MALICIOUS DATA DETECTION AND LOCALIZATION IN STATE ESTIMATION LEVERAGING SYSTEM LOSSES

BY

MIAO LU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Master's Committee:

Research Scientist Rakesh Bobba
Professor Peter W. Sauer

ABSTRACT

In power systems, economic dispatch, contingency analysis, and the detection of faulty equipment rely on the output of the state estimator. Typically, state estimations are made based on the network topology information and the measurements from a set of sensors within the network. The state estimates must be accurate even with the presence of corrupted measurements. Traditional techniques used to detect and identify bad sensor measurements in state estimation cannot thwart malicious sensor measurement modifications, such as malicious data injection attacks. Recent work by Niemira (2013) has compared real and reactive injection and flow measurements as indicators of attacks. In this work, we improve upon the method used in that work to further enhance the detectability of malicious data injection attacks, and to incorporate PMU measurements to detect and locate previously undetectable attacks.

ACKNOWLEDGMENTS

I thank Professor Rakesh Bobba for his guidance and support over the years, and to Professor Peter W. Sauer for his invaluable suggestions, without which this work would not have been possible.

This material is based upon work supported in part by the Department of Energy under Award Number DE-OE0000097.

TABLE OF CONTENTS

1. INTRODUCTION

The power grid is a complex network monitored and controlled by the SCADA (Supervisory Control and Data Acquisition) system. The SCADA system relies on a large number of sensors to collect data to feed into the state estimator in order to estimate the state of the power grid. Accurate state estimates are crucial for economic dispatch, which determines power generation adjustments to match power demands, and for local grid operators to plan control actions in case of contingencies.

State estimators can be either AC or DC. The AC state estimator uses a nonlinear model, incorporating both real and reactive power flows and injections measurements. The DC state estimator uses a linear model that consists of only real power flows and injections measurements. The states consist of bus angles [1].

Ordinary bad data are generally caused by sensor misconfiguration or device failures. This type of bad data is usually large and isolated, which can be detected by traditional bad data detectors with enough measurement redundancy. However, work by Liu *et al.* [2] showed that an attacker, with knowledge of network configurations, can inject coordinated malicious data that are coherent with the DC power flow models without being detected. In [3], the potential success of DC attacks on real EMS (energy management system) software using a nonlinear model was shown. In [4], the sensitivity of real and reactive power measurement residuals in a nonlinear state estimator to false data injection attacks based on a linearized model was examined.

In this work, we show that as the system gets larger, the method used in [4] will have diminished detectability due to measurement noise.   We improve upon that method to further enhance the detectability of malicious data injection attacks, to incorporate PMU measurements to detect and locate previously undetectable attacks.

2. BACKGROUND

2.1 Power Flow

Power flow is a numerical analysis of the flow of electric power. It analyzes power systems in steady-state operation. The power flow solution also sets the initial condition for transient stability analysis.

The AC power injection equations for real power *P* and reactive power *Q* at a bus *i* are:

$$\mathbf{P_i = V_i \sum_{j=1}^{n} V_j\left[G_{ij}\cos\left(\theta_i - \theta_j\right) + B_{ij}\sin\left(\theta_i - \theta_j\right)\right]} \tag{1}$$

$$\mathbf{Q_i = V_i \sum_{j=1}^{n} V_j\left[G_{ij}\sin\left(\theta_i \cdot \theta_j\right) - B_{ij}\cos\left(\theta_i - \theta_j\right)\right]} \tag{2}$$

where *n* is the number of buses, *G* and *B* are the real and imaginary parts of the admittance matrix *Y*.

In this work, the AC power flow analyses were performed on power system test cases from [5] using MATPOWER [6], a flexible and powerful tool for power system research, to get the measurement data needed for state estimation.

2.2 State Estimation

State estimation uses measurements from network sensors to estimate the current state of the system. There is usually redundancy in the number of sensors in case of sensor failure, and to improve accuracy. Each sensor measurement becomes an equation to the state estimation solution. The weighted least-squares errors estimation method is used, which relies on the solution of an overdetermined system of equations, in order to find the solutions that fit best. The weighted least squares problem uses the following estimator:

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

where *W* is a diagonal matrix whose elements are the measurement weights, *H* is the Jacobian matrix representing the network topology, *z* represents the sensor measurements, and $\hat{x}$ represents the estimated states of the system.

2.3 Bad Data Detection

Sensor measurements might be inaccurate due to device misconfiguration or device failures. This type of bad data is usually large and isolated, which can be detected by traditional bad data detectors with enough measurement redundancy. Once the bad data has been found, the erroneous measurement is dropped as long as a set of basic measurements still exits [7]. A set of basic measurements is the minimum number of measurements needed to estimate the *n* state variables.

Many methods for identifying and correcting bad measurements have been proposed. A common approach [8], for detecting bad data is by looking at L2−norm of measurement residual defined as:

$$||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| \tag{4}$$

where $\hat{x}$ is the state estimate and $z - H\hat{x}$ is the measurement residual. If the value of expression in (4) is greater than a certain threshold, it is assumed that bad data is present.

2.4 Malicious Data Attacks

Malicious data injection attacks are those in which an attacker manipulates the sensor measurements to induce a change in the estimated state $\hat{x}$. With knowledge of the network topology, bad data can be injected into DC state estimator without changing the measurement residual, thus cannot be detected by traditional bad data detection schemes. In [2], Liu et al. present false data injection attacks that can bypass the bad data detection.

2.4.1 Attack Principle

Let $\boldsymbol{a}$ be an attack vector, the malicious data the attacker wants to add to the original measurement vector $\boldsymbol{z}$. Then $\boldsymbol{z_a} = \boldsymbol{z} + \boldsymbol{a}$ represents the resulting modified measurement vector. Theorem 1 in [2] shows that if the attack vector, *a*, was chosen to be equal to *Hc*, where *c* is the estimation error introduced, then resulting manipulated measurement $\boldsymbol{z_a} = \boldsymbol{z} + \boldsymbol{a}$ can pass the bad measurement detection scheme described previous:

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

where $\hat{\mathrm{x}}_{\mathrm{bad}}$ is the state estimates using manipulated measurements z_{a}.

2.4.2 Attack Incentives

In power system analysis, accurate state estimates are crucial for economic dispatch, which determines power generation adjustments to match power demands. In [9], it was shown that a data attack on state estimation may disrupt the dispatch operation that controls the system state trajectory by perturbing the economic dispatch solution throughout multiple state estimation periods. If the attack succeeds, it could mislead the control center into thinking that there will be an increase in demand. As result, more expensive units will be used to increase the generation in order to match that demand; and the cost of generation will increase. In [10], the impacts of malicious data attack on real-time electricity market were studied. It was shown that since the real-time price is a function of state estimates, and the real-time locational marginal prices (LMP) is a function of data measured from meters, injecting bad data can affect prices in the real-time market. If an attacker, with knowledge of the topology of the network, can inject malicious data to modify sensor measurements without being detected, then the attacker has the incentive and possesses the capability to alter the prices on the real-time electricity market to make a profit.

Since the power grid is one of our nation’s most critical infrastructures, it is by itself an attractive attack target. Adversaries may attempt to manipulate sensor measurements to cause equipment malfunction, monetary damage, or other malicious actions.

3. DETECTION OF MALICIOUS DATA ATTACKS

3.1 Previous Work and Approach

This is a continuation of work in [4], and thus it is important to briefly describe its approach and results.

3.1.1 Attacker Model

It is assumed that the attacker has the network topology information, at least one column of the Jacobian matrix *H*, in static form available to formulate DC data injection attacks. It is also assumed that the attacker has the ability to manipulate certain sensor measurements in order to launch an attack.

3.1.2 Nonlinear Sensitivity Analysis

Unlike traditional bad data, the malicious data injection attack is designed to fit the sum of squared residual test of a DC state estimator, minimizing the impact on measurement residues, thus avoiding detection. When an AC state estimator is being used against attacks generated based on the DC model, the measurement residues will increase because the AC model accounts for reactive power as well as system losses which the DC model has neglected.

The measurements being considered are the real and reactive power injections and flows. Since the impact of power losses increases with the square of current, the reactive flows would suffer more losses due to higher line reactance compared to line resistance; and since malicious data attack uses a linear model, reactive power estimates were expected to generate more measurement residues.

3.1.3 Establishing Baseline

In order to compare the measurement residues before and after the attack, baseline residual values had to be established. Distribution of residuals due to noise was established using Monte Carlo trails that consist of a normally distributed random variable with zero mean and standard deviation of 1% of the measurement values, which corresponds to 1% measurement noise, resulting in $z*$:

$$\mathbf{z^* = z + n} \tag{6}$$

where n is the measurement noise vector, and $z*$ is the sensor measurements including measurement noise. AC state estimations were performed to record the measurement residues in order to establish a range of acceptable residue values. For any value higher than the value established, malicious data attack is assumed. A cutoff can be chosen based on the percentage of acceptable false alarms. We will use 0% false alarms in our experiment, which means the cutoff chosen is the largest residue value of the established baseline.

3.1.4 Determine Detectability

In order to determine if the system is under malicious data attack, the procedure from the last section was repeated along with added attack vectors from the *H* matrix, resulting in:

$$\mathbf{z^{**} = z + n + a} \tag{7}$$

where *z*** is the measurement vector that includes noise vector *n* and attack vector *a*. In theory any combination of columns of DC *H* matrix can be served as an attack vector.

The detectability of an attack is defined by the percentage of residual above the baseline cutoff established in the last section. If 90% of residuals from the system under attack are above the cutoff value established earlier, then the attack is considered 90% detectable.

3.2 Evaluation of Approach for Larger Test Cases

In [4], analysis was conducted on the IEEE 14-bus test case. We would like to know if the results still hold for larger test cases such as the IEEE 30-bus and the IEEE 57-bus test cases; thus, analysis on these cases was conducted.

3.2.1 Setup

Analysis was conducted on the IEEE 30 and 57-bus test cases. MATPOWER, a MATLAB package developed for power system simulation, was used to perform state estimations. Distribution of residuals due to noise was established using Monte Carlo trails that consist of a normally distributed random variable with zero mean and standard deviation of 1% of the measurement values. We will use 0% false alarms in our experiment, which means the cutoff chosen is the largest residue value of the established baseline.

3.2.2 Establishing Baseline

Sum of squared residues of real and reactive flows and injections (PF, PG, QF, and QG) were recorded separately and as a weighted composite. Histograms of the result were generated. An example histogram for reactive power flows with 5000 random noise samples is shown in Figure 3.1. From these histogram bin counts, a cumulative density function (CDF) plot was created by normalizing the histogram to have an area of 1. The cutoff was established by finding the largest sum of squared residue value of the CDF. An example of CDF with cutoff for real power flows is shown in Figure 3.2.

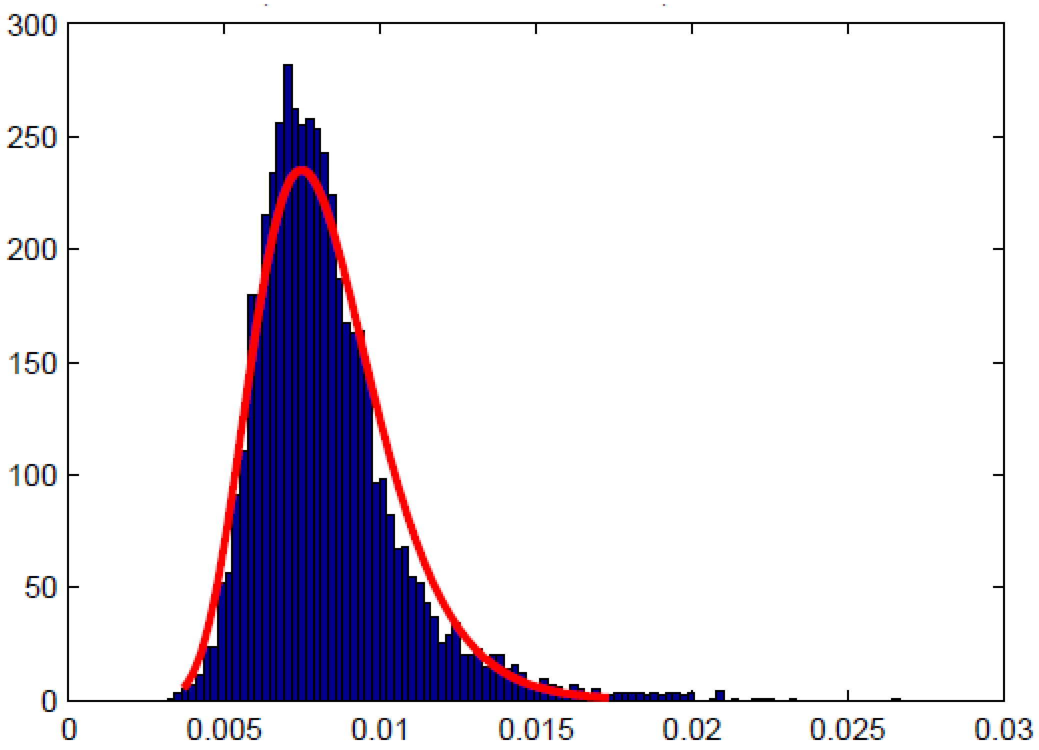

Figure 3.1: An example of histogram of sum of squared residues of reactive power flows (QF) for 5000 random noise vector samples.

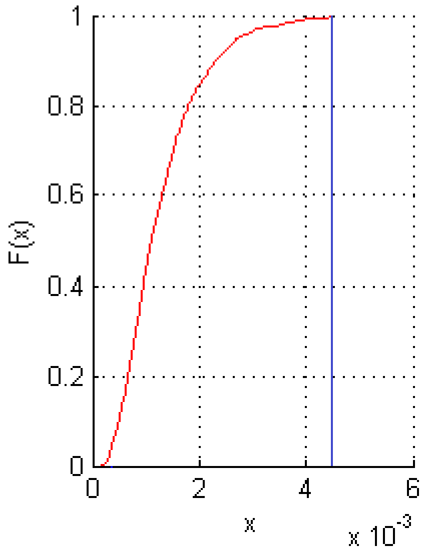

Figure 3.2: An example CDF of sum of squared residues for real power flows (PF), with 1000 random noise vector samples. The vertical line cuts the plot at 100% percentile, indicating the baseline cutoff residue value.

3.2.3 Determine Detectability

In theory any combination of columns of DC *H* matrix can be used to construct the attack vector. For these experiments we have chosen to use only the columns such that the largest entry in each attack vector is scaled to the attack injection level. The scale of attack vectors ranges from 10 MW to 50 MW in 100 MVA base. For every distribution generated, the amount of residues above the baseline cutoff was computed, and the residual type (PF, PG, QF, QG, and weighted composite) with the highest detectability was then recorded.

3.2.4 IEEE 30-Bus Test Case

In Figure 3.3, the grouped bars showed the percentage of attacks detected by different residual types at 10 MW injection level. For each measurement type, the size of residuals can be expected to vary greatly.

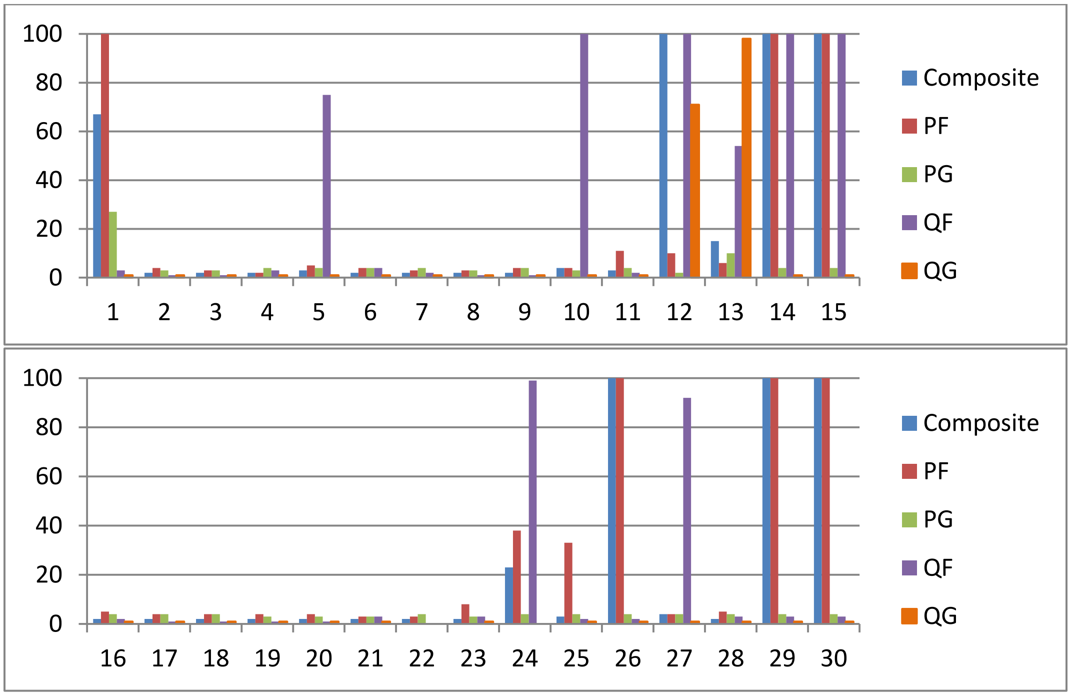

Figure 3.3: Grouped bars indicating percentage of each column DC *H* attack, from column 1 to column 30, detected at 10 MW attack injection level. From left to right, the bars represent the residuals of: the weighted composite, real power flow, real power generation, reactive power flow, and reactive generation.

Figure 3.4 shows the percentage of the total detectability of each attack column type detected, ranging from 10 to 50 MW attack injection level. While the majority of attacks are detectable, attacks based on column 2, 3, 8, 21, and 22 of DC *H* matrix are below 50 percent detectable even at 50 MW injection level. The total generation capacity of the IEEE 30-bus test case is about 335 MW. 50 MW injection is roughly 15% of total generation capacity.

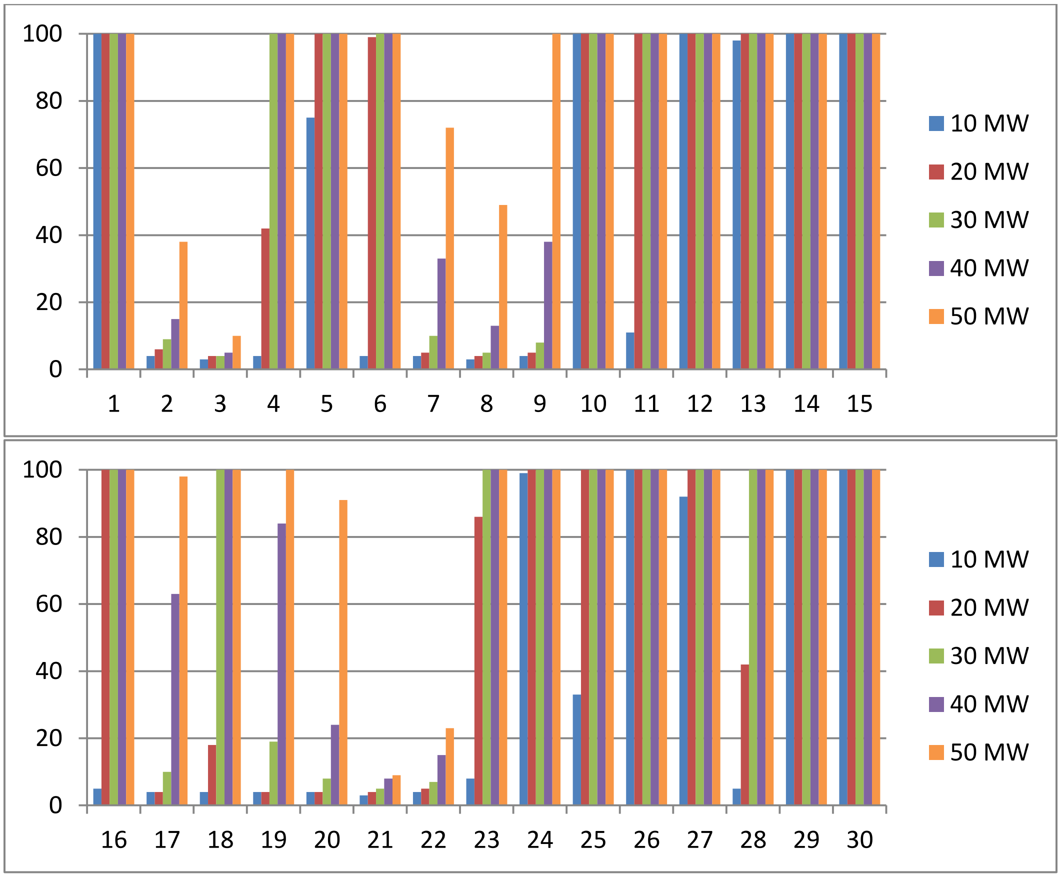

Figure 3.4: Grouped bars indicating the percentage of detectability of each attack column type detected range from 10 to 50 MW attack injection level.

3.2.5 IEEE 57-Bus Test Case

In Figure 3.5 below, the grouped bars showed the percentage of attacks detected by different residual types at 10 MW injection level.

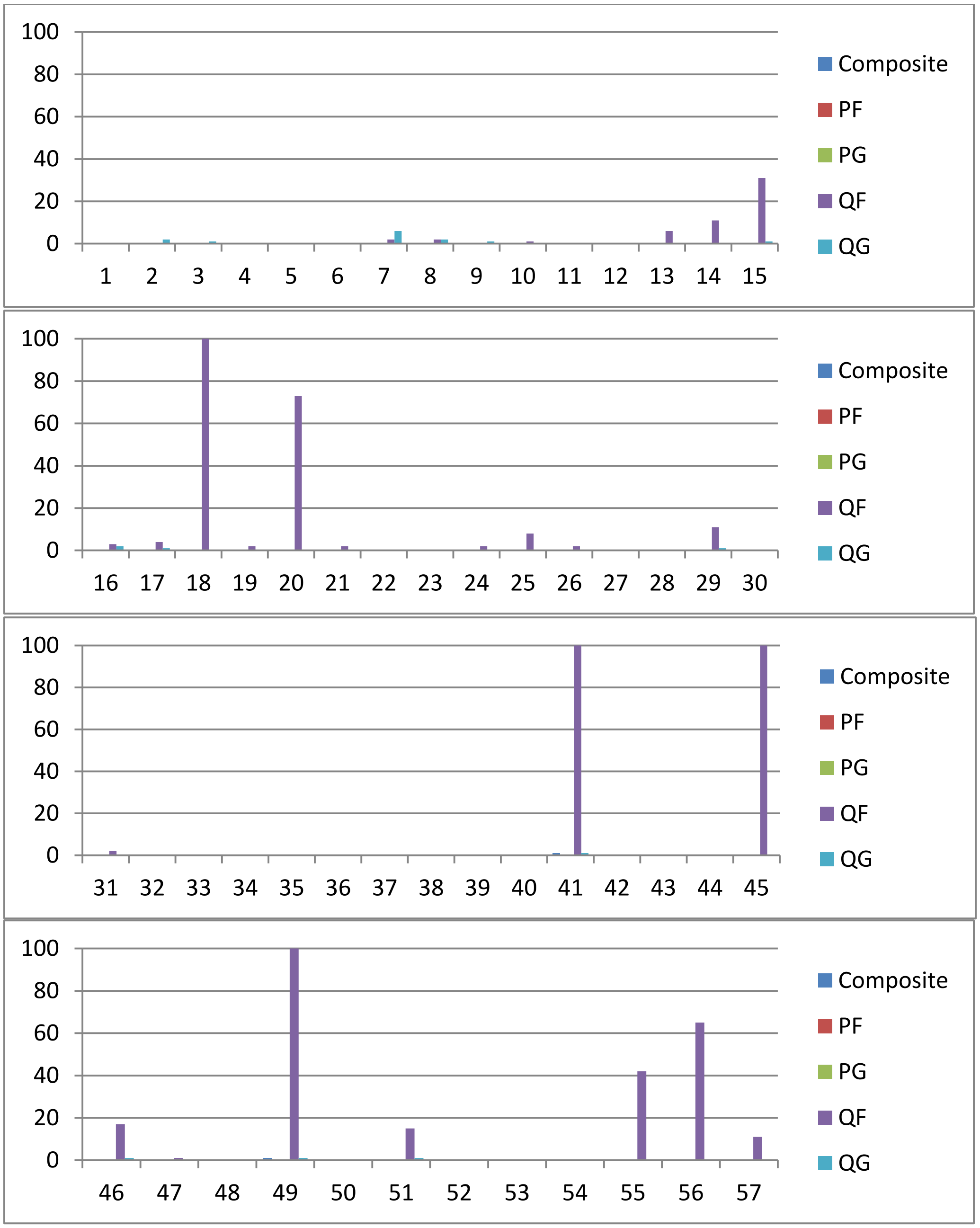

Figure 3.5: Grouped bars indicating percentage of each column DC *H* attack, from column 1 to column 57, detected at 10 MW attack injection level.

Figure 3.6 shows the percentage of the total detectability of each attack column type detected, ranging from 10 to 50 MW attack injection level. The total generation capacity of the IEEE 57-bus test case is about 1975 MW. At 50 MW injection levels, many column attacks remained undetected.

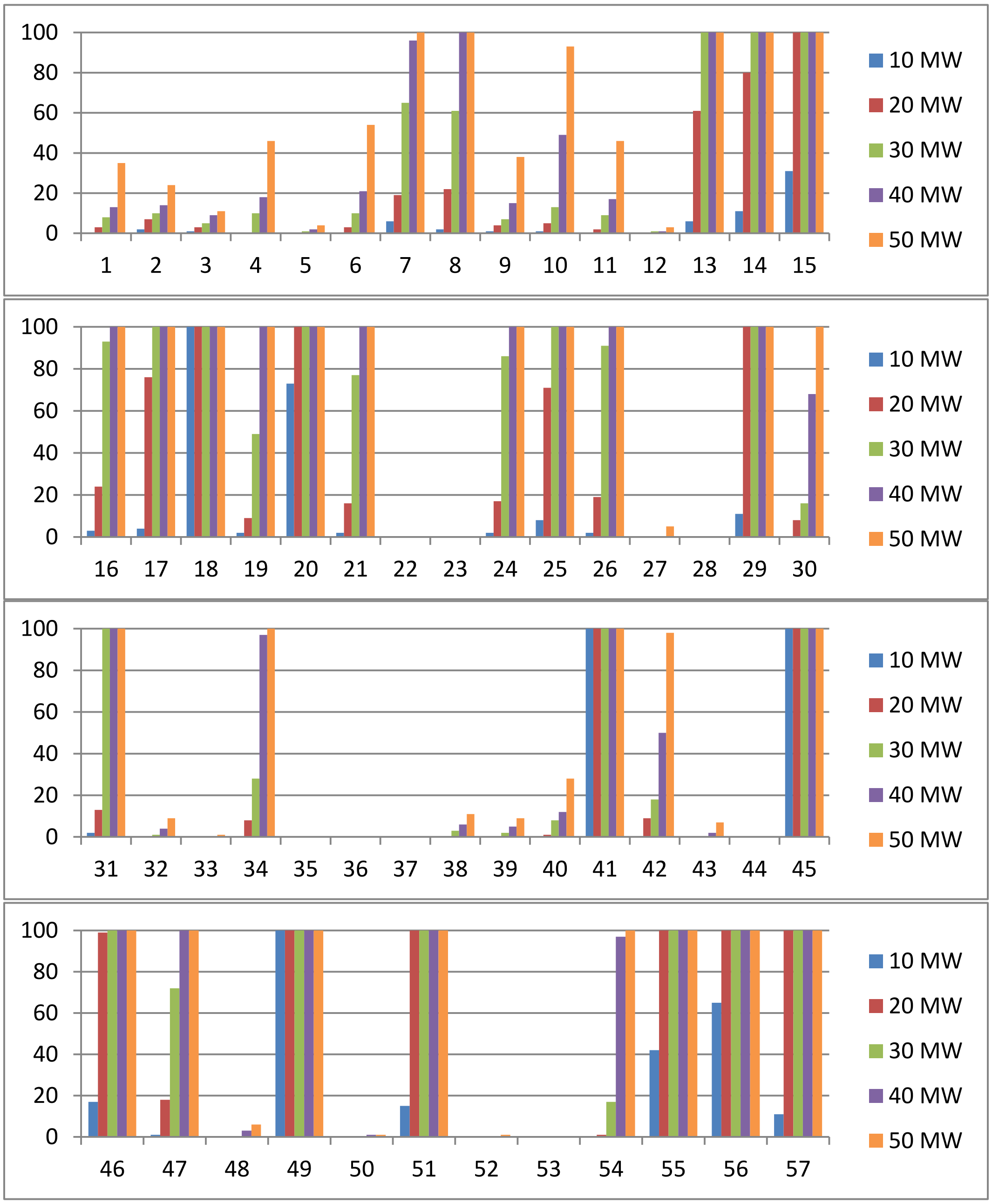

Figure 3.6: Grouped bars indicating the percentage of detectability of each attack column type detected range from 10 to 50 MW attack injection level.

3.2.6 Effect of Measurement Noise on Detectability

Figure 3.7 shows the detectability of QF residual type on attack launched using column 14 of DC H matrix at 100 MW level. The plot at the top contains the CDFs with 1% noise level. The plot at the bottom contains the CDFs with 2% noise level. At 1% noise level, we can clearly distinguish attack from noise, but at 2% noise it is no longer the case. With larger percentage of measurement noise, the CDF plots will further stretch vertically, making the detection of malicious data injection more difficult.

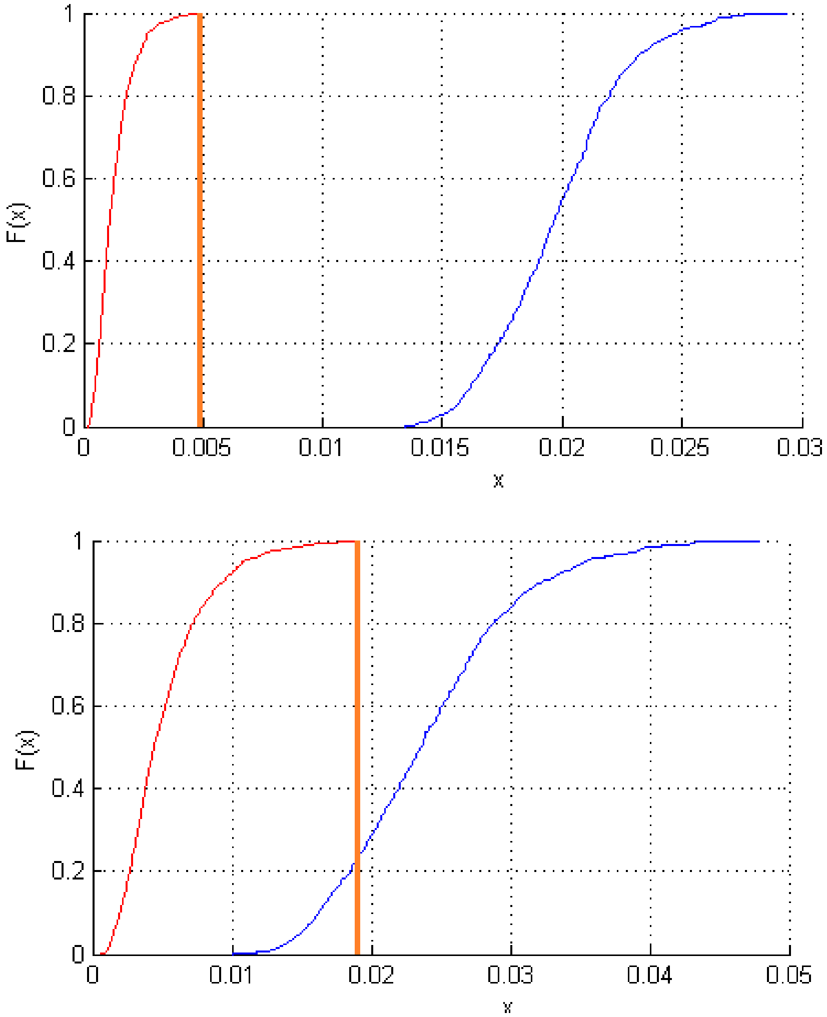

Figure 3.7: Detectability of the same attack due to different measurement noise levels.

3.3 How to Improve the Previous Approach

3.3.1 Note on Detectability

Looking at the figures and data from Section 3.2, many would make the assumption that as the generation capacity increases, the detectability decreases, assuming the attack injection levels were kept constant. The assumption is true, but it is not a major contributor causing attacks to become undetectable. The major contributors to undetectability of many attacks are the random noise assigned to each sensor measurement and the size difference between measured quantities.

For example: for the IEEE 57-bus test case, branch 1 has 102 MW flowing from bus 1 to bus 2, while branch 21 only has 0.67 MW flowing from bus 5 to bus 6. If branch 21 was under malicious data attack, i.e. column 6 of DC *H* matrix was used along with 1% random noise samples, the attack residue from branch 21 has to be extremely large for the attack to be detectable. The reason behind this is that the residue error created at branch 21(0.67 MW) is much smaller than the noise from branch 1 (102 MW) making it almost impossible to distinguish the bad data from noise. Our data from Section 3.2, Figure 3.8 column 6, shows that the detectability of attack using column 6 of DC *H* matrix is about 50% at 50 MW attack injection level.

Figure 3.8 shows the CDFs of sum of squared residues of reactive power flows (QF) for branch 21 alone, without residues from other branches, set to have 1% noise with 200 random noise samples, with and without malicious data attack at 30 MW injection level using column 6 of DC *H* matrix.

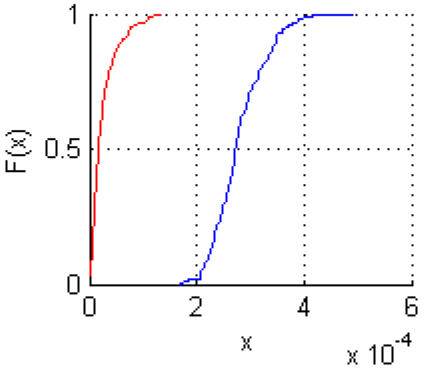

Figure 3.8: CDFs of sum of squared residues of branch 21 for reactive power flows (QF), set to have 1% noise with 200 random noise samples, with and without malicious data attack at 30 MW attack injection level using column 6 of DC *H* matrix.

Looking at Figure 3.8, by analyzing data from branch 21 alone, we can clearly tell that branch 21 was under malicious data attack because we are able to distinguish the sum of squared residues (in blue) from the 1% random noise samples (in red). To be thorough, Figure 3.9 shows the CDFs for weighted QF, sum of squared residues for branches 21 to 40.

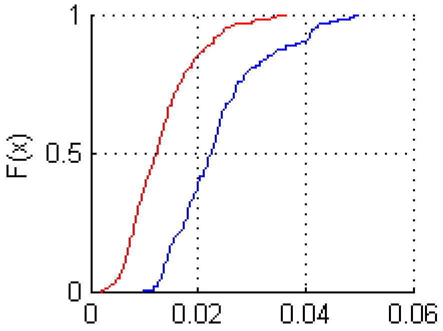

Figure 3.9: CDFs of sum of squared residues for reactive power flows (QF), for branch 20 to branch 40, set to have 1% noise with 200 random noise samples, with and without malicious data attack at 30 MW attack injection level using column 6 of DC *H* matrix.

Looking at Figure 3.9, one cannot distinguish residues below 0.035 from noise, thus making it about 10% detectable, which matches the data from Figure 3.6.

3.3.2 On Improving the Detectability of Malicious Data Attack

It was shown in Section 3.3.1 that the amount of measurement noise and the size difference between measured quantities can have negative effect on the detectability of malicious data attacks. As the system gets larger, it is crucial to come up with better detection schemes without consuming too much computing power.

In this improved approach, the sum of squared residues of all branches connected to each individual bus was calculated, with 1% random noise but without attack to establish baseline cutoffs for each bus. For example: for the IEEE 14-bus test case, the sum of squared residues, from bus 1 to 14, was calculated for each residual type. If the sum of squared residues of any bus exceeds the threshold determined by the baseline case, then we would assume an attack has occurred. The percentage of residues that exceed the baseline threshold is the percentage of attack being detected.

This approach was used because the residue from a single branch may not be large enough to distinguish it from noise, and each column of DC *H* matrix modifies branch data going in and out of a bus. For example: attack constructed using column 14 of DC *H* matrix modifies branch data connected to bus 14. As result, the residue to noise ratio should be the highest around bus 14. More details and simulation results will be provided in the next chapter.

4. IMPROVEMENT TO PREVIOUS METHOD

4.1 Improving the Detectability of Malicious Data Attack

It was shown in Section 3.3.1 that the amount of measurement noise and the size difference between measured quantities can have negative impact on the detectability of malicious data injection attacks. As the system gets larger, it is crucial to come up with better detection schemes without consuming too much computing power. In Section 3.3.2, a method of grouping the branch residues according to their connection to system buses was investigated.

4.2 Locations of Malicious Data Injection Attack

By grouping the residues according to their connection to network buses, if an attack was detected, then we were also able to limit the attack location to a few specific areas. Figure 4.1 showed attacks detected at bus 13 and bus 14, while only branches going in and out of bus 14 where modified. Even though the improved approach was unable to locate the attack at the exact location, it would save a lot of time in locating the sensors being tampered. An attack that is constructed as a combination of columns of DC H matrix will be detected at multiple buses. In the next section, simulation results of the improved approach will be shown.

4.3 Simulation Results

In this section, simulation results will be shown. In theory any combination of columns of DC *H* matrix can be used to construct the attack vector. For these experiments we have chosen to use only the columns such that the largest entry in each attack vector is scaled to the attack injection level. The scale of attack vectors ranges from 10 MW to 50 MW in 100 MVA base. For every distribution generated, the amount of residue above the baseline cutoff was computed, and the residual type (PF, PG, QF, QG, and weighted) with the highest detectability was then recorded.

4.3.1 Setup

The setup was exactly the same as Section 3.2.1.

4.3.2 Establishing Baseline

Squared residues of each real and reactive flow and injection (PF, PG, QF, and QG) were recorded separately. Sum of squared residues was calculated according to their connection to each system bus. Histograms of the result were generated. From these histogram bin counts, a CDF plot was created for each bus. The cutoffs were established by finding the largest sum of squared residue value of each CDF plot. We would then have a percentage of detectability for each bus, and we take the largest percentage to be the percent of attack detected.

For example, Figure 4.1 shows the CDF plots of PF residual type for IEEE 14-bus test case under 30 MW attack injection level using column 14 of DC *H* matrix. Since bus 14 has the most detectability, we would calculate the percentage of attack detected at bus 14, and use that number as the percent of overall attack detected.

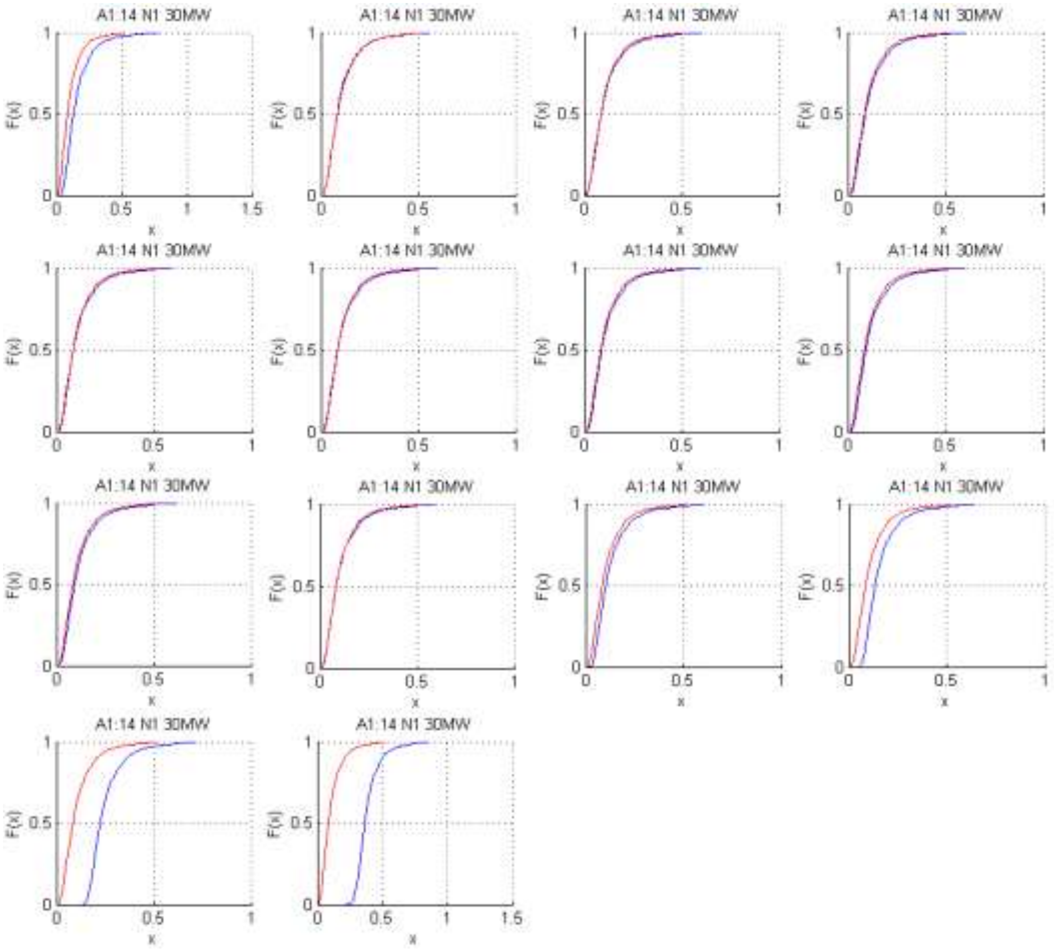

Figure 4.1: CDF plots of PF residual type for IEEE 14-bus test case under 30 MW attack injection level using column 14 of DC H matrix.

4.3.3 IEEE 14-Bus Test Case

Figure 4.2 shows the largest percentage of detectability among PF, PG, QF, and QG residual types for each attack column range from 10 to 50 MW attack injection levels.

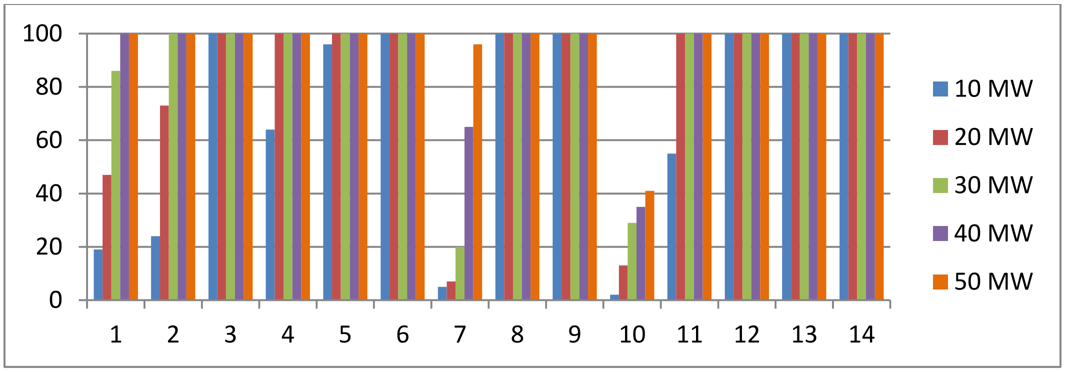

Figure 4.2: Grouped bars indicating the percentage of detectability of each column attack column detected range from 10 to 50 MW attack injection levels.

Comparing results obtained in this section to results obtained from [4], the detectability of attacks has been improved. Since the 14-bus test case is small, the improvement is not obvious. Table 4.1 shows the number of column attacks detected using the method from [4], and the method proposed in this chapter.

Table 4.1: for IEEE 14-bus test case, the number of column attacks detected using method from [4], and the method proposed in this chapter.

Attack Levels	Method used in [4]	Method proposed
10 MW	7	7
30 MW	11	11
50 MW	11	12

For the IEEE 14-bus test case, at 50 MW attack injection level, column attacks 7 and 10 are still not detectable. We will talk about how to detect these attacks in the later chapter.

4.3.4 IEEE 30-Bus Test Case

Figure 4.3 shows the largest percentage of detectability among PF, PG, QF, and QG residual types for each attack column range from 10 to 50 MW attack injection levels.

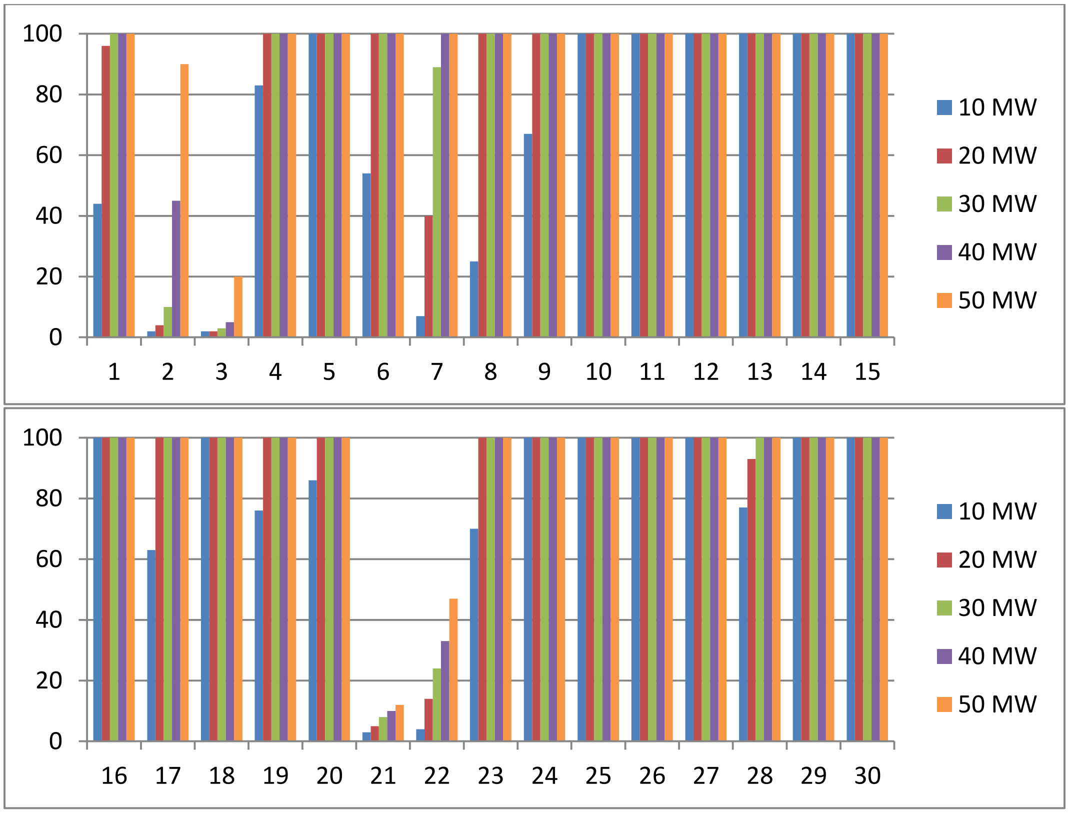

Figure 4.3: Grouped bars indicating the percentage of detectability of each column attack column detected range from 10 to 50 MW attack injection levels.

Compared to results obtained from 3.2.4 using method from [4], the detectability of attacks has been greatly improved, especially at lower attack injection levels. Table 4.2 shows the number of column attacks detected using the method from [4], and the method proposed in this chapter.

Table 4.2: For IEEE 30-bus test case, the number of column attacks detected using method from [4], and the method proposed in this chapter. The detectability of attacks has been improved, especially at lower attack injection levels.

Attack Levels	Method used in [4]	Method proposed
10 MW	8	15
20 MW	15	23
30 MW	20	25
40 MW	20	26
50 MW	22	26

For the IEEE 30-bus test case, at 50 MW attack injection level, column attacks 2, 3, 21, and 22 were still not detectable.

4.3.5 IEEE 57-Bus Test Case

Figure 4.4 shows the largest percentage of detectability among PF, PG, QF, and QG residual types for each attack column range from 10 to 50 MW attack injection levels for the IEEE 57-bus test case.

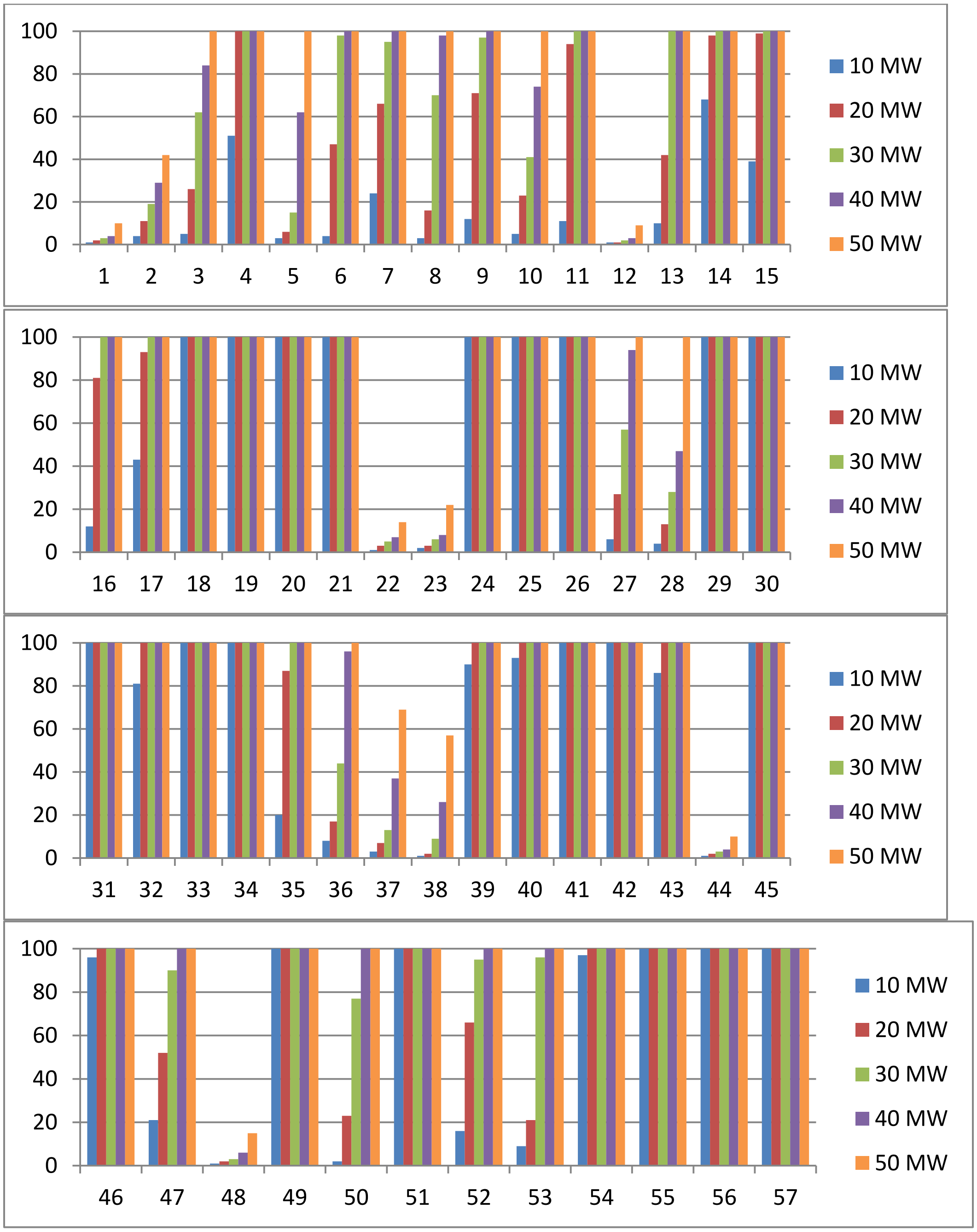

Figure 4.4: Grouped bars indicating the percentage of detectability of each column attack column detected range from 10 to 50 MW attack injection levels.

Compared to results obtained from 3.2.5 using the method from [4], the detectability of attacks has been once again improved. Table 4.3 shows the number of column attacks detected using the method from [4], and the method proposed in this chapter.

Table 4.3: For IEEE 57-bus test case, the number of column attacks detected using method from [4], and the method proposed in this chapter.

Attack Levels	Method used in [4]	Method proposed
10 MW	4	20
20 MW	11	27
30 MW	17	34
40 MW	24	41
50 MW	28	48

The method used in [4] will perform poorly as the bus system gets even larger due to the fact that column attacks are relatively sparse; the sum of squared error from the noise vector will outweigh the attack, thus making it more difficult to distinguish an attack from sensor noise.

5. DETECTION OF MALICIOUS DATA ATTACKS USING PMU

5.1 On Attacks that Are Difficult to Detect

We have noticed during our previous simulations that there were still some column attacks left undetected. As a result, the use of phasor measurement unit (PMU), or synchrophasor, to detect maliciously injected data was investigated.

PMU is considered as one of the most important measuring devices in the future of power systems. When placed on a network bus, it measures the magnitude and phase angle of voltage and current in real time. PMU relies on a GPS time signal for time-stamping of the power system information. Assuming the PMU measurements were unaltered by the attacker, then we can compare these measurements directly against the state estimates without relying on the residues.

Since the PMUs rely on GPS time signal as a reference to measure phase angles while the previous test cases rely on a single reference bus angle, comparing them directly does not make sense. Instead the angle differences between two buses were compared. For example: If an attack was injected using column 10 of DC *H* matrix, by comparing the angle differences, of bus 1 and bus 10, of PMU measurements against those from state estimation, malicious data injection attacks could be detected at much lower injection levels, and the PMU measurements can also provide bad data localization.

5.2 PMU Approach

5.2.1 Setup

The analysis was conducted on the IEEE 14, 30 and 57-bus test cases. MATPOWER was used to perform state estimations. Distribution of noise was established using Monte Carlo trails that consist of a normally distributed random variable with zero mean and standard deviation of 1% of the measurement values. The cutoff chosen was the largest angle difference between two buses from the established baseline, which means 0% false alarm. Calculation of residues was no longer needed here because the PMU measurements were assumed to be secure and accurate. Therefore if the state estimates did not match closely to the PMU measurements, i.e. attack injections were distinguishable from noise, then the attack was detected.

5.2.2 Establishing Baseline

In order to calculate the angle difference between two buses, a reference bus was needed. Since the reference buses for the IEEE test cases were chosen to be bus 1, the same buses were used to calculate the buses angle difference from PMU measurements, which means PMUs were placed at those reference buses. The bus angle difference from PMU measurements and the angle difference from the state estimates were recorded separately. Histograms of the result were generated. From these histogram bin counts, a CDF plot was created for each bus. The cutoffs were established by finding the largest angle difference of each CDF plot. A percentage of detectability for each bus was calculated, and the largest percentage was chosen to be the percent of attack detected.

5.3 Simulation Results

In this section, simulation results will be shown. In theory any combination of columns of DC *H* matrix can be used to construct the attack vector. For these experiments we have chosen to use only the columns such that the largest entry in each attack vector is scaled to the attack injection level. The scale of attack vectors range from 5 MW to 15 MW in 100 MVA base.

5.3.1 Attack on the Reference Bus

Since a bus is being used as the reference bus, if the reference bus was attacked, the PMU placed at the reference bus would not be able to detect such attack, but all other phase angle differences were expected to change thus making the attack even more obvious. Figure 5.1 shows what happens when bus 1 of IEEE 14-bus test case was attacked using column 1 of DC *H* matrix at 10 MW injection level.

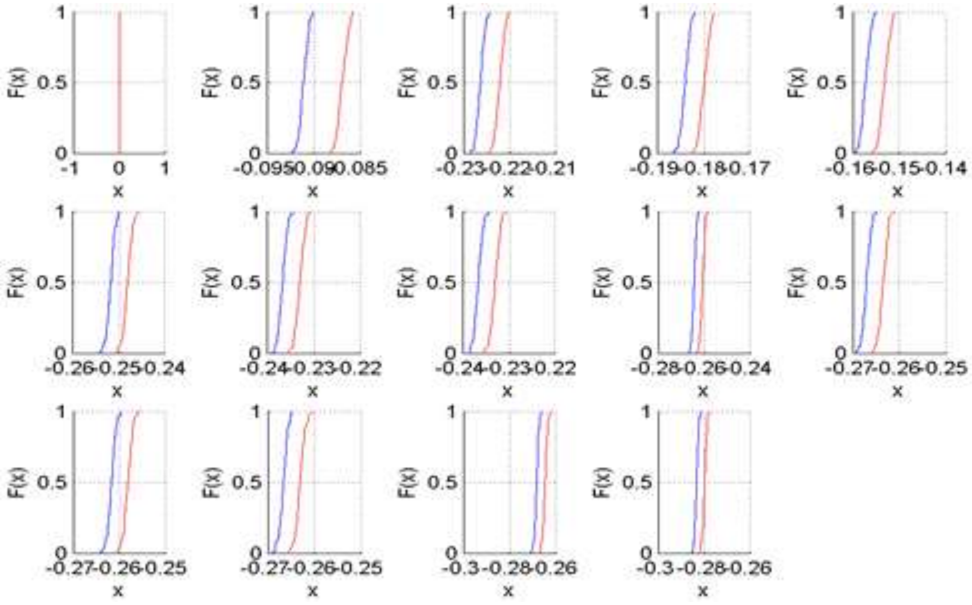

Figure 5.1: CDFs of PMU measurements and state estimates of bus angles for IEEE 14-bus test case when the reference bus was attacked under 10 MW injection level using column 1 of DC H matrix.

Figure 5.2 shows what happens when bus 10 of IEEE 14-bus test case was attacked using column 10 of DC H matrix at 10 MW injection level. In [4], it was shown that even at 80 MW injection level this specific attack was undetectable.

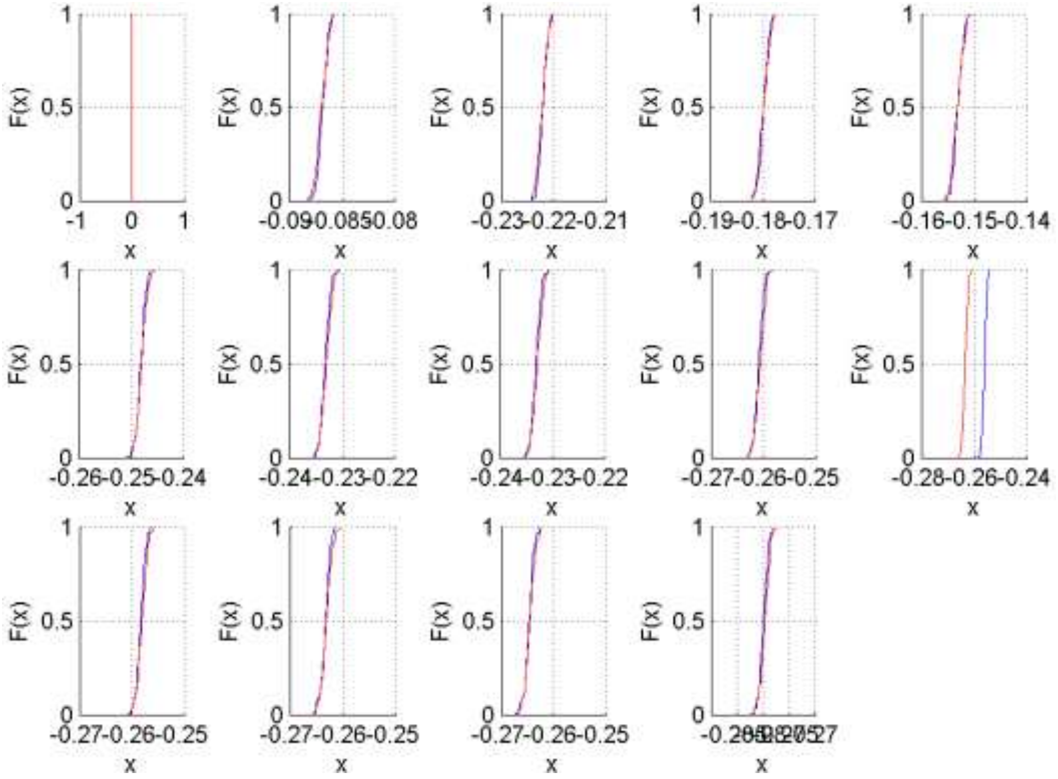

Figure 5.2: CDFs of PMU measurements and state estimates of bus angles for IEEE 14-bus test case when the reference bus was attacked under 10 MW injection level using column 10 of DC H matrix.

Even at only 10 MW attack injection level, the CDF for bus 10 clearly showed that there was something wrong. The exact percentage of detectability will be shown in the next section.

5.3.2 IEEE 14-Bus Test Case

Figure 5.3 shows the percentage of detectability of column attacks ranging from 5 to 15 MW attack injection levels for the IEEE 14-bus test case. Attacks were generated using only columns of DC *H* matrix. At 15 MW injection levels, all column attacks were detected.

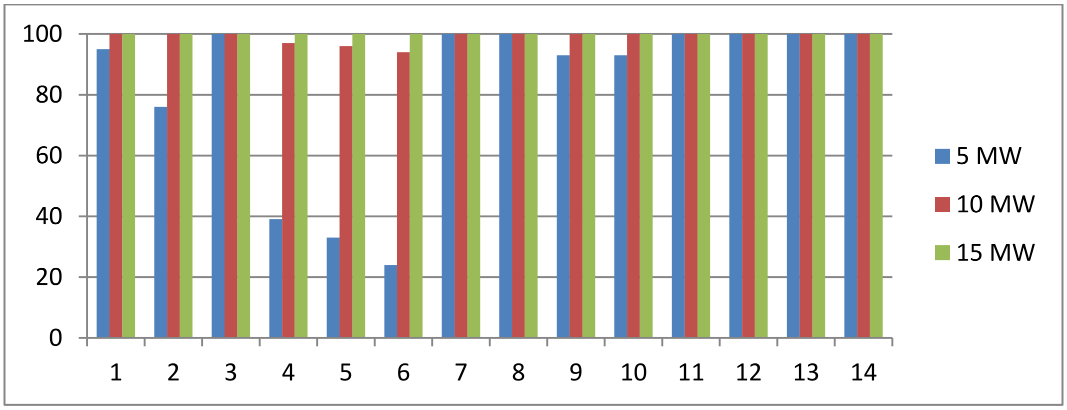

Figure 5.3: Percentage of detectability of column attacks ranging from 5 to 15 MW attack injection levels. Attacks were generated using only columns of DC *H* matrix from the IEEE 14-bus test case.

5.3.3 IEEE 30-Bus Test Case

Figure 5.4 shows the percentage of detectability of column attacks ranging from 5 to 15 MW attack injection levels for the IEEE 30-bus test case. Attacks were generated using only columns of DC *H* matrix. At only 5 MW injection levels, all column attacks were detected.

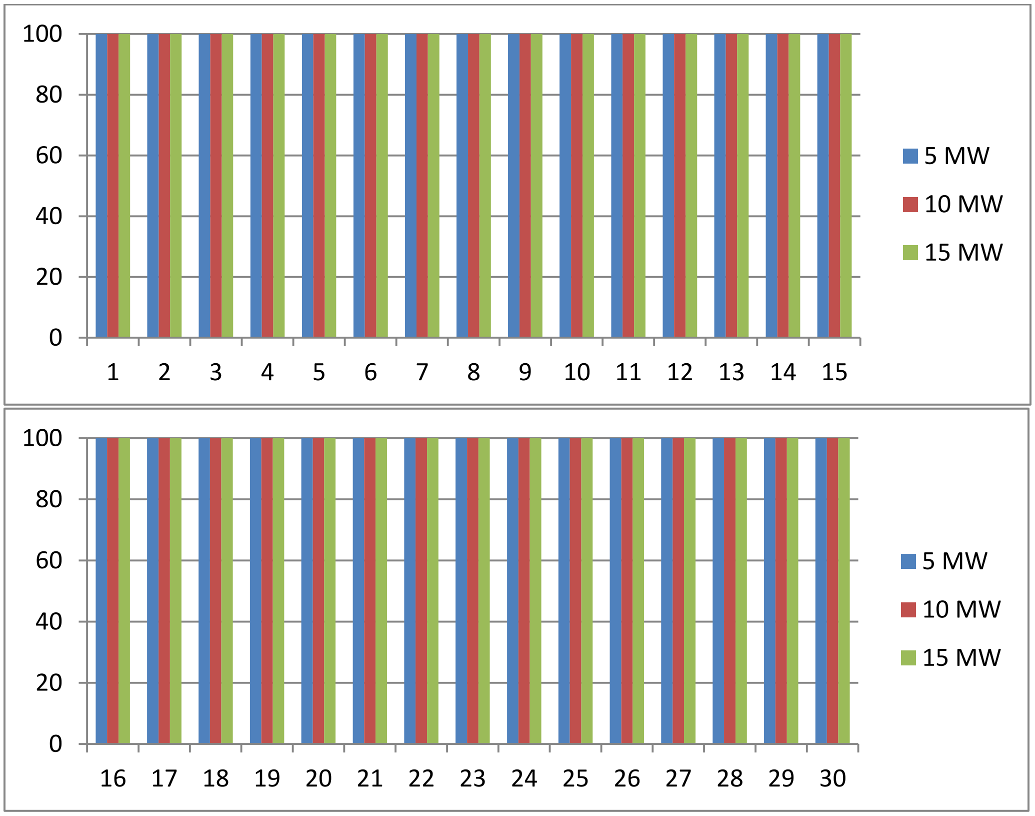

Figure 5.4: Percentage of detectability of column attacks ranging from 5 to 15 MW attack injection levels. Attacks were generated using only columns of DC *H* matrix from the IEEE 30-bus test case.

5.3.4 IEEE 57-Bus Test Case

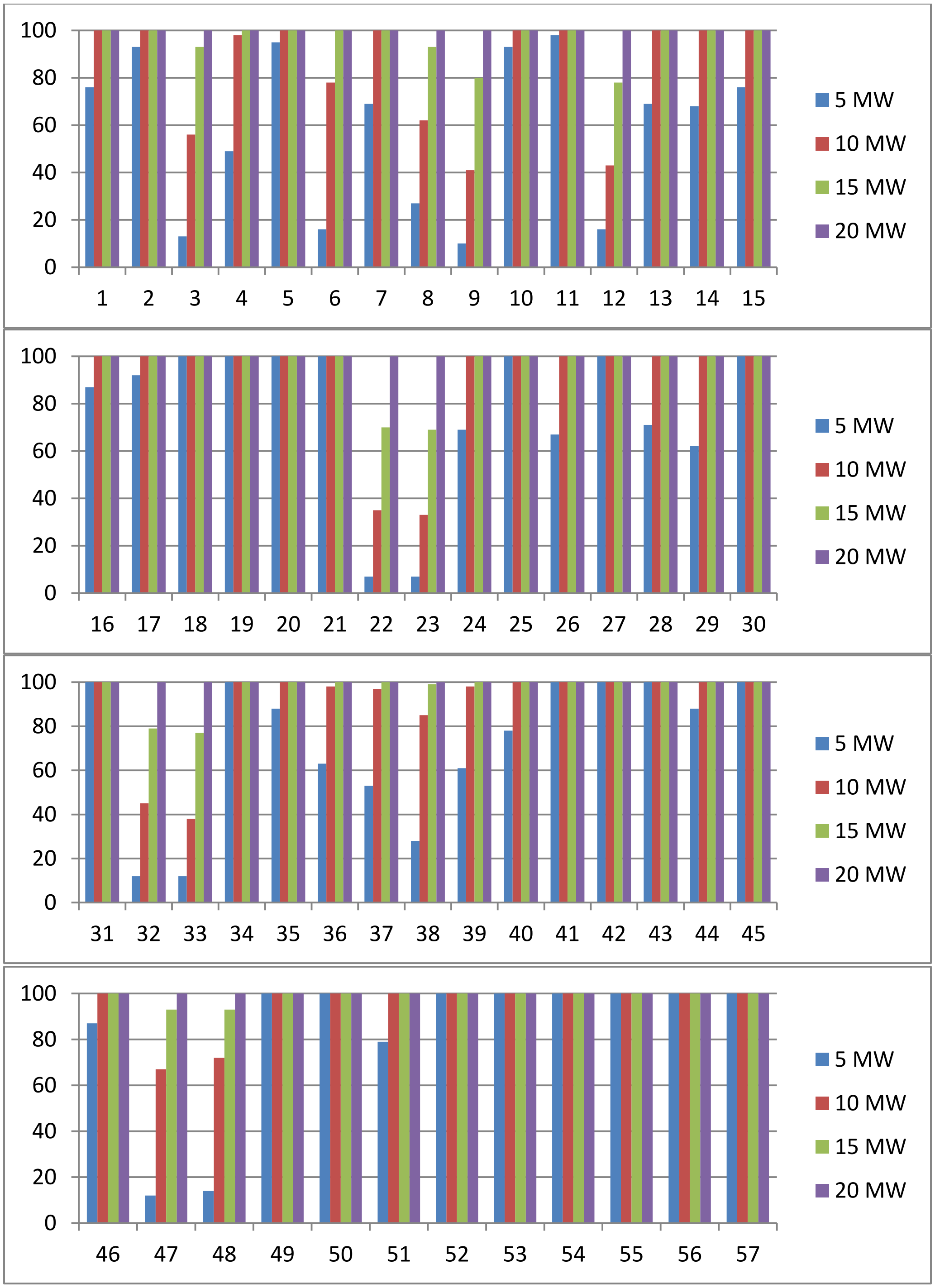

Figure 5.5: Percentage of detectability of column attacks ranging from 5 to 15 MW attack injection levels. Attacks were generated using only columns of DC *H* matrix from the IEEE 57-bus test case.

Figure 5.5 shows the percentage of detectability of column attacks ranging from 5 to 20 MW attack injection levels for the IEEE 57-bus test case. Attacks were generated using only columns of DC *H* matrix. At 20 MW injection levels, all column attacks were detected. The total real power generation of the IEEE 57-bus test case is about 1300 MW. 20 MW is about 1.6% of total real power generated.

5.4 Localization of Malicious Data Attacks using PMU

In section 5.3, it was shown in Figure 5.1 that if the reference bus was under attack, the CDF plots of PMU measurements and state estimates will have large angle differences at other buses. Figure 5.2 showed that PMU placed on bus 10 of the IEEE 14-bus test case could successfully detect attacks launched using column 10 of the DC *H* matrix at 10 MW injection level, which was not detectable even at 80 MW injection level using the method in [4]. Figure 5.2 also showed that none of the other PMUs has detected large bus angle differences, which means the location of the attack has been exposed.

Previously, even if an attack was detected using the method in [4], the location of the attack cannot be derived from the sum of squared of attack residues. It would take much time to locate and fix the damage created by the attacker. Using the approach proposed in Chapter 4, the detectability of attacks was improved; the approach also limits the location of the attack to a few specific areas. For example, Figure 4.1 showed attacks detected at bus 13 and bus 14, while only branches going in and out of bus 14 were modified. Even though the approach in Chapter 4 was unable to locate the attack at the exact location, it would save a lot of time in locating the sensors being tampered with compare to the method used in [4]. By incorporating PMU measurements, the attacks can be pinpointed at the exact bus locations at much lower attack injection level. This is significant because it has improved the attack detectability at much lower attack injection levels, it can save a lot of time finding the attack location, and it provides an extra layer of protection; i.e., in order for the attacks to succeed, the attacker has to tamper with the sensors as well as the PMU measurements.

6. PMU PLACEMENT

6.1 Where to Place the PMUs

Given limited resources, the PMUs need to be strategically placed along with the use of the approach from Chapter 4 to achieve the best protection without running exhaustive simulations. For example: if a minimum attack injection level is given, i.e. any attack above or equal to such level should be detected, then we want to know where to place the PMUs to maximize protection against malicious data attacks, and how many PMUs are needed to complement the method from Chapter 4.

6.1.1 Centrality Measures

The network centralities from [11] were first investigated in order to identify the important nodes in the system where PMUs should be strategically placed in terms of system vulnerabilities. The centralities did not complement the work in [4] or the approach from Chapter 4. One of the reasons could be that the work in [11] did not incorporate system generation and load, which are important in determining the amount of real and reactive flows and injections given the admittance matrix Y.

6.1.2 Using Residues for PMU Placements

To complement the work in [4], baseline cutoffs for each of the residual types (PF, PG, QF, and QG) were found the same way as in Section 3.2.2. Then with specified attack injection levels and 0% measurement noise, attacks generated using the columns of the DC *H* matrix were launched. The sum of squares of attack residues for each residual type can be calculated by running the state estimation once. Since there is no noise, the state estimates would not vary by running state estimation multiple times. Then the sum of squared residues were sorted from smallest to largest; the smallest was the most vulnerable to malicious data injection attack, and the bus that the column attack had modified was the location where the PMU needed to be placed. Note: PMU placement on the reference bus is required in order to calculate bus angle difference.

In order to complement the improved approach from Chapter 4, baseline cutoffs for each of the residual types (PF, PG, QF, and QG) were found the same way as in Section 4.3.2 by grouping the branches according to their connections to each system bus. Then with specified attack injection levels and 0% measurement noise, squared of attack residues for each residual type were found by running the state estimation once. The residues were then grouped according to their connection to each system bus. For each attack launched, we would have a sum of squared residue for each bus. The largest value among the buses was recorded as the attack residue. Attacks were formed using columns of DC *H* matrix. The sum of squares of attack residues for each column of DC *H* matrix were then sorted from smallest to largest, the smallest was the most vulnerable to malicious data injection attack, and is where the PMU needed to be placed.

6.1.3 Number of PMUs Needed

In Section 6.1.2, we introduced a way to strategically place the PMUs to enhance malicious data detectability. Since the PMUs were used to complement the approach from Chapter 4, the PMU placements need to be stopped as soon as the attack residues are large enough to be detected by the approach alone. By comparing the attack residues with the baseline cutoffs determined from Section 6.1.2, the attacks that will be detected by the approach from Chapter 4 can be roughly determined.

In order to distinguish an attack from noise, the largest baseline cutoff from the CDF with no attack but noise has to be smaller than the smallest attack residue from the CDF with attack and noise. By running the column attack only once without noise, the CDF curve for the attack vector injected was nonexistent. As such, a coefficient was needed to be divided by the attack residues in order to find a rough estimate of the leftmost attack residue value. The point of this is to verify that the baseline cutoff, i.e. the largest residue value due to noise alone, is smaller than the smallest attack residue, thus making the difference between noise and attack distinguishable without running exhaustive simulations.

For the purpose of simulation, the coefficient is chosen to be the baseline cutoff divided by the baseline median. How to determine a better coefficient to find the smallest attack residue value of the CDF plot without running exhaustive simulation could be a topic for future work.

6.2 PMU Placements

From Section 5.3.4, it has been shown that for the IEEE 57-bus test case, all attacks generated using the columns of the DC *H* matrix can be detected at 20 MW attack injection levels if PMUs are placed on the buses. In this chapter, the PMU placements that complement the approach from Chapter 4, for attacks generated using the columns of the DC *H* matrix ranging from 20 to 50 MW, will be shown.

6.2.1 IEEE 14-Bus Test Case

Table 6.1 shows where to place the PMUs to further improve the detectability of malicious data injection attacks if the approach from Chapter 4 has been used without running exhaustive simulations. The smaller the number assigned to a bus, the more vulnerable that bus. A zero means no PMU was needed. Since the difference between two bus angles was measured, PMU placement on the reference bus was required, which is bus 1 in this case.

Simulation in Chapter 4 used 500 random noise samples for the IEEE 14-bus test case, which means AC state estimation was performed 500 times to establish the baseline cutoff and 7000 times to simulate the detectability of 14 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 14 column attacks ranging from 20 to 50 MW injection levels, a total of 28500 AC state estimations on the IEEE 14-bus test case were needed. As the test case gets larger, such a method of finding where to place the PMUs is not efficient.

Table 6.1: PMU placement for the IEEE 14-bus test case to complement detection approach used in Chapter 4 without running exhaustive simulations.

	Bus 1	Bus 2	Bus 3	Bus 4	Bus 5	Bus 6	Bus 7
20 MW	1	4	0	0	0	0	2
30 MW	1	0	0	0	0	0	2
40 MW	1	0	0	0	0	0	3
50 MW	1	0	0	0	0	0	3
	Bus 8	Bus 9	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14
20 MW	0	0	3	0	0	0	0
30 MW	0	0	3	0	0	0	0
40 MW	0	0	2	0	0	0	0
50 MW	0	0	2	0	0	0	0

Using the method proposed in this chapter, AC state estimation was performed 500 times to establish the baseline cutoff and 14 times to simulate the detectability of 14 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 14 column attacks ranging from 20 to 50 MW injection levels for PMU placement, a total of 556 AC state estimations on the IEEE 14-bus test case were needed.

Comparing the results from Table 6.1 with simulation results from Chapter 4, the buses that are the most vulnerable to malicious data injection attack were found, the placement order for bus 7 and bus 10 was switched at the 20 MW injection level.

6.2.2 IEEE 30-Bus Test Case

Table 6.2: PMU placements for the IEEE 30-bus test case to complement detection approach used in Chapter 4 without running exhaustive simulations.

	Bus 1	Bus 2	Bus 3	Bus 4	Bus 5	Bus 6	Bus 7
20 MW	1	4	2	0	0	0	6
30 MW	1	4	2	0	0	0	6
40 MW	1	5	2	0	0	0	0
50 MW	1	5	2	0	0	0	0
	Bus 8	Bus 9	Bus 10	Bus 11	Bus 12	Bus 13	Bus 14
20 MW	0	0	0	0	0	7	0
30 MW	0	0	0	0	0	0	0
40 MW	0	0	0	0	0	0	0
50 MW	0	0	0	0	0	0	0
	Bus 15	Bus 16	Bus 17	Bus 18	Bus 19	Bus 20	Bus 21
20 MW	0	0	0	0	0	0	3
30 MW	0	0	0	0	0	0	3
40 MW	0	0	0	0	0	0	3
50 MW	0	0	0	0	0	0	3
	Bus 22	Bus 23	Bus 24	Bus 25	Bus 26	Bus 27	Bus 28
20 MW	5	0	0	0	0	0	0
30 MW	5	0	0	0	0	0	0
40 MW	4	0	0	0	0	0	0
50 MW	4	0	0	0	0	0	0
	Bus 29	Bus 30					
20 MW	0	0					
30 MW	0	0					
40 MW	0	0					
50 MW	0	0					

Table 6.2 shows where to place the PMUs to further improve the detectability of malicious data injection attacks if the approach from Chapter 4 has been used without running exhaustive simulations. The smaller the number assigned to a bus, the more vulnerable was that bus. A zero means no PMU was needed. Bus 1 was used as reference bus.

Simulation in Chapter 4 used 500 random noise samples for the IEEE 30-bus test case, which means AC state estimation was performed 500 times to establish the baseline cutoff and 15,000 times to simulate the detectability of 30 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 30 column attacks ranging from 20 to 50 MW injection levels, a total of 60500 AC state estimations on the IEEE 30-bus test case were needed.

Using the method proposed in this chapter, AC state estimation was performed 500 times to establish the baseline cutoff and 30 times to simulate the detectability of 30 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 30 column attacks ranging from 20 to 50 MW injection levels for PMU placement, a total of 620 AC state estimations on the IEEE 30-bus test case were needed.

Comparing the results from Table 6.2 with simulation results from Chapter 4, at 20 MW attack injection level, there should have been a PMU placement for bus 28, but it was missing, so a redundant PMU was placed on bus 13 instead. Attack generated using the 28th column of the DC *H* matrix was above 90% detectable using the method from Chapter 4. For attack injection levels ranging from 30 to 50 MW, the buses that are the most vulnerable to malicious data injection attack were found correctly.

6.2.3 IEEE 57-Bus Test Case

Table 6.3 shows where to place the PMUs to further improve the detectability of malicious data injection attacks if the approach from Chapter 4 has been used without running exhaustive simulations. Bus 1 was used as reference bus.

Table 6.3: The order of PMU placements for the IEEE 57-bus test case to complement the detection approach used in Chapter 4 without running exhaustive simulations.

Bus Protection Order	Bus Number							
20 MW	1	12	44	48	38	22	23	2
	37	5	28	10	36	27	3	50
	8	53	13	6	47	7	52	9
	16	35	17	11	32	33		
30 MW	1	12	44	48	22	23	38	2
	37	28	5	10	36	27	3	8
	50	53	6	11	13	35	32	33
40 MW	1	12	44	48	22	23	2	38
	37	28	10	5	3	36	53	50
	6	33						
50 MW	1	12	44	48	22	23	2	38
	37	28	10	36				

Simulation in Chapter 4 used 500 random noise samples for the IEEE 57-bus test case, which means AC state estimation was performed 500 times to establish the baseline cutoff and 28,500 times to simulate the detectability of 57 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 30 column attacks ranging from 20 to 50 MW injection levels, a total of 114,500 AC state estimations on the IEEE 57-bus test case were needed.

Using the method proposed in this chapter, AC state estimation was performed 500 times to establish the baseline cutoff and 57 times to simulate the detectability of 57 column attacks for an arbitrary attack injection level. In order to simulate the detectability of 57 column attacks ranging from 20 to 50 MW injection levels for PMU placement, a total of 728 AC state estimations on the IEEE 30-bus test case were needed.

Table 6.4: The order of PMU placements for the IEEE 57-bus test case determined using simulation results from Chapter 4.

Bus Protection Order	Bus Number							
20 MW	1	12	38	44	48	22	23	5
	37	2	28	8	36	53	10	50
	3	27	13	6	47	7	52	9
	16	35	17	11	14	15		
30 MW	1	12	44	48	22	23	38	37
	5	2	28	10	36	27	3	8
	50	47	7	52	53	9	6	
40 MW	1	12	44	48	22	23	38	2
	37	28	5	10	3	27	36	8
50 MW	1	12	44	22	48	23	2	38
	37							

Table 6.4 shows the actual PMU placements needed after simulation by running exhaustive AC state estimations from Chapter 4. Comparing the results from Table 6.3 and Table 6.4 along with simulation results from Chapter 4, at 20 MW attack injection level, there should have been two PMUs placed on bus 14 and 15, but instead two PMUs were placed on bus 32 and 33. Attacks generated using the 14th and 15th columns of DC *H* matrix were above 95% detectable using the method from Chapter 4.

At 30 MW attack injection level, there should have been four PMUs placed on bus 7, 9, 47, and 52, but instead five PMUs were placed on bus 11, 13, 32, 33, and 35. Attacks generated using the 7th, 9th, and 52nd columns of the DC *H* matrix were above 95% detectable using the method from Chapter 4. The 47th had detectability above 90%.

At 40 MW attack injection level, there should have been two PMUs placed on bus 8 and 27, but instead four PMUs were placed on bus 6, 33, 50, and 53. Attacks generated using the 8th and 27th columns of DC *H* matrix were above 95% detectable using the method from Chapter 4.

At 50 MW attack injection level, the buses that are the most vulnerable to malicious data injection attack were found correctly but three extra PMUs were placed on bus 10, 28 and 36.

For the IEEE 57-bus test case, most of the PMU placements were correct. For those missed locations, the simulation from Chapter 4 shows detectability above 90%

6.3 Protecting the Basic Set

In [7], the authors showed that protecting a strategically selected set of sensor measurements or state variables prevented attackers from launching unobservable attacks. The PMU placements introduced in this chapter could be used to complement the approach used in Chapter 4. For example: Figure 4.2 showed that attacks on bus 7 and 10 were difficult to detect even at 50 MW attack injection level using the approach from Chapter 4, so in order to complement the approach, PMUs were needed. Table 6.1 showed that at 50 MW attack injection level, PMUs were needed on bus 1, 7, and 10. This is important because instead of protecting a basic set of sensor measurements, or the equal number of state variables, the number of state variables that required protection has now been limited to only three.

7. CONCLUSION AND FUTURE WORK

This work followed the idea from [4] that, by using an AC state estimator against attacks generated using a linear model, measurement residues were introduced. The difference between the measurement residues was leveraged to detect malicious data injection attacks. The detection method was tested on larger bus systems, and was shown to have diminished sensitivity as the system got larger due to sensor noise introduced by larger measurements, which outweighed the residues generated by smaller measurements that were being attacked.

To improve detectability for larger bus cases, a method of grouping the branch residues according to their connection to network buses was developed. This method has shown improved detectability for larger bus cases. Like the method used in [4], a few attacks were difficult to detect even at higher attack injection levels. PMU was incorporated, and was shown to not only detect the attacks at lower attack injection levels, but also provide attack localization. If measurements have been tampered with, then the PMUs placed at the bus nearby would detect the abnormality.

Future work could include the detection of attacks that were difficult to detect without relying on the PMUs or development of an algorithm that distinguishes residual distributions from ordinary bad data.

REFERENCES

[1] A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 1999.

[2] Y. Liu, M. K. Reiter, and P. Ning, “False data injection attacks against state estimation in electric power grids," in *Proc. of the 16th ACM Conference on Computer and Communications Security,* 2009.

[3] A. Teixeira, G. Dán, H. Sandberg, and K. H. Johansson, “Cyber security study of a SCADA energy management system: Stealthy deception attacks on the state estimator,” in *18th IFAC World Congress, Milan, Italy,* 2011.

[4] W. Niemira, R. Bobba, P. Sauer, and W. Sanders, “Malicious data detection in state estimation leveraging system losses & estimation of perturbed parameters,” *IEEE SmartGridComm*, 2013.

[5] R. Christie, “Power systems test case archive.” Univ. Washington. [Online]. Available: https://www.ee.washington.edu/research/pstca/, 1993.

[6] R. Zimmerman and C. Murillo-Sánchez, “Matpower user's manual.” [Online]. Available: http://http://www.pserc.cornell.edu/matpower/

[7] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, “Detecting false data injection attacks on dc state estimation,” *Secure Control Systems Workshop, CPSWeek* (Apr. 2010).

[8] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*, 2nd *ed*. John Wiley and Sons, 1996.

[9] J. Kim, L. Tong, and R. J. Thomas, “Dynamic attacks on power systems economic dispatch,” *2014 Asilomar Conference on Signals, Systems, and Computers*, November 2014.

[10] L. Jia, R. J. Thomas, and L. Tong, “Impacts of malicious Data on real-time price of electricity market operations,” *Hawaii Intl. Conf. on System Sciences,* Jan, 2012.

[11] Z. Wang, A. Scaglione and R. Thomas, “Electricalcentrality measures for electric power grid vulnerability analysis,” *Proc. 49th IEEE Conf. Decision Control*, 2010.