system throughout the year. Note that both the AE and WGAN models are trained with this data, as indicated in Algorithm 2. Fig. 3.4a shows 100 measurement samples from the real dataset and 100 created fake measurements for the 9-bus test case. We can see in Fig. 3.4a generated fake measurements compared with real measurements from our dataset; the fake measurements (in red) follow the same pattern or distribution as the real ones (in blue); in fact, they overlap the real measurements, but they are not exactly the same. This means that the WGAN learned the true power system measurement distribution instead of memorizing the dataset. Note that Theorem 1 guarantees the model convergence with enough training data. In our numerical experiments, we trained our models by creating training and testing datasets of 7,760 and 1,000 samples, respectively. With these training datasets our models successfully learned

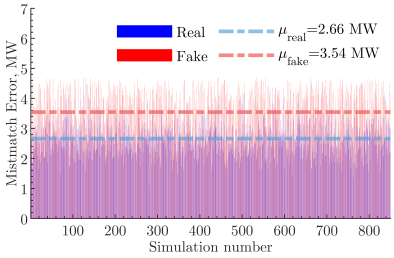
the underlying power system measurement distribution. Also note that the our procedure to create the dataset produces rich distributions of sensor measurements, Fig. 3.4a. For example, the measurement no. 1 has a range from 0p.u. to over 1p.u. (which corresponds to an active branch power flow measurement).

To assess if the trained WGAN learned the implicit power system measurement distribution, we carry out a power flow mismatch analysis, as follows. If we add power injection measurements in the set of measurements, the power flow balance at the i-th bus should be $\sum_{j\in\delta^+(i)} f^p_{(i,j)} + e_j = p^{\text{inj}}_i + e_i$, where $\delta^+(i)$ is the set of adjacent buses to bus i, $f^p_{(i,j)}$ is the power flow on branch (i,j), and e_j and e_i are the measurement errors associated to active power flow and injection, respectively. Under this setting, the power flow mismatch will not be zero due to measurement errors, that is, $\left|\sum_{j\in\delta^+(i)} f^p_{(i,j)} - p^{\text{inj}}_i\right| > 0$. We compute this power mismatch error $\left|\sum_{j\in\delta^+(i)} f^p_{(i,j)} - p^{\text{inj}}_i\right|$ for all the buses in the system for both real and fake tampered measurements. Fig. 3.4b shows the results, where each bar, blue for real and red for fake measurements, indicates the average power flow mismatch in the whole system for one simulation. In the same figure, we can see that the power flow mismatches of the real and tampered fake measurements are very close: 2.66 MW for the real measurements and 3.54 MW for the tampered fake measurements. This is remarkable because the WGAN does not know the power system topology, and it does not have information about which measurements should comply with the power flow balance. Yet, the WGAN produces fake tampered measurements that are within 1 MW, on average, with respect to the real measurements, as shown in Fig. 3.4b.

Including variable renewable sources such as wind and solar generation that vary significally from one day to the next could produce a more diverse sensor measurement distribution. To test this idea, we use the 9-bus test case, and we take the normalized wind and solar aggregated generation data from the RTS-GMLC Preston and Barrows



(a) Real vs. fake tampered measurements for the 9-bus test case. Note that the fake measurements look like the real ones.



(b) Power flow mismatch error for the real and fake measurements.

Figure 3.4: Learning an Implicit Power System Model with the Proposed Wgan Architecture for the 9-bus Test Case Using Real Load Profiles from Ercot Electric Reliability Council of Texas, (ERCOT) (2022).

(2018). Then, we include the wind generation on bus 5 and the solar generation on bus 6 with different penetration values. For a penetration of 30%, we can see the sensor measurement distribution in Fig. 3.5. This distribution looks a little bit wider than the one without VRES in Fig. 3.4a. Notice that both sensor measurement distributions look alike, which means that our original procedure to generate datasets creates rich sensor measurement distributions. Thus, the datasets for the remaining experiments will be created without adding VRES into the simulations.

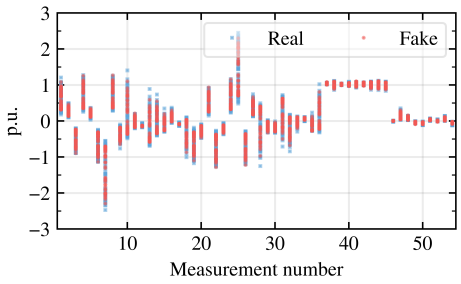


Figure 3.5: Sensor Measurement Distribution with Vres for the 9-bus Test System.

Analyzing Attack's Vector Sparsity

We can test the attack vector's sparsity by taking the absolute difference between the real and tampered measurement vectors, that is, $|\mathbf{z} - \tilde{\mathbf{z}}|$. To test this idea, we take the real and tampered measurements for the 9-bus test case, with $w_{\text{sparse}} = 0$, and we show two examples of specific sets of real and tampered measurements in Fig. 3.6. In the top part of the Figure, we can see the real and tampered measurements. In the inferior part of the Figure, we can see the absolute difference vectors, $|\mathbf{z} - \tilde{\mathbf{z}}|$. Note that even though $w_{\text{sparse}} = 0$, these vectors contain many zero values indicating the property of sparsity.

We train the WGAN following the same procedure for the 9-bus test system with the addition of the sparsity regularizer with a weight of 0.5, that is, $w_{\text{sparse}} = 0.5$. To test the sparsity of the results, we follow the same experiment design from the last example. Specifically, we take the real and tampered measurements for the 9-bus test case, and we show two examples of specific sets of real and tampered measurements in Fig. 3.7. In the top part of the Figure, we can see the real and tampered measurements. In the inferior part of the Figure, we can see the absolute difference vectors, $|\mathbf{z} - \tilde{\mathbf{z}}|$. As expected, when sparsity is explicitly taken into account, the attack vectors (absolute difference vectors in Fig. 3.7) present more sparsity than those

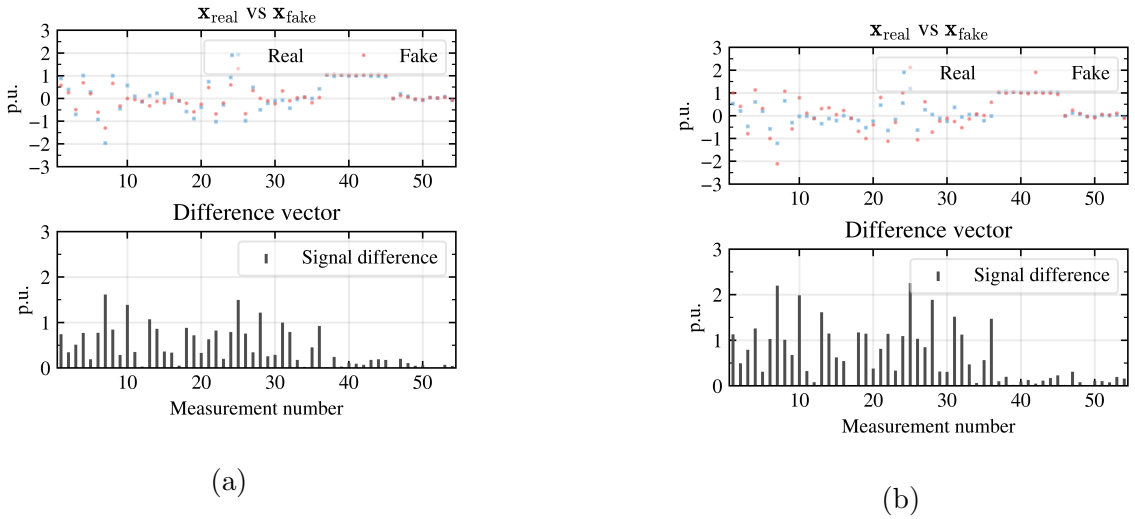


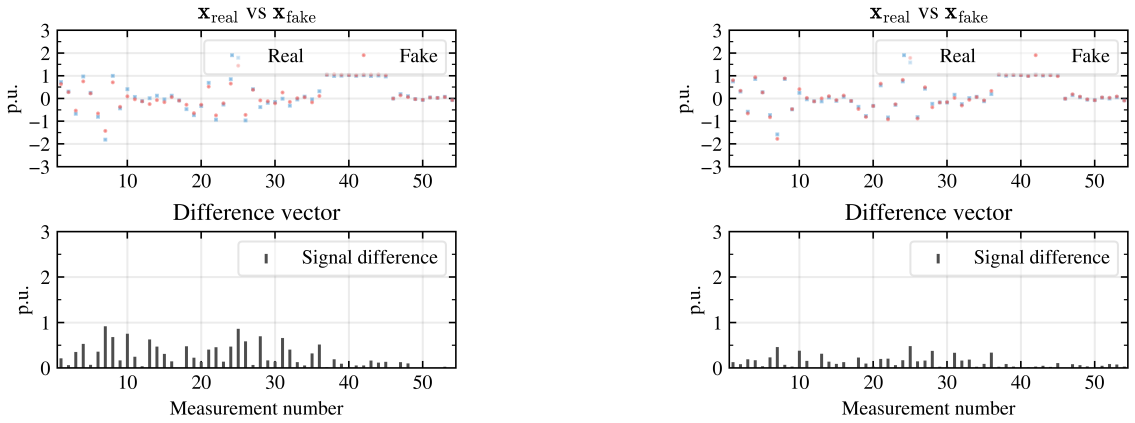
Figure 3.6: Examples of Absolute Difference Vectors.

in Fig. 3.6, where no sparsity is expressly considered in the model. However, the differences between real and tampered measurement vectors for the sparse FDIA are smaller than the FDIA that does not explicitly take into account the sparsity.

The model's results without including sparsity, $w_{\text{sparse}} = 0$, present sparsity and produce more changes in the tampered measurements. Thus, the remaining experiments will be done without explicitly including sparsity.

Analyzing Attack Vector

We can assess an attack vector's impact by taking the absolute difference between the real and tampered measurement vectors, that is, $|\mathbf{z} - \tilde{\mathbf{z}}|$. To test this idea, we take $1,000$ real and tampered measurements for the 9-bus test case, and we show two examples of specific sets of real and tampered measurements in Fig. 3.6. In the top part of the Figure, we can see the real and tampered measurements. In the inferior part of the Figure, we can see the absolute difference vectors, $|\mathbf{z} - \tilde{\mathbf{z}}|$. Note that in the $1,000$ samples, the mean magnitude of the attack vector is 15.05 units. Also, the



(a) (b)

Figure 3.7: Examples of Absolute Difference Vectors with the Sparsity Regularizer With $w_{\text{sparse}} = 0.5$.

attack vector, in specific sensor measurements, dramatically changes the real values. Under this context, the operator could take wrong corrective actions that will interfere with the correct and safe operation of the electric grid. This means that the attack will damage the system and lead to catastrophic events.

3.5.3 Deploying FDIAs without Power System Knowledge

In the last section, we showed that a WGAN can learn the power system measurement distribution. This section shows how we deploy a FDIA with our proposed framework, which is given by equation 3.9 and equation 3.10.

Deploying a FDIA with fake tampered measurements

Our objective is to create fake tampered measurements $\tilde{\mathbf{z}}$ that generate estimated measurements and state variables as different as possible from the real ones. At the same time, for an attack to be successful, these measurements should pass the residual

error test. Fig. 3.9 shows an instance of a real measurement vector and a created fake one for the 9-bus test network. The fake tampered measurements are within the historical range from the dataset and look similar to the real ones. However, they produce significant changes in voltage magnitudes v and voltage angles δ with respect to the real states, as shown in Fig. 3.10. Furthermore, the fake measurements pass the test in equation 3.2, which means that the control center will not notice the FDIA.

Comparison against other FDIA methods

To assess the advantages and differences between our proposed model-free FDIA framework, we compare it against the model-based FDIA presented in Hug and Giampapa (2012) and described by equation 3.3—we will refer to this FDIA as Method 1. This model-based attack has the same residual error as the original measurements as proven in Hug and Giampapa (2012). However, the Method 1 produces measurements that are out of the historical range from the historical measurements.

To prove the last point, we perform the following experiment. We use the fake vector in Fig. 3.10, where we can see that the voltage magnitude in bus 5 goes from 1 to 1.05 p.u. We use Method 1 to tamper the state $v_5 = 1.05$ p.u. using equation 3.3. Fig. 3.9 shows the real measurements (in blue), the created tampered measurements by our proposed framework (in red), the created tampered measurements by Method 1, and the historical measurement range from our data generation (gray bar). In the same Figure, we see that the created measurements by the WGAN are within or very close to the historical range. In contrast, some tampered measurements by Method 1 are far away from the real historical measurements. In specific, we see in Fig. 3.9 that measurements 18 and 36 show a large distance from the historical range. The key observation is: Even though Method 1 produces measurements with the same residual error as the real ones, these measurements will still look suspicious.

The power system operator would realize that the tampered measurements 18 and 36 are outliers with respect to the historical ones, as shown in Fig. 3.9. In contrast, in the same Figure, we can see that our fake tampered measurements are within the range of historical measurements and also pass the residual error test (for a confidence of $p = 0.95$). Thus, making them less suspicious for the power system operator. This means that our attack design is more advantageous at the stealth level.

We also carried out a sensitivity analysis for different confidence values p. In this sensitivity analysis, we compare our method against three techniques in the literature: Method 1 introduced in Hug and Giampapa (2012), Method 2 from Chin *et al.* (2017), and Method 3 proposed in Liu and Li (2017). This sensitivity analysis is carried out with the residual error test. Thus, the results only depend on the residual error produced by the FDIA approaches. In other words, the range of historical measurements does not affect the success rate. Methods 1 and 2 produce the same residual error as the real measurements; this means that if the real measurement passes the residual error test, the tampered measurements by these methods will pass as well. Method 3 is an attack on a specific area, and we chose to delimit this area by the buses 5 and 6. An important characteristic of this technique is that the residual error of the tampered measurements can be lower than the real residual. The authors in Liu and Li (2017) attribute it to the fact that the tampered measurements will be more consistent (i.e., free of noise errors); thus, reducing the overall residual error.

To compare these methods, we made $1,000$ simulations with the same procedure described in section 3.5.1, and we tamper the real noisy measurements with our proposed approach and Methods 1, 2, and 3. For a given confidence value p, we compute its corresponding threshold $\tau = \chi^2_{k,p}$, and obtain the probability of each measurement to pass the residual error test for the specified threshold, that is, $\Pr\left(J(\mathbf{z}) \geq \tau\right)$. We repeat this process for each simulation and each aforementioned method, and we ob-

tain the success rate of passing the residual error test. This is the probability of the simulations to pass the error test, and we call it p_{pass}. We repeat this experiment for several values $p \in (0, 1)$, and the result is shown in Fig. 3.8. We can see that as the threshold τ increases, the probability to pass the residual error test p_{pass} increases as well. Given that Methods 1 and 2 (in brown and purple, respectively) tampered the measurements such that the residual error is the same as the real one (in blue), they (almost) follow perfectly the real curve. Method 3 (in green) is close to the real curve but just slightly above due to the behavior of this technique, as we previously explained. Note that Methods 1 and 2 produce the same p_{pass} as the real noisy measurements in Fig. 3.8. This is because both methods are guaranteed to have the same residual error as the real noisy measurements by design, as indicated in equation 3.3 (see proof in Hug and Giampapa (2012)).

It is important to note that we trained our model with noisy measurements, and the method did not have access to the underlying power system model. The key finding is that despite using only noisy measurements, our approach produces tampered measurements with lower residual errors, outperforming all other methods. We ascribe this due to the regularization term that contains the AE in equation 3.9, $\|\tilde{\mathbf{z}} - \text{AE}(\tilde{\mathbf{z}})\|_2^2$. As discussed in Section 3.3.2, an autoencoder has a denoising effect on the on the noisy measurements. This will be proved with an ablation study in 3.5.5. A summary of the qualitative traits of each of the aforementioned methods is shown in Table 3.1, where it is shown that our proposed algorithm is the only one that tampers measurements so that they are within the historical range.

Comparison Against Other Model-free FDIA Method

To make a fair comparison, we train our proposed model with the same methodology indicated before with the difference that we use the DC power flow model as the

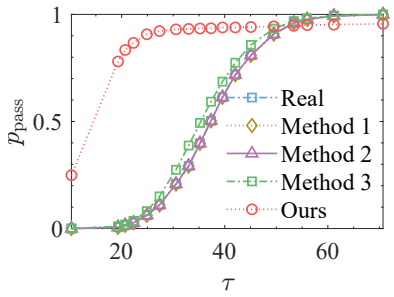


Figure 3.8: Comparison of Passing the Residual Error Test with Different Methods for the 9-bus Test Case.

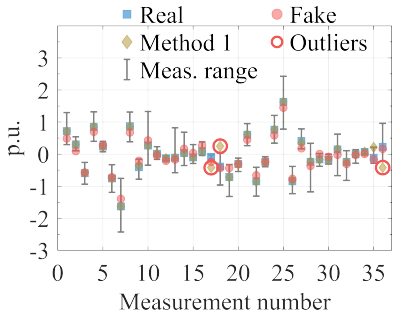


Figure 3.9: Comparison of the Tampered Measurements by the Model-based Method 1 Hug and Giampapa (2012) With Our Model-free Approach for the 9-bus Test Case.

work in Mohammadpourfard *et al.* (2020) does. This framework requires normal and tampered measurements to train a conditional adversarial network (cGAN). However, the work in Mohammadpourfard *et al.* (2020) does not clearly indicate how the dataset of tampered measurements is obtained. For simplicity, we use the well-known FDIA proposed in Hug and Giampapa (2012) to create the dataset of tampered measurements. We evaluate both approaches on the 14-bus test. For a given confidence value p, we compute its corresponding threshold $\tau = \chi^2_{k,p}$, and obtain the probability of each measurement to pass the residual error test for the specified threshold, that is,

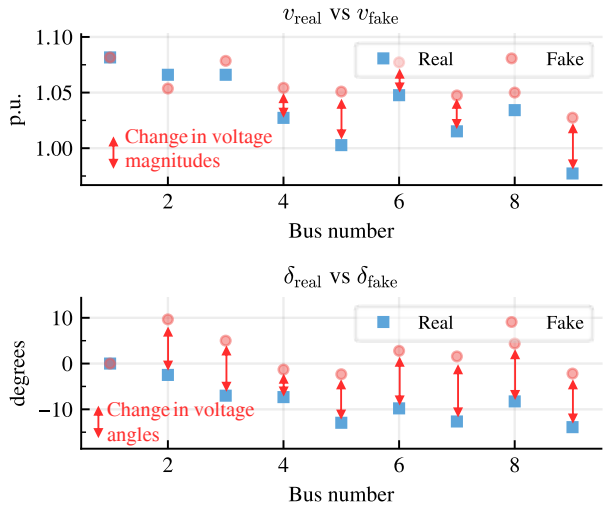


Figure 3.10: Example of a Real and a Fake Measurement Vector for the 9-bus Test Case.

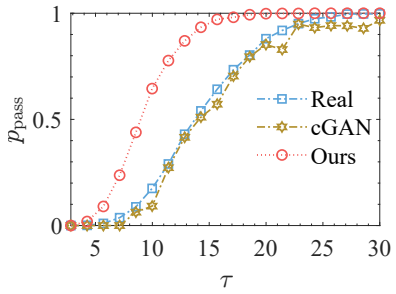


Figure 3.11: Comparison of Passing the Residual Error Test with The cGAN, Mohammadpourfard *et al.* (2020), For the 14-bus Test Case.

$\Pr(J(\mathbf{z}) \geq \tau)$. We repeat this process for each simulation and each aforementioned method, and we obtain the success rate of passing the residual error test. This is the probability of the simulations to pass the error test, and we call it p_{pass}. We repeat this experiment for several values $p \in (0, 1)$, and the result is shown in Fig. 3.11. We carry out the same experiments for the IEEE 9-, 57-, 118-, and 300-bus test cases for a confidence value $p = 0.95$. The results are shown in Table 3.2.

Validate Scalability of the Proposed Approach

Finally, we show that our approach scales to bigger power system networks. To demonstrate it, we test our model-free FDIA on the IEEE 118-bus network. The created fake tampered measurements pass the residual error test, and Fig. 3.12 shows that the created fake measurements provoke significant changes in the voltage angles, leading to a successful FDIA.

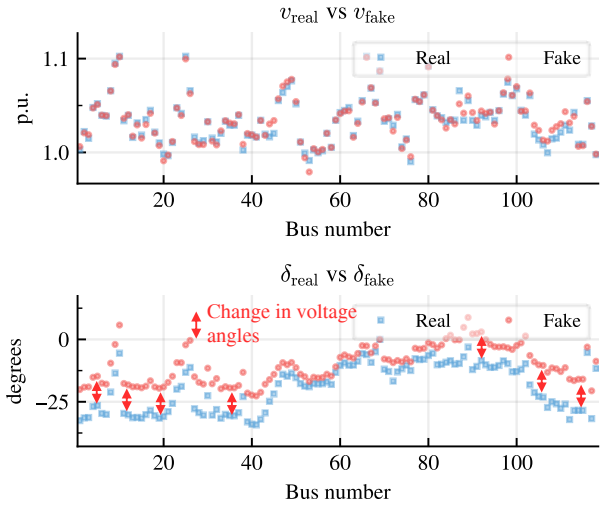
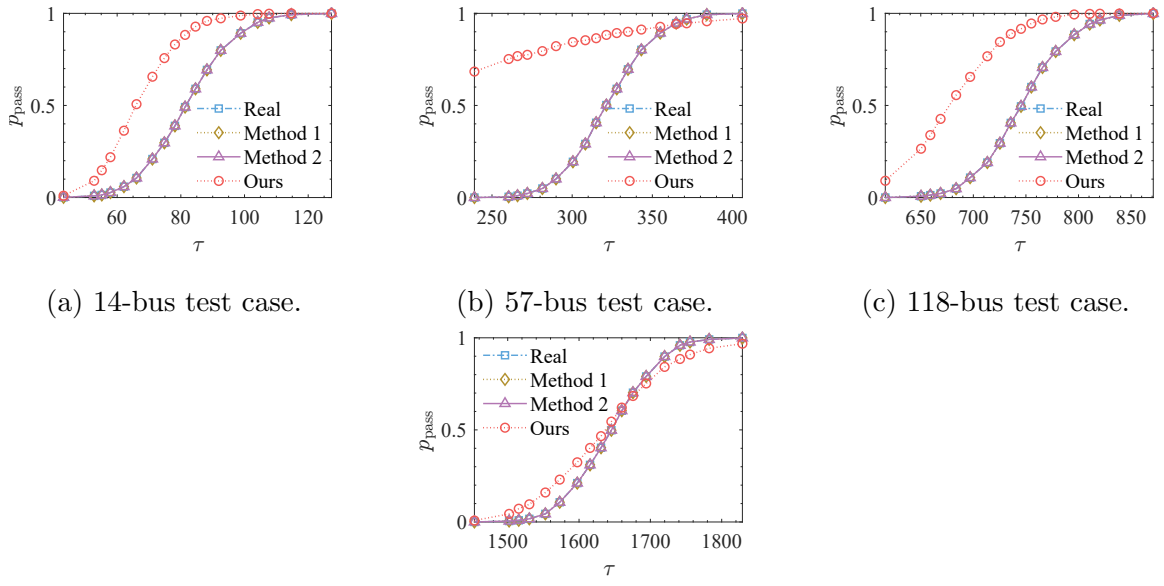


Figure 3.12: Example of a Real Vs a Fake Measurement for the 118-bus Test Case. Note That the Fake Measurements Produce Different States.

Also, a sensitivity analysis, like the one in the previous section, is carried out for the IEEE 9-, 14, 57-, 118-, and 300-bus test cases, and the results are shown in Fig. 3.13. In the same Figure, we can see that our FDIA method outperforms the ones proposed in the literature.

Finally, we validate the scalability of our proposed approach. As previously mentioned, the AE and the WGAN models for all the test cases are trained for 10 and 100, respectively. The number of training samples and the number of iterations for



(a) 14-bus test case.

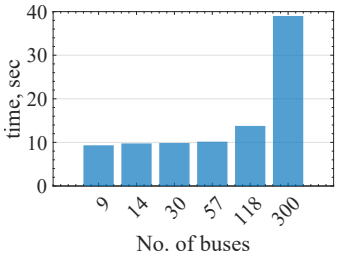
(b) 57-bus test case.

(c) 118-bus test case.

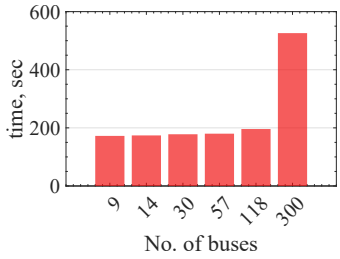
(d) 300-bus test case.

Figure 3.13: CDF Comparison for Many Test Cases.

all test cases are fixed since we used real yearly load data from the Electric Reliability Council of Texas (ERCOT) for 2021 Electric Reliability Council of Texas, (ERCOT) (2022). Also, the number of layers is fixed to be 5 for both the generator and discriminator for all the experiments. The only component that varies is the dimensionality, which depends upon the power system size. Thus, our proposed approach presents good scalability with respect to the power system size. We can test this by measuring the training times for the AE and WGAN models. Fig. 3.14 shows such training times. We can see that training the surrogate state estimator (i.e., AE) for 10 epochs takes less than 40 sec for all test cases. Training the WGAN model for 100 epochs takes less than 530 sec for all test cases. We can see that the training times for the models' convergence for 1 year of data are low. Thus, our proposed attack could be easily deployed in real-world settings.



(a) Auto-encoder training times.



(b) WGAN training times.

Figure 3.14: Training times for Ae and Wgan for Different Test Cases.

3.5.4 Comparison of Different Defenses

The Chi-squared test could be, in some cases, inaccurate due to the approximations of errors by residuals Abur and Exposito (2004). So, in this section, we show how our proposed algorithm performs against more sounding defenses. In the literature, there exist numerous defenses with different traits. For example, defenses that do not use temporal correlations and ones that make use of them. In the realm of defenses that exploit temporal patterns to detect FDIAs, there are works such as the moving-target defense (MTD) Zhang *et al.* (2019); Lakshminarayana and Yau (2020) or the work in He *et al.* (2017). However, our proposed FDIA scheme does not take into account inter-temporal correlation, so it would be unfair to test our attack against such defenses. Thus, in this section, we choose defenses that utilize data measurements at a specific time interval to detect spurious data. Specifically, we test our proposed attack against the largest normalized residual statistical test (LNRT) Abur and Exposito (2004); Zhao and Mili (2018) and a recent deep learning-based detector that consists of an adversarial autoencoder Zhang *et al.* (2020).

Largest normalized residual statistical test (LNRT)

The LNRT is more robust than the classical Chi-squared test for bad data detection and identification Abur and Exposito (2004); Zhao and Mili (2018). The normalized value of the residual for the measurement i can be computed as $r_i^{\text{norm}} = \frac{|r_i|}{\sqrt{\Omega_{ii}}}$, where $\sqrt{\Omega_{ii}}$ is the diagonal entry in the residual covariance matrix. This normalized residual entry has a standard normal distribution, that is, $r_i^{\text{norm}} \sim \mathcal{N}(0, 1)$. Then, the largest element in the set $\{r_i^{\text{norm}}\}_{i=1}^{M}$ is compared against a chosen threshold to decide if bad data is presented. If this threshold is set to 3, then the confidence level is 99.7%. We carry out this test for the 14-bus test system for each real and fake measurement, and the results are shown in Fig. 3.15, where the average is 99.75% for real measurements and 99.79% for tampered measurements with our proposed method. We carry out the same experiments for the IEEE 9-, 57-, 118-, and 300-bus test cases for a confidence value $p = 0.997$. The results are shown in Table 3.3.

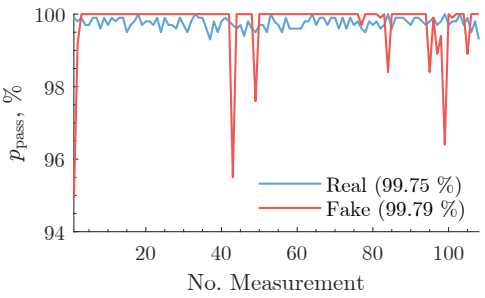


Figure 3.15: Largest Normalized Residual Statistical Test for the 14-bus Test System.

Deep Learning-based detector

There are recent learning-based detectors to detect FDIAs. The work in Zhang *et al.* (2020), for example, proposed a scheme that consists in an adversarial autoencoder (AEE). The AAE network is trained in three stages: the reconstruction phase, the adversarial phase, and the supervised phase. For a model-based FDIA, this AEE

has a detection accuracy of 96.25% and 97.85% for the 13- and 123-bus distribution networks. We test this defense against our proposed model-free FDIA for the IEEE 9-, 14-, 57-, 118-, and 300-bus test cases, and the results are shown in Table 3.3. In this table, we can see that our proposed approach has a lower success rate for the AAE defense than for the Chi-squared and LNRT. Nonetheless, our method still exhibits a high success rate (above 80%) for all the tested cases.

3.5.5 Ablation Study

This section presents an ablation study to show the impact of the SE's surrogate model in the proposed framework. The experiment design is similar to the one presented in previous sections. We made $1,000$ simulations with the same procedure described in section 3.5.1. For a given confidence value p, we compute its corresponding threshold $\tau = \chi^2_{k,p}$, and obtain the probability of each measurement to pass the residual error test for the specified threshold, that is, $\Pr\left(J(\mathbf{z}) \geq \tau\right)$. We repeat this process for each simulation for the real and proposed framework with and without AE for the 9-bus test case. Next, we obtain the success rate of passing the residual error test, p_{pass}. We repeat this experiment for several values $p \in (0, 1)$, and the result is shown in Fig. 3.16. In the same Figure, we can see that the model without the AE has a lower probability of passing the residual error test throughout all the thresholds. We can also see that the model without the AE (green line) always have around the same or lower probability of passing the residual error test than the real measurements. As discussed in Sections 3.3.2 and 3.5.3, whereas the model with the AE has a denoising effect the model without the AE can only learn from the noisy measurement data. We carry out the same experiments for the IEEE 14-, 57-, 118-, and 300-bus test cases for a confidence value $p = 0.95$. The results are shown in Table 3.4, which shows that the model with the AE has a higher success rate than

the one without it.

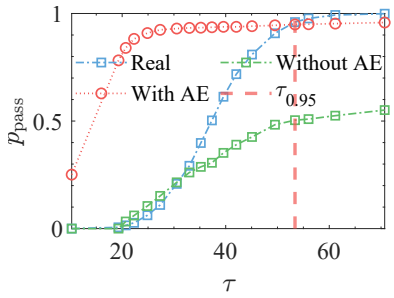


Figure 3.16: Probability of Passing the Residual Error Test for 9-bus Test Case with and Without AE.

3.6 Conclusion

We presented an architecture to create tampered measurement vectors to carry out a FDIA without knowing the power system underlying information. The architecture is framed into an optimization framework that considers the WGAN loss function and two regularization terms to control the attack measurement vectors. We validated our proposed framework with several power systems, in which we created fake measurements to create a bad data injection attack without knowing the underlying power system model. These fake measurements passed the residual error test to detect bad data and gave completely wrong estimated state variables and measurements, which would compromise the electric grid's reliability. This work proves that for an attacker, it is not required to have access to all power system information. Thus, more research is needed to keep power systems safe from these attacks.

Algorithm 2: Training Process of the Proposed Scheme to Create Tampered Measurements to Deploy a FDIA.

Inputs : Dataset $\mathcal{M} = \{\mathbf{z}_i \in \mathbb{R}^m\}_{i=1}^{L}$, batch size b, number of iterations of the critic per generator iteration n_{critic}, generator and discriminator learning rates α, clipping parameter c.

Output: Generator (G) network.

Train an AE with the real measurements from the dataset $\mathcal{M}$ and the loss function $\mathcal{L} = \|\mathbf{z} - \mathrm{AE}(\mathbf{z})\|^2$.

for *number of training iterations* **do**

  for $k = 1, \ldots, n_{critic}$ **do**

    Sample a minibatch of b samples $\left\{\mathbf{z}_D^{(1)}, \ldots, \mathbf{z}_D^{(b)}\right\} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(b)}\right\} \sim \mathbb{P}_r$ from the measurement dataset $\mathcal{M}$.

    Sample a different minibatch of b samples and create a minibatch of fake measurements $\left\{G\left(\mathbf{z}_G^{(1)}\right), \ldots, G\left(\mathbf{z}_G^{(b)}\right)\right\} = \left\{\tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{z}}^{(b)}\right\} \sim \mathbb{P}_g$.

    Train the critic (or discriminator): Gradient ascent on the critic:

    $$\max_{D \in \mathcal{D}} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_g} D(\mathbf{z}) - D(\tilde{\mathbf{z}}).$$

    Clip discriminator weights in the range $[-c, c]$.

  end

  Sample real and fake measurements: $\left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(b)}\right\} \sim \mathbb{P}_r$ and $\left\{\tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{m}}^{(b)}\right\} \sim \mathbb{P}_g$.

  Train the Generator: Gradient descent on generator:

  $$\min_{G} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_g} \Big[D(\mathbf{z}) - D(\tilde{\mathbf{z}}) + w_d \cdot d(\mathbf{z}, \tilde{\mathbf{z}}) + \|\tilde{\mathbf{z}} - \mathrm{AE}(\tilde{\mathbf{z}})\|_2^2 \Big].$$

end

Get generator G that creates tampered measurements.

Table 3.1: Comparison of Different FDIAs.

	Ours	M1	M2	M3
Is power system model needed?	✗	✓	✗	✓
Same residual as originals?	✗	✓	✗	✓
Measurements needed to deploy attack	All or Area under attack	All	All	Area under attack
Tampered measurements within historical range?	✓	✗	✗	✗

Table 3.2: Comparison of Passing the Residual Error Test with The cGAN, Mohammadpourfard *et al.* (2020).

Test Case	Success Rate (%)	
	Ours	cGAN
9-bus	95.5	92.7
14-bus	95.7	95.78
57-bus	89.3	93.6
118-bus	97	91.4
300-bus	91	93.1

Table 3.3: Comparison of Different Defense Mechanisms Against A FDIA. $^{*}p = 0.997$, $^{\dagger}p = 0.95$.

Test Case	Success Rate (%)		
	LNRT*	Chi-squared†	AEE Zhang *et al.* (2020)
9-bus	98.3	95.5	93
14-bus	99.79	95.7	92.6
57-bus	92	89.3	86.5
118-bus	99.4	97	92.3
300-bus	93.5	91	84.4

Table 3.4: Impact of Including an AE.

Test Case	Success Rate (%)	
	With AE	Without AE
9-bus	95.5	63.7
14-bus	95.7	81.1
57-bus	89.3	61
118-bus	97	54.33
300-bus	91	70.6

Chapter 4

ATTACK ON THE DC STATE ESTIMATOR WITHOUT SYSTEM INFORMATION AND PERFORMANCE GUARANTEE

The previous chapter introduced a model-free attack using an AE and a WGAN to create tampered measurements. However, this work lacks a formal analysis of the mechanism of the attack. While the presented model-free is innovative, relaxing the assumption on the attacker's system knowledge, it is apparent that there are still limitations that warrant further study. For example, fundamental questions are not answered, such as why the AE reduces the residual error. They also don't include assessing the model hyper-parameters selection to regulate the attack's aggressiveness and success rate.

To address such limitations, in this chapter, we introduce a FDIA that (1) does not need any grid information, (2) does not require a dataset of perturbed measurements, (3) maximizes the attack impact while being stealthy, and (4) has formal performance guarantees. Our proposed framework is composed of (i) a GAN to create realistic, high-quality samples, (ii) an attack regularization term to maximize the attack impact, (iii) a residual error regularization term based on an autoencoder that ensures the attack's stealthiness, and (iv) formal analysis on the framework's performance.

Simulations on the IEEE 14-bus, 118-bus, RTS-GMLC, and 200-bus Illinois synthetic model test cases verify the performance of the proposed model-free FDIA. Also, to contrast the differences and advantages between our approach and the existing ones in the literature, we carry out comparisons between our proposed FDIA and three other successful methods reported in Hug and Giampapa (2012); Chin *et al.* (2017); Mohammadpourfard *et al.* (2020); Costilla-Enriquez and Weng (2022).

4.1 Problem Formulation

To show the proposed model-free FDIA attack, we first review the model-based approaches based on DC state estimation.

4.1.1 State Estimation with DC Power Flow Model

State estimation (SE) infers the state variables (i.e., voltage angles and voltage magnitudes) $\mathbf{x} = (x_1, \ldots, x_n)$ from a set of measurements $\mathbf{z} = (z_1, \ldots, z_m)$ Wood *et al.* (2013), where n is the number of buses or nodes in the grid, and m is the number of measurements. Mathematically, we can describe the problem as $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}$, where $\mathbf{H}$ is the physical (linear) relationship between state variables and measurements, and $\mathbf{e}$ is a vector representing white noise from the collected measurements (e.g., SCADA or PMU). In practice, measurements are collected and sent to the power system operator, which obtains the estimated states $\hat{\mathbf{x}}$ by solving Tarali and Abur (2012); Weng *et al.* (2017):

$$\begin{aligned} \hat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}) &= (\mathbf{z} - \mathbf{H}\mathbf{x})^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x}), \\ &= \sum_{i=1}^{m} \frac{(z_i - \mathbf{H}_i \mathbf{x})^2}{\sigma_i^2}, \end{aligned} \tag{4.1}$$

where $\mathbf{H}_i$ is the i-th row of the matrix $\mathbf{H}$. However, the vector of measurements $\mathbf{z}$ may contain bad or wrong data due to telecommunication failures, meter errors, or even FDIAs Abur and Exposito (2004); Wang *et al.* (2020). To confidently estimate the states, the SE possesses a Bad Data Detector (BDD) module to detect and filter suspicious data.

Bad Data Detector (BDD)

The measurement errors are assumed to follow a Gaussian distribution $e_i \sim \mathcal{N}(0, \sigma_i)$ Abur and Exposito (2004) (where σ_i is the standard deviation of the i-th measurement).

Therefore, the squared measurement residual error $\mathbf{r} = \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$ follows a Chi-square distribution χ_k, where k represents the number of independent variables in the power system, and $\hat{\mathbf{z}} = H\hat{\mathbf{x}}$ is the vector of estimated measurements. Then, the presence of errors in the measurements can be detected with the Chi-square test (or residual error test) Abur and Exposito (2004); Weng *et al.* (2016). This test works as follows: (i) Select the detection confidence probability p (e.g., 0.95), and compute its associated threshold value $\tau = \chi^2_{(m-n),p}$ with $p = \Pr\left(J\left(\hat{\mathbf{x}}\right) \leq \chi^2_{(m-n),p}\right)$. (ii) Compute the normalized measurement error $J(\hat{\mathbf{x}}) = \sum_{i=1}^{m} \left(z_i - \mathbf{H}\hat{x}_i\right)^2 / \sigma_i^2$. (iii) If the inequality in equation 4.2 holds, bad data will be suspected, or else the measurements are assumed to be free of bad data.

$$J\left(\hat{\mathbf{x}}\right) \geq \tau \tag{4.2}$$

4.1.2 Model-based FDIAs

A FDIA modifies the estimated states $\hat{\mathbf{x}}$ or measurements $\hat{\mathbf{z}}$ by changing the original measurements $\mathbf{z}$ with a maliciously tampered measurement vector, that is, $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$, where $\mathbf{a}$ is an attack vector. The attacker designs this attack vector to compromise the system's reliability by creating a wrong state estimate. For a FDIA to be successful, it must circumvent the bad data detector Eq. (3.2) He *et al.* (2017). The assumptions in the literature for a model-based FDIA about the attacker's knowledge are the following Liu *et al.* (2011); Hug and Giampapa (2012); Zhang and Sankar (2016): (1) the attackers can intercept and alter the power system measurements that are used to obtain the estimated states in the grid; (2) the attacker has access to the power system model, which includes transmission line parameters and topology information; and (3) the attacker possesses the SE model or can obtain the estimated states of the network. Under these strong assumptions, the attacker could launch a perfect FDIA Wang *et al.* (2020). In this perfect FDIA, the attacker can define the

attack vector as $\mathbf{a} = \mathbf{Hc}$, where $\mathbf{c}$ is an arbitrary vector of changes in the estimated states. In this scenario, if the original measurements $\mathbf{z}$ can pass the residual-based bad data detector test in Eq. (3.2), the corrupted measurements $\mathbf{z}_a$ will also pass this test Hug and Giampapa (2012).

4.2 Proposed Method

In this section, the model-free attack from last chapter is introduced.

4.2.1 Creating Realistic and High-Quality Samples

As in the previous chapter, we will use the WGAN. These models have two distinct neural networks, a generator G and a discriminator D (or critic). The minimax objective of the WGAN is

$$\min_{G} \max_{D \in \mathcal{D}} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\boldsymbol{\lambda} \sim \mathbb{P}_\lambda} \left[D\left(\mathbf{z}\right) - D\left(G(\boldsymbol{\lambda})\right) \right], \tag{4.3}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions Arjovsky *et al.* (2017); $\mathbb{P}_r$ is the real data distribution; $\boldsymbol{\lambda}$ is known as the latent space, and it is sampled from an independent distribution $\mathbb{P}_\lambda$. The generator G learns the real distribution $\mathbb{P}_r$, which, in our context, this real distribution is the set of historically observed measurements $\mathcal{Z} = \{\mathbf{z}_i \in \mathbb{R}^m\}_{i=1}^{L}$ (where L is the number of elements in the dataset), where $\mathbf{z}_i = \mathbf{H}\mathbf{x}_i + \mathbf{e}_i$.

4.2.2 Minimizing the Residual Error in the State Estimator

To gain trust from the power system operator, the created tampered measurements, $\tilde{\mathbf{z}} = G(\boldsymbol{\lambda})$, must pass the residual error test in Eq. (4.2). This normalized residual error ε for the tampered measurements is given as

$$\varepsilon = \left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\right) = \sum_{i=1}^{m} \frac{\left(z_i - \mathbf{H}_i\hat{\mathbf{x}}\right)^2}{\sigma_i^2}. \tag{4.4}$$

As Eq. (4.2) suggests, the smaller the residual error ε, the bigger the probability of passing the test for a given tampered measurement. In other words, a given vector of tampered measurements, $\tilde{\mathbf{z}}$, should produce a small residual error. However, in the model-free attack setting, the attacker does not know the state estimator model or system parameters $\mathbf{H}$. Thus, for an attacker, it is not possible to compute the residual error ε because it depends on the system model, $\mathbf{H}$. Nonetheless, based on universal power system knowledge, we know there exists a mapping function from measurement space to state space $h : \mathbb{R}^m \mapsto \mathbb{R}^n$ and mapping back from state space to measurement space $\mathrm{SE} : \mathbb{R}^n \mapsto \mathbb{R}^m$ that makes the squared residual error zero: $\varepsilon = \|\boldsymbol{z} - h(\mathrm{SE}(\boldsymbol{z}))\|^2$. Then, it is possible to learn such implicit functions through data with an autoencoder.

Based on the hidden dimension size, a trained AE with real measurements will minimize the residual error in Eq. (4.4). A formal analysis is presented in the following section on Theorem 4. Once the autoencoder is trained (denoted as AE^*), the AE loss function can be embedded into Eq. (4.3) to incentivize the generation of tampered measurements that will produce similar estimated measurements and thus lower the residual error. This can be done by adding the regularization term $\|\tilde{\mathbf{z}} - \mathrm{AE}^*(\tilde{\mathbf{z}})\|_2^2$ in Eq. (4.3):

$$\min_{G} \max_{D \in \mathcal{D}} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\boldsymbol{\lambda} \sim \mathbb{P}_\lambda} \Big[D(\mathbf{z}) \quad - \quad D(\tilde{\mathbf{z}}) \quad + \quad \lambda^{\mathrm{AE}} \quad \cdot \quad \|\tilde{\mathbf{z}} \quad - \quad \mathrm{AE}^*(\tilde{\mathbf{z}})\|^2 \Big], \quad (4.5)$$

where $\tilde{\mathbf{z}} = G(\boldsymbol{\lambda})$ and λ^{AE} is the autoencoder regularization weight.

4.2.3 Maximize the FDIA Impact

To successfully deploy an FDIA, we want to incentivize the generator to construct measurements with a high attack vector norm $\|\boldsymbol{a}\|^2$. This will provoke the SE with a high likelihood to produce erroneous estimated states, the main objective in a FDIA.

To accomplish this, we can incentivize the model to generate such fake measurements with the regularization term $\lambda^{\text{attack}} \cdot d(\mathbf{z}, \tilde{\mathbf{z}})$ in Eq. (3.9), where $\tilde{\mathbf{z}} = G(\mathbf{z})$, $d(\mathbf{z}, \tilde{\mathbf{z}}) = \|\boldsymbol{a}\|^2$ represents attack vector size, and λ^{attack} is a hyper-parameter that represents the weight regularization term. This regularization term incentivizes the WGAN to produce a tampered measurement vector $\tilde{\mathbf{z}}$ that will generate completely wrong estimated measurements. This leads to the following loss function

$$\min_{G} \max_{D \in \mathcal{D}} \quad \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_r} \mathbb{E}_{\tilde{\mathbf{z}} \sim \mathbb{P}_g} \Big[D(\mathbf{z}) - D(\tilde{\mathbf{z}}) + \lambda^{\text{AE}} \cdot \|\tilde{\mathbf{z}} - \text{AE}(\tilde{\mathbf{z}})\|_2^2 - \lambda^{\text{attack}} \cdot d(\mathbf{z}, \tilde{\mathbf{z}}) \Big]. \quad (4.6)$$

4.2.4 Summary of our Proposed Model-free Attack

To summarize, our proposed architecture is shown in Fig. 4.1 with two stages. First, an autoencoder is trained with historical measurement data to minimize the residual error in the state estimator. Second, the WGAN is trained with the same data and the two regularization terms: (1) one incentivizes the WGAN to produce measurements that will pass the residual error test and (2) another to maximize the attack's impact. In the next section, we formally analyze the proposed attack framework.

4.3 Mathematical Analysis for the Proposed Method

The last section presented our proposed framework to create fake power system measurements to deploy a FDIA. However, to successfully deploy a FDIA without relying upon the underlying power system model, we need to be confident that our learned model will produce measurements that (i) look legit, (ii) cause an impact, and (iii) bypass the bad data detector. This section formally analyzes our proposed framework to demonstrate that it satisfies all the required properties to create effective

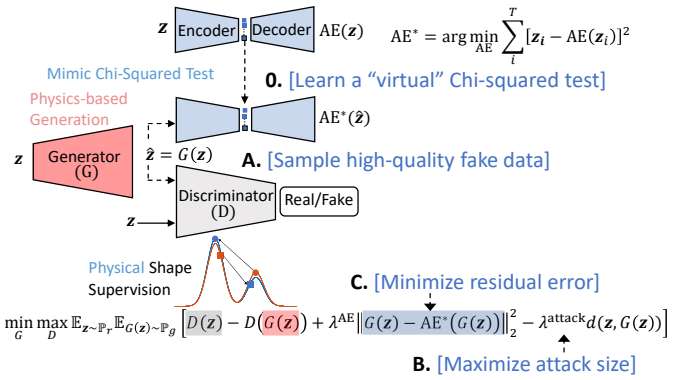


Figure 4.1: Proposed Model-free Architecture with a Wgan and Two Regularization Terms to Deploy an FDIA.

attacks. Such requirements are met and summarized in Theorem 3. However, we must first formally analyze (i) the autoencoder and (ii) the state estimator's residual error.

4.3.1 Connection between PCA and Autoencoders

We first analyze the connection between the autoencoder and PCA. Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ be a set of zero-centered data points $\boldsymbol{x}_i \in \mathbb{R}^{n\times 1}$. We define the data matrix as $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n\times m}$. PCA uses the top eigenspace of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ to approximate/fit the dataset Li *et al.* (2020). The autoencoder is a neural network model for unsupervised learning composed of an encoder and a decoder. The simplest case consists of a linear encoder $\boldsymbol{A} \in \mathbb{R}^{p\times n}$ that usually maps the input into a low dimensional space (latent space with a p-dimension) and a linear decoder $\boldsymbol{B} \in \mathbb{R}^{n\times p}$ that maps back the latent space to the original space, that is, $\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{B}\boldsymbol{X}$. This problem is formulated as

$$\min_{\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{X}\|_F^2 . \tag{4.7}$$

The work in Baldi and Hornik (1989) identified the connection between autoen-

coders and PCA: *under mild nondegeneracy conditions, any* $\boldsymbol{B}$ *at a local minimizer recovers the top rank-p eigen space of* $\boldsymbol{X}\boldsymbol{X}^T$. The problem in Eq. (4.7) can be expressed as $\min_{\boldsymbol{B},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{Z}\|_F^2$, where $\boldsymbol{Z} \in \mathbb{R}^{p\times m}$. It can be proved that the top rank-p eigenspace can be recovered from any local minimizer. Therefore, the autoencoder, from a geometric point of view, performs PCA on $\boldsymbol{X}$.

Theorem 2. Equivelancy of autoencoder and PCA. *Assume that* $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ *(with* $m \geq n$*) is full-rank and that* $\boldsymbol{X}\boldsymbol{X}^T$ *has distinct eigenvalues. Then, at any local minimizer of the optimization problem*

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times p},\, \boldsymbol{Z}\in\mathbb{R}^{p\times m}} \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{Z}\|_F^2 \qquad p \leq n, \tag{4.8}$$

$\boldsymbol{B}$ *spans the top rank-p eigenspace of* $\boldsymbol{X}\boldsymbol{X}^T$. *Furthermore, all these local minimums are also global minimums Li* et al. *(2020).*

Low-ranking dataset approximation

Now that we know that the autoencoder is equivalent to the PCA, Theorem 2, to pave our way to understand the residual error in the state estimator, we analyze the norm of the autoencoded measurements utilizing the singular value decomposition. Let's assume the data has been properly centered and scaled for this analysis. The SVD of the dataset measurement matrix, $\boldsymbol{Z}$, is expressed as Brunton and Kutz (2019): $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \sum_{i=1}^{m} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$, where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times m}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with non-negative entries. The SVD provides an optimal low-rank approximation to the matrix $\boldsymbol{Z}$. Specifically, we can obtain the rank-r approximation by keeping the leading r singular values and vectors and discarding the rest: $\boldsymbol{Z} \approx \boldsymbol{Z}_{(r)} = \boldsymbol{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\boldsymbol{V}_{(r)}^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$, where $\boldsymbol{Z}_{(r)} \in \mathbb{R}^{m\times T}$ is the rank-r approximation of the dataset matrix $\boldsymbol{Z}$, $\boldsymbol{U}_{(r)} \in \mathbb{R}^{m\times r}$ is the truncated $\boldsymbol{U}$

matrix, $\boldsymbol{V}_{(r)} \in \mathbb{R}^{n\times r}$ is the truncated $\boldsymbol{V}$ matrix, and $\boldsymbol{\Sigma}_{(r)} \in \mathbb{R}^{r\times r}$ is the truncated $\boldsymbol{\Sigma}$ with the lead r singular values.

4.3.2 Residual error analysis

This subsection analyzes the state estimator's residual error. Let $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{T}$ be the measurement dataset with each sample $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$. We define the matrix of collected measurements as $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_T \end{bmatrix} \in \mathbb{R}^{m\times T}$. The data generation process for each sample is the following

$$\boldsymbol{z}_i = \boldsymbol{H}\boldsymbol{x}_i^* + \boldsymbol{e}_i, \tag{4.9}$$

where $\boldsymbol{x}_i^* \in \mathbb{R}^{n\times 1}$ is the vector of the underlying system states, $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ $(n < m)$ represents the physical relationship between state variables and measurements, $\boldsymbol{e}_i \sim \mathcal{N}(0, \boldsymbol{R})$ are measurement errors, and $\boldsymbol{R} = \operatorname{diag}(\sigma_i^2, \ldots, \sigma_m^2)$ is the covariance matrix of the measurement errors. The model in Eq. (4.9) can be written in matrix form as $\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{X}^* + \boldsymbol{E}$, where $\boldsymbol{X}^* = \begin{bmatrix} \boldsymbol{x}_1^* & \boldsymbol{x}_2^* & \cdots & \boldsymbol{x}_T^* \end{bmatrix} \in \mathbb{R}^{n\times T}$ is the matrix of system states, and $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_T \end{bmatrix} \in \mathbb{R}^{m\times T}$, $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ is the matrix of measurement noises.

State estimation

State estimation (SE) infers the system state variables $\boldsymbol{x}_i \in \mathbb{R}^{n\times 1}$ from a measurement vector $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$. The state estimation minimizes

$$\hat{\boldsymbol{x}}_i = \arg\min_{\boldsymbol{x}} (\boldsymbol{z}_i - \boldsymbol{H}\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{z}_i - \boldsymbol{H}\boldsymbol{x}_i). \tag{4.10}$$

The solution of Eq. (4.10) is

$$\hat{\boldsymbol{x}}_i = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}\right)^{-1} \boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{z}_i = \boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{z}_i, \tag{4.11}$$

where $\boldsymbol{G} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H} \in \mathbb{R}^{n\times n}$. The estimated vector measurement is

$$\hat{\boldsymbol{z}}_i = \boldsymbol{H}\hat{\boldsymbol{x}}_i = \boldsymbol{H}\boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{z}_i = \boldsymbol{K}\boldsymbol{z}_i, \tag{4.12}$$

where $\boldsymbol{K} = \boldsymbol{H}\boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1} \in \mathbb{R}^{m\times m}$ is known as the *hat* matrix Abur and Exposito (2004).

The measurement residual vector for the i-th sample $\boldsymbol{\varepsilon}_i \in \mathbb{R}^{m\times 1}$ is expressed as follows

$$\boldsymbol{\varepsilon}_i = \boldsymbol{z}_i - \hat{\boldsymbol{z}}_i = \boldsymbol{z}_i - \boldsymbol{K}\boldsymbol{z}_i = (\boldsymbol{I} - \boldsymbol{K})\,\boldsymbol{z}_i = \boldsymbol{S}\boldsymbol{z}_i, \tag{4.13}$$

where the matrix $\boldsymbol{S} = (\boldsymbol{I} - \boldsymbol{K}) \in \mathbb{R}^{m\times m}$ is called the *residual sensitivity matrix*. The residual matrix for the whole dataset matrix can be succinctly written as

$$\mathcal{E} = \boldsymbol{K}\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 & \boldsymbol{\varepsilon}_2 & \cdots & \boldsymbol{\varepsilon}_T \end{bmatrix} \in \mathbb{R}^{m\times T}. \tag{4.14}$$

Bad Data Detector (BDD)

The state estimator SE possesses a Bad Data Detector (BDD) module to detect and filter suspicious data. The vector of measurement errors is assumed to follow a Gaussian distribution and be independent $\boldsymbol{e}_i \sim \mathcal{N}(0, \boldsymbol{R})$ Abur and Exposito (2004). Therefore, the normalized squared measurement residual error of the i-th sample

$$\|\boldsymbol{\varepsilon}_i\|^2 = \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2 \sim \chi^2_\nu \tag{4.15}$$

follows a chi-squared distribution χ^2_ν, where $\nu = m - n$ is a positive integer that specifies the number of degrees of freedom, representing the number of independent variables in the power system. Note that in Eq. (4.15), we assume that all the diagonal elements of $\boldsymbol{R}$ are equal for simplicity. If this is not the case, then $\|\boldsymbol{\varepsilon}_i\|^2 = \left\|\boldsymbol{R}^{-\frac{1}{2}}\,(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)\right\|^2$.

Based on the statistical properties of $\|\boldsymbol{\varepsilon}_i\|^2$, the presence of errors in the measurements can be detected with the chi-squared test (or residual error test) Abur and Exposito (2004); Weng *et al.* (2016). This test works as follows:

(i) Select the detection confidence probability p (e.g., 0.95), and compute its associated threshold value $\tau = \chi^2_{\nu,p}$ with $p = \Pr\left(J\left(\hat{\mathbf{x}}\right) \leq \chi^2_{\nu,p}\right)$.

(ii) Compute the normalized measurement error $\|\boldsymbol{\varepsilon}_i\|^2$.

(iii) If the inequality in Eq. (4.16) holds, bad data will be suspected, or else the measurements are assumed to be free of bad data.

$$\|\boldsymbol{\varepsilon}_i\|^2 \geq \tau \tag{4.16}$$

4.3.3 Performance guarantees

Theorem 3. FDIA in fully-observable case. *Given a measurement vector* $\mathbf{z}$*, and assuming that the system is observable from this set of measurements, our proposed model can generate false data* $\tilde{\mathbf{z}}$ *that satisfies:*

(i) It generates high-quality fake measurements that look "real," Lemma 1.

(ii) The attack size follows $\|\tilde{\mathbf{z}} - \mathbf{K}\tilde{\mathbf{z}}\|^2 \geq \mathcal{O}\left(\lambda^{attack}\right)$*, where* λ^{attack} *is the penalty term of* $\tilde{\mathbf{z}}$ *and* $\mathbf{z}$ *being too close, leading to* $\|\tilde{\mathbf{z}} - \mathbf{z}\|^2 \geq \mathcal{O}\left(\lambda^{attack}\right)$ *in DC grids, Lemma 2.*

(iii) The residual error of $\tilde{\mathbf{z}}$ *is lower or equal to the residual error of* $\mathbf{z}$*, i.e.,* $\mathbb{E}\|\tilde{\mathbf{z}} - \mathbf{K}\tilde{\mathbf{z}}\|^2 \leq \mathbb{E}\|\mathbf{z} - \mathbf{K}\mathbf{z}\|^2$ *provided sufficient data and training capacity, where* $\mathbf{K}$ *is the matrix related to the state estimation process. Thus, the probability of* $\tilde{\mathbf{z}}$ *passing the Chi-squared test is lower or equal to that of* $\mathbf{z}$ *passing the Chi-squared test, Theorem 4.*

Lemma 1. Distribution Match of GAN. *The JS divergence between the learned distribution* $g_{GAN}(\mathbf{z})$ *and the noisy distribution* $g_{noisy}(\mathbf{z})$ *is minimized with sufficient data and training capacity, that is,* $\mathcal{O}\left(\lambda^{attack}\right) \cdot JS(g_{noisy}(\mathbf{z}) \mid\mid g_{GAN}(\mathbf{z})) \leq JS(g_{noisy}(\mathbf{z}) \mid\mid g_{GAN}(\mathbf{z}))$.

Lemma 2. Larger Attack via Regularization. *The attack impact, quantified by* $\|\mathbf{z}-\tilde{\mathbf{z}}\|^2$, *has a lower bound* $\|\mathbf{z}-\tilde{\mathbf{z}}\|^2 \geq \mathcal{O}\left(\lambda^{attack}\right)$, *where* λ^{attack} *is the penalty term of* $\mathbf{z}$ *and* $\tilde{\mathbf{z}}$ *being too close. Thus, a larger penalty* λ^{attack} *leads to a larger attack impact on power.*

Theorem 4. Residual error of autoencoded measurements (lineal autoencoder). *Given a matrix of collected measurements* $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_T \end{bmatrix} \in \mathbb{R}^{m\times T}$, *where* $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$ *is the* i*-th sample, the residual error of the* r*-rank compression of the dataset* $\left\|\mathcal{E}_{(r)}\right\|_F^2$ *is smaller than the residual error of the original dataset* $\|\mathcal{E}\|_F^2$*:* $\left\|\mathcal{E}_{(r)}\right\|_F^2 < \|\mathcal{E}\|_F^2 \qquad r < m$.

Proof. Proof of Theorem 4. The squared residual error of the measurement dataset Eq. (4.14) is given as $\|\mathcal{E}\|_F^2 = \|\boldsymbol{K}\boldsymbol{Z}\|_F^2 = \sum_{i=1}^{T} \|\boldsymbol{\varepsilon}_i\|^2$.

The squared residual error of the r-rank dataset is given as

$$\begin{aligned} \left\|\mathcal{E}_{(r)}\right\|_F^2 &= \left\|\boldsymbol{K}\boldsymbol{Z}_{(r)}\right\|_F^2 = \left\|\boldsymbol{K}\boldsymbol{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\boldsymbol{V}_{(r)}^{\mathrm{T}}\right\|_F^2 \\ &= \left\|\boldsymbol{K}\boldsymbol{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\right\|_F^2 \\ &= \left\|\boldsymbol{K}\begin{bmatrix} \boldsymbol{u}_1\sigma_1 & \boldsymbol{u}_2\sigma_2 & \cdots & \boldsymbol{u}_r\sigma_r \end{bmatrix}\right\|_F^2 \\ &= \sum_{i=1}^{r} \|\boldsymbol{K}\boldsymbol{u}_i\sigma_i\|^2 . \end{aligned} \tag{4.17}$$

From Eq. (4.17), we conclude that the dataset residual error $\left\|\mathcal{E}_{(r)}\right\|_F^2$ is proportional to r. Thus

$$\left\|\mathcal{E}_{(r)}\right\|_F^2 < \|\mathcal{E}\|_F^2 \qquad r < m. \tag{4.18}$$

■

Lemma 3. Reconstruction error. *The reconstruction error* $\varepsilon\left(\boldsymbol{Z}, \boldsymbol{Z}_{(r)}\right)$ *is inversely proportional to* r*, as follows* $\varepsilon\left(\boldsymbol{Z}, \boldsymbol{Z}_{(r)}\right) = \left\|\boldsymbol{Z} - \boldsymbol{Z}_{(r)}\right\|_F^2$.

4.4 Numerical Experiments

This section shows how we deploy FDIAs on power grids with our proposed framework without knowing their mathematical or physical model. We conducted extensive experiments on different power networks to show the contributions and generality of our approach.

We train a WGAN with historical measurements with the model in Eq. (4.6) to demonstrate that (i) the model produces realistic, high-quality samples, (ii) the fake measurements successfully pass the residual error test with a high success rate, corroborating the mathematical analysis, and (iii) show that the trained WGAN creates different measurements (and therefore states) from the actual ones. This shows that the regularization terms work, maximizing the attack's impact and reducing the residual error in the state estimator. We carried out the aforementioned experiments in various test cases with similar results. Specifically, we use the IEEE 14-bus case, the IEEE 118-bus case, the Reliability Test System - Grid Modernization Lab Consortium (RTS-GMLC) test system, and the 200-bus Illinois synthetic model.

4.4.1 Data Generation and Model Architecture

Data Generation

For all test cases, we consider the DC power flow model and obtain all the active power flow measurements through transmission lines and transformers as measurements. The IEEE 14-bus case is composed of 20 measurements and 14 states, the IEEE 118-bus case has 186 measurements and 118 states, the RTS-GMLC test system contains 120 measurements and 73 states, and the 200-bus Illinois synthetic model has 245

measurements and 200 states.

We obtain the power systems' measurements by solving L times the DC power flow under different load conditions using MATPOWER Zimmerman *et al.* (2011). To simulate the 24-hour fluctuation, we use the real yearly load data from the Electric Reliability Council of Texas (ERCOT) for 2021 Electric Reliability Council of Texas, (ERCOT) (2022). For our simulations, we multiply each busload with the normalized loading parameter associated with a randomly selected area, γ, obtained from these realistic profiles. Similarly, we also adjust generation by scaling the generation profiles by multiplying them by the same loading parameter, γ, Ajjarapu and Christy (1992); Milano (2008). To make it more realistic, we add white noise to all measurements according to the standard deviation associated with the measurement devices. That is active power flow: 0.02 p.u., according to Shahriar *et al.* (2018). Finally, if we do not find an AC power flow solution, we do not include it in the dataset. This data generation approach will give us rich data variety with the power system under different load conditions.

Model Architecture

The architecture of our proposed WGAN model is inspired by the architecture of the DCGAN Radford *et al.* (2015) with the following modifications to adapt it to our power system data. Since the sensor measurement vectors are one-dimensional, we use fully connected layers instead of convolutional layers. The generator, G, consists of 5 layers with ReLU activation function for all layers except for the output, which uses tanh. The discriminator, D, comprises 5 layers with LeakyReLU activations with the slope of the leak set to 0.2.

4.4.2 Validation of Performance Guaranties

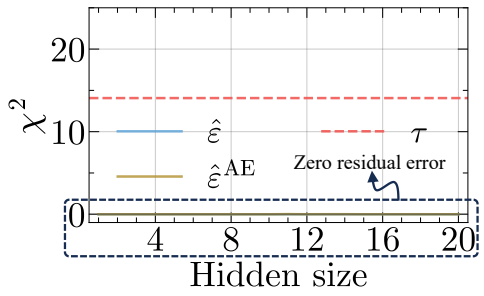
To demonstrate the stealthiness of our framework, we evaluate the performance guarantees in Theorem 4 and Lemma 3. These numerical quantifications are the residual and reconstruction errors for the 14-bus test case, with 13 system states and 20 measurements for the noiseless and noisy cases. Since the performance guarantees are based on the number of singular values used to reconstruct the measurements, we evaluate such quantities when varying the AE's latent dimension from 1 to 20.

Noiseless case

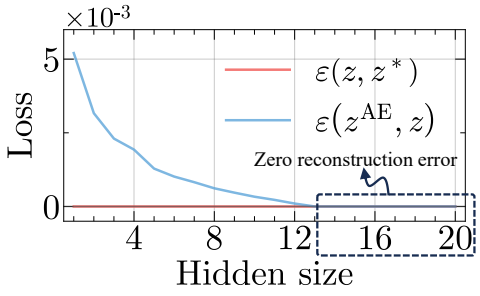
Theorem 4 states that the residual error, ε, is proportional to the AE's size of the hidden dimension. However, in the noiseless case, the state estimation for the original measurement will always be perfect. Thus producing a zero residual error, $\hat{\varepsilon}$. Since the autoencoded residual error, $\hat{\varepsilon}^{\text{AE}}$, is upper bounded by the original residual error, $\hat{\varepsilon}$, then $\hat{\varepsilon}^{\text{AE}} = 0$ for all hidden sizes. This is illustrated in Fig. 4.2a. Lemma 3 states that the reconstruction error is inversely proportional to the hidden dimension size. This makes sense since an AE with a hidden dimension size equal to the input's size can reconstruct any input by trivially learning an identity matrix. Fig. 4.2b shows the reconstruction error for different numbers of hidden dimensions. It can be seen that the reconstruction error decreases as the hidden size increases. Such error achieves zero with a hidden size of 13. The reason is that there is no noise in the dataset, and the measurements were constructed with 13 states.

Noisy case

Theorem 4 states that the residual error, ε, is proportional to the AE's size of the hidden dimension. We know that the autoencoded residual error, $\hat{\varepsilon}^{\text{AE}}$, is upper bounded



(a) Residual error, $\hat{\varepsilon}^{\mathrm{AE}}$. This error is proportional to the size of the latent dimension. $\hat{\varepsilon}^{\mathrm{AE}}$ brown line Eq. (4.18).



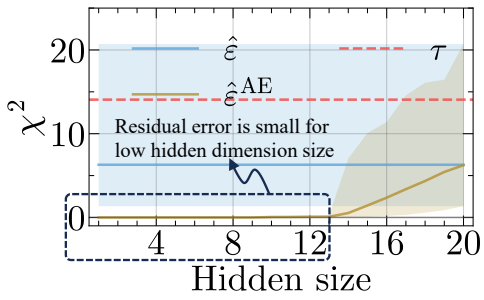
(b) Reconstruction error is inversely proportional to the hidden size. $\varepsilon\left(\boldsymbol{Z}, \tilde{\boldsymbol{Z}}^{\mathrm{AE}}\right)$ blue line, Lemma 3.

Figure 4.2: Residual and Reconstruction Errors for Clean 14-bus Dataset.

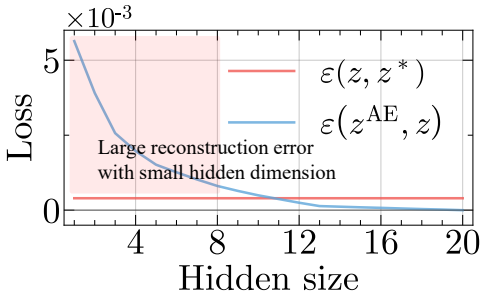
by the original residual error, $\hat{\varepsilon}$. This implies that the residual error will be the smallest when the hidden dimension is the smallest. This is illustrated in Fig. 4.3a, where the solid lines represent the mean of the samples, and the surrounding areas show the lowest and highest values. It can be seen that $\hat{\varepsilon}^{\mathrm{AE}} \leq \hat{\varepsilon}$ along all the hidden dimensions, as expected by Theorem 4. Lemma 3 states that the reconstruction error is inversely proportional to the hidden dimension size. Fig. 4.3b shows the reconstruction error for different numbers of hidden dimensions. It can be seen that the reconstruction error decreases as the hidden size increases. The blue line represents the AE's reconstruction error, achieving zero for a hidden size of 20. This is expected as the AE has enough capacity to learn all the information, including noise.

4.4.3 The Trade-off of Regularization Weights

To design an effective attack, the fake measurements must (1) pass the chi-squared test and (2) create an impactful attack vector. We measure these two requirements by the success rate and the attack vector size. Thus, we need a high success rate and a high attack vector for an effective attack. From the model in Eq. (4.6), we can see



(a) Residual error, $\hat{\varepsilon}^{\mathrm{AE}}$. This error is proportional to the size of the latent dimension, $\hat{\varepsilon}^{\mathrm{AE}}$ brown line Eq. (4.18).



(b) Reconstruction error is inversely proportional to the hidden size. $\varepsilon\left(\boldsymbol{Z}, \tilde{\boldsymbol{Z}}^{\mathrm{AE}}\right)$ blue line, Lemma 3.

Figure 4.3: Residual and Reconstruction Errors for the Noisy 14-bus Test Case.

that λ^{AE} and $\lambda^{\mathrm{attack}}$ have contradictive objectives. λ^{AE} is proportional to the success rate and inversely proportional to the attack vector size. Similarly, $\lambda^{\mathrm{attack}}$ is inversely proportional to the success rate and proportional to the attack size. To illustrate this, let λ^{AE} to have a fixed value, and $\lambda^{\mathrm{attack}}$ to monotonically increase. It can be seen in Fig. 4.4 that as $\lambda^{\mathrm{attack}}$ gets larger, the attack impact increases, but the success rate decreases. Fig. 4.4a shows a *conservative* set of regularization weights that produce a high success rate of 100%. Fig. 4.4b shows a set of regularization weights that produce a high success rate of 96%, which is similar to the real data. Fig. 4.4c shows an *agressive* set of regularization weights that produce a low success rate of 60%.

Therefore, for the model in Eq. (4.6) to optimally work, it is essential to carefully choose $\lambda^{\mathrm{attack}}$ and λ^{AE}. To choose an appropriate weight configuration, we explore the sensitivity of the success rate and the attack vector size for $\lambda^{\mathrm{attack}} \in [0, 1]$ and $\lambda^{\mathrm{AE}} \in [0, 1]$. The sensitivity of the success rate is shown in Fig. 4.5a, and the sensitivity of the attack vector size is shown in Fig. 4.5b. We first want to deploy a stealthy attack. This means we want a specified high success rate, $p^{\mathrm{specified}}$. Secondly, given our specified success rate, we want to produce the largest feasible attack vector, $\|\mathbf{a}\|^2$.

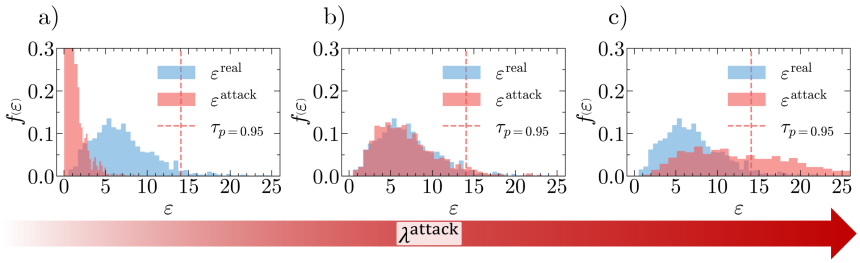
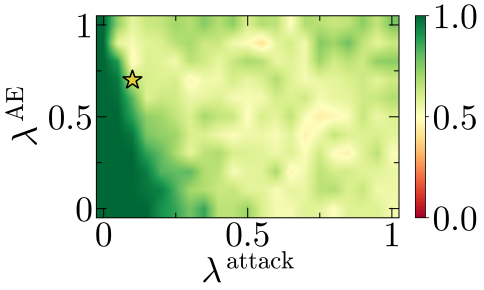


Figure 4.4: Attack Regularization Weight Effect on Residual Error Distribution.

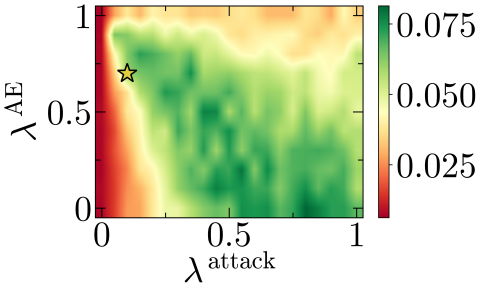
This can be cast as the following optimization problem:

$$\begin{aligned} \max_{\lambda^{\text{AE}},\lambda^{\text{attack}}} \quad & \left\|\mathbf{a}\left(\lambda^{\text{AE}},\lambda^{\text{attack}}\right)\right\|^2, \\ \text{s.t.} \quad & p\left(\lambda^{\text{AE}},\lambda^{\text{attack}}\right) = p^{\text{specified}}. \end{aligned} \tag{4.19}$$

We solve the optimization problem in Eq. (4.19) with $p^{\text{specified}} = 0.99$ and we obtain the optimal regularization weights of $\lambda^{\text{AE}^*} = 0.7$ and $\lambda^{\text{attack}^*} = 0.1$, which are indicated in Fig. 4.5. Thus, we use these regularization weights in the rest of the work to carry out the simulations.



(a) Sensitivity of attack success rate with respect to λ^{AE} and λ^{attack}.



(b) Sensitivity of attack vector norm with respect to λ^{AE} and λ^{attack}.

Figure 4.5: Sensitivity Regularization Weights. $\lambda^{\text{AE}^*} = 0.7$ and $\lambda^{\text{attack}^*} = 0.1$.

4.4.4 Validation of Attack's Performance

This section analyzes the attack's performance on different test cases. Specifically, it analyzes (1) the quality of the created samples, (2) the change in system states, (3) the attack stealthiness via the success rate, and (4) the attack sensitivity with respect to the number of measurements in the system.

Quality of Created Samples

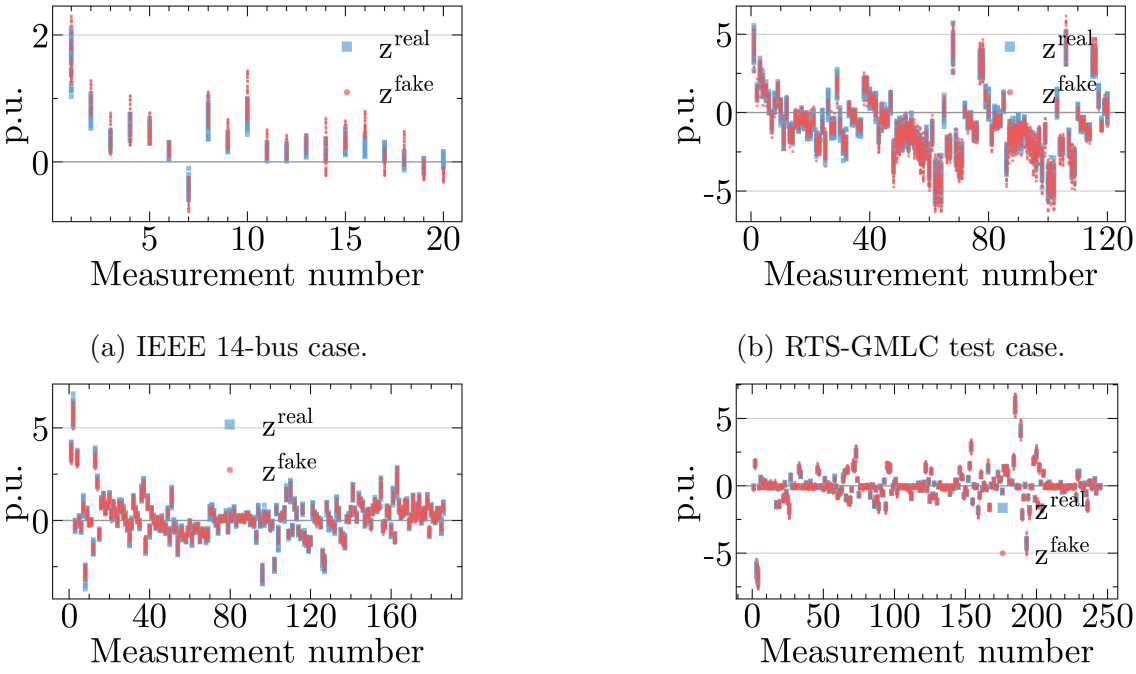
Fig. 4.6 shows the real measurements (in blue) and the fake measurements (in red) produced by our proposed framework. In the same figure, we can see that the fake measurements look like real ones but do not completely overlap. This means that our framework produces samples not in the original dataset. This is expected as the attack regularization term incentivizes the GAN to produce such measurements, Lemma 2.

Perturbation of System States

Fig. 4.7 shows the real system states (in blue) and the states produced by the fake measurements (in red). It can be seen that the fake states are more widespread than the real states. This accomplishes the attacker's target of dramatically changing the system states by exclusively tampering the measurements.

Evaluating Attack Stealthiness

Fig. 4.8 shows the residual error's PDF for the real (in blue) and fake measurements (in red). It is important to underscore that the residual errors of the fake measurements are consistently smaller than those of the real measurements. This validates the Theorem 4. That is, the autoencoded measurements produce smaller residual errors than the real ones. Thus, the fake measurements are more likely to pass the



(a) IEEE 14-bus case.

(b) RTS-GMLC test case.

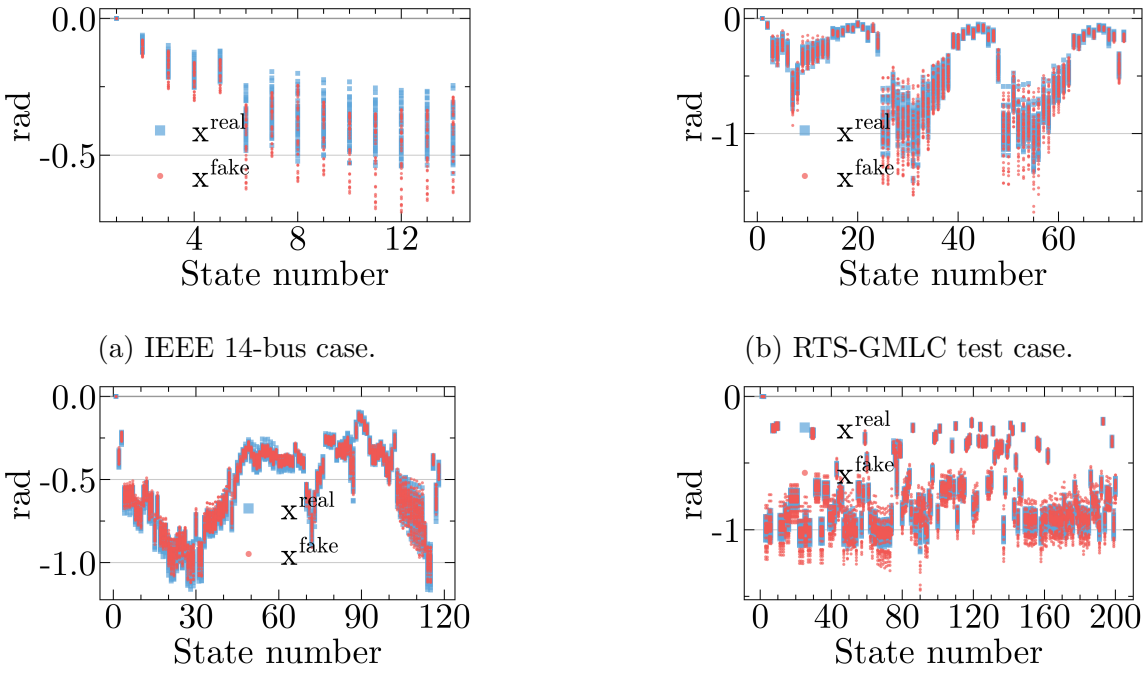
(c) IEEE 118-bus case.

(d) 200-bus Illinois synthetic model.

Figure 4.6: Measurement Distribution for Different Test Networks.

chi-squared test. We also carry out a sensitivity analysis of the success rate with respect to a given threshold. the experiment design is as follows. For a given confidence value p, we compute its corresponding threshold $\tau = \chi^2_{k,p}$, and obtain the probability of each measurement to pass the residual error test for the specified threshold, that is, $\Pr\left(J(\mathbf{z}) \geq \tau\right)$. We repeat this process for each simulation and obtain the success rate of passing the residual error test. This is the simulation's probability of passing the error test. We repeat this experiment for several values $p \in (0, 1)$. Fig. 4.9 shows the result of this sensitivity for the real (in blue) and the fake (in red) measurements for different test grids.

It can be seen that the fake measurements always have a higher success rate than the real ones. Thus, the attack is not likely to be detected for any chosen threshold τ.



(a) IEEE 14-bus case.

(b) RTS-GMLC test case.

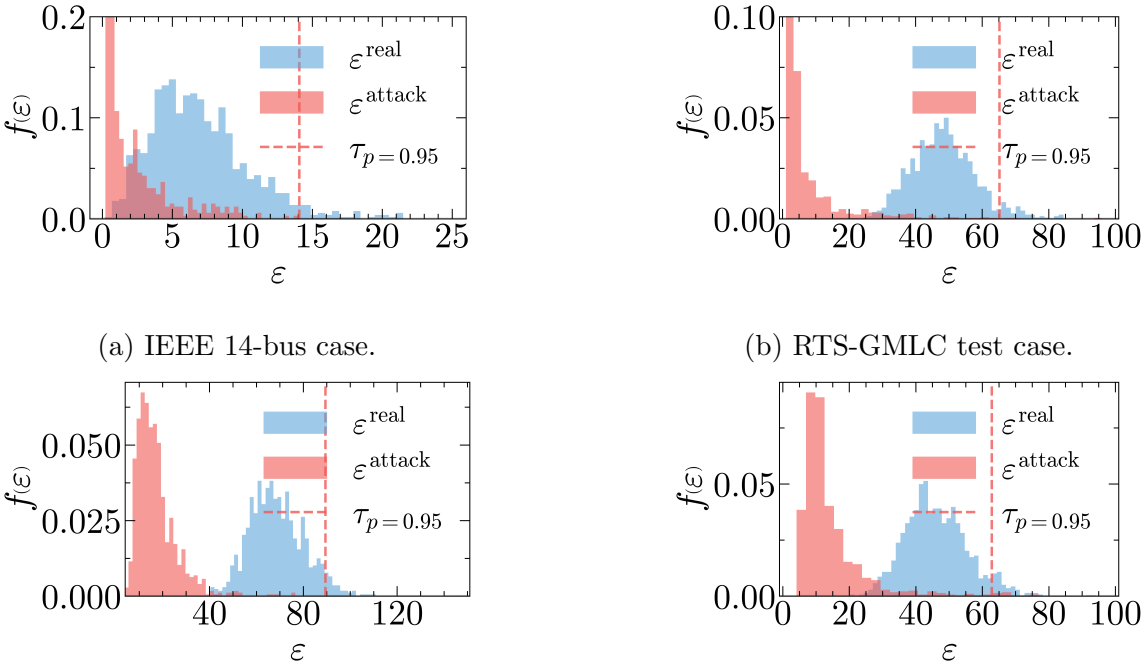
(c) IEEE 118-bus case.

(d) 200-bus Illinois synthetic model.

Figure 4.7: Real and Fake States Distribution for Different Test Networks.

Sensitivity of Attack's Regularization Weight

In this part, we explore the sensitivity of the attack's regularization weight λ^{attack} while keeping λ^{AE} fixed, as previously illustrated in Fig. 4.4. Fig. 4.8 shows the results for a set of conservative weights that produce a high success rate. Fig. 4.10 shows the results for a larger attack weight that shifts the residual error distribution to the right to overlap the real underlying distribution. Fig. 4.11 shows the results for an aggressive attack weight that creates larger attack vectors but produces lower success rates.



(a) IEEE 14-bus case.

(b) RTS-GMLC test case.

(c) IEEE 118-bus case.

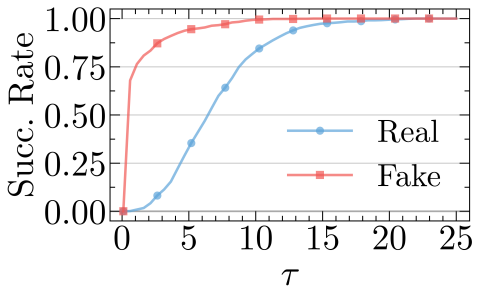
(d) 200-bus Illinois synthetic model.

Figure 4.8: Residual Error's PDFs.

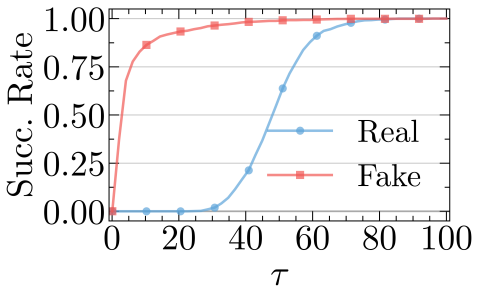
Attack Analysis with Limited Tampered Measurements

Risk (ρ) is the expectation of loss Wald (1945); DeGroot (2005), and in this context, it can be interpreted as the attack impact. This is given as $\rho = P \times \xi$, where P is the probability of bypassing the residual error test, and ξ is the magnitude of the attack. P is the attack success rate and $\xi = \|\boldsymbol{a}\|^2$ is the attack vector magnitude. P is obtained by computing the attack's success rate when n^{unknown} measurements are in the system. In this case, the defender has access to all the measurements in the system, and the attacker can only modify a subset of the measurements.

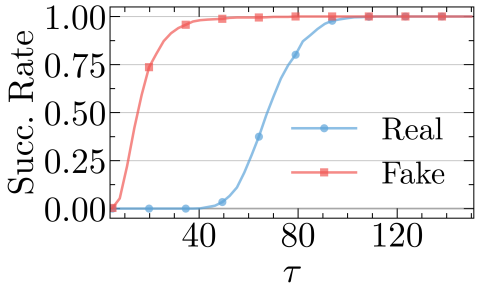
To assess the attack's impact, two scenarios are tested with respect to the attacker's measurement knowledge. In the first scenario, the attacker can see all the measurement information in the system but can only modify a subset of the observed



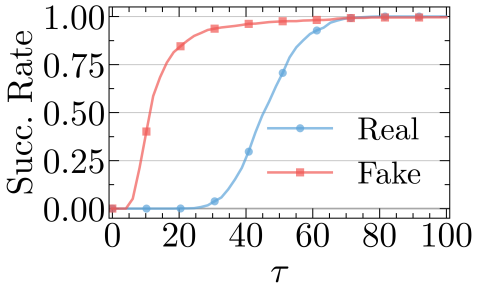
(a) IEEE 14-bus case.



(b) RTS-GMLC test case.



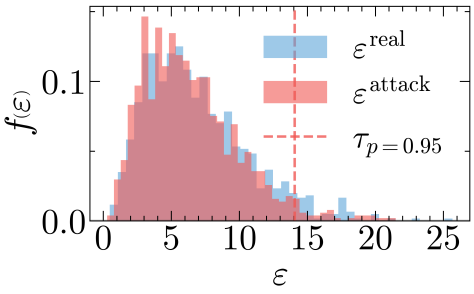
(c) IEEE 118-bus case.



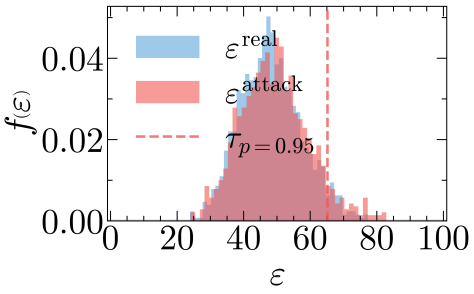
(d) 200-bus Illinois synthetic model.

Figure 4.9: Residual Error's CDFs.

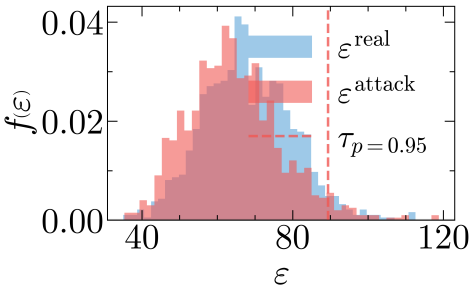
measurements. Under this scenario, for the IEEE 14-bus test case, Fig. 4.12a shows the success rate and Fig. 4.12b shows the attack vector magnitude $\xi = \|\boldsymbol{a}\|^2$. Based on these results, the risk is computed as $\rho = P \times \xi$, and the results for the IEEE 14-bus test case and for the 200-bus Illinois synthetic model are shown in Figs. 4.13a and 4.13b. In the second scenario, the attacker can only see and modify a subset of the measurements in the system. The results for the IEEE 14-bus test case and for the 200-bus Illinois synthetic model are shown in Figs. 4.13c and 4.13d. It can be observed that the attack impact is lower than in the first scenario. In the first scenario, the attacker trains the models with all the measurement information, but in the second case, the attacker is myopic to train the models with reduced measurement information.



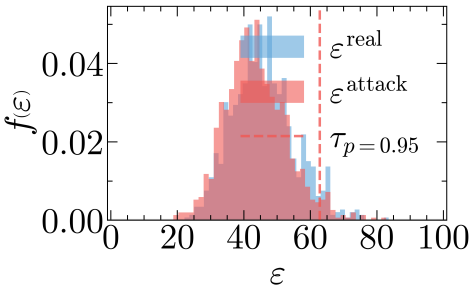
(a) IEEE 14-bus case.



(b) RTS-GMLC test case.



(c) IEEE 118-bus case.

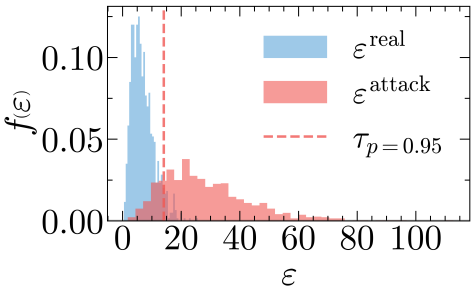


(d) 200-bus Illinois synthetic model.

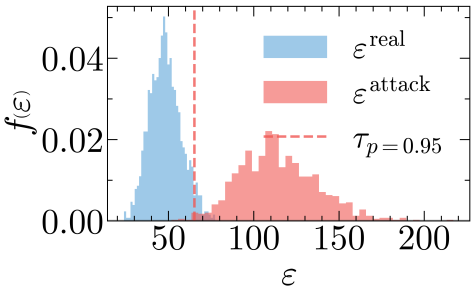
Figure 4.10: Residual Error's PDFs with λ^{attack} Tuned to Mimic the Residual Error Distribution.

Sensitivity with respect to Measurement coverage

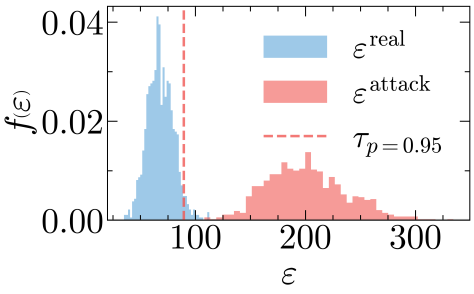
Theorem 3 states that our framework can produce high-quality samples that bypass the residual error test given that the system is observable. In this part, we analyze the sensitivity of this claim by evaluating the attack success rate with respect to the measurement coverage. To do this, we randomly remove $n^{\text{unknown}} \in \mathbb{Z}$ measurements and carry out the attack. This process is done for $n^{\text{unknown}} \in [0, m]$. Fig. 4.14 shows the test cases' success rates. The trend is the same for all test cases: The success rate decreases as the number of removed measurements increases. This is expected as the attack is carried out with less information. Thus, it is more challenging. Another observation is that the success rate drops to zero after a certain number of



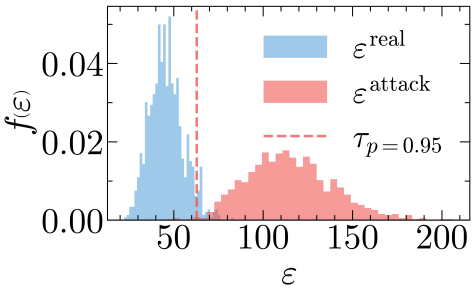
(a) IEEE 14-bus case.



(b) RTS-GMLC test case.



(c) IEEE 118-bus case.



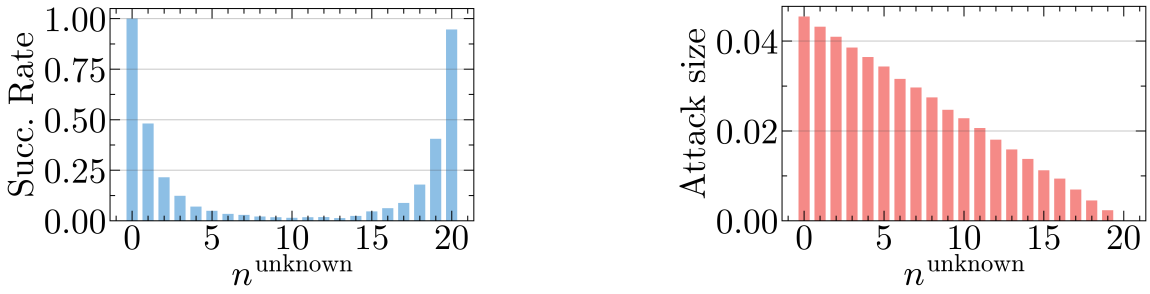
(d) 200-bus Illinois synthetic model.

Figure 4.11: Residual Error's PDFs with aggressive λ^{attack} That Skews the Distribution to Produce a Low Success Rate.

removed measurements. This is because the system becomes unobservable, and the state estimation problem becomes underdetermined. For instance, Fig. 4.14a shows the results for the 14-bus test case, where we can see that when $n^{\text{unknown}} \geq 7$, the success rate drops to zero. As this system has 20 measurements and 13 states, if 7 or more measurements are removed, then the system becomes unobservable. The same is true for the other cases.

Comparison against other FDIA methods

To assess the advantages and differences between our proposed model-free FDIA framework, we compare our method against two model-based and two model-free attacks. The model-based attacks are: MB 1 introduced in Hug and Giampapa (2012)



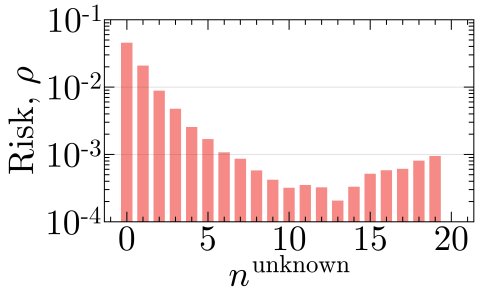
(a) Success rate.

(b) Attack size vector.

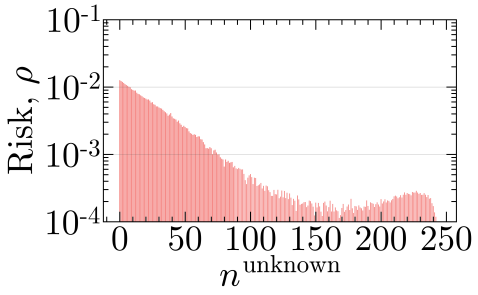
Figure 4.12: Success Rate and Attack Vector Size with Partial Observability for the IEEE 14-bus Test Case.

and MB 2 from Chin *et al.* (2017). The model-free attacks are: MF 1 introduced in Mohammadpourfard *et al.* (2020) and MF 2 proposed in Costilla-Enriquez and Weng (2022). For the model-based attacks, we inject random attack vectors. For the model-free attacks, we train them with exclusively historical measurements. To compare these methods, we generate samples with the same procedure described in section 4.4.1 and tamper the real noisy measurements with our proposed approach and Methods: MB 1, MB 2, MF 1, MF 2. We track two essential metrics in these comparisons: (i) attack success rate and (ii) attack vector size.

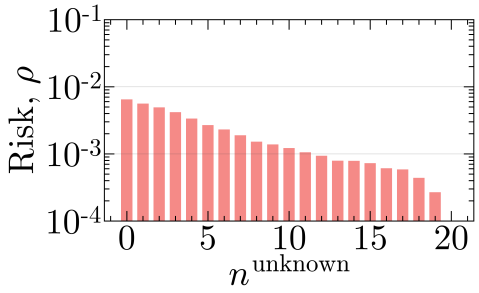
To evaluate the success rate of all approaches, we carry out the following procedure. For a given confidence value p, we compute its corresponding threshold $\tau = \chi^2_{k,p}$, and obtain the probability of each measurement to pass the residual error test for the specified threshold, that is, $\Pr\left(J(\mathbf{z}) \geq \tau\right)$. Specifically, we select a confidence value $p = 0.95$. The results are shown in Table 4.1. It can be observed that our method produces success rates above 95%. This is because our framework reduces the residual error of tampered samples. The model-based methods, MB 1 and MB 2, have success rates around 95%. This is because both methods are guaranteed to have the same residual error as the real noisy measurements by design (see proof in Hug and Gi-



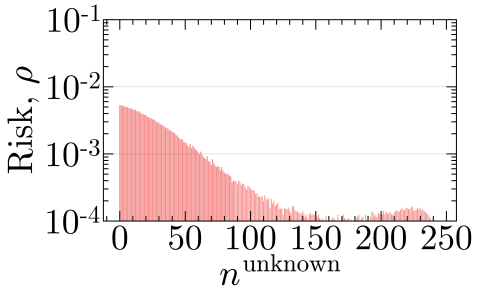
(a) Attack impact with knowledge of all measurements for IEEE 14-bus test case.



(b) Attack impact with knowledge of all measurements for 200-bus Illinois synthetic model.



(c) Attack impact with limited knowledge measurements for IEEE 14-bus test case.

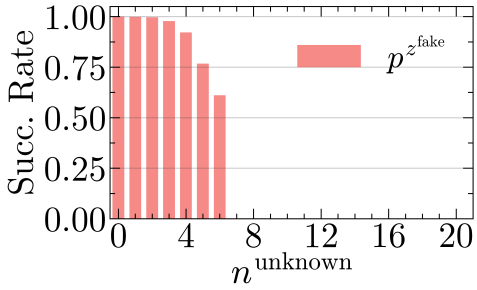


(d) Attack impact with limited knowledge measurements for 200-bus Illinois synthetic model.

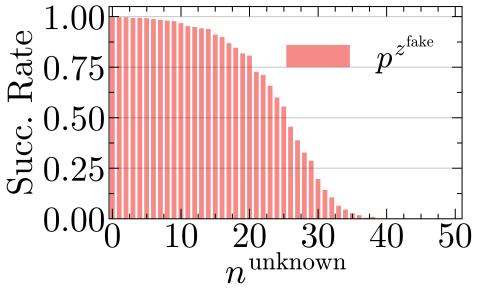
Figure 4.13: Attack Impact Analysis with Partial Observability.

ampapa (2012)). Similarly, the MF 1 framework has success rates around 95% with more variability due to the training process and lack of mathematical guarantees. The MF 2 approach has higher success rates than 95% due to its architecture that reduces residual errors.

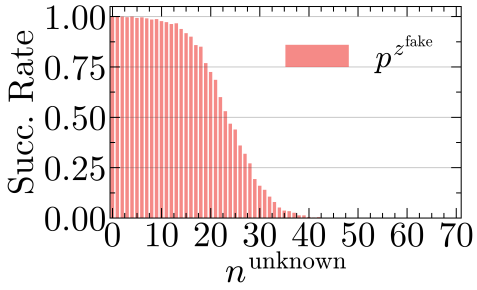
Now, we evaluate the attack vector size $\|\boldsymbol{a}\|^2$. For this evaluation, we only report the attack vector norms for the model-free attacks. The reason is that model-free attacks can inject an arbitrarily large attack vector. The results for our framework, MF 1, and MF 2 are shown in Table 4.2, where it can be seen that our approach



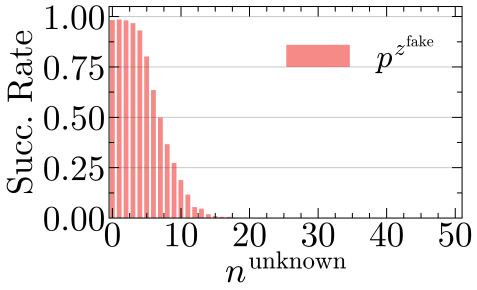
(a) IEEE 14-bus case.



(b) RTS-GMLC test case.



(c) IEEE 118-bus case.



(d) 200-bus Illinois synthetic model.

Figure 4.14: Sensitivity of Measurement Coverage.

produces the largest attack vectors. Thus, our attack is the most effective.

4.5 Conclusion

This chapter formally analyzed the proposed model-free FDIA to create tampered measurement vectors to carry out a False Data Injection Attack (FDIA) without knowing any underlying information about the power system. This work proves that an attacker can deploy a guaranteed attack and is not required to have access to all power system information, which poses a significant threat to the security of power systems.

Table 4.1: Comparison of Attack Success Rates.

Test Case	Success Rate (%)				
	Ours	MB 1	MB 2	MF 1	MF 2
14-bus	100	94.7	94.7	94.4	95.7
RTS-GMLC	99.7	94.4	94.4	93.3	98.3
118-bus	100	95.2	95.2	91.4	97.1
200-bus	98.2	94.2	94.2	92.2	96.3

Table 4.2: Comparison of Attack Vector Size.

Test Case	Attack vector norm, $\|\boldsymbol{a}\|^2$		
	Ours	MF 1	MF 2
14-bus	0.0224	0.0106	0.0186
RTS-GMLC	0.0275	0.0185	0.0276
118-bus	0.0421	0.0274	0.0365
200-bus	0.0355	0.0385	0.0309

Chapter 5

ATTACK ON THE AC STATE ESTIMATOR WITHOUT SYSTEM INFORMATION AND PERFORMANCE GUARANTEE

5.1 Introduction

The last chapter introduced a formal analysis of a model-free FDIA on a linear model. However, the linear model does not truly capture the non-linear power system nature. To address such limitations, based on general knowledge about the power system, this chapter introduces a FDIA on an AC model that only requires one piece of information: (i) historical system measurements. The proposed framework relies on the fact that the footprint of the system model may be hidden in the historical data implicitly, as depicted in Fig. 5.1.

Extensive simulations on both transmission (IEEE 14-bus, 118-bus, 300-bus, and RTS-GMLC) and distribution networks (22-bus, 85-bus, 123-bus, and 144-bus radial) model test cases verify the performance of the proposed model-free FDIA. Furthermore, to highlight the distinctions and benefits of our approach compared to those documented in the existing literature, we conduct comparative assessments involving our proposed FDIA and three other well-established methods outlined in Hug and Giampapa (2012); Chin *et al.* (2017); Mohammadpourfard *et al.* (2020); Costilla-Enriquez and Weng (2022).

5.2 Mathematical Analysis for the Proposed Method

The previous section introduced our framework for generating counterfeit power system measurements to deploy a FDIA. To ensure the successful deployment of a

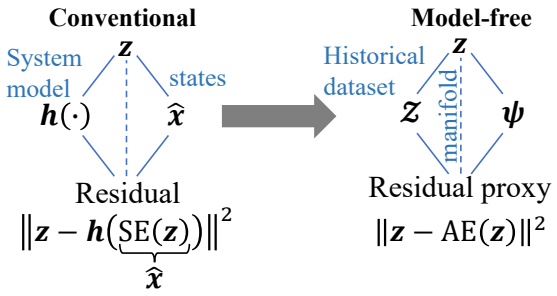


Figure 5.1: Information Flow of the Proposed Model-Free Architecture FDIA.

FDIA without relying on the power system model, our framework must guarantee that its learned model can produce measurements that (i) appear legitimate, (ii) have a substantial impact, and (iii) evade detection by the bad data detector. This section conducts a formal analysis of our framework to demonstrate its ability to meet these requirements. These prerequisites are formally examined and summarized in Theorem 6. However, before that, we perform a formal analysis of (i) the autoencoder and (ii) the residual error of the state estimator.

5.2.1 Non-linear autoencoder

The non-linear autoencoder is a generalization of the linear autoencoder into a non-linear form

$$\min_{\boldsymbol{v},\boldsymbol{w}} \|\boldsymbol{X} - g_{\boldsymbol{v}}\left(f_{\boldsymbol{w}}\left(\boldsymbol{X}\right)\right)\|_F^2, \tag{5.1}$$

where $f_{\boldsymbol{w}} : \mathbb{R}^m \mapsto \mathbb{R}^r$ is a nonlinear encoder parametrized by $\boldsymbol{w}$ and $g_{\boldsymbol{v}} : \mathbb{R}^r \mapsto \mathbb{R}^m$ is a nonlinear decoder parametrized by $\boldsymbol{v}$. The encoder $f_{\boldsymbol{w}}$ transforms an input $\boldsymbol{x} \in \mathbb{R}^{m\times 1}$ into a hidden representation $\boldsymbol{y} = f_{\boldsymbol{w}}\left(\boldsymbol{x}\right) \in \mathbb{R}^{r\times 1}$. The resulting hidden representation $\boldsymbol{y}$ is then mapped back to a reconstructed vector $\hat{\boldsymbol{x}} = g_{\boldsymbol{v}}\left(\boldsymbol{y}\right) \in \mathbb{R}^{m\times 1}$. Generally, $\hat{\boldsymbol{x}}$ is not an exact reconstruction of $\boldsymbol{x}$. The optimization problem in Eq. (5.1) can

be trivially minimized by setting $\boldsymbol{Y} = \boldsymbol{X}$. An autoencoder where $\boldsymbol{Y}$ is of the same dimensionality as $\boldsymbol{X}$ (or larger) can perfectly reconstruct the input by simply learning the identity mapping. Non-trivial solutions arise when further constraints are applied to the autoencoder. The typical autoencoders use a *bottleneck* to produce an under-complete representation where $r < m$. The resulting lower dimensional $\boldsymbol{Y}$ then is a *lossy compressed* representation of $\boldsymbol{X}$. Such representation will reconstruct a *lossy compressed* signal $\hat{\boldsymbol{X}}$ Vincent *et al.* (2010). This will then result in a non-zero reconstruction error.

Autoencoders with nonlinear encoder functions f and nonlinear decoder functions g can thus learn a more powerful nonlinear generalization of PCA Goodfellow *et al.* (2016). In other words, autoencoders can learn to represent data in a lower-dimensional space while preserving the most important information, even if the data has complex nonlinear relationships. The reason, autoencoders are feedforward networks and the universal approximation theorem applies to them. This theorem guarantees that a feedforward neural network with at least one hidden layer can represent an approximation of any function (within a broad class) to an arbitrary degree of accuracy, provided that it has enough hidden units. these advantages also apply to autoencoders Hornik *et al.* (1989); Cybenko (1989). Experimentally, deep autoencoders yield much better compression than corresponding shallow or linear autoencoders Hinton and Salakhutdinov (2006).

5.2.2 An AE can Implicitly Learn the Power System Model

Assumption 1. *The manifold assumption states that the (high-dimensional) data lie (roughly) on a low-dimensional manifold Chapelle* et al. *(2009). According to our data generation process in Eq.* (5.4)*, the measurement data, $\boldsymbol{z} \in \mathbb{R}^m$, can be fully characterized by the set of variables $\boldsymbol{x} \in \mathbb{R}^n$. Since $n < m$ then the high-dimensional*

measurement data lie on a low-dimensional manifold. Thus, the manifold assumption applies to our analysis.

Theorem 5. Zero residual error for a non-linear AE. *For a fully observable system (i.e., redundant measurements), we know there exists a mapping function from measurement space to state space* $h : \mathbb{R}^m \mapsto \mathbb{R}^n$ *and a mapping back from state space to measurement space* $SE : \mathbb{R}^n \mapsto \mathbb{R}^m$ *that makes the squared residual error zero*

$$\varepsilon = \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|^2 = \|\boldsymbol{z} - h(\hat{\boldsymbol{x}})\|^2 = \|\boldsymbol{z} - h(SE(\boldsymbol{z}))\|^2 \tag{5.2}$$

Then, it is possible to learn such functions with an autoencoder

$$\varepsilon^{\mathrm{AE}} = \left\|\boldsymbol{z} - \boldsymbol{z}^{\mathrm{SE}}\right\|^2 = \|\boldsymbol{z} - g(f(\boldsymbol{z}))\|^2, \tag{5.3}$$

where $f_{\boldsymbol{w}}$ is an encoder and $g_{\boldsymbol{v}}$ is a decoder. The decoder transforms an input $\boldsymbol{z} \in \mathbb{R}^{m\times 1}$ into a hidden representation $\boldsymbol{\psi}_i = f_{\boldsymbol{w}}(\boldsymbol{z}) \in \mathbb{R}^{n\times 1}$. The resulting hidden representation $\boldsymbol{\psi}_i$ is then mapped back to a reconstructed vector $\boldsymbol{z}^{\mathrm{AE}} = g_{\boldsymbol{v}}(\boldsymbol{\psi}_i) \in \mathbb{R}^{m\times 1}$.

Proof. Proof of Theorem 5. For an AE, we know that its objective is to minimize the reconstruction error for any measurement vector $\boldsymbol{z}$ based on the given dimension in the hidden layer. This means that in an ideal case (i.e., a good DNN approximation for the encoder and decoder functions) $\|\boldsymbol{z} - g(f(\boldsymbol{z}))\|^2$ is minimized: achieving the global optimum.

Assume we know the number of states in the system, n. Let's design the dimension in the latent layer to be n. Let's also assume that there is a function $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ that can universally approximate $h^{-1} = \mathrm{SE} : \mathbb{R}^m \mapsto \mathbb{R}^n$, where f maps the measurements $\boldsymbol{z}$ to the state variables $\boldsymbol{x}$. Additionally, let's assume there is a function $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ that can universally approximate $h : \mathbb{R}^n \mapsto \mathbb{R}^m$, where g maps the state variables $\boldsymbol{x}$ to the measurements $\boldsymbol{z}$. This means the AE can have a zero reconstruction error based on the above assumptions. Note that these are valid assumptions because the

encoder f and decoder g are feedforward networks, and the universal approximation theorem applies to them. This theorem guarantees that a feedforward neural network with at least one hidden layer can represent an approximation of any function (within a broad class) to an arbitrary degree of accuracy, provided that it has enough hidden units Hornik *et al.* (1989); Cybenko (1989). Thus, both the encoder and decoder can approximate any arbitrary function. Based on the AE's model, i.e., minimizing the reconstruction error Eq. (5.1), we know that an AE with a latent layer dimension equal to the number of system states n shall be able to achieve zero reconstruction error. ■

5.2.3 Residual Error on the State Estimator

5.2.4 Residual error analysis

This subsection analyzes the state estimator's residual error. Let $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{T}$ be the measurement dataset with each sample $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$. We define the matrix of collected measurements as $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_T \end{bmatrix} \in \mathbb{R}^{m\times T}$. Consider the linearized measurement equations

$$\Delta \boldsymbol{z}_i = \boldsymbol{H}\Delta \boldsymbol{x}_i^* + \boldsymbol{e}_i, \tag{5.4}$$

where $\boldsymbol{x}_i^* \in \mathbb{R}^{n\times 1}$ is the vector of the underlying system states, $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ ($n < m$) represents the physical relationship between state variables and measurements, $\boldsymbol{e}_i \sim \mathcal{N}(0, \boldsymbol{R})$ are measurement errors, and $\boldsymbol{R} = \operatorname{diag}(\sigma_i^2, \ldots, \sigma_m^2)$ is the covariance matrix of the measurement errors. The model in Eq. (5.4) can be written in matrix form as $\Delta \boldsymbol{Z} = \boldsymbol{H}\Delta \boldsymbol{X}^* + \boldsymbol{E}$, where $\boldsymbol{X}^* = \begin{bmatrix} \boldsymbol{x}_1^* & \boldsymbol{x}_2^* & \cdots & \boldsymbol{x}_T^* \end{bmatrix} \in \mathbb{R}^{n\times T}$ is the matrix of system states, and $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_T \end{bmatrix} \in \mathbb{R}^{m\times T}$, $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ is the matrix of measurement noises.

Then, the linearized state vector from Eq. (3.1) is given by

$\Delta\hat{\boldsymbol{x}}_i = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\Delta\boldsymbol{z}_i = \boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\Delta\boldsymbol{z}_i$, where $\boldsymbol{G} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H} \in \mathbb{R}^{n\times n}$. The estimated vector measurement is $\Delta\hat{\boldsymbol{z}}_i = \boldsymbol{H}\hat{\boldsymbol{x}}_i = \boldsymbol{H}\boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\Delta\boldsymbol{z}_i = \boldsymbol{K}\Delta\boldsymbol{z}_i$, where $\boldsymbol{K} = \boldsymbol{H}\boldsymbol{G}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1} \in \mathbb{R}^{m\times m}$ is known as the *hat* matrix Abur and Exposito (2004). The measurement residual vector for the i-th sample $\boldsymbol{\varepsilon}_i \in \mathbb{R}^{m\times 1}$ is expressed as follows

$$\boldsymbol{\varepsilon}_i = \Delta\boldsymbol{z}_i - \Delta\hat{\boldsymbol{z}}_i = \Delta\boldsymbol{z}_i - \boldsymbol{K}\Delta\boldsymbol{z}_i = \left(\boldsymbol{I} - \boldsymbol{K}\right)\Delta\boldsymbol{z}_i = \boldsymbol{S}\Delta\boldsymbol{z}_i, \tag{5.5}$$

where the matrix $\boldsymbol{S} = (\boldsymbol{I} - \boldsymbol{K}) \in \mathbb{R}^{m\times m}$ is called the *residual sensitivity matrix*. The residual matrix for the whole dataset matrix can be succinctly written as

$$\mathcal{E} = \boldsymbol{K}\Delta\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{\varepsilon}_1 & \boldsymbol{\varepsilon}_2 & \cdots & \boldsymbol{\varepsilon}_T\end{bmatrix} \in \mathbb{R}^{m\times T}. \tag{5.6}$$

5.2.5 Performance guarantees

Theorem 6. FDIA in the fully-observable case for the AC state estimator model. *Given a measurement vector* $\mathbf{z}$*, and assuming that the system is observable from this set of measurements, our proposed model can generate false data* $\tilde{\mathbf{z}}$ *that satisfies:*

(i) It generates high-quality fake measurements that look "real," Lemma 1.

(ii) Around an linealized point, the attack size follows $\|\Delta\tilde{\mathbf{z}} - \mathbf{K}\Delta\tilde{\mathbf{z}}\|^2 \geq \mathcal{O}\left(\lambda^{attack}\right)$*, where* λ^{attack} *is the penalty term of* $\tilde{\mathbf{z}}$ *and* $\mathbf{z}$ *being too close, leading to* $\|\tilde{\mathbf{z}} - \mathbf{z}\|^2 \geq \mathcal{O}\left(\lambda^{attack}\right)$ *in AC grids, Lemma 2.*

(iii) The residual error of $\tilde{\mathbf{z}}$ *is lower or equal to the residual error of* $\mathbf{z}$*, i.e.,* $\mathbb{E}\|\tilde{\mathbf{z}} - \mathbf{K}\tilde{\mathbf{z}}\|^2 \leq \mathbb{E}\|\mathbf{z} - \mathbf{K}\mathbf{z}\|^2$ *provided sufficient data and training capacity, where* $\mathbf{K}$ *is the matrix related to the state estimation process. Thus, the probability of* $\tilde{\mathbf{z}}$ *passing the Chi-squared test is lower or equal to that of* $\mathbf{z}$ *passing the Chi-squared test, Theorems 5 and 7.*

Assumption 2. *Autoencoders with nonlinear encoder functions f and nonlinear decoder functions g can thus learn a more powerful nonlinear generalization of PCA Goodfellow* et al. *(2016). In other words, autoencoders can learn to represent data in a lower-dimensional space while preserving the most important information, even if the data has complex nonlinear relationships.*

Theorem 7. Residual error of compressed measurements (non-lineal autoencoder). *Given a matrix of collected measurements* $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_T \end{bmatrix} \in \mathbb{R}^{m \times T}$*, where* $\boldsymbol{z}_i \in \mathbb{R}^{m \times 1}$ *is the i-th sample, and an autoencoder that produces an under-complete reconstruction of the input data* $\boldsymbol{Z}_{(r)}^{AE} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ *(Assumption 2), where* $r < m$*, then the squared residual error of under-complete reconstruction* $\left\| \mathcal{E}_{(r)}^{AE} \right\|_F^2$ *is smaller than the residual error of the original dataset* $\|\mathcal{E}\|_F^2$ $\left\| \mathcal{E}_{(r)}^{AE} \right\|_F^2 < \|\mathcal{E}\|_F^2 \qquad r < m.$

Proof. Proof of Theorem 7. The squared residual error of the measurement dataset Eq. (5.6) is given as

$$\begin{aligned} \|\mathcal{E}\|_F^2 &= \|\boldsymbol{K}\boldsymbol{Z}\|_F^2 \\ &= \sum_{i=1}^{T} \|\boldsymbol{\varepsilon}_i\|^2 \end{aligned} \tag{5.7}$$

The squared residual error of under-complete reconstruction of the dataset $\boldsymbol{Z}_{(r)}^{\mathrm{AE}}$ is given as

$$\begin{aligned} \left\| \mathcal{E}_{(r)}^{\mathrm{AE}} \right\|_F^2 &= \left\| \boldsymbol{K}\boldsymbol{Z}_{(r)}^{\mathrm{AE}} \right\|_F^2 = \left\| \boldsymbol{K}\boldsymbol{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\boldsymbol{V}_{(r)}^{\mathrm{T}} \right\|_F^2 \\ &= \left\| \boldsymbol{K}\boldsymbol{U}_{(r)}\boldsymbol{\Sigma}_{(r)} \right\|_F^2 = \sum_{i=1}^{r} \|\boldsymbol{K}\boldsymbol{u}_i\sigma_i\|^2 . \end{aligned} \tag{5.8}$$

From Eq. (5.8), we conclude that the dataset residual error $\left\| \mathcal{E}_{(r)} \right\|_F^2$ is proportional to r. Note that if the autoencoder produces a *lossy compressed* reconstruction, then we have $\boldsymbol{Z}_{(r)}^{\mathrm{AE}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$, where $r < m$. Thus, the following inequality holds

$$\left\| \mathcal{E}_{(r)} \right\|_F^2 < \|\mathcal{E}\|_F^2 \qquad r < m. \tag{5.9}$$

■

5.3 Numerical Experiments

This section shows how we deploy FDIAs on power grids with our proposed framework without knowing their mathematical or physical model. We conducted extensive experiments on different power networks to show our approach's contributions and generality.

We train a WGAN with historical measurements with the model in Eq. (4.6) to demonstrate that (i) the model produces realistic, high-quality samples, (ii) the fake measurements successfully pass the residual error test with a high success rate, corroborating the mathematical analysis, and (iii) show that the trained WGAN creates different measurements (and therefore states) from the actual ones. This shows that the regularization terms work, maximizing the attack's impact and reducing the residual error in the state estimator. We carried out the aforementioned experiments in various test cases with similar results. Specifically, we use the IEEE 14-bus case, the IEEE 118-bus case, the Reliability Test System - Grid Modernization Lab Consortium (RTS-GMLC) test system, and the 200-bus Illinois synthetic model.

5.3.1 Data Generation and Model Architecture

Data Generation

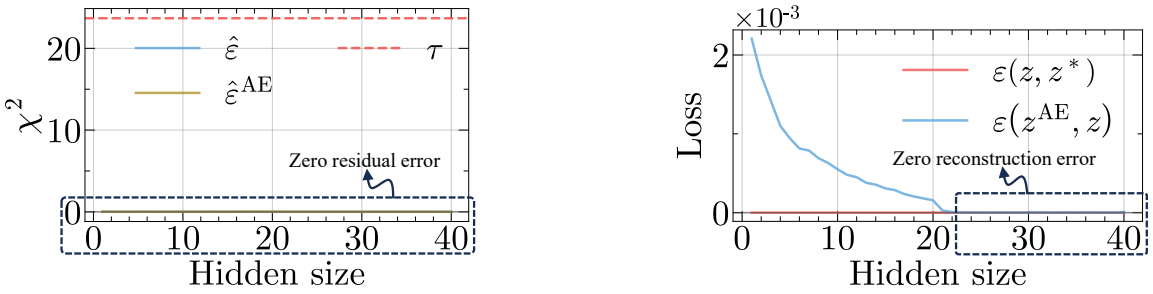
For all test cases, we consider the AC power flow model and obtain all the active and reactive power flow measurements through transmission lines and transformers as measurements for four transmission grids: IEEE 14-bus, 118-bus, 300-bus, and RTS-GMLC; and four distribution systems: 22-bus, 85-bus, 123-bus, and 144-bus radial. The data generation process is the same as in the previous chapter.

5.3.2 *Validation of Performance Guaranties*

To demonstrate the stealthiness of our framework, we evaluate the performance guarantees in Theorem 7 and Lemma 3. These numerical quantifications are the residual and reconstruction errors for the 14-bus test case for the noiseless and noisy cases.

Noiseless case

In Theorem 7, it is noted that the residual error, ε, is proportional to the hidden dimension of the AE. In the noiseless case, the original measurement results in perfect state estimation, yielding a zero residual error, $\hat{\varepsilon}$. The autoencoded residual error, $\hat{\varepsilon}^{\text{AE}}$, is always upper bounded by $\hat{\varepsilon}$, resulting in $\hat{\varepsilon}^{\text{AE}} = 0$ for all hidden sizes, as shown in Fig. 5.2a. Furthermore, Lemma 3 states that the reconstruction error is inversely proportional to the hidden dimension size. This relationship is illustrated in Fig. 5.2b for various hidden dimensions.



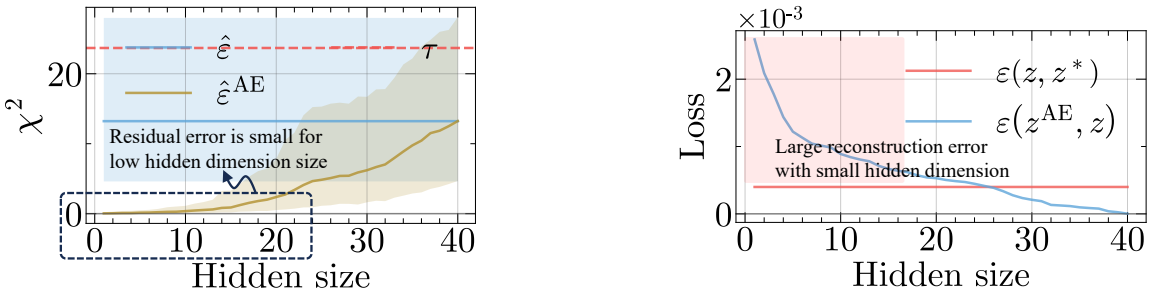
(a) Residual error.

(b) Reconstruction error.

Figure 5.2: Errors for Clean 14-bus Dataset.

Noisy case

Theorem 7 asserts that ε is proportional to AE's hidden dimension. $\hat{\varepsilon}^{\text{AE}} \leq \hat{\varepsilon}$ implies the smallest error with the smallest hidden dimension, as in Fig. 5.3a. This aligns with Theorem 7. Lemma 3 states reconstruction error inversely scales with hidden dimension. In Fig. 5.3b, the error decreases as the hidden size increases.



(a) Residual error.

(b) Reconstruction error.

Figure 5.3: Errors for the Noisy 14-bus Test Case.

Relationship between System States and AE's Latent Variables

In this section, we test the Assumption 1 and Theorem 5. Specifically, we measure the distance/error between the underlying system states and the AE's latent representations. Let $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$ be a measurement sample and $\boldsymbol{x}_i \in \mathbb{R}^{n\times 1}$ be the vector of the underlying system states associated with the i-th measurement vector $\boldsymbol{z}_i$.

In Eq. (5.1), the encoder $f_{\boldsymbol{w}}$ transforms an input $\boldsymbol{z}_i \in \mathbb{R}^{m\times 1}$ into a hidden representation $\boldsymbol{\psi}_i = f_{\boldsymbol{w}}(\boldsymbol{z}_i) \in \mathbb{R}^{n\times 1}$. The resulting hidden representation $\boldsymbol{\psi}_i$ is then mapped back to a reconstructed vector $\boldsymbol{z}_i^{\text{AE}} = g_{\boldsymbol{v}}(\boldsymbol{\psi}_i) \in \mathbb{R}^{m\times 1}$. In the given setup, we seek the connection between system states $\boldsymbol{v}_i$ and AE's hidden representations $\boldsymbol{\psi}_i$. To bridge the difference in their learned representations, we apply a linear transformation: $\boldsymbol{x}_i = \boldsymbol{T}\boldsymbol{\psi}_i$. where $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ is a matrix that represents the linear transformation.

This problem can formulated as $\min_{\boldsymbol{T}} \|\boldsymbol{v}_i - \boldsymbol{T}\boldsymbol{\psi}_i\|^2$. After solving this problem for each sample, we obtain the mean absolute percentage error (MAPE) between system states and transformed hidden representations. This metric MAPE $(\cdot)$ is widely used in the literature due to its properties, such as scale-independency and interoperability. Table 5.1 shows the results for all the test systems, where we can see that all the MAPEs have small values, giving indication that Assumption 1 and Theorem 5 hold in practice.

Table 5.1: MAPE Between System States and Transformed Ae's Latent Representations.

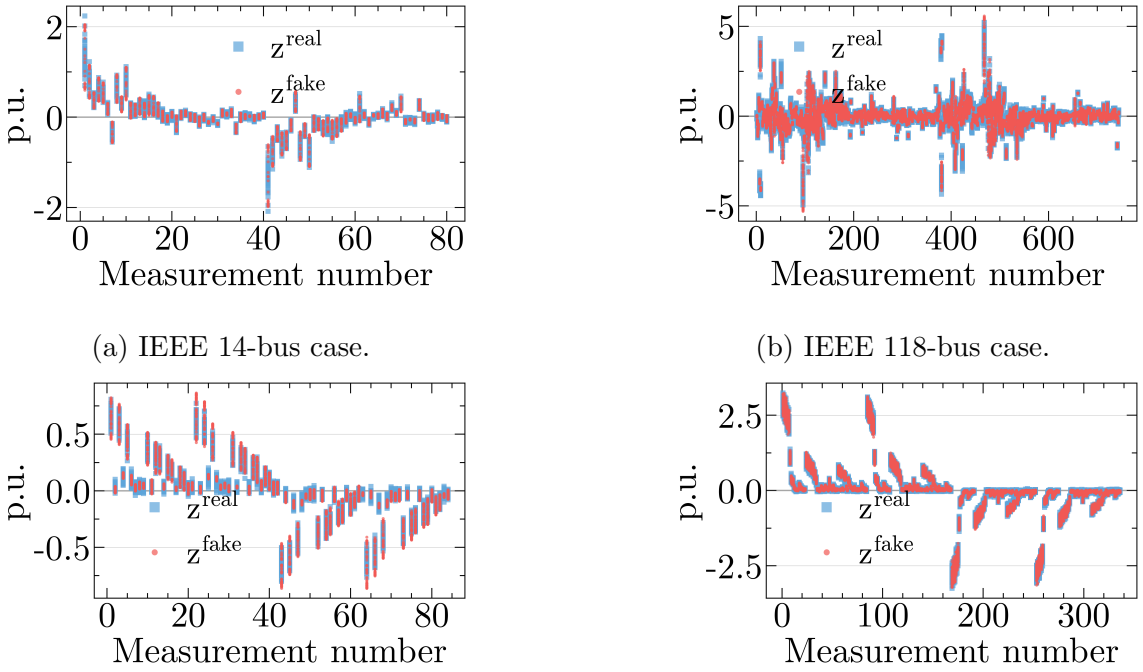
	Test case	MAPE (%)
Trans.	14-bus	1.0462
	RTS-GMLC	6.9421
	118-bus	4.9311
	300-bus	8.6370
Dist.	22-bus radial	2.046
	85-bus radial	0.7224
	123-bus radial	0.2386
	141-bus radial	5.8980

5.3.3 Evaluating Attack Performance

In this section, we assess the attack's performance across various test cases, focusing on (1) sample quality, (2) system state alteration, (3) attack stealthiness measured by success rate, and (4) attack sensitivity concerning the number of system measurements.

Quality of Created Samples

In Fig. 5.4, real measurements (blue) and fake measurements (red) generated by our framework are displayed. While the fake measurements resemble real ones, they do not entirely overlap, indicating the generation of novel samples outside the original dataset. This aligns with our expectations due to the attack regularization term's influence, as discussed in Lemma 2.



(a) IEEE 14-bus case.

(b) IEEE 118-bus case.

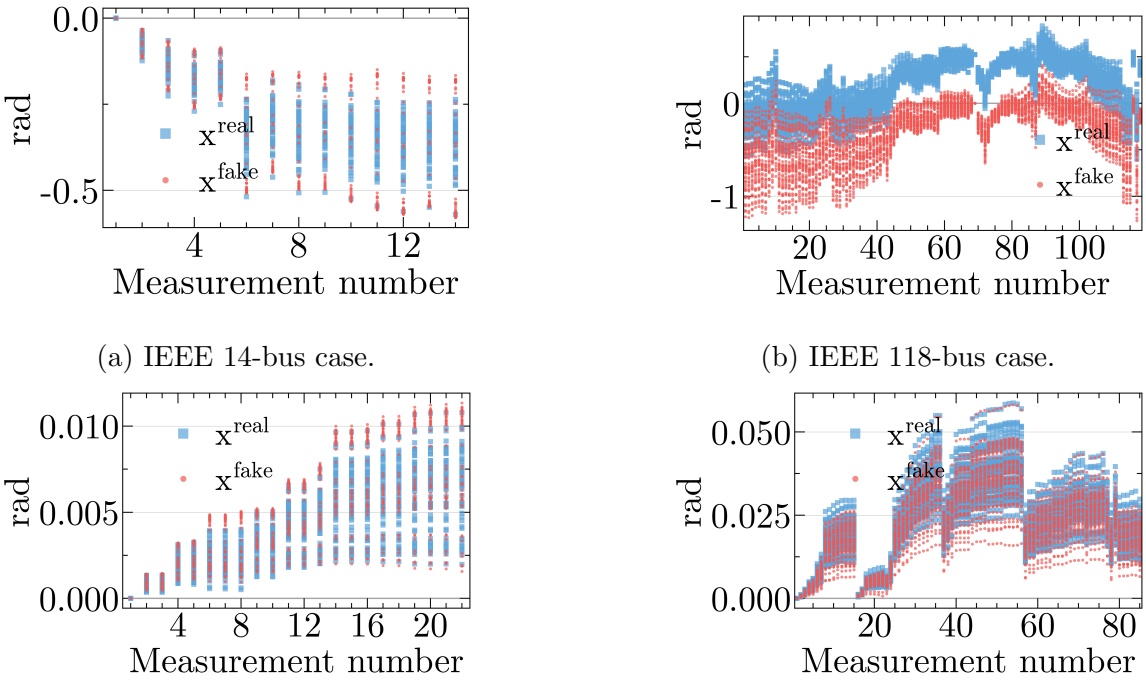
(c) 22-bus radial distribution system.

(d) 85-bus radial distribution system.

Figure 5.4: Measurement Distribution for Different Test Networks.

Perturbation of System States

In Fig. 5.5, real system states (blue) and fake measurements' states (red) are compared. Fake states exhibit increased dispersion, fulfilling the attacker's objective of significantly altering system states solely through measurement tampering.



(a) IEEE 14-bus case.

(b) IEEE 118-bus case.

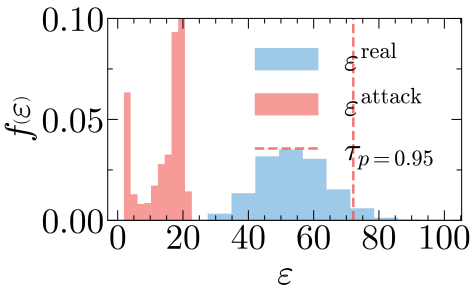
(c) 22-bus radial distribution system.

(d) 85-bus radial distribution system.

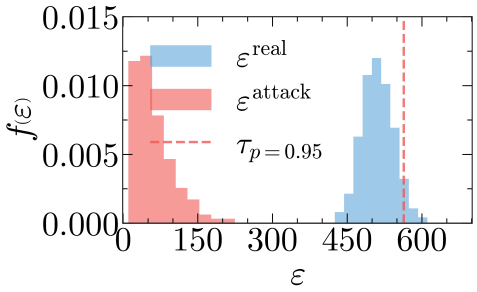
Figure 5.5: Real and Fake States Distribution for Different Test Networks.

Assessing Attack Stealth

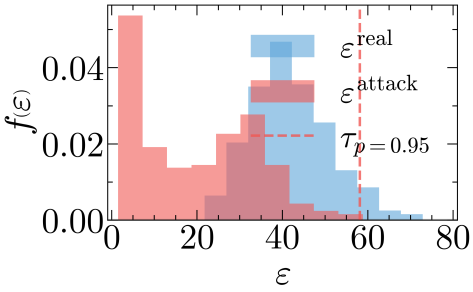
In Fig. 5.6, the PDF of residual errors for real (blue) and fake (red) measurements is depicted. Notably, fake measurements consistently yield smaller residual errors than real ones, validating the concept described in Theorem 7. In other words, autoencoded measurements exhibit smaller residual errors than real ones, making fake measurements more likely to pass the chi-squared test. We conduct a sensitivity analysis regarding the success rate with a given threshold. The experiment involves determining the threshold τ corresponding to a confidence value p (i.e., $\tau = \chi^2_{k,p}$) and calculating the probability of each measurement passing the residual error test at this threshold, denoted as $\Pr\left(J(\mathbf{z}) \geq \tau\right)$. This process is repeated for each simulation to yield the success rate of passing the error test, representing the simulation's probabil-



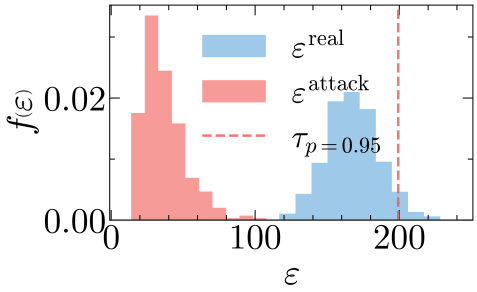
(a) IEEE 14-bus case.



(b) IEEE 118-bus case.



(c) 22-bus radial distribution system.



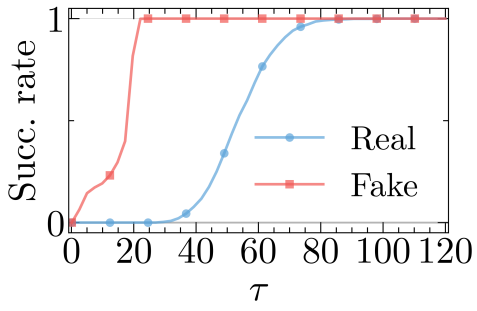
(d) 85-bus radial distribution system.

Figure 5.6: Residual Error's PDFs.

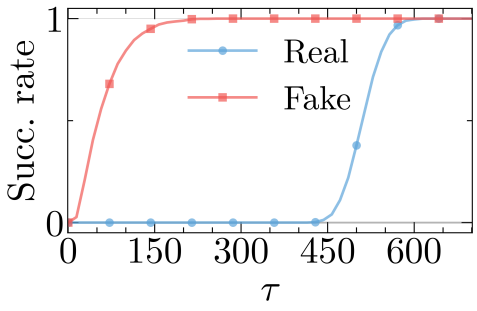
ity of success. We repeat this experiment for various p values in the range (0,1). The results, depicted in Fig. 5.7 illustrates this sensitivity analysis for both real (blue) and fake (red) measurements across different test grids. Fake measurements consistently exhibit a higher success rate than real ones, making it unlikely for the attack to be detected with any chosen threshold (τ).

Measurement Coverage Sensitivity

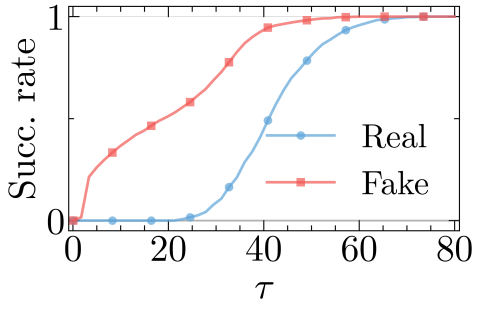
Theorem 6 confirms our framework's ability to generate quality samples that evade the residual error test when the system is observable. We assess the validity of this assertion by examining the attack's success rate in relation to measurement coverage. To accomplish this, we randomly eliminate $n^{\text{unknown}} \in \mathbb{Z}$ measurements and conduct the attack, repeating the process for $n^{\text{unknown}} \in [0, m]$. In Fig. 5.8, the success rates



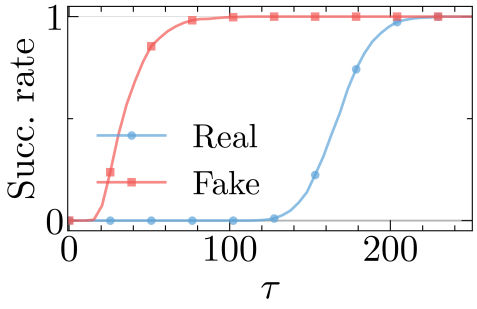
(a) IEEE 14-bus case.



(b) IEEE 118-bus case.



(c) 22-bus radial distribution system.



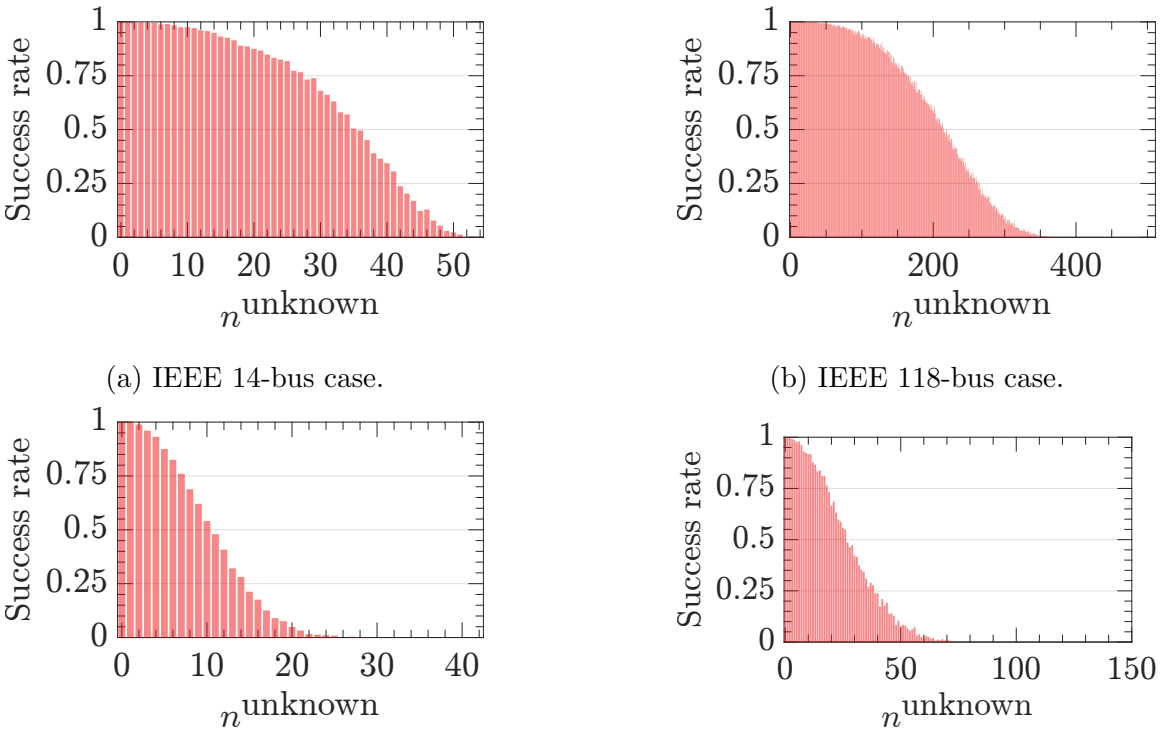
(d) 85-bus radial distribution system.

Figure 5.7: Residual Error's CDFs.

for test cases follow a consistent pattern: they decrease as more measurements are removed, making the attack more challenging due to reduced information. Additionally, the success rate drops to zero after a certain threshold of removed measurements as the system becomes unobservable, resulting in an underdetermined state estimation problem.

Attack Analysis with Limited Tampered Measurements

Risk (ρ) is the expectation of loss Wald (1945); DeGroot (2005); in this context, it can be interpreted as the attack impact. This is given as $\rho = P \times \xi$, where P is the probability of bypassing the residual error test, and ξ is the magnitude of the attack. P is the attack success rate and $\xi = \|\boldsymbol{a}\|^2$ is the attack vector magnitude. P



(a) IEEE 14-bus case.

(b) IEEE 118-bus case.

(c) 22-bus radial distribution system.

(d) 85-bus radial distribution system.

Figure 5.8: Sensitivity of Measurement Coverage.

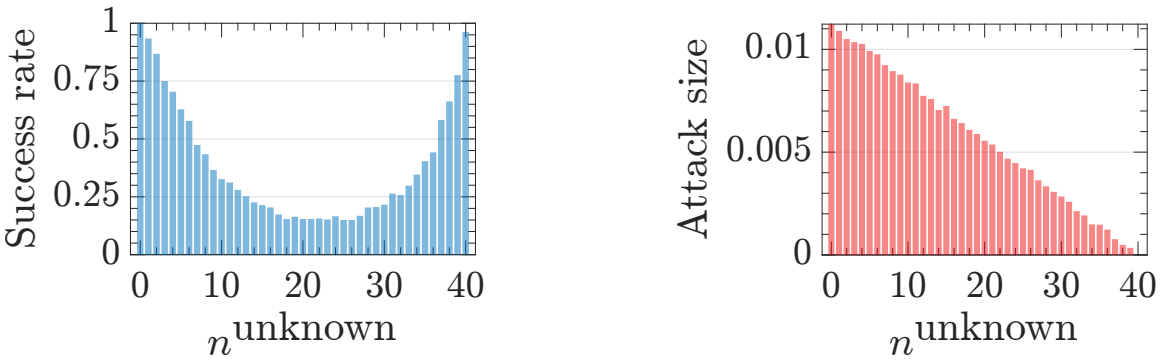
is obtained by computing the attack's success rate when n^{unknown} measurements are in the system. In this case, the defender has access to all the measurements in the system, and the attacker can only modify a subset of the measurements.

To assess the attack's impact, three scenarios are evaluated with respect to the attacker's measurement knowledge. *Scenario A*: The attacker can see all the measurement information in the system but can only modify a subset of the observed measurements. Under this scenario, the attacker has enough knowledge about the number of states that are directly related to the attack. Thus, the AE's latent dimension is set as the number of observed states by the measurements under attack. *Scenario B*: The attacker can see and change only a subset of observed measurements. He also has enough system knowledge to infer the number of states associated with

the observed measurements. Thus, the AE's latent dimension is set as the number of observed states by the measurements under attack. *Scenario C*: The attacker can see and change only a subset of observed measurements. In this case, he lacks system knowledge to infer the number of states associated with the observed measurements. Thus, the AE's latent dimension is set as the number of states in the system. If the number or measurements is lower than the summer of states in the system, then the AE is omitted in the model.

The proposed model is evaluated under the three aforementioned cases with different unknown measurements. Specifically, for each case, we randomly eliminate $n^{\text{unknown}} \in \mathbb{Z}$ measurements and conduct the attack, repeating the process for $n^{\text{unknown}} \in [0, m]$. The result success rate P and the attack size $\|\boldsymbol{a}\|^2$ for the IEEE 14-bus test case are shown in Fig. 5.9a and Fig. 5.9b, respectively. Based on these results, the risk is computed as $\rho = P \times \xi$, and the results for the three cases are shown in Fig. 5.10a. The same procedure is carried out on the 85-bus radial distribution system, and the results for the three cases are shown in Fig. 5.10b. Figs. 5.10a and 5.10b show the same trend: The attack impact is the highest for *scenario A* and decreases for *scenario B* and *scenario C* being the latter the one with the lowest attack impact. In other words, reducing the attacker's knowledge lessens the attack's impact.

The available measurements for the previous simulations were the active and reactive power flow on transmission lines on one end. We also investigate the effect of different measurements on the attack under the three aforementioned scenarios. More measurements are available to evaluate such an effect. Specifically, active and reactive flow measurements on both line ends. Additionally, measurements for voltage magnitudes and angles are included in this case. Fig. 5.11 shows the results for the three scenarios for the IEEE 14-bus test case. Despite including more diverse mea-



(a) Success rate.

(b) Attack size vector.

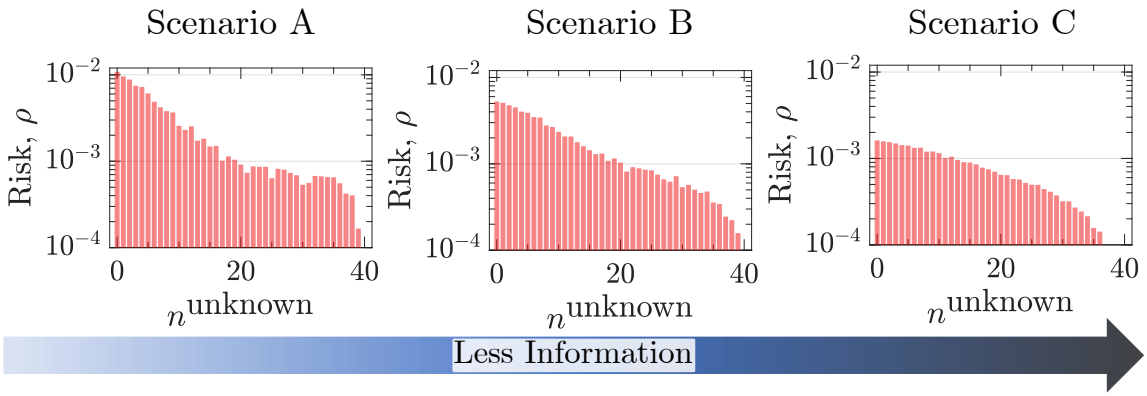
Figure 5.9: Success Rate and Attack Vector Size with Partial Observability for the IEEE 14-bus Test Case.

surements, the same trend continues: The system risk decreases as less information is available for the attacker.

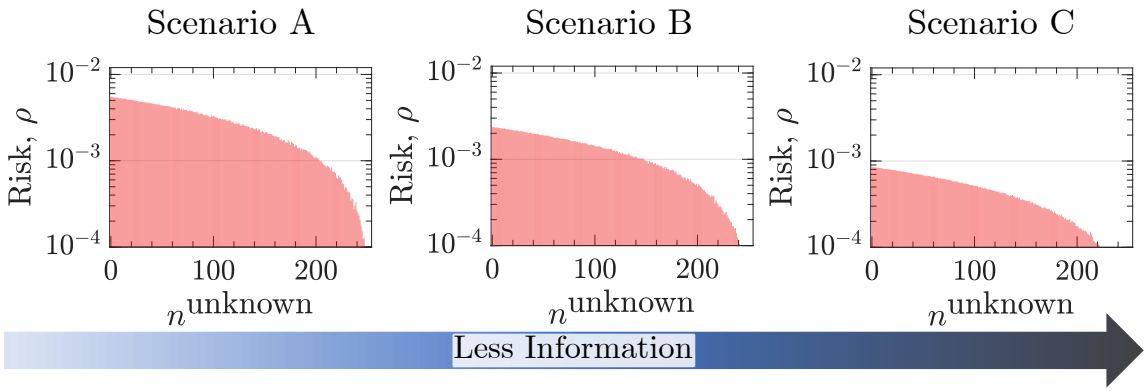
Comparison with alternative FDIA methods

To evaluate our model-free FDIA framework, we compare it to two model-based (MB 1 introduced in Hug and Giampapa (2012) and MB 2 from Chin *et al.* (2017)) and two model-free attacks (MF 1 introduced in Mohammadpourfard *et al.* (2020) and MF 2 proposed in Costilla-Enriquez and Weng (2022)). For model-based attacks, random attack vectors are injected, while model-free attacks are trained exclusively on historical measurements. To facilitate a comparison between these methods, we follow the sample generation process outlined in Section 3.5.1 and apply our proposed approach along with Methods MB 1, MB 2, MF 1, and MF 2 to tamper the real noisy measurements. During these comparisons, we monitor two critical metrics: (i) the attack success rate and (ii) the size of the attack vector.

We assess success rates for all methods by calculating the probability of measurements passing a residual error test at a confidence level of 0.95, using the threshold



(a) Attack impact for the IEEE 14-bus test case.



(b) Attack impact for the 85-bus radial distribution system.

Figure 5.10: Attack Impact Analysis with Partial Observability for Different Levels of Knowledge.

$\tau = \chi^2_{k,0.95}$. Results are presented in Table 5.2. Our method consistently achieves success rates above 95% by reducing tampered sample error. Model-based methods MB1 and MB2 also maintain a 95% success rate, guaranteed by their design (proof in Hug and Giampapa (2012)). MF1 shows around 95% success with variability due to training and no guarantees. MF2 exceeds 95% success, thanks to error-reducing architecture.

Note that model-based attacks can introduce arbitrarily large attack vectors. Thus, we assess the attack vector size ($|\boldsymbol{a}|^2$) only for model-free attacks. The results

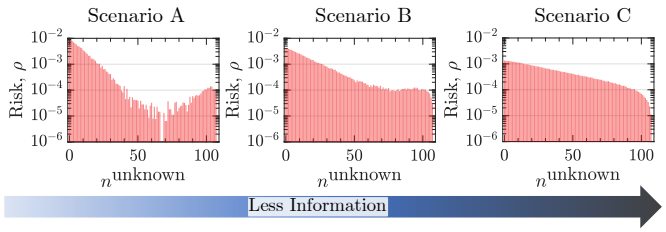


Figure 5.11: Attack Impact Analysis with Partial Observability for Different Levels of Knowledge with All Measurements for the IEEE 14-bus Test Case.

Table 5.2: Comparison of Attack Success Rates.

	Test case	Success Rate (%)				
		Ours	MB 1	MB 2	MF 1	MF 2
Trans.	14-bus	100	94.9	96.6	94	98.2
	RTS-GMLC	97	94.5	95.9	93.8	96.8
	118-bus	100	95.7	95	95	98.6
	300-bus	97.7	94.9	95.9	94	98.4
Dist.	22-bus radial	99.6	94.2	95.8	96.8	96.4
	85-bus radial	100	95	95.6	93.4	96.2
	123-bus radial	96	94	95.3	95.4	91.6
	141-bus radial	98.7	95.9	94	91.2	93.2

in Table 5.3 reveal that our approach generates the largest attack vectors, demonstrating its superior effectiveness.

5.4 Conclusion

This study introduced an innovative architecture for generating tampered measurement vectors, enabling a FDIA without prior knowledge of the power system's

Table 5.3: Comparison of Attack Vector Size.

	Test case	Attack vector norm, $\|\boldsymbol{a}\|^2$		
		Ours	MF 1	MF 2
Trans.	14-bus	0.0460	0.0114	0.0442
	RTS-GMLC	0.7419	0.4555	0.5701
	118-bus	0.1272	0.0571	0.1088
	300-bus	0.5431	0.4857	0.3619
Dist.	22-bus radial	0.0069	0.0035	0.0066
	85-bus radial	0.0144	0.0079	0.0104
	123-bus radial	0.0055	0.0048	0.0036
	141-bus radial	0.0014	0.0008	0.0012

details. Our approach is embedded within an optimization framework incorporating the WGAN and two regularization terms for controlling the attack vectors. We validate this framework across various power systems, demonstrating its ability to introduce deceptive measurements for a bad data injection attack without the need for in-depth knowledge of the underlying power system model. These manipulated measurements successfully pass the residual error test, resulting in inaccurate estimated state variables and measurements, thereby posing a significant threat to the reliability of the electric grid. This research underscores that attackers can compromise power system security without complete access to system information.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Modern power systems must adapt to the ongoing changes, such as introducing active devices (e.g., PV panels, wind generators, and energy storage devices) and the burgeoning integration of measurement devices to monitor, control, and protect the power grid. However, increasing uncertainties and stress can make analytical solutions fail, and new sensing and communication capability may expose the power grid to rising cyber-attacks. To solve the first problem, Chapter 2 proposed a novel hybrid method that improves the power flow problem's convergence. This method combines the Newton-Raphson and stochastic gradient algorithms, which improves the convergence of the power flow problem when the initialization points are poor or when the system is stressed. To solve the second cyber-security problem, Chapter 3 exposed a vulnerability in security on power systems. Specifically, this work shows that an attacker is not required to access all power system information to launch a FDIA successfully. Chapters 4 and 5 formally analyzed the architecture to create tampered measurement vectors to carry out a FDIA on DC and AC models without knowing the power system's underlying information. Overall, this work aims to explore and propose approaches to enhance the robust system planning, control, and operation with increasingly distributed energy resources. To add a layer of security simultaneously, we analyze the system vulnerability for systematic solutions for defenses. If successful, this work can lay down the foundation for security-aware system operation in the future.

6.2 Future Work

We now highlight directions for future work for (I) the proposed hybrid method to solve the power flow problem and (II) to analyze and defend the system against false data injection attacks.

Hybrid Newton-Raphson and Stochastic Gradient Descend Method

- Investigate the proposed hybrid method utilizing the Jacobian approximation at each iteration.
- Find feasible operating points on the AC model based on an optimal solution from a DC model.

Model-Free False Data Injection Attack

- Consider inter-temporal restrictions in the attack model.
- Assess the attack's impact on the system. For example, to determine if an attacker can lead the system to a blackout.
- Develop a theoretically sound defense to detect both model-based and model-free attacks. Such a defense model should consider the lack of attack samples in the training dataset.

REFERENCES

Abhyankar, S., Q. Cui and A. J. Flueck, "Fast power flow analysis using a hybrid current-power balance formulation in rectangular coordinates", in "IEEE PES T&D Conference and Exposition", pp. 1–5 (2014).

Abur, A. and A. G. Exposito, *Power system state estimation: theory and implementation* (CRC press, 2004).

Ahmadian, S., H. Malki and Z. Han, "Cyber attacks on smart energy grids using generative adverserial networks", in "Global Conference on Signal and Information Processing", pp. 942–946 (IEEE, 2018).

Ajjarapu, V. and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis", IEEE transactions on Power Systems **7**, 1, 416–423 (1992).

Arjovsky, M., S. Chintala and L. Bottou, "Wasserstein generative adversarial networks", in "Proceedings of the International Conference on Machine Learning", vol. 70, pp. 214–223 (2017).

Arrillaga, J. A. and B. Harker, *Computer modelling of electrical power systems* (John Wiley & Sons, Inc., 1983).

Bacher, R. and W. Tinney, "Faster local power flow solutions: the zero mismatch approach", IEEE Transactions on Power Systems **4**, 4, 1345–1354 (1989).

Baldi, P. and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima", Neural networks **2**, 1, 53–58 (1989).

Battiti, R., "First-and second-order methods for learning: between steepest descent and newton's method", Neural computation **4**, 2, 141–166 (1992).

Bertsekas, D. P., "Nonlinear programming", Journal of the Operational Research Society **48**, 3, 334–334 (1997).

Braz, L. M., C. A. Castro and C. Murati, "A critical evaluation of step size optimization based load flow methods", IEEE Transactions on Power Systems **15**, 1, 202–207 (2000).

Brunton, S. L. and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control* (Cambridge University Press, 2019).

Chapelle, O., B. Scholkopf and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]", IEEE Transactions on Neural Networks **20**, 3, 542–542 (2009).

Chen, Y., S. Huang, F. Liu, Z. Wang and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control", IEEE Transactions on Smart Grid **10**, 2, 2158–2169 (2018).

Chen, Y. and C. Shen, "A Jacobian-free newton method with adaptive preconditioner and its application for power flow calculations", IEEE Transactions on Power Systems **21**, 3, 1096–1103 (2006).

Chin, W.-L., C.-H. Lee and T. Jiang, "Blind false data attacks against ac state estimation based on geometric approach in smart grid communications", IEEE Transactions on Smart Grid **9**, 6, 6298–6306 (2017).

Costilla-Enriquez, N. and Y. Weng, "Exposing cyber-physical system weaknesses by implicitly learning their underlying models", in "Asian Conference on Machine Learning", pp. 1333–1348 (PMLR, 2021).

Costilla-Enriquez, N. and Y. Weng, "Attack power system state estimation by implicitly learning the underlying models", IEEE Transactions on Smart Grid **14**, 1, 649–662 (2022).

Costilla-Enriquez, N., Y. Weng and B. Zhang, "Combining newton-raphson and stochastic gradient descent for power flow analysis", IEEE Transactions on Power Systems **36**, 1, 514–517 (2020).

Cybenko, G., "Approximation by superpositions of a sigmoidal function", Mathematics of control, signals and systems **2**, 4, 303–314 (1989).

Da Costa, V. and A. Rosa, "A comparative analysis of different power flow methodologies", IEEE PES Transmission and Distribution Conference and Exposition: Latin America pp. 1–7 (2008).

da Costa, V. M., N. Martins and J. L. R. Pereira, "Developments in the newton raphson power flow formulation based on current injections", IEEE Transactions on power systems **14**, 4, 1320–1326 (1999).

DeGroot, M. H., *Optimal statistical decisions*, vol. 82 (John Wiley & Sons, 2005).

Electric Reliability Council of Texas, (ERCOT), "Hourly Load Data Archives", `https://www.ercot.com/gridinfo/load/load_hist/`, Accessed: 06/14/2022 (2022).

Expósito, A. G. and E. R. Ramos, "Augmented rectangular load flow model", IEEE Transactions on Power Systems **17**, 2, 271–276 (2002).

Ferraty, F. and P. Vieu, *Nonparametric functional data analysis: theory and practice* (Springer Science & Business Media, 2006).

Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)* (The MIT Press, 2016).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets", in "Advances in Neural Information Processing Systems", vol. 27, pp. 2672–2680 (2014).

He, Y., G. J. Mendis and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism", IEEE Transactions on Smart Grid **8**, 5, 2505–2516 (2017).

Hinton, G. E. and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", science **313**, 5786, 504–507 (2006).

Hollander, M. and J. Sethuraman, "Nonparametric statistics: Advanced computational methods", (2001).

Hornik, K., M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", Neural networks **2**, 5, 359–366 (1989).

Hug, G. and J. A. Giampapa, "Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks", IEEE Transactions on Smart Grid **3**, 3, 1362–1370 (2012).

Ji, G., M. C. Hughes and E. B. Sudderth, "From patches to images: a nonparametric generative model", in "International Conference on Machine Learning", pp. 1675–1683 (PMLR, 2017).

Jia, L., R. J. Thomas and L. Tong, "On the nonlinearity effects on malicious data attack on power system", in "IEEE Power and Energy Society General Meeting", pp. 1–8 (2012).

Jin, M., J. Lavaei and K. H. Johansson, "Power grid ac-based state estimation: Vulnerability analysis against cyber attacks", IEEE Transactions on Automatic Control **64**, 5, 1784–1799 (2019).

Kim, J., L. Tong and R. J. Thomas, "Subspace methods for data attack on state estimation: A data driven approach", IEEE Transactions on Signal Processing **63**, 5, 1102–1114 (2014).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", International Conference on Learning Representations (2014).

Kosut, O., Liyan Jia, R. J. Thomas and Lang Tong, "Limiting false data attacks on power system state estimation", in "Conference on Information Sciences and Systems", pp. 1–6 (2010).

Kothari, D. P. and I. Nagrath, *Modern power system analysis* (Tata McGraw-Hill Education, 1989).

Kundur, P., J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor *et al.*, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions", IEEE transactions on Power Systems **19**, 3, 1387–1401 (2004).

Lakshminarayana, S., S. Sthapit and C. Maple, "Application of physics-informed machine learning techniques for power grid parameter estimation", Sustainability **14**, 4, 2051 (2022).

Lakshminarayana, S. and D. K. Yau, "Cost-benefit analysis of moving-target defense in power grids", IEEE Transactions on Power Systems **36**, 2, 1152–1163 (2020).

Li, T., R. Mehta, Z. Qian and J. Sun, "Rethink autoencoders: robust manifold learning", in "ICML Workshop on Uncertainty and Robustness in Deep Learning", (2020).

Li, X., X. Liang, R. Lu, X. Shen, X. Lin and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges", IEEE Communications Magazine **50**, 8, 38–45 (2012).

Liang, G., J. Zhao, F. Luo, S. R. Weller and Z. Y. Dong, "A review of false data injection attacks against modern power systems", IEEE Transactions on Smart Grid **8**, 4, 1630–1638 (2017).

Liu, X. and Z. Li, "False data attacks against ac state estimation with incomplete network information", IEEE Transactions on Smart Grid **8**, 5, 2239–2248 (2017).

Liu, Y., P. Ning and M. K. Reiter, "False data injection attacks against state estimation in electric power grids", ACM Transactions on Information and System Security **14**, 1, 1–33 (2011).

Lourenco, E. M., A. S. Costa and R. R. P Jr, "Steady-state solution for power networks modeled at bus section level", IEEE Transactions on Power Systems **25**, 1, 10–20 (2010).

Luo, G.-X. and A. Semlyen, "Efficient load flow for large weakly meshed networks", IEEE Transactions on Power Systems **5**, 4, 1309–1316 (1990).

Milano, F., "Continuous newton's method for power flow analysis", IEEE Transactions on Power Systems **24**, 1, 50–57 (2008).

Milano, F., "Continuous newton's method for power flow analysis", IEEE Transactions on Power Systems **24**, 1, 50–57 (2009).

Mo, Y., T. H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure", Proceedings of the IEEE **100**, 1, 195–209 (2012).

Mohammadpourfard, M., F. Ghanaatpishe, M. Mohammadi, S. Lakshminarayana and M. Pechenizkiy, "Generation of false data injection attacks using conditional generative adversarial networks", in "Innovative Smart Grid Technologies Europe", pp. 41–45 (IEEE, 2020).

Mohammadpourfard, M., A. Sami and Y. Weng, "Identification of false data injection attacks with considering the impact of wind generation and topology reconfigurations", IEEE Transactions on Sustainable Energy **9**, 3, 1349–1364 (2018).

Mohammadpourfard, M., Y. Weng and M. Tajdinian, "Benchmark of machine learning algorithms on capturing future distribution network anomalies", IET Generation, Transmission Distribution **13**, 8, 1441–1455 (2019).

Molzahn, D. K., V. Dawar, B. C. Lesieutre and C. L. DeMarco, "Sufficient conditions for power flow insolvability considering reactive power limited generators with applications to voltage stability margins", in "IREP Symposium Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid", pp. 1–11 (2013).

National Academies of Sciences, Engineering, and Medicine, *Enhancing the resilience of the nation's electricity system* (National Academies Press, 2017).

Pirnia, M., C. A. Cañizares and K. Bhattacharya, "Revisiting the power flow problem based on a mixed complementarity formulation approach", IET Generation, Transmission & Distribution **7**, 11, 1194–1201 (2013).

Preston, E. and C. Barrows, "Evaluation of year 2020 IEEE RTS generation reliability indices", in "IEEE International Conference on Probabilistic Methods Applied to Power Systems", pp. 1–5 (2018).

Radford, A., L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", arXiv preprint arXiv:1511.06434 (2015).

Sasson, A. M., C. Trevino and F. Aboytes, "Improved newton's load flow through a minimization technique", IEEE Transactions on Power Apparatus and Systems , 5, 1974–1981 (1971).

Semlyen, A., "Fundamental concepts of a krylov subspace power flow methodology", IEEE Transactions on Power Systems **11**, 3, 1528–1537 (1996).

Shahriar, M. S., I. O. Habiballah and H. Hussein, "Optimization of phasor measurement unit (pmu) placement in supervisory control and data acquisition (scada)-based power system for better state-estimation performance", Energies **11**, 3, 570 (2018).

Stott, B., "Effective starting process for newton-raphson load flows", **118**, 8, 983–987 (1971).

Stott, B., "Review of load-flow calculation methods", Proceedings of the IEEE **62**, 7, 916–929 (1974).

Stott, B. and O. Alsac, "Fast decoupled load flow", IEEE transactions on power apparatus and systems , 3, 859–869 (1974).

Styczynski, J. and N. Beach-Westmoreland, "When The Lights Went Out: A Comprehensive Review Of The 2015 Attacks On Ukrainian Critical Infrastructure", URL `https://www.boozallen.com/s/insight/thought-leadership/lessons-from-ukranians-energy-grid-cyber-attack.html` (2019).

Tarali, A. and A. Abur, "Bad data detection in two-stage state estimation using phasor measurements", in "IEEE PES Innovative Smart Grid Technologies Europe", pp. 1–8 (2012).

Teixeira, A., K. C. Sou, H. Sandberg and K. H. Johansson, "Secure control systems: A quantitative risk management approach", IEEE Control Systems Magazine **35**, 1, 24–45 (2015).

Tinney, W. F. and C. E. Hart, "Power flow solution by newton's method", IEEE Transactions on Power Apparatus and systems , 11, 1449–1460 (1967).

Valenzuela, J., J. Wang and N. Bissinger, "Real-time intrusion detection in power system operations", IEEE Transactions on Power Systems **28**, 2, 1052–1062 (2013).

Van Cutsem, T. and C. Vournas, *Voltage stability of electric power systems* (Springer Science & Business Media, 2007).

Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.", Journal of machine learning research **11**, 12 (2010).

Wainwright, K. *et al.*, *Fundamental methods of mathematical economics/Alpha C. Chiang, Kevin Wainwright.* (Boston, Mass.: McGraw-Hill/Irwin,, 2005).

Wald, A., "Statistical decision functions which minimize the maximum risk", Annals of Mathematics **46**, 2, 265–280, URL `http://www.jstor.org/stable/1969022` (1945).

Wang, W. and Z. Lu, "Cyber security in the smart grid: Survey and challenges", Computer networks **57**, 5, 1344–1371 (2013).

Wang, Z., H. He, Z. Wan and Y. Sun, "Detection of false data injection attacks in ac state estimation using phasor measurements", IEEE Transactions on Smart Grid pp. 1–1 (2020).

Wasserman, L., *All of nonparametric statistics* (Springer Science & Business Media, 2006).

Weng, Y., R. Negi, C. Faloutsos and M. D. Ilić, "Robust data-driven state estimation for smart grid", IEEE Transactions on Smart Grid **8**, 4, 1956–1967 (2016).

Weng, Y., R. Negi and M. D. Ilić, "Probabilistic joint state estimation for operational planning", IEEE Transactions on Smart Grid **10**, 1, 601–612 (2017).

Weng, Y., R. Rajagopal and B. Zhang, "A geometric analysis of power system loadability regions", IEEE Transactions on Smart Grid (2019).

Wood, A. J. and B. F. Wollenberg, *Power generation, operation, and control* (John Wiley & Sons, 2012).

Wood, A. J., B. F. Wollenberg and G. B. Sheblé, *Power generation, operation, and control* (John Wiley & Sons, 2013).

Xie, L., Y. Mo and B. Sinopoli, "False data injection attacks in electricity markets", in "IEEE International Conference on Smart Grid Communications", pp. 226–231 (2010).

Xie, L., Y. Mo and B. Sinopoli, "Integrity data attacks in power market operations", IEEE Transactions on Smart Grid **2**, 4, 659–666 (2011).

Yu, Z.-H. and W.-L. Chin, "Blind false data injection attack using pca approximation method in smart grid", IEEE Transactions on Smart Grid **6**, 3, 1219–1226 (2015).

Yuan, Y., Z. Li and K. Ren, "Modeling load redistribution attacks in power systems", IEEE Transactions on Smart Grid **2**, 2, 382–390 (2011).

Zhang, B. and D. Tse, "Geometry of injection regions of power networks", IEEE Transactions on Power Systems **28**, 2, 788–797 (2013).

Zhang, J. and L. Sankar, "Physical system consequences of unobservable state-and-topology cyber-physical attacks", IEEE Transactions on Smart Grid **7**, 4, 2016–2025 (2016).

Zhang, P., Q. Liu, D. Zhou, T. Xu and X. He, "On the discrimination-generalization tradeoff in GANs", in "International Conference on Learning Representations", (2018).

Zhang, Y., J. Wang and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach", IEEE Transactions on Smart Grid **12**, 1, 623–634 (2020).

Zhang, Y., L. Wang, Y. Xiang and C. Ten, "Power system reliability evaluation with SCADA cybersecurity considerations", IEEE Transactions on Smart Grid **6**, 4, 1707–1721 (2015).

Zhang, Z., R. Deng, D. K. Yau and P. Chen, "Zero-parameter-information data integrity attacks and countermeasures in iot-based smart grid", IEEE Internet of Things Journal **8**, 8, 6608–6623 (2021).

Zhang, Z., R. Deng, D. K. Yau, P. Cheng and J. Chen, "Analysis of moving target defense against false data injection attacks on power grid", IEEE Transactions on Information Forensics and Security **15**, 2320–2335 (2019).

Zhao, J. and L. Mili, "Vulnerability of the largest normalized residual statistical test to leverage points", IEEE Transactions on Power Systems **33**, 4, 4643–4646 (2018).

Zhou, Y., Y. Liu and S. Hu, "Smart home cyberattack detection framework for sponsor incentive attacks", IEEE Transactions on Smart Grid **10**, 2, 1916–1927 (2019).

Zimmerman, R. D., C. E. Murillo-Sánchez and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education", IEEE Transactions on Power Systems **26**, 1, 12–19 (2011).