False Data Injection Attacks against State Estimation in Electric Power Grids*

Yao Liu, Peng Ning
Department of Computer Science
North Carolina State University
Emails: {yliu20, pning}@ncsu.edu

Michael K. Reiter
Department of Computer Science
University of North Carolina, Chapel Hill
Email: reiter@cs.unc.edu

ABSTRACT

A power grid is a complex system connecting electric power generators to consumers through power transmission and distribution networks across a large geographical area. System monitoring is necessary to ensure the reliable operation of power grids, and *state estimation* is used in system monitoring to best estimate the power grid state through analysis of meter measurements and power system models. Various techniques have been developed to detect and identify bad measurements, including the *interacting bad measurements* introduced by *arbitrary, non-random* causes. At first glance, it seems that these techniques can also defeat malicious measurements injected by attackers.

In this paper, we present a new class of attacks, called *false data injection attacks*, against state estimation in electric power grids. We show that an attacker can exploit the configuration of a power system to launch such attacks to successfully introduce *arbitrary* errors into certain state variables while bypassing existing techniques for bad measurement detection. Moreover, we look at two realistic attack scenarios, in which the attacker is either constrained to some specific meters (due to the physical protection of the meters), or limited in the resources required to compromise meters. We show that the attacker can systematically and efficiently construct attack vectors in both scenarios, which can not only change the results of state estimation, but also modify the results in *arbitrary* ways. We demonstrate the success of these attacks through simulation using IEEE test systems. Our results indicate that security protection of the electric power grid must be revisited when there are potentially malicious attacks.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; E.m [**Data**]: Miscellaneous

General Terms

Algorithms, Security

Keywords

Power grids, state estimation, vulnerability

1. INTRODUCTION

A power grid is a complex system connecting a variety of electric power generators to customers through power transmission and distribution networks across a large geographical area, as illustrated in Figure 1. The security and reliability of power grids has critical impact on society and people's daily life. For example, on August 14, 2003, a large portion of the Midwest and Northeast United States and Ontario, Canada, experienced an electric power blackout, which affected an area with a population of about 50 million people. The estimated total costs range between $4 billion and $10 billion (U.S. dollars) in the United States, and $2.3 billion (Canadian dollars) in Canada [30].

System monitoring is necessary to ensure the reliable operation of power grids. It provides pertinent information on the condition of a power grid based on the readings of meters placed at important area of the power grid. The meter measurements may include bus voltages, bus real and reactive power injections, and branch reactive power flows in every subsystem of a power grid. These measurements are typically transmitted to a *control center*, a component that retains crucial system data and provides centralized monitoring and control capability for the power grid. Measurements are usually stored in a *telemetry system*, which is also known as *Supervisory Control And Data Acquisition (SCADA)* system. *State estimation* is used in system monitoring to best estimate the power grid state through analysis of meter measurement data and power system models.

State estimation is the process of estimating unknown state variables in a power grid based on the meter measurements. The output of state estimation is typically used in contingency analysis, which will then be used to control the power grid components (e.g., to increase the yield of a power generator) to maintain the reliable operation even if some faults (e.g., a generator breakdown) may occur next.

It is possible for an attacker to compromise meters to introduce malicious measurements. For example, there is an online video[1] that teaches people how to manipulate elec-

*This work is supported by the National Science Foundation (NSF) under grant CNS-0831302. The authors would like to thank the anonymous reviewers for their helpful suggestions.

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

[1] `http://www.metacafe.com/watch/811500/electric_`

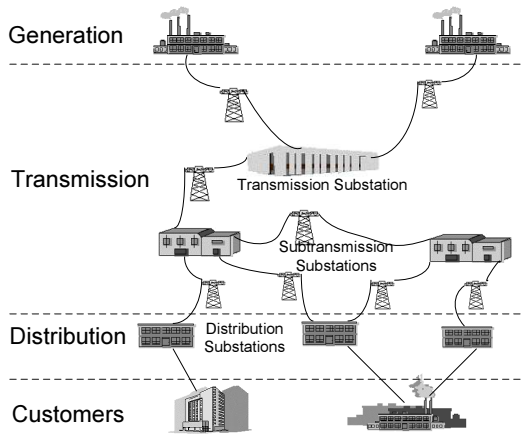

Figure 1: A power grid connecting power plants to customers via power transmission and distribution networks (revised from [2])

tric meters to cut their electricity bills. Though this meter-hacking tutorial is about meters at the end consumers, it is conceivable that attackers have the same kind of ability to modify the meters in the power grid to introduce bad measurements if they have access to these meters. If these bad measurements affect the outcome of state estimation, they can mislead the power grid control algorithms, possibly resulting in catastrophic consequences such as blackouts in large geographic areas.

Power systems researchers have realized the threat of bad measurements and developed techniques for processing them (e.g., [16, 21–25]). These techniques first detect if there are bad measurements, and then identify and remove the bad ones if there are any. Some of these techniques (e.g., [21,23,25]) were targeted at *arbitrary*, interacting (i.e., correlated) bad measurements. At first glance, it seems that these approaches can also defeat the malicious measurements injected by attackers, since such malicious measurements can be considered as interacting bad measurements.

However, in this paper, we discover that all existing techniques for bad measurement detection and identification can be bypassed if the attacker knows the configuration of the power system. The fundamental reason for this failure is that all existing techniques for bad measurement detection rely on the same assumption that "when bad measurements take place, the squares of differences between the observed measurements and their corresponding estimates often become significant [16]." Unfortunately, our investigation indicates that this assumption is not always true. With the knowledge of the power system configuration, the attacker can systematically generate bad measurements so that the above assumption is violated, thus bypassing bad measurements detection.

In this paper, we present a new class of attacks, called *false data injection attacks*, against state estimation in electric power systems. By taking advantage of the configuration information of a power system, an attacker can inject malicious measurements that will mislead the state estimation process without being detected by any of the existing techniques for bad measurement detection.

State estimation uses power flow models. A *power flow model* is a set of equations that depicts the energy flow on each transmission line of a power grid. An *AC power flow model* is a power flow model that considers both real and reactive power and is formulated by nonlinear equations. For large power systems, state estimation using an AC power flow model is computationally expensive and even infeasible in many cases. Thus, power system engineers sometimes only consider the real power and use a linearized power flow model, *DC power flow model*, to approximate the AC power flow model [14,18]. A DC power flow model is less accurate, but simpler and more robust than an AC model due to the linearity [14]. In this paper, as the first step in our research, we focus on attacks against state estimation using DC power flow models. We expect the results of this paper to serve as the foundation for future research for generalized power flow models.

We present false data injection attacks from the attacker's perspective. We first show that it is possible for the attacker to inject malicious measurements that can bypass existing techniques for bad measurement detection. We then look at two realistic attack scenarios. In the first attack scenario, the attacker is constrained to accessing some specific meters due to, for example, different physical protection of the meters. In the second attack scenario, the attacker is limited in the resources required to compromise meters. We consider two realistic attack goals: *random false data injection attacks*, in which the attacker aims to find any attack vector as long as it can lead to a wrong estimation of state variables, and *targeted false data injection attacks*, in which the attacker aims to find an attack vector that can inject *arbitrary* errors into certain state variables. We show that the attacker can systematically and efficiently construct attack vectors for false data injection attacks in both attack scenarios with both attack goals.

We validate these attacks through simulation using the IEEE test systems, including IEEE 9-bus, 14-bus, 30-bus, 118-bus, and 300-bus systems [37]. The simulation results demonstrate the success of these attacks. For example, to inject a specific malicious value into one target state variable, the attacker only needs to compromise 10 meters in most cases in the IEEE 300-bus system, which has 1,122 meters in total.

Practical Implication: We would like to point out that the false data injection attacks do pose strong requirements for the attackers. It requires that the attackers know the configuration of target power system, which is in general not easy to access. Moreover, the attackers have to manipulate some meters or their measurements before they are used for state estimation. Nevertheless, it is critical for power engineers and security people to be aware of this threat. Existing state estimation and the follow-up processes such as contingency analysis assume near-perfect detection of large bad measurements, while our results indicate that the attackers can always bypass the detection by manipulating the measurement values. Such a discrepancy may be amplified in the processes following state estimation and lead to catastrophic impacts (e.g., blackouts in large geographic areas).

The rest of the paper is organized as follows. Section 2 gives background and related work. Section 3 presents the basic principle of false data injection attacks, and gives the approaches for both random and targeted false data injection attacks in the two attack scenarios. Section 4 demonstrates the success of these attacks through simulation. Sec-

`meter_hack_how_to_cut_your_electricity_bill_in_half/`

tion 5 concludes this paper and points out future research directions.

2. PRELIMINARIES

Power System (Power Grid): A *power transmission system* (or simply a *power system*) consists of electric generators, transmission lines, and transformers that form an electrical network [31]. This network is also called a *power grid*. It connects a variety of electric generators together with a host of users across a large geographical area. Redundant paths and lines are provided so that power can be routed from any power plant to any customers, through a variety of routes, based on the economics of the transmission path and the cost of power. A control center is usually used to monitor and control the power system and devices in a geographical area.

State Estimation: Monitoring power flows and voltages in a power system is important in maintaining system reliability. To ensure that a power system continues to operate even when some components fail, power engineers use meters to monitor system components and report their readings to the control center, which estimates the state of power system variables from these meter measurements. Examples of state variables include bus voltage angles and magnitudes.

The state estimation problem is to estimate power system state variables $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ based on the meter measurements $\mathbf{z} = (z_1, z_2, ..., z_m)^T$, where n and m are positive integers and $x_i, z_j \in \mathcal{R}$ for $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$ [31]. More precisely, assume $\mathbf{e} = (e_1, e_2, ..., e_m)^T$ with $e_j \in \mathcal{R}$, $j = 1, 2, ..., m$, are measurement errors, the state variables are related to the measurements through the following model

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}, \tag{1}$$

where $\mathbf{h}(\mathbf{x}) = (h_1(x_1, x_2, ..., x_n), ..., h_m(x_1, x_2, ..., x_n))^T$ and $h_i(x_1, x_2, ..., x_n)$ is a function of $x_1, x_2, ..., x_n$. The state estimation problem is to find an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ that is the best fit of the measurement $\mathbf{z}$ according to Equation (1).

For state estimation using the DC power flow model, Equation (1) can be represented by a linear regression model

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{2}$$

where $\mathbf{H} = (h_{i,j})_{m \times n}$. Three statistical estimation criteria are commonly used in state estimation: *the maximum likelihood criterion*, *the weighted least-square criterion*, and *the minimum variance criterion* [31]. When meter error is assumed to be normally distributed with zero mean, these criteria lead to an identical estimator with the following matrix solution

$$\hat{\mathbf{x}} = (\mathbf{H^T W H})^{-1} \mathbf{H^T W z}, \tag{3}$$

where $\mathbf{W}$ is a diagonal matrix whose elements are reciprocals of the variances of meter errors. That is,

$$\mathbf{W} = \begin{bmatrix} \sigma_1^{-2} & & & & \\ & \sigma_2^{-2} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \sigma_m^{-2} \end{bmatrix}, \tag{4}$$

where σ_i^2 is the variance of the i-th meter ($1 \le i \le m$).

Bad Measurement Detection: Bad measurements may be introduced due to various reasons such as meter failures and malicious attacks. Techniques for bad measurements detection have been developed to protect state estimation [23, 31]. Intuitively, normal sensor measurements usually give an estimate of the state variables close to their actual values, while abnormal ones may "move" the estimated state variables away from their true values. Thus, there is usually "inconsistency" among the good and the bad measurements. Power systems researchers proposed to calculate the *measurement residual* $\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}$ (i.e., the difference between the vector of observed measurements and the vector of estimated measurements), and use its L_2-norm $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ to detect the presence of bad measurements. Specifically, $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ is compared with a threshold τ, and the presence of bad measurements is assumed if $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$.

The selection of τ is a key issue. Assume that all the state variables are mutually independent and the meter errors follow the normal distribution. It can be mathematically shown that $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|^2$, denoted $\mathcal{L}(\mathbf{x})$, follows a $\chi^2(v)$-distribution, where $v = m - n$ is the degree of freedom. According to [31], τ can be determined through a hypothesis test with a significance level α. In other words, the probability that $\mathcal{L}(\mathbf{x}) \ge \tau^2$ is equal to α. Thus, $\mathcal{L}(\mathbf{x}) \ge \tau^2$ indicates the presence of bad measurements, with the probability of a false alarm being α.

2.1 Related Work

Many researchers have considered the problem of bad measurements detection and identification in power systems (e.g., [4,6,7,9,10,12,21–25,28,29,32–36]). Early power system researchers realized the existence of bad measurements and observed that a bad measurement usually led to large normalized measurement residual. After the presence of bad measurements is detected, they mark the measurement having the largest normalized residual as the suspect and remove it [9, 10, 24, 28, 29, 32–34]. For example, Schweppe et al. [29] filter one measurement having the largest normalized residual at each loop, and then rerun the same process on the reduced measurement set until the detection test is passed. Handschin et al. [9] proposed a grouped residual search strategy that can remove all suspected bad measurements at the same time.

It was found that the largest normalized residual criterion only worked well for independent, non-correlated bad measurements called *non-interacting bad measurements* [21, 23, 25]. In practice, there exist correlated bad measurements, which make the normalized residual of a good measurement the largest. Such bad measurements are called *interacting bad measurements*. The largest normalized residual method does not work satisfactorily in dealing with interacting bad measurements. To address this problem, Hypothesis Testing Identification (HTI) [21] and Combinatorial Optimization Identification (COI) [4, 12, 25] were developed. HTI selects a set of suspected bad measurements according to their normalized residuals, and then decide whether an individual suspected measurement is good or bad through hypothesis testing. COI uses the framework from the decision theory to identify multiple interacting bad measurements. For example, Asada et al. [4] proposed an intelligent bad data identification strategy based on tabu search to deal with multiple interacting bad measurements.

Recently, the focus in bad measurement processing is on the improvement of the robustness using phasor measurement units (PMUs) [6,7,35,36]. For example, Chen et al. [7]

used PMUs to transform the critical measurements into redundant measurements such that the bad measurements can be detected by the measurement residual testing and the system is still observable.

It seems that at least the approaches targeting at arbitrary, interacting bad measurements (e.g., [4,12,21,25]) can also defeat the malicious ones injected by attackers, since such malicious measurements are indeed arbitrary, interacting bad measurements. However, despite the variations in these approaches, all of them use the same method (i.e., $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$) to detect the existence of bad measurements. In this paper, we show that an attacker can systematically bypass this detection method, and thus all existing approaches.

3. FALSE DATA INJECTION ATTACKS

We assume that there are m meters that provide m measurements $z_1, ..., z_m$. We also assume that there are n state variables $x_1, ..., x_n$. The relationship between these m meter measurements and n state variables can be characterized by an $m \times n$ matrix $\mathbf{H}$, as discussed in Section 2. In general, the matrix $\mathbf{H}$ of a power system is a constant matrix determined by the topology and line impedances of the system. How the control center constructs $\mathbf{H}$ is illustrated in [23].

We assume that the attacker knows the matrix $\mathbf{H}$ of the target power system. For example, the attacker can obtain $\mathbf{H}$ by intruding into the control center of the target system. The attacker generates malicious measurements based on the matrix $\mathbf{H}$, and then injects the malicious measurements into the compromised meters to undermine the state estimation process. The injected malicious measurements can introduce arbitrary errors into the output of state estimation without being detected by the existing approaches.

As discussed earlier, we consider two realistic attack goals: *random false data injection attacks*, in which the attacker aims to find any attack vector as long as it can result in a wrong estimation of state variables, and *targeted false data injection attacks*, in which the attacker aims to find an attack vector that can inject a specific error into certain state variables.

We use the following two realistic attack scenarios to facilitate the discussion on how the attacker can construct attack vectors to bypass the current bad measurement detection scheme. Note, however, that the false data injection attacks are not constrained by these attack scenarios.

- **Scenario I – Limited Access to Meters:** The attacker is restricted to accessing some specific meters due to, for example, different physical protection of meters.
- **Scenario II – Limited Resources to Compromise Meters:** The attacker is limited in the resources required to compromise meters. For example, the attacker only has resources to compromise up to k meters (out of all the meters). Due to the limited resources, the attacker may also want to minimize the number of meters to be compromised.

In the following, we first show the basic principle of false data injection attacks. We then focus on the two attack scenarios and show how the attacker can construct attack vectors for both random and targeted false data injection attacks.

3.1 Basic Principle

Let $\mathbf{z_a}$ represent the vector of observed measurements that may contain malicious data. $\mathbf{z_a}$ can be represented as $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$, where $\mathbf{z} = (z_1, ..., z_m)^T$ is the vector of original measurements and $\mathbf{a} = (a_1, ..., a_m)^T$ is the malicious data added to the original measurements. We refer to $\mathbf{a}$ as an *attack vector*. The i-th element a_i being non-zero means that the attacker compromises the i-th meter, and then replaces its original measurement z_i with a phony measurement $z_i + a_i$.

The attacker can choose any non-zero arbitrary vector as the attack vector $\mathbf{a}$, and then construct the malicious measurements $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$. Let $\hat{\mathbf{x}}_{\mathbf{bad}}$ and $\hat{\mathbf{x}}$ denote the estimates of $\mathbf{x}$ using the malicious measurements $\mathbf{z_a}$ and the original measurements $\mathbf{z}$, respectively. $\hat{\mathbf{x}}_{\mathbf{bad}}$ can be represented as $\hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c}$ is a non-zero vector of length n. Note that $\mathbf{c}$ reflects the estimation error injected by the attacker.

As discussed in Section 2, the bad measurement detection algorithm computes the L_2-norm of the corresponding measurement residual to check whether there exist bad measurements or not. However, if the attacker can use $\mathbf{Hc}$ as the attack vector $\mathbf{a}$ (i.e., $\mathbf{a} = \mathbf{Hc}$), then the L_2-norm of the measurement residual of $\mathbf{z_a}$ is equal to that of $\mathbf{z}$, as shown in Theorem 1. In other words, if the attacker can choose $\mathbf{a}$ as a linear combination of the column vectors of $\mathbf{H}$, $\mathbf{z_a}$ can pass the detection as long as $\mathbf{z}$ can pass the detection.

THEOREM 1. *Suppose the original measurements* $\mathbf{z}$ *can pass the bad measurement detection. The malicious measurements* $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$ *can pass the bad measurement detection if* $\mathbf{a}$ *is a linear combination of the column vectors of* $\mathbf{H}$ *(i.e.,* $\mathbf{a} = \mathbf{Hc}$*).*

PROOF. Since $\mathbf{z}$ can pass the detection, we have $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \leq \tau$, where τ is the detection threshold. $\hat{\mathbf{x}}_{\mathbf{bad}}$, the vector of estimated state variables obtained from $\mathbf{z_a}$, can be represented as $\hat{\mathbf{x}} + \mathbf{c}$. If $\mathbf{a} = \mathbf{Hc}$, i.e., $\mathbf{a}$ is a linear combination of the column vectors $\mathbf{h}_1, ..., \mathbf{h}_n$ of $\mathbf{H}$, then the resulting L_2-norm of the measurement residual is

$$\begin{aligned} \|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{Hc})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \leq \tau. \end{aligned} \quad (5)$$

Thus, the L_2-norm of the measurement residual of $\mathbf{z_a}$ is less than the threshold τ. This means that $\mathbf{z_a}$ can also pass the bad measurement detection. □

In this paper, we refer to an attack in which the attack vector $\mathbf{a}$ equals $\mathbf{Hc}$, where $\mathbf{c}$ is an arbitrary non-zero vector, as a *false data injection attack*. By launching false data injection attacks, the attacker can manipulate the injected false data to bypass the bad measurement detection and also introduce arbitrary errors into the output of the state estimation (since each element of $\mathbf{c}$ could be an arbitrary number).

3.2 Scenario I – Limited Access to Meters

We assume that the attacker has access to k specific meters. Assume $\mathcal{I}_m = \{i_1, ..., i_k\}$ is the set of indices of those meters. In other words, the attacker can modify z_{i_j}, where $i_j \in \mathcal{I}_m$. To launch a false data injection attack without being detected, the attacker needs to find a non-zero attack vector $\mathbf{a} = (a_1, ..., a_m)^T$ such that $a_i = 0$ for $i \notin \mathcal{I}_m$ and

$\mathbf{a}$ is a linear combination of the column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{Hc}$).

3.2.1 Random False Data Injection Attack

As discussed earlier, the non-zero attack vector $\mathbf{a}$ satisfies the condition $\mathbf{a} = (a_1, ..., a_m)^T = \mathbf{Hc}$ with $a_i = 0$ for $i \notin \mathcal{I}_m$. In a random false data injection attack, the vector $\mathbf{c}$ (i.e., the errors introduced to the state variables) can be any value.

The attacker can find an attack vector $\mathbf{a}$ as follows. First, the attacker can compute an equivalent form of the relation $\mathbf{a} = \mathbf{Hc}$ by eliminating $\mathbf{c}$. Let $\mathbf{P} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T}$, and $\mathbf{B} = \mathbf{P} - \mathbf{I}$. It is easy to see that $\mathbf{PH} = \mathbf{H}$. The attacker can simply multiply $\mathbf{P}$ to both sides of the relation $\mathbf{a} = \mathbf{Hc}$ to obtain a sequence of equivalent forms, as shown below:

$$\begin{aligned} \mathbf{a} = \mathbf{Hc} &\Leftrightarrow \mathbf{Pa} = \mathbf{PHc} \Leftrightarrow \mathbf{Pa} = \mathbf{Hc} \Leftrightarrow \mathbf{Pa} = \mathbf{a} \\ &\Leftrightarrow \mathbf{Pa} - \mathbf{a} = \mathbf{0} \Leftrightarrow (\mathbf{P} - \mathbf{I})\mathbf{a} = \mathbf{0} \\ &\Leftrightarrow \mathbf{Ba} = \mathbf{0}. \end{aligned} \quad (6)$$

This means that a vector $\mathbf{a}$ satisfies the relation $\mathbf{a} = \mathbf{Hc}$ if and only if it satisfies the relation $\mathbf{Ba} = \mathbf{0}$. So the attacker needs to find a non-zero attack vector $\mathbf{a}$ such that $\mathbf{Ba} = \mathbf{0}$ and $a_i = 0$ for $i \notin \mathcal{I}_m$.

There are many known methods to obtain attack vectors from the above equation. Here is a simple one: Represent $\mathbf{a}$ as $\mathbf{a} = (0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_2}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T$, where $a_{i_1}, a_{i_2}, ..., a_{i_k}$ are the unknown variables. Let $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_m)$, where $\mathbf{b}_i$ $(1 \le i \le m)$ is the i-th column vector of $\mathbf{B}$. Thus, $\mathbf{Ba} = 0 \Leftrightarrow (..., \mathbf{b}_{i_1}, ..., \mathbf{b}_{i_2}, ..., \mathbf{b}_{i_k}, ...)(0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_2}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T = \mathbf{0}$. Let the $m \times k$ matrix $\mathbf{B'} = (\mathbf{b}_{i_1}, ..., \mathbf{b}_{i_k})$ and the length k vector $\mathbf{a'} = (a_{i_1}, ..., a_{i_k})^T$. We have

$$\mathbf{Ba} = \mathbf{0} \Leftrightarrow \mathbf{B'a'} = \mathbf{0}.$$

If the rank of $\mathbf{B'}$ is less than k, $\mathbf{B'}$ is a rank deficient matrix, and there exist infinite number of non-zero solutions $\mathbf{a'}$ that satisfy the relation $\mathbf{B'a'} = \mathbf{0}$ [20]. According to [20], the solution is $\mathbf{a'} = (\mathbf{I} - \mathbf{B'^{-}B})\mathbf{d}$, where $\mathbf{B'^{-}}$ is the Matrix 1-inverse of $\mathbf{B'}$ and $\mathbf{d}$ is an arbitrary non-zero vector of length k.

If the rank of $\mathbf{B'}$ is equal to k, then $\mathbf{B'}$ is not a rank deficient matrix and the relation $\mathbf{B'a'} = \mathbf{0}$ has a unique solution $\mathbf{a'} = \mathbf{0}$ [20]. This means that no error can be injected into the state estimation, and the attacker vector does not exist.

Existence of Attack Vectors: It is possible that the attack vector does not exist if k is too small. However, if $k \ge m - n + 1$, the attack vector always exists, as shown in Theorem 2. Moreover, as long as the attacker can compromise $m - n + 1$ or more meters, she can always construct an attack vector to bypass the detection.

THEOREM 2. *If the attacker can compromise k specific meters, where $k \ge m-n+1$, there always exist attack vectors $\mathbf{a} = \mathbf{Hc}$ such that $\mathbf{a} \neq \mathbf{0}$ and $a_i = 0$ for $i \notin \mathcal{I}_m$.*

PROOF. According to Equation (6), $\mathbf{a} = \mathbf{Hc} \Leftrightarrow \mathbf{Ba} = \mathbf{0}$, where $\mathbf{B} = \mathbf{P} - \mathbf{I} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T} - \mathbf{I}$. $\mathbf{H}$ should be an $m \times n$ full rank matrix to allow the estimation of $\mathbf{x}$ from $\mathbf{z}$ [31]. Without loss of generality, we further assume $m \ge n$. Thus, $\text{rank}(\mathbf{H}) = n$. Since $\mathbf{P} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T}$, $\mathbf{P}$ is a projection matrix of $\mathbf{H}$. Thus, $\text{rank}(\mathbf{P}) = \text{rank}(\mathbf{H}) = n$, and n eigenvalues of $\mathbf{P}$ are 1's and the remaining $m - n$ eigenvalues of $\mathbf{P}$ are 0's [20]. Obviously, for $\mathbf{B} = \mathbf{P} - \mathbf{I}$, $m - n$ eigenvalues of $\mathbf{B}$ are 1's and n eigenvalues of $\mathbf{B}$ are 0's. Therefore, $\text{rank}(\mathbf{B}) = m - n$. The matrix $\mathbf{B'}$ is a $m \times k$ matrix. So $\text{rank}(\mathbf{B'}) \le m - n$. Further considering $k \ge m - n + 1$, we have $\text{rank}(\mathbf{B'}) < k$. Thus, $\mathbf{B'}$ is rank deficient matrix and there exist infinite number of non-zero solutions for $\mathbf{a'}$ that satisfy the relation $\mathbf{B'a'} = \mathbf{0}$. This means there exist many non-zero attack vectors $\mathbf{a}$ in which $a_i = 0$ for $i \notin \mathcal{I}_m$. □

Lightweight Construction of Attack Vectors: When $k \ge m - n + 1$, the attacker does not need to compute the matrices $\mathbf{B}$ and $\mathbf{B'}$ to solve $\mathbf{B'}a' = 0$. Instead, the attacker can perform column transformations on $\mathbf{H}$ directly such that some column vectors in the resulting matrix become linear combinations of column vectors in $\mathbf{H}$ and at the same time, the elements corresponding to the meters not controlled by the attacker are eliminated (i.e., $a_i = 0$ for $i \notin \mathcal{I}_m$). Each of such vectors can be used as an attacker vector.

Specifically, let $\bar{\mathcal{I}}_m = \{j | 1 \le j \le m, j \notin \mathcal{I}_m\}$, and $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_n)$, where $\mathbf{h}_i = (h_{1,i}, ..., h_{m,i})^T$ for $1 \le i \le n$. For a random $j \in \bar{\mathcal{I}}_m$ (i.e., the meter not under the attacker's control), the attacker first scans $\mathbf{H}$ to look for a column vector whose j-th element is not zero. If the attacker can find such a vector, the attacker swaps it with the first column vector $\mathbf{h}_1$. Then, the attacker can construct an $m \times (n-1)$ matrix $\mathbf{H^1} = (\mathbf{h^1}_1, ..., \mathbf{h^1}_{n-1})$ by performing column transformations on $\mathbf{H}$ (to zero out the j-th element in all column vectors):

$$\mathbf{h^1}_i = \begin{cases} \mathbf{h}_1 - \frac{h_{j,1}}{h_{j,i+1}}\mathbf{h}_{i+1}, & \text{if } h_{j,i+1} \neq 0, 1 \le i \le n-1 \\ \mathbf{h}_{i+1}, & \text{if } h_{j,i+1} = 0, 1 \le i \le n-1 \end{cases} \quad (7)$$

If the j-th element is zero for all the column vectors of $\mathbf{H}$, then $\mathbf{h^1}_i = \mathbf{h}_i$ for $1 \le i \le n-1$. As a result, the j-th row of $\mathbf{H^1}$ are all zeros. The attacker repeats this process to the reduced matrix $\mathbf{H^1}$ and the reduced matrices thereafter using a different element in $\bar{\mathcal{I}}_m$, until all elements in $\bar{\mathcal{I}}_m$ are exhausted. Finally, the attacker can get a matrix having at least one column vector, since $m - k \le n - 1$. Obviously, the column vectors of the final matrix are linear combinations of the column vectors of $\mathbf{H}$, and the $m - k$ rows with index $j \in \bar{\mathcal{I}}_m$ of this matrix consist of all 0's. Any column vector can be used as an attacker vector.

3.2.2 Targeted False Data Injection Attack

In a targeted false data injection attack, the attacker intends to inject specific errors into the estimation of certain chosen state variables. This attack can be represented mathematically as follows. Let $\mathcal{I}_v = \{i_1, ..., i_r\}$, where $r < n$, denote the set of indexes of the r target state variables chosen by the attacker. (That is, the attacker has chosen $x_{i_1}, x_{i_2}, ..., x_{i_r}$ to compromise.) In this attack, the attacker intends to construct an attack vector $\mathbf{a}$ such that the resulting estimate $\hat{\mathbf{x}}_{\mathbf{bad}} = \hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c} = (c_1, c_2, ..., c_n)^T$ and c_i for $i \in \mathcal{I}_v$ is the specific error that the attacker has chosen to inject to $\hat{x}_i$. That is, the attacker wants to replace $\hat{x}_{i_1}, ...,$ and $\hat{x}_{i_r}$ with $\hat{x}_{i_1} + c_{i_1}, ...,$ and $\hat{x}_{i_r} + c_{i_r}$, respectively.

We consider two cases for the targeted false data injection attack: A *constrained* and an *unconstrained* case. In the constrained case, the attacker wants to launch a targeted false data injection attack that only changes the target state variables but does not pollute the other state variables. The constrained case represents the situations where the control center (software or operator) may know ways to verify the estimates of the other state variables. In the unconstrained

case, the attacker has no concerns on the impact on the other state variables when attacking the chosen ones.

Constrained Case: The construction of an attack vector $\mathbf{a}$ becomes rather simple in the constrained case. Consider the relation $\mathbf{a} = \mathbf{Hc}$. As discussed earlier, the attack vector $\mathbf{a}$ must satisfy the condition that $a_i = 0$ where $i \notin \mathcal{I}_m$. Note that every element c_i in $\mathbf{c}$ is fixed, which is either the chosen value when $i \in \mathcal{I}_v$ or 0 when $i \notin \mathcal{I}_v$. Thus, the attacker can substitute $\mathbf{c}$ back into the relation $\mathbf{a} = \mathbf{Hc}$, and check if $a_i = 0$ for $\forall i \notin \mathcal{I}_m$. If yes, the attacker succeeds in constructing the (only) attack vector $\mathbf{a}$. Otherwise, the attack is impossible.

Unconstrained Case: In this case, only the elements c_i of $\mathbf{c}$ for $i \in \mathcal{I}_v$ are fixed; the other elements c_j for $j \notin \mathcal{I}_v$ can be any values. The attacker can first transform $\mathbf{a} = \mathbf{Hc}$ into an equivalent form without having $\mathbf{c}$, and then solve $\mathbf{a}$ from the equivalent form.

Note that $\mathbf{a} = \mathbf{Hc} = \sum_{i \notin \mathcal{I}_v} \mathbf{h}_i c_i + \sum_{j \in \mathcal{I}_v} \mathbf{h}_j c_j$. Let $\mathbf{H_s} = (\mathbf{h}_{j_1}, ..., \mathbf{h}_{j_{n-r}})$ and $\mathbf{c_s} = (c_{j_1}, ..., c_{j_{n-r}})^T$, where $j_i \notin \mathcal{I}_v$ for $1 \le i \le n-r$. Let $\mathbf{b} = \sum_{j \in \mathcal{I}_v} \mathbf{h}_j c_j$, $\mathbf{P_s} = \mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T}$, $\mathbf{B_s} = \mathbf{P_s} - \mathbf{I}$, and $\mathbf{y} = \mathbf{B_s b}$. Thus, the relation $\mathbf{a} = \mathbf{Hc}$ can be transformed into the following equivalent form:

$$\begin{aligned}
\mathbf{a} = \mathbf{Hc} \quad &\Leftrightarrow \quad \mathbf{a} = \sum_{i \notin \mathcal{I}_v} \mathbf{h}_i c_i + \sum_{j \in \mathcal{I}_v} \mathbf{h}_j c_j = \mathbf{H_s c_s} + \mathbf{b} \\
&\Leftrightarrow \quad \mathbf{P_s a} = \mathbf{P_s H_s c_s} + \mathbf{P_s b} \\
&\Leftrightarrow \quad \mathbf{P_s a} = \mathbf{H_s c_s} + \mathbf{P_s b} \\
&\Leftrightarrow \quad \mathbf{P_s a} = \mathbf{a} - \mathbf{b} + \mathbf{P_s b} \\
&\Leftrightarrow \quad (\mathbf{P_s} - \mathbf{I})\mathbf{a} = (\mathbf{P_s} - \mathbf{I})\mathbf{b} \\
&\Leftrightarrow \quad \mathbf{B_s a} = \mathbf{B_s b} \Leftrightarrow \mathbf{B_s a} = \mathbf{y}. \qquad (8)
\end{aligned}$$

This implies that $\mathbf{a}$ satisfies the relation $\mathbf{a} = \mathbf{Hc}$ if and only if $\mathbf{a}$ satisfies the relation $\mathbf{B_s a} = \mathbf{y}$. (It is easy to see that $\mathbf{B_s}$ is an $m \times m$ matrix.) Thus, the attacker needs to find an attack vector $\mathbf{a}$ such that $\mathbf{B_s a} = \mathbf{y}$ where $\mathbf{a} = (a_1, a_2, ..., a_m)^T$ and $a_i = 0$ for $i \notin \mathcal{I}_m$.

There are k unknown elements in $\mathbf{a}$ at positions $i_1, ..., i_k$, where $i_1, ..., i_k \in \mathcal{I}_m$. Thus, the vector $\mathbf{a}$ can be written as $\mathbf{a} = (0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T$, where a_{i_j}'s are unknown elements to be solved. Suppose $\mathbf{B_s} = (\mathbf{b}_{\mathbf{s}_1}, ..., \mathbf{b}_{\mathbf{s}_m})$, where $\mathbf{b}_{\mathbf{s}_i}$ $(1 \le i \le m)$ is the i-th column vector of $\mathbf{B_s}$. We follow the same reasoning as in Section 3.2.1 to denote $\mathbf{B'_s} = (\mathbf{b}_{\mathbf{s}_{i_1}}, ..., \mathbf{b}_{\mathbf{s}_{i_k}})$ and $\mathbf{a'} = (a_{i_1}, ..., a_{i_k})^T$. Then we have

$$\mathbf{B'_s a'} = \mathbf{y} \Leftrightarrow \mathbf{B_s a} = \mathbf{y} \Leftrightarrow \mathbf{a} = \mathbf{Hc}.$$

Thus, to construct an attack vector, the attacker needs to check if the rank of $\mathbf{B}'_s$ is the same as the rank of the augmented matrix $(\mathbf{B}'_s|\mathbf{y})$. If yes, the relation $\mathbf{B'_s a'} = \mathbf{y}$ is a consistent equation. According to [20], there exist infinite number of solutions $\mathbf{a'} = \mathbf{B'_s}^{-}\mathbf{y} + (\mathbf{I} - \mathbf{B'_s}^{-}\mathbf{B_s})\mathbf{d}$ that satisfy the relation $\mathbf{B'_s a'} = \mathbf{y}$, where $\mathbf{B}'^{-}_s$ is the Matrix 1-inverse of $\mathbf{B}'_s$ and $\mathbf{d}$ is an arbitrary non-zero vector of length k. The attacker can construct an attack vector $\mathbf{a}$ from any $\mathbf{a'} \neq \mathbf{0}$. If the rank of $\mathbf{B}'_s$ is not the same as the rank of the augmented matrix $(\mathbf{B'_s}|\mathbf{y})$, then the relation $\mathbf{B'_s a'} = \mathbf{y}$ is not a consistent equation, and thus has no solution. This means that the attacker cannot construct an attack vector to inject the specific errors into the chosen state variables.

3.3 Scenario II – Limited Resources to Compromise Meters

In Scenario II, we assume the attacker has limited resources to compromise up to k meters. Unlike Scenario I, there is no restriction on what meters can be chosen. For the sake of presentation, we call a length-m vector a *k-sparse vector* if it has at most k non-zero elements. Thus, the attacker needs to find a k-sparse, non-zero attack vector $\mathbf{a}$ that satisfies the relation $\mathbf{a} = \mathbf{Hc}$. As in Scenario I, we consider both random and targeted false data injection attacks in Scenario II.

3.3.1 Random False Data Injection Attack

With the resources to compromise up to k meters, the attacker may use a brute-force approach to construct an attack vector. That is, the attacker may try all possible $\mathbf{a}$'s consisting of k unknown elements and $m - k$ zero elements. For each candidate $\mathbf{a}$, the attacker may check if there exists a non-zero solution of $\mathbf{a}$ such that $\mathbf{Ba} = \mathbf{0}$ using the same method as discussed in Section 3.2.1. If yes, the attacker succeeds in constructing an attack vector. Otherwise, the attack vector does not exist. However, the brute-force approach could be time consuming. In the worst case, the attacker needs to examine $\binom{m}{k}$ candidate attack vectors.

To improve the time efficiency, the attacker may take advantage of the following observation. Since a successful attack vector is a linear combination of the column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{Hc}$), the attacker can perform column transformations to $\mathbf{H}$ to reduce the non-zero elements in the transformed column vectors. As this process continues, more column vectors in the transformed $\mathbf{H}$ will have fewer non-zero elements. The column vectors with no more than k non-zero elements can be used as attack vectors. In particular, when the matrix $\mathbf{H}$ is a sparse matrix (which is usually the case in real power systems), it does not take many column transformations to construct a desirable attack vector.

A Heuristic Approach: We give a heuristic approach to exploit this observation: The attacker can initialize a size n queue with the n column vectors of $\mathbf{H}$. The attacker then repeats the following process: Take the first column vector $\mathbf{t}$ out from the queue. If $\mathbf{t}$ is a k-sparse vector, the algorithm returns and $\mathbf{t}$ can be used as the attack vector. If not, for each column vector $\mathbf{s}$ in the queue, the attacker checks if linearly combining $\mathbf{t}$ and $\mathbf{s}$ can result in a column vector with less zero elements than $\mathbf{t}$. If so, the attacker appends the resulting vector into the queue. The attacker repeats this process until a k-sparse vector is found or the set is empty. It is easy to see that a k-sparse vector constructed in this way must be a linear combination of some column vectors of $\mathbf{H}$, and can serve as an attack vector.

The heuristic approach could be quite slow for a general $\mathbf{H}$. However, it works pretty efficiently for a sparse matrix $\mathbf{H}$, which is usually the case for real-world power systems. For example, in our simulation, when $k = 12$ in the IEEE 300-bus test system, it takes the heuristic approach about 16.63 seconds on a regular PC to find an attack vector after computing 596 linear combinations of column vectors.

The heuristic approach does not guarantee the construction of an attack vector even if it exists, nor does it guarantee the construction of an attack vector that has the minimum number of non-zero elements. Nevertheless, it runs pretty quickly when it can construct an attack vector, and thus could still be a useful tool for the attacker.

Ideally, in order to reduce the attack costs, the attacker would like to compromise as few meters as possible. In other

words, the attacker wants to find the optimal attack vector $\mathbf{a}$ with the minimum number of non-zero elements. The attacker may use the brute-force approach discussed at the beginning of Section 3.3.1 with k being 1 initially, and gradually increase k until an attack vector is found. Apparently, such an attack vector gives the optimal solution with the minimum number of compromised meters. There are possibilities to improve such a brute-force approach, for example, using a binary search in identifying the minimum k.

3.3.2 Targeted False Data Injection Attack

We follow the notation used in Scenario I to describe the targeted false data injection attack. Let $\mathcal{I}_v = \{i_1, ..., i_r\}$, where $r < n$, denote the set of indexes of the r target state variables chosen by the attacker. In this attack, the attacker intends to construct an attack vector $\mathbf{a}$ to replace $\hat{x}_{i_1}, ...,$ and $\hat{x}_{i_r}$ with $\hat{x}_{i_1} + c_{i_1}, ...,$ and $\hat{x}_{i_r} + c_{i_r}$, respectively, where $c_{i_1}, ..., c_{i_r}$ are the specific errors to be injected.

Similar to Scenario I, we consider both constrained and unconstrained cases.

Constrained Case: As discussed earlier, in the constrained case, the attacker intends to only change the estimation of the chosen target state variables, but does not modify the others. Thus, all elements of $\mathbf{c}$ are fixed. So the attacker can substitute $\mathbf{c}$ into the relation $\mathbf{a} = \mathbf{Hc}$. If the resulting $\mathbf{a}$ is a k-sparse vector, the attacker succeeds in constructing the attack vector. Otherwise, the attacker fails. The attack vector derived in the constrained case is the only possible attack vector; there is no way to further reduce the number of compromised meters.

Unconstrained Case: In the unconstrained case, only the elements c_i of $\mathbf{c}$ for $i \in \mathcal{I}_v$ are fixed; the other c_j for $j \notin \mathcal{I}_v$ can be any values. According to Equation (8), $\mathbf{a} = \mathbf{Hc} \Leftrightarrow \mathbf{B_s a} = \mathbf{y}$. (Note that the derivation of Equation (8) does not assume any specific compromised meters. Thus, Equation (8) also holds in the unconstrained case in Scenario II.)

To construct an attack vector, the attacker needs to find a k-sparse attack vector $\mathbf{a}$ that satisfies the relation $\mathbf{B_s a} = \mathbf{y}$. A closer look at this problem reveals that it is the *Minimum Weight Solution for Linear Equations problem* [11], which is an NP-Complete problem: Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, compute a vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ such that $\mathbf{x}$ has at most k non-zero elements.

Several efficient heuristic algorithms have been developed to deal with the above problems, for example, the Matching Pursuit algorithm [19, 26, 27], the Basis Pursuit algorithm [8, 13], and the Gradient Pursuit algorithm [5]. The attacker can use these algorithms to find a near optimal attack vector. In our simulation, we choose to use the Matching Pursuit algorithm, since it is the most popular algorithm for computing the sparse signal representations and has exponential rate of convergence [15].

The attacker may want to minimize the number of meters to be compromised, i.e., to find an attack vector $\mathbf{a}$ with the minimum number of non-zero elements that satisfies $\mathbf{a} = \mathbf{Hc}$ such that the chosen elements in $\mathbf{c}$ have the specific values. This problem is in fact the MIN RVLS$^=$ problem [3]: Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, compute a vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ such that $\mathbf{x}$ has as few non-zero elements as possible. Matching Pursuit Algorithm can again be used to find an attack vector, since this problem is the optimization version of the minimum weight solution for linear equations problem discussed earlier.

3.4 Requirements and Practical Implications

We would like to point out that the false data injection attacks do pose strong requirements for the attackers. In particular, it requires that the attackers know the configuration of the target power system. Such information is usually kept secret by power companies at control centers or other places with physical security measures. Thus, it is non-trivial for the attackers to obtain the system configuration information to launch these attacks. Nevertheless, it would be definitely wrong to assume that the attackers cannot access such information at all. For example, an attacker can obtain the configuration of the North American power grid from the POWERmap mapping system, which contains information about every power plant, major substation, and 115-765 kV power line of the North American power grid [17]. An attacker may also take advantage of publicly available sources such as satellite photos or through social engineering approaches to obtain the desired information.

Another requirement for the attackers is the manipulation of the meter measurements. The attackers may physically tamper the meters, or manipulate the meter measurements before they are used for state estimation in the control center. Again, due the existing protection in the power grid, this is non-trivial. However, assuming that this is impossible will definitely give us a false sense of security and will pave ways for catastrophes in the future.

Despite the difficulty for launching false data injection attacks, it is critical for power engineers and security people to be aware of this threat. Existing state estimation and the follow-up processes assume a near-perfect detection of large bad measurements. However, our work in this paper indicates that an attacker can systematically bypass detection. This discrepancy may be amplified in the later processes following state estimation, leading to catastrophic impacts. Additional research is necessary to clarify the implication of such attacks.

4. EXPERIMENTAL RESULTS

In this section, we validate the false data injection attacks through experiments using IEEE test systems, including the IEEE 9-bus, 14-bus, 30-bus, 118-bus, and 300-bus systems. We are primarily interested in the feasibility of constructing attack vectors in various situations as well as the efforts required for a successful attack vector construction.

In our experiments, we simulate attacks against state estimation using the DC power flow model. We extract the configuration of the IEEE test systems (particularly matrix $\mathbf{H}$) from MATPOWER, a MATLAB package for solving power flow problems [37][2]. We perform our experiments based on matrix $\mathbf{H}$ and meter measurements obtained from MATPOWER. For each test system, the state variables are voltage angles of all buses, and the meter measurements are real power injections of all buses and real power flows of all branches. We simulate the behavior of compromising the i-th meter by adding an offset (chosen by the attack) to the i-th measurement.

Additional information of these IEEE test systems is given

[2]In MATPOWER, the shift injection vector is set to $\mathbf{0}$ for state estimation using the DC power flow model.

in Appendix A. All the experiments are simulated in MATLAB 7.4.0 on a DELL PC running Windows XP, which has a 3.0 GHz Pentium 4 processor and 1 GB memory.

4.1 Results of Scenario I

As mentioned earlier, in Scenario I, the attacker is limited to accessing k specific meters. In other words, the attacker can only modify the measurements of these k meters. Our evaluation objective in this scenario is mainly two-fold. First, we would like to see how likely the attacker can use these k meters to achieve his/her attack goal. Second, we want to see the computational efforts required for finding an attack vector. In our evaluation, we consider (1) random false data injection attacks, (2) targeted false data injection attacks in the unconstrained case, and (3) targeted false data injection attacks in the constrained case.

Based on our evaluation objective, we use two evaluation metrics: the *probability* that the attacker can successfully construct an attack vector given the k specific meters, and the *execution time* required to either construct an attack vector or conclude that the attack is infeasible.

We perform the experiments as follows. For random false data injection attacks, we let the parameter k range from 1 to the maximum number of meters in each test system. (For example, k ranges from 1 to 490 in the IEEE 118-bus system.) For each k, we randomly choose k specific meters to attempt an attack vector construction. We repeat this process 100 times for both IEEE 118-bus and 300-bus systems and 1,000 times for the other systems[3], and estimate the *success probability* p_k as $p_k = \frac{\#\ successful\ trials}{\#\ trials}$.

Let R_k denote the percentage of the specific meters under attacker's control (i.e., $\frac{k}{total\ number\ of\ meters}$). Figure 2 shows the relationship between p_k and R_k for random false data injection attacks. We can see that p_k increases sharply as R_k is larger than a certain value in all systems. For example, p_k of the IEEE 300-bus system increases quickly when R_k exceeds 20%. Moreover, the attacker can generate the attack vector with the probability close to 1 when R_k is large enough. For example, p_k is almost 1 when R_k is greater than 60% and 40% in the IEEE 118-bus and 300-bus systems, respectively. Finally, larger systems have higher p_k than smaller systems for the same R_k. For example, p_k is about 0.6 for IEEE 300-bus system and 0.1 for IEEE 118-bus system when the attacker can compromise 30% of the meters in both systems.

For targeted false data injection attacks in the unconstrained case, we also let the parameter k range from 1 to the maximum number of meters in each test system, and perform the following experiments for each k. We randomly pick 10 target state variables for each test system (8 for the IEEE 9-bus system, since it only has 8 state variables). For each target state variable, we perform multiple trials (1,000 trials for the IEEE 9-bus, 14-bus, and 30-bus systems, 100 trials for the IEEE 118-bus system, and 20 trials for the IEEE 300-bus system)[4]. In each trial, we randomly

[3]It takes significantly more time to exhaustively examine the IEEE 118-bus and 300-bus systems with all possible k's. We reduce the number of trials for these systems so that the simulation can finish within a reasonable amount time.

[4]In this case, it take even more time than random false data injection attacks to exhaustively examine the IEEE 118-bus and 300-bus systems with all possible k's. Thus, we reduce the number of trials for these two systems so that the simulation can finish within a reasonable amount time.

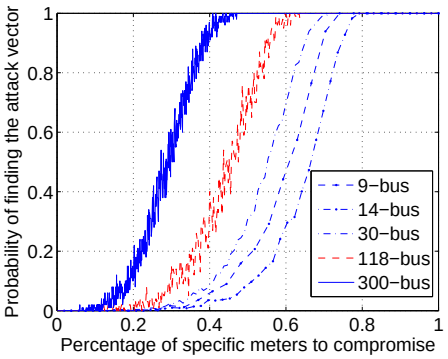

Figure 2: Probability of finding an attack vector for random false data injection attacks

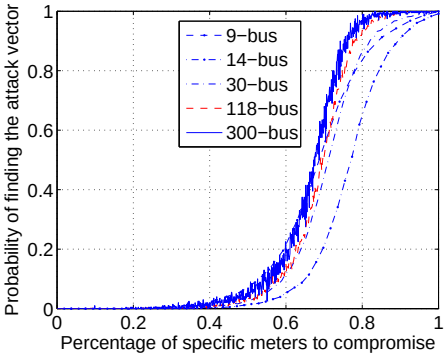

Figure 3: Probability of finding an attack vector for compromising a single state variable in targeted false data injection attacks (unconstrained case)

choose k meters and test if an attack vector that injects false data into this target variable can be generated. If yes, we mark the experiment as successful. After these trials, we can compute the success probability $p_{k,v}$ for this particular state variable v as $p_{k,v} = \frac{\#\ successful\ trials}{\#\ trials}$. Finally, we compute the overall success probability p_k as the average of $p_{k,v}$'s for all the chosen state variables.

Figure 3 shows the relationship between p_k and R_k for targeted false data injection attacks in the unconstrained case. We observe the same trend in this figure as in Figure 2, though the probability in this case is in general lower than that in Figure 2. For example, p_k increases sharply as R_k is larger than 60% for both the IEEE 118-bus and 300-bus systems. Moreover, for both systems, the probability that the attacker can successfully generate the attack vector is larger than 0.6 when R_k is larger than 70%. For targeted false data injection attacks, larger systems also tend to have higher p_k than smaller systems for the same R_k.

It is critical to note that Figures 2 and 3 represent the success probabilities of "blind trials". In this case, an attacker needs to compromise 30–70% of the meters to get a reasonable probability to construct an attacker vector. However, as shown later in Section 4.2.1, when an attacker targets the "weakest link" of a power system, she only needs to compromise 4 meters in these test systems.

The targeted false data injection attack in the constrained

Table 1: Timing results in Scenario I (ms)

Test system	Random attack	Targeted attack (unconstrained)
IEEE 9-bus	0.17–2.4	0.21–2.2
IEEE 14-bus	0.16–5.6	0.26–11.3
IEEE 30-bus	0.35–14.9	0.24–31.4
IEEE 118-bus	0.34–867.9	0.42–1,874.5
IEEE 300-bus	0.55–8,549.6	0.73–18,510

case is the most challenging one for the attacker. Due to the constraints on the specific meters, the targeted state variables, and the necessity of no impact on the remaining state variables, the probability of constructing a successful attack vector is in fact very small, though still possible. We perform experiments for this case slightly differently. We randomly pick 6 sets of meters for the IEEE 118-bus and 300-bus systems. In each set, there are 350 meters and 700 meters for the IEEE 118-bus and 300-bus systems, respectively. We then check the number of individual target state variables that can be affected by each set of meters without affecting the estimation of the remaining state variables. The results show that the attacker can affect 8–11 and 13–16 individual state variables in the IEEE 118-bus and 300-bus systems, respectively. Thus, though the targeted false data injection attack in the constrained case is hard, it is still possible to modify some target state variables.

In Scenario I, all attacks can be performed fairly quickly. When the attack is feasible, it takes again little time to actually construct an attack vector. Table 1 shows the execution time required by the random false data injection attack and the targeted false data injection attack in the unconstrained case. For example, the time needed for the random false data injection attack to either construct an attack vector or conclude the infeasibility of the attack ranges from 0.34ms to 867.9 ms for the 118-bus system. The time required for the targeted false data injection attack in the constrained case is very small, since the computational task is just the multiplication of a matrix and a column vector. For example, the time required for the IEEE 300-bus system ranges from 1.2ms to 11ms. We do not give the specific numbers in this paper.

4.2 Results of Scenario II

In Scenario II, the attacker has limited resources to compromise up to k meters. Compared with Scenario I, the restriction on the attacker is relaxed in the sense that any k meters can be used for the attack. Similar to Scenario I, we would also like to see how likely the attacker can use the limited resources to achieve his/her attack goal, and at the same time, examine the computation required for attacks. We use two evaluation metrics in our experiments: (1) number of meters to compromise in order to construct an attack vector, and (2) execution time required for constructing an attack vector.

Due to the flexibility for the attacker to choose different meters to compromise in Scenario II, the evaluation of Scenario II generally requires more experiments to obtain the evaluation results. In the following, we examine (1) random false data injection attacks, (2) targeted false data injection attacks in the constrained case, and (3) targeted false data injection attacks in the unconstrained case, respectively.

4.2.1 Results of Random False Data Injection Attacks

Random false data injection attacks are the easiest one among the three types of attacks under evaluation, mainly due to the least constraints that the attacker has to follow. We perform a set of experiments to construct attack vectors for random false data injection attacks in the IEEE test systems. We assume the attacker wants to minimize the attack cost by compromising as few meters as possible. This means the attacker needs to find the attack vector having the minimum number of non-zero elements. The brute-force approach is too expensive to use for finding such an attack vector due to its high time complexity. For example, it needs to examine about 2^{27} combinations for the IEEE 9-bus test system. Thus, in our experiment, we use the heuristic algorithm discussed in Section 3.3.1 to find an attack vector that has near minimum number of non-zero elements for each system.

Table 2: Random false data injection attacks

Test system	# meters to compromise
IEEE 9-bus	4
IEEE 14-bus	4
IEEE 30-bus	4
IEEE 118-bus	4
IEEE 300-bus	4

Table 2 shows the results. In all test systems, the number of meters that need to be compromised is surprisingly small. For all test systems, the attacker can construct an attack vector for random false data injection attacks by only compromising *4 meters*. We look into the experimental data, and find that this is mainly due to the fact that the $\mathbf{H}$ matrices of all these IEEE test systems are sparse. For example, the $\mathbf{H}$ matrix of the IEEE 300-bus system is a 1,122×300 matrix, but most of the entries are 0's. In particular, the sparsest column in $\mathbf{H}$ only has 4 non-zero elements. This column is eventually selected by the algorithm as the attack vector. Note that power systems with sparse $\mathbf{H}$ matrices are not rare cases. In practice, components in a power system that are not physically adjacent to each other are usually not connected. As a result, the $\mathbf{H}$ matrices of the power systems are often sparse.

4.2.2 Results of Targeted False Data Injection Attacks in Constrained Case

Similar to Scenario I, targeted false data injection attacks in the constrained case are the most challenging one among all types of attacks due to all the constraints the attacker has to follow in attack vector construction. In the constrained case, the attacker aims to change specific state variables to specific values and keep the remaining state variables as they are.

In our experiments, we randomly choose l ($1 \leq l \leq 10$) target state variables and generate malicious data for each of them. The malicious values are set to be 100 times larger than the real estimates of the state variables. We then examine how many meters need to be compromised in order to inject the malicious data (without changing the other non-target state variables). For each value of l, we perform the above experiment 1,000 times to examine the distribution of the number of meters that need to be compromised.

Figure 4 shows the result of the IEEE 300-bus system. We use box plot[5] to show the relationship between the number

[5]In a box plot [1], each box describes a group of data through their five summaries: minimum, first quartile, median, third

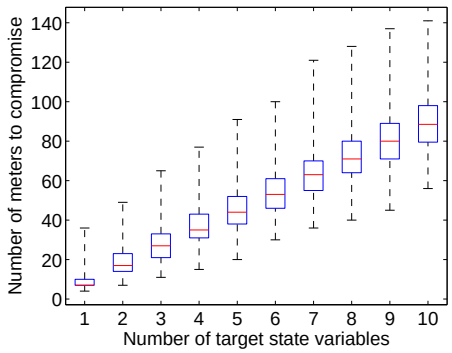

Figure 4: Constrained case: Number of meters to compromise to inject false data into l state variables in the IEEE 300-bus system

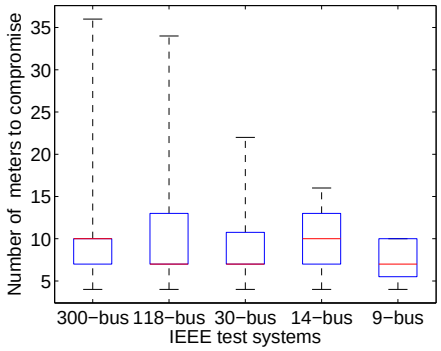

Figure 5: Constrained case: Number of meters to compromise to inject false data into one state variable

of target state variables and the number of meters to compromise. In the worst case, to inject malicious data into as many as 10 state variables, the attacker needs to compromise 55–140 meters in the IEEE 300-bus system. Given 1,122 meters in the IEEE 300-bus system, the attacker only needs to compromise a small fraction of the meters to launch targeted false data injection attacks even in the constrained case.

We also exhaustively examine a special situation of targeted false data injection attacks in the constrained case. Specifically, for each state variable, we examine the number of meters that need be compromised if the attacker aims at this variable. Figure 5 shows the results. We can see that the attacker can inject malicious data into any single state variable using less than 35 meters for the IEEE 118-bus system and less than 40 meters for the IEEE 300-bus system. For all the systems, none of the median values is greater than 10. This means that the attacker can affect most of the state variables by using at most 10 compromised meters.

In the constrained case, since $\mathbf{c}$ is fixed, the attack vectors can be directly computed. Thus, the execution time in all the experiments is very short. For example, it costs only 1.2 ms on the test computer to generate an attack vector that injects false data into 10 state variables in the IEEE 300-bus system.

quartile, and maximum. They are represented as horizontal lines at the very bottom, at the lower end, inside the box, at the upper end, and at the very top of the box, respectively.

4.2.3 *Results of Targeted False Data Injection Attacks in Unconstrained Case*

In the unconstrained case, the attacker wants to inject malicious data into specific state variables, but the attacker does not have to keep the other state variables unchanged. As discussed in Section 3.3.2, we use the Matching Pursuit algorithm [19, 26, 27] to find attack vectors. We perform the same set of experiments as in Section 4.2.2 to obtain the two evaluation metrics: the number of meters to compromise and the execution time. Note that in the unconstrained case, it takes significantly more time to find a near minimum number of meters than the previous experiments. Thus, we show more detailed results on execution time in this case.

Figure 6 shows the relationship between the number of meters to compromise and the number of specific state variables to compromise for the IEEE 300-bus system. Figure 7 shows the corresponding execution time of the Matching Pursuit algorithm for finding an attack vector successfully. We can see that the attacker needs to compromise 55–140 meters for the IEEE 300-bus system, if the attacker wants to inject malicious data into as many as 10 state variables. These meters can be quickly identified within 8 seconds.

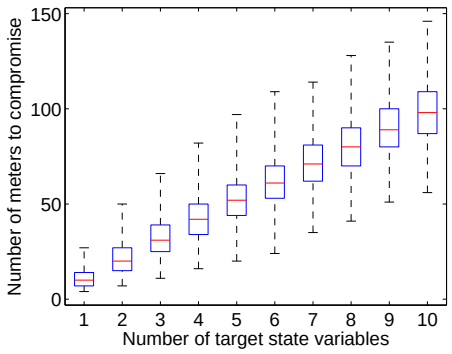

Figure 6: Unconstrained case: Number of meters to compromise to inject false data into l state variables in the IEEE 300-bus system

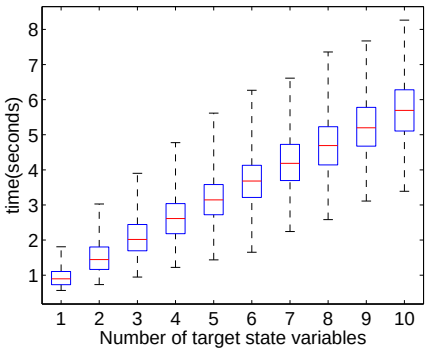

Figure 7: Unconstrained case: Execution time of finding an attack vector to inject false data into one state variable in the IEEE 300-bus system

We also exhaustively examine the special situation of injecting malicious data into a single state variable for all the

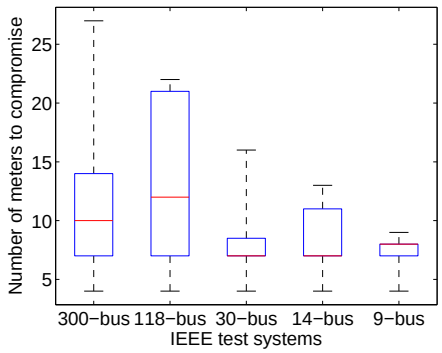

Figure 8: Unconstrained case: Number of meters to compromise to inject false data into one state variable

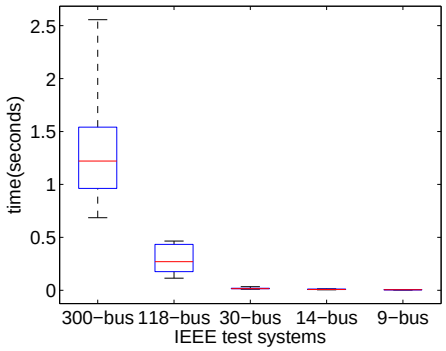

Figure 9: Unconstrained Case: Execution time of finding an attack vector to inject false data into one state variable

IEEE test systems, as in the constrained case. Figures 8 and 9 show the number of meters to compromise for these systems and the corresponding execution time, respectively. As shown in Figures 8 and 9, for example, the attacker can inject malicious data into any single state variable of the IEEE 300-bus system by compromising 27 meters, and it costs less than 2.6 seconds to find the attack vector.

These experimental results indicate that the false data injection attacks are practical and easy to launch if the attacker has the configuration information of the target system and can modify the meter measurements.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a new class of attacks, called *false data injection attacks*, against state estimation in electric power systems. We show that an attacker can take advantage of the configuration of a power system to launch such attacks to bypass the existing techniques for bad measurement detection. We considered two realistic attack scenarios, where the attacker is either constrained to some specific meters, or limited in the resources required to compromise meters. We showed that the attacker can systematically and efficiently construct attack vectors in both scenarios, which can not only change the results of state estimation, but also modify the results in a predicted way. We performed simulation on IEEE test systems to demonstrate the success of these attacks. Our results in this paper indicate that security protection of the electric power grid must be revisited when there are potentially malicious attacks.

In our future work, we would like to extend our results to state estimation using AC power flow models. Moreover, we will also investigate the possibility of adapting network anomaly detection techniques to identify false data injection attacks.

6. REFERENCES

[1] *Box Plot: Display of Distribution*. `http://www.physics.csbsju.edu/stats/box2.html`.

[2] *Electric Power Risk Assessment*. `http://www.solarstorms.org/ElectricAssessment.html`.

[3] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, December 1998.

[4] E. N. Asada, A. V. Garcia, and R. Romero. Identifying multiple interacting bad data in power system state estimation. In *IEEE Power Engineering Society General Meeting*, pages 571–577, June 2005.

[5] T. Blumensath and M. Davies. Gradient pursuits. *IEEE Transactions on Signal Processing*, 56(6):2370–2382, June 2008.

[6] J. Chen and A. Abur. Improved bad data processing via strategic placement of PMUs. In *IEEE Power Engineering Society General Meeting*, pages 509–513, June 2005.

[7] J. Chen and A. Abur. Placement of PMUs to enable bad data detection in state estimation. *IEEE Transactions on Power Systems*, 21(4):1608–1615, November 2006.

[8] S. S. Chen. *PhD thesis: Basis Pursuit*. Department of Statistics, Stanford University, 1995.

[9] E.Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter. Bad data analysis for power system state estimation. *IEEE Transactions on Power Apparatus and Systems*, 94(2):329–337, April 1975.

[10] A. Garcia, A. Monticelli, and P. Abreu. Fast decoupled state estimation and bad data processing. *IEEE Transactions on Power Apparatus and Systems*, 98(5):1645–1652, September 1979.

[11] M. R. Garey and D. S. Johnson. *Computer and Intractability: a guide to the theory of NP-Completeness*. W.H.Freeman and Company, 1979.

[12] S. Gastoni, G. P. Granelli, and M. Montagna. Multiple bad data processing by genetic algorithms. In *IEEE Power Tech Conference*, pages 1–6, June 2003.

[13] P. Georgiev and A. Cichoki. Sparse component analysis of overcomplete mixtures by improved basis pursuit method. In *the 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, pages 5:37–40, May 2004.

[14] D. V. Hertem, J. Verboomen, K. Purchala, R. Belmans, and W. L. Kling. Usefulness of DC power flow for active power flow analysis with flow controlling devices. In *The 8th IEE International Conference on AC and DC Power Transmission*, pages 58–62, March 2006.

[15] P. S. Huggins and S. W. Zucker. Greedy basis pursuit. *IEEE Transactions on Signal Processing*, 55(7):3760–3772, July 2007.

[16] L. Jeu-Min and P. Heng-Yau. A static state estimation approach including bad data detection and identification in power systems. In *IEEE Power Engineering Society General Meeting*, pages 1–7, June 2007.

[17] R. Kinney, P. Crucitti, R. Albert, and V. Latora. Modeling cascading failures in the north american power grid. *The European Physical Journal B – Condensed Matter and Complex Systems*, 46:101–107, 2005.

[18] M. Li, Q. Zhao, and P. B. Luh. DC power flow in systems with dynamic topology. In *Power and Energy Society General Meeting–Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–8, 2008.

[19] L. Lovisolo, E. A. B. da Silva, M. A. M. Rodrigues, and P. S. R. Diniz. Efficient coherent adaptive representations of monitored electric signals in power systems using damped sinusoids. *IEEE Transactions on Signal Processing*, 53(10):3831–3846, October 2005.

[20] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

[21] L. Mili, T. V. Cutsem, and M. Ribbens-Pavella. Hypothesis testing identification: A new method for bad data analysis in power system state estimation. 103(11):3239–3252, November 1984.

[22] L. Milli, T. V. Cutsem, and M. R. Pavella. Bad data identification methods in power system state estimation, a comparative study. *IEEE Transactions on Power Apparatus and Systems*, 103(11):3037–3049, November 1985.

[23] A. Monticelli. *State Estimation in Electric Power Systems, A Generalized Approach*. Kluwer Academic Publishers, 1999.

[24] A. Monticelli and A. Garcia. Reliable bad data processing for real-time state estimation. *IEEE Transactions on Power Apparatus and Systems*, 102(5):1126–1139, May 1983.

[25] A. Monticelli, F. F. Wu, and M. Y. Multiple. Bad data identification for state estimation by combinatorial optimization. *IEEE Transactions on Power Delivery*, 1(3):361–369, July 1986.

[26] B. K. Natarajan. Sparse approximate solutions to linear system. *SIAM Journal on Computing*, 24(2):227–234, April 1995.

[27] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *the 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

[28] V. H. Quintana, A. Simoes-Costa, and M. Mier. Bad data detection and identification techniques using estimation orthogonal methods. *IEEE Transactions on Power Apparatus and Systems*, 101(9):3356–3364, September 1982.

[29] F. C. Schweppe, J. Wildes, and D. B. Rom. Power system static state estimation. parts 1, 2, 3. *IEEE Transactions on Power Apparatus and Systems*, 89(1):120–135, January 1970.

[30] U.S.-Canada Power System Outage Task Force. *Final report on the August 14, 2003 blackout in the United States and Canada*. `https://reports.energy.gov/B-F-Web-Part1.pdf`, April 2004.

[31] A. Wood and B. Wollenberg. *Power generation, operation, and control*. John Wiley and Sons, 2nd edition, 1996.

[32] N. Xiang and S. Wang. Estimation and identification of multiple bad data in power system state estimation. In *the 7th Power Systems Computation Conference, PSCC*, pages 1061–1065, July 1981.

[33] N. Xiang, S. Wang, and E. Yu. A new approach for detection and identification of multiple bad data in power system state estimation. *IEEE Transactions on Power Apparatus and Systems*, 101(2):454–462, Febuary 1982.

[34] N. Xiang, S. Wang, and E. Yu. An application of estimation-identification approach of multiple bad data in power system state estimation. In *IEEE Power Engineering Society Summber Meeting*, July 1983.

[35] L. Zhao and A. Abur. Multi area state estimation using synchronized phasor measurements. *IEEE Transactions on Power Systems*, 20(2):611–617, May 2005.

[36] J. Zhu and A. Abur. Bad data identification when using phasor measurements. In *IEEE Power Tech Conference*, pages 1676–1681, July 2007.

[37] R. D. Zimmerman and C. E. Murillo-Sánchez. *MATPOWER, A MATLAB Power System Simulation Package*. `http://www.pserc.cornell.edu/matpower/manual.pdf`, September 2007.

APPENDIX

A. IEEE TEST SYSTEMS

We validate the false data injection attacks through experiments using IEEE test systems, including the IEEE 9-bus, 14-bus, 30-bus, 118-bus, and 300-bus systems. We extract the configuration of these test systems (particularly the matrix $\mathbf{H}$) from MATPOWER, a MATLAB package for solving power flow problems [37]. The information regarding the topology, bus data, and branch data can be found from source files of MATPOWER. The names of these source files are `case9.m`, `case14.m`, `case30.m`, `case118.m`, and `case300.m`.

Table 3 shows the number of state variables and the number of measurements in the IEEE test systems. All these systems are assumed to be fully measured. The matrix $\mathbf{H}$'s for the test systems are space consuming; we do not include them here.

Table 3: Number of state variables and measurements in the IEEE test systems

Test system	# State Variables	# Measurements
IEEE 9-bus	8	27
IEEE 14-bus	13	54
IEEE 30-bus	29	112
IEEE 118-bus	117	490
IEEE 300-bus	299	1,122