TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Detecting False Data Injection Attacks Against Smart Grid Wide Area Monitoring Systems

PhD THESIS

submitted in partial fulfillment of the requirements for the degree of

Doctor of Technical Sciences

by

Dipl.-Ing. Sarita Paudel
Registration Number 01029083

to the Faculty of Electrical Engineering and Information Technology

at the TU Wien

Advisor: Univ. Prof. Dipl.-Ing. Dr-Ing. Tanja Zseby
Second advisor: Dr. Paul Smith (Austrian Institute of Technology)

Reviewers:
Dr. Kieran McLaughlin. Queen's University Belfast, United Kingdom.
Univ. Prof. Dr. Wolfgang Kastner. TU Wien, Austria.

Vienna, 25th May, 2021

Declaration of Authorship

Dipl.-Ing. Sarita Paudel

I hereby declare that this thesis is in accordance with the Code of Conduct rules for good scientific practice (in the current version of the respective newsletter of the TU Wien). In particular it was made without the unauthorized assistance of third parties and without the use of other than the specified aids. Data and concepts directly or indirectly acquired from other sources are marked with the source. The work has not been submitted in the same or in a similar form to any other academic institutions.

Vienna, 25th May, 2021

Acknowledgements

I would like to thank my supervisors Prof. Tanja Zseby and Dr. Paul Smith for their consistent support and guidance during the running of this project. Without your assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished. I would like to thank you very much for your support and understanding over these past four years.

The research in this dissertation is supported by the Austrian Research Promotion Agency (FFG) dissertation project AdA (Adaptive Anomaly Detection in Smart Grids), project number-854296.

Getting through my dissertation required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past three years. I cannot begin to express my gratitude and appreciation for their friendship.

I would like to thank my family and friends for supporting me during the compilation of this dissertation. My special thanks goes to my parents and parents-in-law, who have always been there, at all good and bad times to support me and my plans. My husband Deepak has always supported me and his affectionate support has shaped my career more than anything else. Last but not least, I would like to thank my daughter Sofia. Without her tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

Kurzfassung

Wide Area Monitoring Systems (WAMS) werden verwendet, um Synchrophasordaten an verschiedenen Standorten zu messen und den Betreibern ein nahezu Echtzeitbild des Geschehens im System zu geben. Da Stromnetze kritische Infrastrukturen sind, sind WAMS verlockende Ziele für alle Arten von Angreifern, einschließlich gut organisierter und motivierter Gegner wie Terroristengruppen oder verfeindete Staaten.

Wir möchten die Sicherheit des Stromversorgungssystems verbessern, indem wir FDI-Angriffe (False Data Injection) gegen WAMS erkennen. Durch die Einführung geeigneter statistischer Methoden wollen wir die Leistungsfähigkeit bei der Erkennung von Anomalien verbessern und gleichzeitig die Auswirkungen von Angriffen auf die Zustandsschätzung (State Estimation - SE) abschwächen. Wir analysieren zunächst Smart-Grid-Bedrohungen mit Hilfe von Angriffsbäumen und formulieren ein Modell, um verschiedene FDI-Angriffe darstellen zu können. Dann untersuchen wir verschiedene Anomalieerkennungsmethoden hinsichtlich ihrer Fähigkeit, FDIs zu erkennen. Um zu untersuchen, wie solche Angriffe erkannt werden können, verwenden wir Methoden zur Erkennung von Zustandsschätzungen (SE) und fehlerhaften Daten (Bad Data - BD). Danach untersuchen wir die Eignung einer statischen SE-Methode für gewichtete kleinste Quadrate (Weighted Least Squares - WLS) und einer rekursiven SE-Methode für Kalman Filter (KFs). Anschließend untersuchen wir die Eignung von Residuen aus WLS und DKF zur Erkennung fehlerhafter Messungen. Drei Verfahren, einfache Pre-fit Residuen, L2-Norm- und normalisierte Residuen-basierte Verfahren, werden zum Erfassen von fehlerhaften Messungen verwendet. Dann untersuchen wir die Eignung verschiedener einfacher statistischer Methoden zur Erkennung von Anomalien, mittlere absolute Abweichung vom Median (MAD), Kullback-Leibler-Divergenz (KLD) und kumulative Summe (CUSUM). Die in den verschiedenen Experimenten verwendeten Daten stammen von Phasor Measurement Units (PMUs) aus einem realen Stromnetz. Desweiteren untersuchen wir die Verbesserung der Anomalieerkennung durch eine Kombination von Methoden mit einer gewichteten Abstimmung. Schließlich wird eine Analyse der Minderung der Auswirkungen von Angriffen auf die Zustandsschätzung durch Ersetzen der detektierten fehlerhaften Daten durchgeführt.

Die Anwendungen für die Forschungsergebnisse sind vielfältig: Die Überwachung und Steuerung von Smart Grids kann von der im Rahmen unserer Forschung durchgeführten Bedrohungsanalyse profitieren. Darüber hinaus können die verschiedenen statistischen Methoden, die in den Experimenten untersucht und verwendet wurden, bei der Identifizierung des geeigneten Analysewerkzeugs für die Erkennung von Anomalien helfen. Unsere Untersuchungen zeigen, dass für einen vertrauenswürdigen Mechanismus zur Erkennung von Anomalien eine Kombination verschiedener Methoden erforderlich ist.

Abstract

Wide area monitoring systems (WAMSs) are used to measure synchrophasor data at different locations and give operators a near-real-time picture of what is happening in the system. Since power grids are critical infrastructures, WAMSs are tempting targets for all kinds of attackers, including well-organized and motivated adversaries such as terrorist groups or adversarial nation states. Attacks on WAMSs can trigger wrong decisions and severely impact grid stability, overall power supply, and physical devices.

We aim to improve power system security by detecting false data injection (FDI) attacks against WAMSs. Through adoption of adequate statistical methods, we aim to enhance anomaly detection performance and at the same time mitigate the effects of attacks on state estimation (SE). We first analyze smart grid threats with the use of attack trees and formulate a model to express different FDI attacks. Then we investigate different anomaly detection methods with regard to their ability to detect FDIs. In order to investigate how such attacks can be detected, we first look into SE and bad data (BD) detection methods. We then investigate the suitability of a static SE method based on weighted least squares (WLS) and a recursive SE method based on Kalman filters (KFs), and analyse the suitability of using residuals from WLS and DKF for detecting bad measurements. Three methods, i.e., plain pre-fit residuals, L2-norm and normalized residuals based methods are used for detecting bad measurements. We then investigate the suitability of different lightweight statistical anomaly detection methods median absolute deviation (MAD), Kullback-leibler divergence (KLD) and cumulative sum (CUSUM). The data used in the different experiments come from phasor measurement units (PMUs) installed in a real power grid. Further, we investigate improving anomaly detection performance with a combination of methods based on weighted voting. Finally, an analysis of mitigating the effects of attacks on SE by replacing detected BD is conducted.

The impacts of this research are manifold: smart grid monitoring and control can benefit from the threat analysis conducted as part of our research. Additionally, all the different statistical methods investigated and utilised in the experiments can help in the identification of the proper analytical tool for anomaly detection. Last but not least, our research suggests that a combination of different methods are needed for a trustworthy anomaly detection in smart grids.

List of Acronyms

CPS	cyber physical system
ICS	industry control system
EPS	electric power system
SG	smart grid
ICT	information and communication technology
WAMS	wide area monitoring system
SCADA	supervisory control and data acquisition
PMUs	Phasor Measurement Units
SE	state estimation
LSE	linear state estimation
RB	residual-based
BDD	bad data detection
AD	anomaly detection
WLS	weighted least squares
LWLS	linear weighted least squares
KF	Kalman filter
KG	Kalman gain
DKF	discrete Kalman filter
EKF	extended Kalman filter
APT	advanced persistent threat
CC	control center
SA	situation awareness
SE	state estimation
SSE	static state estimation
DSE	dynamic state estimation
IEDs	intelligent electronic devices
PDCs	phasor data concentrators
SDCs	super data concentrators
PGWs	phasor gateways
FDI	false data injection
TSO	transmission system organization
DSO	distributed system organization

MAD	median absolute deviation
KLD	Kullback-Leibler divergence
CUSUM	cumulative sum
RMS	root mean square
GPS	positioning system
TCP	transmission control protocol
UDP	user datagram protocol
GOOSE	generic object oriented events
GSE	generic substation events
SMV	sampled measured values
MMS	manufacturing message specification
GSSE	generic substation status event
HV	high voltage
MV	medium voltage
LV	low voltage
DG	distributed generation
NERC	North American Electric Reliability Corporation
CIP	Critical Infrastructure Protection
NIST	National Institute of Standards and Technology
NESCOR	National Electric Sector Cybersecurity Organization Resource
NASPI	North American SynchroPhasor Initiative
DCS	distributed control system
PLC	programmable logic controllers
RTU	remote terminal unit
BGP	border gateway protocol
AMI	advanced metering infrastructures
BA	benign anomaly
MA	malicious anomaly
ADN	active distribution network
p.u.	per unit

Contents

CHAPTER 1

Introduction

Notice of adoption from previous publications in Chapter 1
Parts of the contents of this chapter have been published in the following papers:

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. 4th International Symposium for ICS and SCADA Cyber Security Research*, 2016*

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM*

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*, pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
Text from motivation are based on the work done in [129], [130], [132] and [133]. A part of text while describing research activities and performance metrics is based on the work done in [129], [130] and [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

1.1 Motivation

Integrated electric power system (EPS) networks consist of several components for power generation and distribution. Natural disasters (e.g., earthquakes, hurricanes or flooding) can cause malfunctions and damage to power systems equipment or installations like power lines, power units and substations [26]. Failures of EPS components can lead to critical states in grid operation, endanger devices, and even cause blackouts.

The integration of information and communication technology (ICT) into EPSs supports enhanced monitoring and control capabilities. An important application of ICT in this context is to enable situational awareness with respect to the EPS's state. Situational awareness [72] "relates to the perception of changes in environment with respect to time (or space), projection of the status after changes". Sharing of information is one aspect of situational awareness.

Smart grids (SGs) improve the efficiency of the traditional power grids by adopting modern communication and control technologies. Different devices from various vendors are connected in multiple layers, and communicate via proprietary or open standard protocols. Though integration of ICT helps power grids to be smarter, it introduces security issues.

To enable situational awareness, a wide area monitoring system (WAMS) can be deployed that includes distributed sensors, which measure power system state, and communication technologies that enable the transmission of this state to a control center (CC). The data that is collected using a WAMS can be used to support real-time decision making by operators, e.g., in order to respond to a fault, and to facilitate grid planning.

An important technology that has emerged in recent years is phasor measurement units (PMUs). These devices can measure the power system state – e.g., voltage, power, and phase angle – at very high frequencies (50Hz), and can be used to support real-time situational awareness. The data from PMUs can be transmitted over a wide-area network using specialized protocols, such as IEC 61850 [63], to a CC. One important use of the data that has been collected by PMUs is to estimate the state of the power system, e.g., where there is a lack of monitoring capability or to mitigate measurement noise. There are several approaches to state estimation (SE).

WAMSs improve situational awareness in SGs and provide information to prevent critical incidents [183]. They also support planning and optimization of grid operation. WAMSs collect clock-synchronized measurement values from widely distributed PMUs, and

provide input to various applications in the grid, e.g., as direct input to control functions, feedback in control loops, or stored for future planning and post-incident analysis. The measurement values are processed and decisions regarding appropriate grid control actions are made in the CC. As a consequence, utilities are affected by the decisions in the CC.

WAMSs constitute a suite of different solutions consisting of various combinations of components, such as intelligent electronic devices (IEDs), PMUs, phasor data concentrators (PDCs and super PDCs), communication equipment, applications, visualization tools and many more [80]. Phasor gateways (PGWs) offer a publish-subscribe framework for sharing phasor measurements among different utilities or CCs [25]. Therefore, WAMSs integrate many different components in different topological settings [156], and all devices can be entry points for attacks. Global positioning system (GPS) synchronized PMUs in the power grid provide accurate and time-synchronized measurements. They are required for fine grained control and monitoring applications [49] for SE, fault detection, and voltage and frequency stability.

PMUs help to secure operation, but various cyber attacks are possible to compromise PMU devices, PMU measurements, communication protocols, and applications used for monitoring, protection and control. For example, voltage manipulation attacks can cause over voltage and under voltage in a power system. We assume all voltage magnitudes are close to 1 per unit (p.u.). According to the European Standard EN 50260 [51], the acceptable voltage fluctuation in normal operation is between 0.9 p.u. to 1.1 p.u.. A voltage higher than 1.1 p.u. is considered as over voltage and less than 0.9 p.u. is considered as under voltage. SE in real-time is often used to monitor the grid and achieve situational awareness [33]. Increasingly, this task is realized using Kalman filters, in order to account for variations in sensor measurements [139, 113].

Since power grids are critical infrastructures, WAMS is a tempting target for all kinds of attackers, including well-organized and motivated adversaries, such as terrorist groups or adversarial nation states. Such groups posses sufficient resources to launch sophisticated attacks. With the introduction of new technology, there is a corresponding increase in risk from cyber attacks. An example is the 2015 cyber attack on the Ukraine power grid [96] that caused a regional blackout.

Attackers can perform malicious cyber attacks using existing vulnerabilities in SG devices, hardware and software, or the communication channels. Different devices, communication channels between the devices, hardware, software, and many more components in a SG might be compromised to perform successful cyber attacks. Attackers can also gather information by sniffing communication networks and use the information for attack preparation. Data integrity attacks on WAMSs can lead to incorrect control decisions and actions.

Cyber attacks to a WAMS could have significant consequences – in the short-term, if they are used to support fault isolation, incorrect switching decisions could be made; and in the longer-term, if the measurements derived from them are used to support grid planning, sub-optimal and expensive investment strategies could be employed. For

instance, the consequences of over voltage and under voltage include failures and damage, which endanger grid operation.

Information about threats, vulnerabilities and indicators of compromise is a valuable resource for system administrators of complex and interconnected ICT systems. Besides detecting an attack, the detection delay is important. The faster an attack is detected, the faster can countermeasures be put in place or decisions can be delayed until the data is verified.

An important class of attacks to a WAMS are false data injection (FDI) attacks, wherein an attacker manipulates data (e.g., voltage and power measurements) to misdirect the processes and systems that use it. Moreover, researchers have investigated a class of FDI attacks that are unobservable to algorithms that aim to detect bad data, normally caused by measurement noise. It has been shown that these unobservable attacks can result in significant consequences to an EPS [178, 168].

Application of countermeasures to the detected anomalies can help in maintaining the correctness of a system state. Replacement of detected bad measurements can support in mitigating the effect of attacks on monitoring and control applications like SE. As SE for grid operation is critical, trustworthy estimated states need to be sent to the operators in a CC.

With the increasing importance of securing SGs against the growing number and evolving cyber attacks, awareness of security issues and countermeasures against attempted disruption of the SGs has gained the attention of the research community. It is very important to improve the security solutions developed for power systems. Thus in this research, we aim to develop a model for improving the security of power systems.

1.2 Research Questions

The objective of this research is to improve the security of a power system. We investigate the cyber threats, effect and countermeasures of FDI attacks in a WAMS. Our investigation with the objective of improving power system security sets the following research questions:

RQ 1: What are possible attacks on wide area monitoring systems (WAMSs)? WAMSs consist of multiple devices with different interfaces and therefore can provide many entry points for different types of attacks. In this work, we analyse existing literature and investigate which vulnerabilities and attack entry points exist in a WAMS, and provide a comprehensive overview of vulnerabilities and attack possibilities. We then focus our work on FDI attacks.

For FDI attacks, additional challenges arise from the fact that WAMSs usually deploy SE methods and bad data detection (BDD) algorithms. Bad data algorithms interact with attacks and attack detection mechanisms because the algorithms use residuals from the SE for detecting bad data and the detection of bad data can be an indicator of an

attack, so alarms should be triggered if needed. Therefore also the effects of FDI attacks on such algorithms have to be investigated. This research question has following two sub-questions:

- **RQ 1.1**: How can an attacker cause false data injection attacks in a wide area monitoring system?
- **RQ 1.2**: How can multiple different false data injection attack forms be expressed in one comprehensive attack model?

The research questions **RQ 1.1** and **RQ 1.2** will be addressed in Chapter 5. We will further split up the research questions **RQ 1.1** and **RQ 1.2** into different sub-questions in the corresponding Chapters.

RQ 2: How can one detect false data injection (FDI) attacks in WAMSs data?
Many different methods exist to detect deviations from normal behavior. We need to investigate which methods are suitable for which types of attacks regarding detection performance and detection delay. Since detection methods are often specialized to detect specific changes it is unlikely to find one detection method that performs well for all attacks. Therefore, we also investigate a combination of methods. Additionally, the interaction of attack detection with existing algorithms (SE, BDD) and possibilities for attackers to influence such algorithms need to be analyzed. This research question has three sub-questions as follows:

- **RQ 2.1**: To what extent can residual-based bad data detection methods detect different FDI attacks?
- **RQ 2.2**: Can stealthy attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100] be detected by residual-based methods?
- **RQ 2.3**: Is it possible to detect the injected attacks with lightweight statistical methods?

The research question **RQ 2.1** will be addressed in Chapter 7. BDD methods are used to check for errors in the measurements that would influence SE or to detect actively manipulated data. In Chapter 7, we check if the attacks would raise any alarm at all (i.e. detect at least one anomalous data point), how fast the methods detect that something is wrong and then how many of the manipulated data points are recognized as anomalies.

The research question **RQ 2.2** will be addressed in Chapter 8. We want to experiment if an attack of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ (as described in [100]) is undetected using residuals of linear weighted least squares (LWLS) and discrete Kalman filter (DKF) for SE.

The research question **RQ 2.3** will be addressed in Chapter 9. In order to prevent stealthy attacks, we propose using multiple statistical anomaly detection methods in an overall effort to achieve effective detection.

We will split the research questions **RQ 2.1**, **RQ 2.2** and **RQ 2.3** into different sub-questions in the corresponding Chapters.

RQ 3: How can the effects of FDI attacks on state estimation (SE) be mitigated?
If due to an FDI attack, manipulated values are used for SE, then the estimated state can differ from the real state. The fake states can lead to wrong control decisions which can cause impact to devices and human lives. To avoid such impacts, we investigate different methods to replace bad data in order to preserve the SE. This research question has only one sub-question as follows:

- **RQ 3.1**: To what extent can the effects of FDI attacks on state estimation in electric power systems (EPSs) be mitigated by replacing detected anomalies with values derived from past data?

The research question **RQ 3.1** will be addressed in Chapter 10. We will split the research questions **RQ 3.1** into different sub-questions in the corresponding Chapter.

The goal of this thesis is to answer the research questions formulated above.

1.3 Methodology and Approach

An overview of our research approach is depicted in Fig. 1.1. Our research approach consists of four phases that are described in following sections. The research methodology carried out in our study is depicted in Fig. (1.2). The activities shown in the research methodology are mainly from the second, third and fourth phases of the research approach.

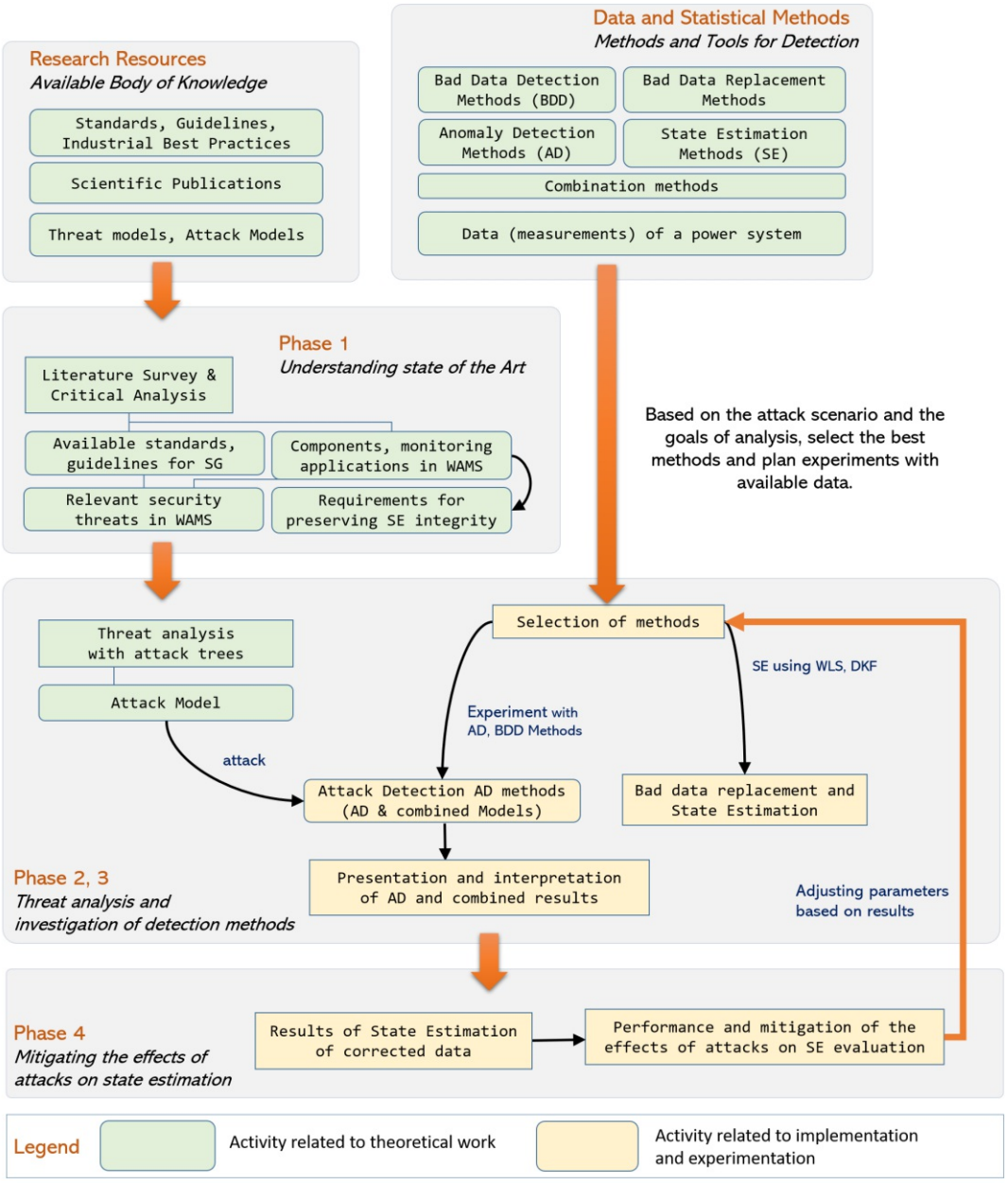


Figure 1.1: An overview of the research approach.

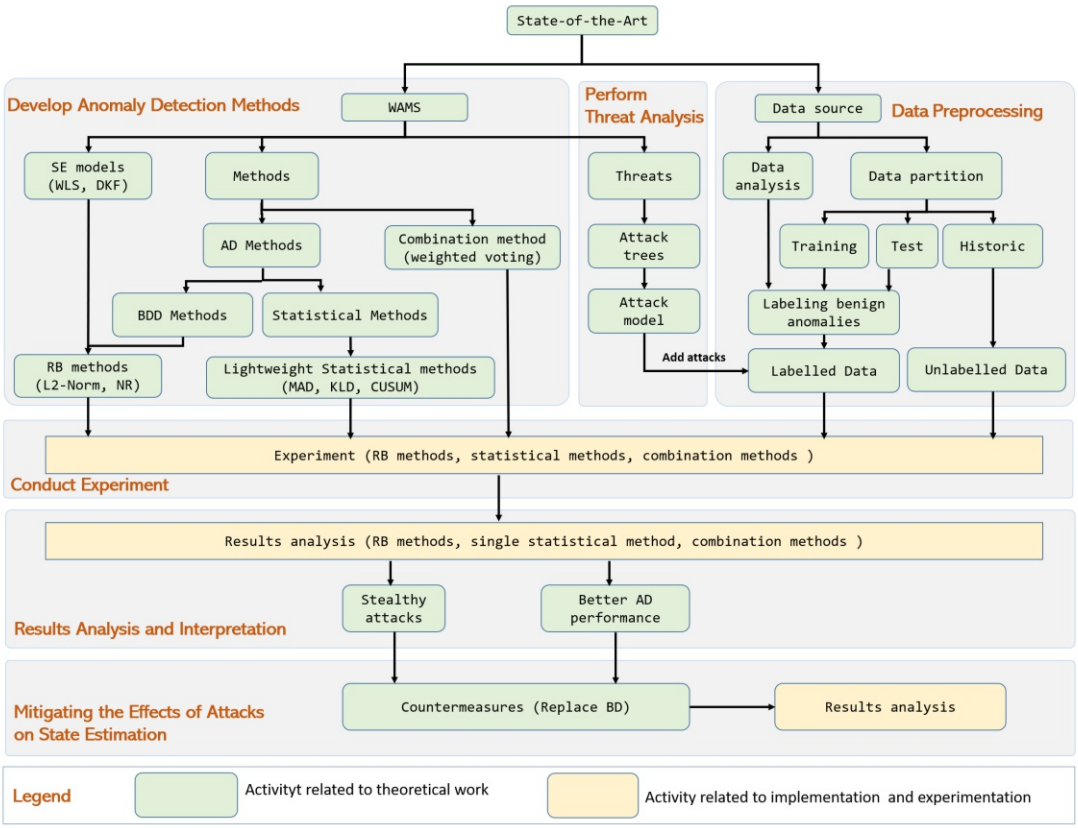


Figure 1.2: Major activities in our research.

1.3.1 Phase 1: Understanding State of the Art

The first phase includes a thorough understanding of theoretical concepts and preliminaries for conducting the proposed research. As a first step of this phase, we conduct a literature survey for better understanding the state-of-the-art. The undirected connection between boxes in the Fig. 1.1 means knowledge gained from an upper box is used in a lower box. Preliminaries of a power system like phasor measurements, WAMSs topologies, and communication standards, safety limits etc. are studied. At the same time an overview of available standards, guidelines and their support for improving security of SGs is abstracted and relevant security threats in WAMS are identified. A study of WAMS key components, applications used for monitoring and control functions is carried out which helps us better understand the decisions and control actions based on the decisions in a CC.

We identify vulnerabilities and potential attacks on the WAMSs based on the literature survey. This is presented in detail in Chapters 2 and 3.

1.3.2 Phase 2: Threat Analysis

The second phase of our research consists of a threat analysis. Details of the second phase can be found in Chapters 4 and 5.

After identifying the threats in WAMS, we perform a threat analysis that helps us in understanding the nature of attacks. We investigate advanced persistent threats (APTs), and how physical and cyber attacks are combined for launching an attack. An attacker can compromise software/hardware, or both. A detailed study of attack vectors compromising field devices, like PMU, compromising a software which is based on input data, compromising the communication between the components of WAMS etc. is carried out.

After having gained knowledge from the study of the attack vectors in the wide area network in SGs, we further investigate attack scenarios using the attack vectors. Our study continues on the methods for presenting vulnerabilities, attack techniques and the attackers' goal. We model the dependencies and building blocks of APTs on WAMSs using attack trees. The WAMSs infrastructure includes classical IT components, clock synchronization, and data collection and aggregation points such as PDCs [8, 163]. We consider the entire WAMS infrastructure. Since SGs are cyber-physical systems, we consider physical perturbations, in addition to cyber attacks in our models. We develop attack trees to analyze and assess the different paths an attacker can take. Generic attack trees and specific attack trees are developed for different attack scenarios.

After understanding the nature of attacks, we defined specific FDI attack scenarios as use cases for our research on detection methods and developed a comprehensive attack model to specify different forms of FDI attacks. The model describes a generic way how different attacks on PMU data can be expressed and generates types of false data injection attacks. Attack parameters and the types of attacks are discussed with specific values.

1.3.3 Phase 3: Investigation of Detection Methods

The third phase of our research consists of an investigation of detection methods. Details of the third phase can be found in Chapters 7, 8 and 9.

From the literature, we selected suitable data sets for our experiments. Our study aims to develop anomaly detection (AD) methods that could be deployed online and can help to secure a real power system. We stick to our goal and use data from a real power system in our experiments, which help us in understanding the behavior of a power system. We use a data set from EPFL that is a representative network of a SG, an active distribution network deployed on the EPFL campus [47].

Analysis of the real power system measurements (EPFL PMU measurements) helps us in understanding the characteristics of the measurements. We use MATLAB, Python and R for the analysis.

We partition the data into historical, training and test data. Historical data is only used for building a reference histogram for one AD method (Kullback-Leibler divergence). Training data is used for setting thresholds for the BDD and AD methods. We then proceed to BDD and AD experiments. Manipulated test data is used for BDD and AD.

Our SE experiment is carried out using two methods, a DKF and a LWLS. We analyse the influence that attacks have on SE. We then look at BDD methods, which are often integrated into SE and check to which extent those methods are useful to detect attacks.

Residuals from the SE are used for BDD. We use residual-based (RB) BDD methods, L2-norm and normalized residuals. Since some attacks are not detected by BDD methods, we investigate alternative methods for AD. Similarly, we investigate selected lightweight statistical AD methods, median absolute deviation (MAD), Kullback-Leibler divergence (KLD) and cumulative sum (CUSUM). These methods are based on different features of data. MAD checks individual measurement values, KLD compares distribution of measurements and CUSUM focuses on changes in the mean over time.

We then study the applicability of BDD methods and AD methods in our use case and analyze the combination of methods in order to improve the AD performance. For having a final AD decision, we combine results from the lightweight statistical AD methods. A combination method, weighted voting, is used for combining the results. The final decision is taken into consideration for mitigating the effects of attacks on SE. The results from the mitigation of the effects of attacks on SE experiment shows how the effects of attacks on SE is maintained.

In order to assess a detection method's performance, we compare the original labels of the manipulated test data with the predicted labels from the method. From this we derive: a) true positives (TP), i.e. anomaly correctly identified as an anomaly, b) true negatives (TN), i.e. normal data points correctly classified as normal, c) false positives (FP), i.e. how many normal data points are classified as anomalies and d) false negatives (FN), i.e. how many anomalies we miss (anomalies classified as normal). A confusion matrix visualizes all of the above mentioned detection performance in a table as shown in Tab. 1.1.

Detected as	**Labeled as**	
	Non-malicious	Malicious
Non-malicious	True negative	False negative
Malicious	False positive	True positive

Table 1.1: Confusion matrix

From the TP, TN, FP, FN we then calculate recall or true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), accuracy and precision as follows:

$$Recall = \frac{TP}{TP + FN} \tag{1.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{1.2}$$

$$TNR = \frac{TN}{TN + FP} \tag{1.3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.4}$$

$$Precision = \frac{TP}{TP + FP} \tag{1.5}$$

We then determine the detection delay and show how fast an attack is detected by calculating the difference between the real start of the attack k_s and the k_a measurement when the first data point of the attack was classified as an anomaly.

1.3.4 Phase 4: Mitigating the Effects of Attacks on SE

The fourth phase of our research consists of mitigation of the effects of attacks on SE. Details of the fourth phase can be found in Chapter 10.

In this phase, an evaluation of mitigation of the effects of attacks on SE is performed. In a first step, we investigate on the methods for mitigating the effects of attacks on SE. In a second step, we investigate the applicability of the bad data replacement methods in our use case, then in the last step analyze mitigation of the effects of attacks on SE. For this, we analyze SE with and without AD methods and the replacement of detected anomalous data before sending them to SE.

1.4 Contribution

Here we describe the scientific contribution of this work and give an overview of the methods used for our investigations. Table 1.2 illustrates an overview of the research questions, methods and the contributions made in different chapters of this thesis. Different methods are used to answer the research questions of this research. Details of the methods and the contributions are available in different sections as shown in the last column of the Tab. 1.2. Here, each of the contribution is briefly described as follows.

1.4.1 Threat Analysis and FDI Attack Model

Threat Analysis: We develop attack vectors, attack scenarios and attack trees in order to derive vulnerabilities and attack scenarios in WAMS (see Tab. 1.2). For the attack vectors, we provide how an attacker can use different attack entry points to compromise hardware or software on compromising the physical device and communication. We develop generic attack trees for compromising a device and develop specific attack trees

for causing a blackout and manipulating sensor data (e.g., phase angle). In contrast to existing research, we do a threat analysis on WAMS, which could be considered in the transmission system organization (TSO) and the distributed system organization (DSO).

False Data Injection Attack Model: We design false data injection attack types that poison the measurement data without exceeding any safety limits, as such attacks can remain stealthy and result in wrong decisions in the CC. In addition, attackers can change the states of the power grid by modifying the sensor readings (or injecting false data in sensor measurements). We develop a false data injection (FDI) attack model for generating different attack types: randomizing signal, adding constant offset, adding incremental constant or random offsets to signal. In contrast to existing work, we specify different types of attacks in a single FDI attack model.

1.4.2 State Estimation Methods Investigation

For investigating linear weighted least squares (LWLS) and Kalman filters (KF) (as shown in Tab. 1.2), we implement them in MATLAB. For KF, we generate an example to exemplify the influence of parameters settings and dependencies for estimating states from observed measurements, modify the example and adopt it to our use case. We apply the modified KF for estimating states based on real PMU data. We apply the LWLS and the DKF in our use case with real data and find that the DKF better represents measurements changes in residuals.

Table 1.2: Methods and contribution for answering the research questions; Sec. = section (sub-research questions of research questions are in brackets; Exp. = experiment).

Research Question	**Method**		**Contribution**	**Sec.**
What are possible attacks on WAMSs? (**RQ 1.1**, **RQ 1.2**)	Attack vectors		- Threat analysis particularly for WAMS architecture - FDI attack model for generating different types of attacks	5.1.1
	Attack scenarios			5.1.2
	Attack trees			5.1.3
	Attack model			5.3
How can one detect false data injection (FDI) attacks in WAMSs data? (**RQ 2.1**, **RQ 2.2**, **RQ 2.3**)	Weighted least squares applied to real data		- Detection method based on DKF residuals for stealthy attack of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100] (where LWLS residuals are used)	4.1
	Discrete Kalman filter example model for parameter investigation			4.2.2
	Discrete Kalman filter applied to the real measurement			4.2.1
	Exp. with real data	Plain residuals	- RB detection method applied to real data (previously used for simulated data in [139]) - An adversary's techniques for circumventing RB BDD detection - Application of lightweight statistical detection methods - Weighted voting scheme for combining statistical methods (previously used for combining machine learning methods in [101]) with higher precision	7.1.1
		L2-norm residuals		7.1.2
		Normalized residuals		7.1.3
		Median absolute deviation (MAD)		9.1.2
		Kullback-Leibler divergence (KLD)		9.1.3
		Cumulative sum (CUSUM)		9.1.4
		Weighted voting		9.4.1
How can SE integrity be preserved in the presence of FDI attacks? (**RQ 3.1**)	Replace bad data by prediction		- A method for mitigating the effects of attacks on voltage SE using Kalman filter model	10.1.4
	Voltage estimation with corrected values			

1.4.3 Anomaly Detection

Bad Data Detection: We investigate the suitability of using residuals from LWLS and a DKF for detecting bad measurements. After the investigation on the suitability of plain residuals, L2-norm residuals and normalized residuals for detecting bad measurements, we implement the methods in MATLAB. We reproduce an AD approach that has been proposed in [139] which works with simulation data. We adopt the AD method and apply it in our use case with real data. After that we analyze AD results of the promising methods in literature; the L2-norm and normalized residuals methods with our real data and find that some attacks cannot be detected. In contrast to existing research, we implemented stealthy attacks as defined by Liu et al. in [100] for WLS also for pre-fit residuals of KFs. We found that stealthy attack of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100] using residuals of LWLS does not remain stealthy using residuals of DKF.

Anomaly Detection: In SG (critical infrastructure setup), it is beneficial to rapidly detect attacks with the few computational resources. Further, it is beneficial that a human operator is able to understand the results from the detection methods and have them readily explainable (in contrast to, for example, some machine learning and deep learning methods where explainability remains an open issue). We investigate the suitability of three lightweight statistical methods: MAD, KLD and CUSUM for detecting FDI attacks. We analyze AD results performance of the methods KLD, CUSUM and MAD on different attack types and investigate the attack parameter which affects the AD.

Methods Combination: We investigate whether a combination of methods can improve the detection performance. We investigate the suitability of weighted voting method for combining the AD methods. We analyze AD performance of the weighted voting method for combination. In contrast to existing work, we apply the combination of lightweight statistical methods in SGs. We use the weighted voting scheme in [101] previously used for combining machine learning methods to combine statistical methods. An overview of contributions made in AD is shown in Tab. 1.2.

1.4.4 Mitigating the Effects of Attacks on SE Analysis

We investigate methods for correcting bad (anomalous) measurement. We investigate the suitability of bad measurement replacement by the prediction of DKF. We analyze SE results after replacing anomalous measurement by predicted value of DKF in attack types. In contrast to existing work (e.g., in [85, 84]), we use the prediction of Kalman filter for mitigating the effects of attacks on SE and show how the effects of attacks on voltage estimation is mitigated based on the AD results of the statistical methods (see Tab. 1.2). Our approach covers the future work in [85]).

1.5 Structure

The structure of this thesis is as follows:

Chapter 2 provides an overview of preliminary knowledge on SGs for conducting this dissertation research. Phasor measurements, WAMSs topologies and communication standards, safety limits of measurements (e.g., voltage, frequency) based on existing standard are the main building blocks of this work. Understanding vulnerabilities and potential attacks on SGs is an important step to carry out this research.

In Chapter 3, we present state-of-the-art on BDD, stealthy attacks, usage of SE for detecting attacks, applicability of data and events aggregation for detecting the attack against key components of a WAMS. Further, some relevant existing AD techniques for detecting anomalies in SGs are presented in this chapter.

Chapter 4 presents SE methods in the context of power systems. In particular, a static SE method weighted least squares (WLS) and a recursive SE method Kalman filters (KFs) are discussed. Then, we present our use case of SE with experimental results.

In Chapter 5, we present the threat analysis and FDI attacks model. A threat analysis is carried out for understanding potential attack vectors of WAMSs and developing attack trees. Attack vectors (e.g., compromising hardware, software, communication) against key components (e.g., PMU, PDC, Gateway) of a WAMS, how an attacker can reach his/her ultimate goal (e.g., compromise field device, cause blackout) and achieve intermediate goals are presented using attack trees. Moreover, we develop a FDI attack model, consisting of generated types of FDI attacks.

Chapter 6 presents an analysis of the real PMU data we use. Analysis of voltage, frequency and phase angle helps us in selecting a representative set of data for our experiment. This chapter presents the analysis with figures and illustrates the selected historic, training and test data. Further, we present data preprocessing.

Chapter 7 focuses on RB BDD methods. To this end, we look at plain pre-fit residuals, L2-norm, and normalized residuals. First, we discuss the RB BDD methods; second we present the experimental setup of how thresholds are defined for plain pre-fit residuals based method, L2-norm, normalized residuals methods; and last demonstrate experimental results.

Chapter 8 presents FDI attacks against SE and stealthy attacks from the literature. We present FDI attacks in voltage and current measurements, and the FDI attacks' effect on SE using LWLS and DKF. Theoretical discussion is followed by preliminary experimental results and a discussion on identification of attacks using residuals. Moreover, we present stealthy attacks on voltage and current measurements.

Chapter 9 focuses on AD using lightweight statistical methods. We present an AD model developed in this work. The model executes RB BDD methods and different lightweight statistical methods for detecting anomalies. First, we briefly present the model and

present lightweight statistical methods for AD. For the lightweight statistical methods we use - MAD, KLD and CUSUM. Second we present the experimental setup of how thresholds are defined for MAD, KLD and CUSUM; third we demonstrate experimental results of single methods and then present the findings based on the results; fourth ROC curves of the methods are presented. Then we focus on improving AD performance using a combination of methods. To this end, we use weighted voting for combining AD results of the statistical methods. Further, we present the combined results, and show the results analysis - how the AD performance is improved using the weighted voting method.

Chapter 10 focuses on mitigating the effects of attacks on SE. In this chapter, first we introduce proposed approach for mitigating the effects of attacks on SE. We then show how the SE can be effected by FDI attacks. Finally, we show how AD and bad data replacement support mitigating the effects of attacks and then present experimental results for mitigating voltage estimation.

In Chapter 11, we summarize and conclude the dissertation research. The main outcomes are summarized and an overview of future work is presented.

1.6 Support

The research in this dissertation is supported by the Austrian Research Promotion Agency (FFG)[1] dissertation project AdA (Adaptive Anomaly Detection in Smart Grids)[2], project number-854296.

This project focuses on providing novel methods for efficient anomaly detection in the SGs, both relevant domains of anomaly detection and SG security.

[1] https://www.ffg.at/
[2] https://projekte.ffg.at/projekt/1359933

CHAPTER 2

Background

Notice of adoption from previous publications in Chapter 2
Parts of the contents of this chapter have been published in the following papers:

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. 4th International Symposium for ICS and SCADA Cyber Security Research*, 2016*

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM*

[131] S. Paudel, P. Smith, and T. Zseby. Data Attacks in Wide Area Monitoring System. Symposium on Innovative Smart Grid Cybersecurity Solutions*, 2017*

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*, pages 16:1–16:10, 2018*

Explanation text, on what parts were adopted from previous publications:
The voltage phasor measurements described in this chapter are based on the work done in [132]. The wide area monitoring structure described in this chapter is based on the work done in [129] and [131]. Some part of safety limits described in this chapter is based on [132].

S. Paudel performed theoretical considerations together with all co-authors, the text and figures in the papers were created together by all authors.

In this chapter, we provide background knowledge for conducting our research. We provide an overview of monitoring and control functions and major components of the smart grid. Wide area monitoring system and its components, phasor measurements, and safety limits are addressed.

Wide area monitoring systems (WAMSs) are used to measure synchrophasor data at different locations and give operators a near-real-time picture of what is happening in the system. The measurement data is periodically collected via communication channels to monitor, predict and control the power consumption, and detect any problems in the power grid.

A WAMS provides an essential building block for supervision and control. WAMSs collect clock-synchronized measurement values from distributed phasor measurement units (PMUs), and provide input to various applications in the grid, e.g., as direct input to control functions, or are stored for future planning and post-incident analysis.

2.1 Phasor Measurements

In this section, we introduce the voltage phasor, its synchrophasor representation and conversion of polar voltage to rectangular coordinates. A sinusoidal signal $x(t)$ in a power system can be represented as [5]

$$x(t) = X_{max} \cdot cos(wt + \theta) \tag{2.1}$$

where X_{max} is the amplitude of the wave, w is the angular frequency, and θ is the phase angle at $t = 0$. A *phasor* represents a sinusoidal wave in the form of a complex number, which can be expressed using polar or rectangular coordinates. A phasor of the sinusoidal wave given in Eq. (2.1) can be represented using Eq. (2.2), wherein the underscore ($\underline{X}$) is used to denote a complex number. As proposed by previous work [5, 23], the root mean square (RMS), i.e. $X_{max}/\sqrt{2}$, of the waveform is used for the phasor definition, instead of the amplitude.

$$\underline{X} = (X_{max}/\sqrt{2}) \cdot e^{j\theta} \tag{2.2}$$

A synchrophasor representation of the signal $x(t)$ is the value of $\underline{X}$, where θ is interpreted as the phase angle relative to a synchronized cosine function. The cosine function has a maximum value at $t = 0$ i.e., phase angle is 0 degrees. Figure 2.1 shows the synchronized cosine function for two different phase angles (θ and θ_1). The curves in this figure have X_{max} at different time.

For our investigations, an attacker is assumed to be manipulating voltage phasors that are measured by PMUs. The voltage phasor is defined based on the RMS, and we denote the RMS of the voltage as V. The voltage phasor $\underline{V}$ is defined as

$$\underline{V} = V \cdot e^{j\theta} \tag{2.3}$$

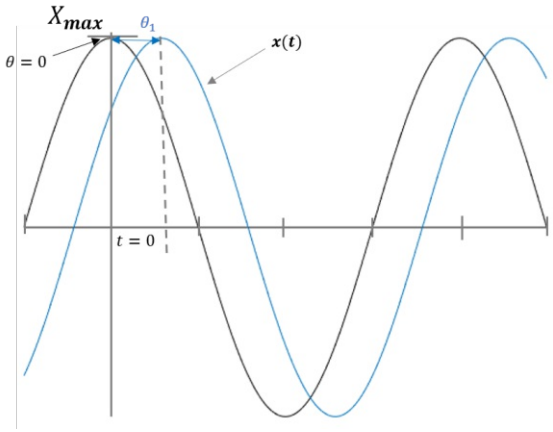


Figure 2.1: Synchrophasor representation

In order to apply linear state estimation using Kalman filters, the voltage phasor measurements need to be converted from polar to rectangular coordinates [176, 149]. So we express the voltage as real part and imaginary part:

$$\underline{V} = V_{re} + jV_{im} \tag{2.4}$$

The real and imaginary part can be calculated as the projection of the polar voltage V to the x-axis (Eq. (2.5)) and the y-axis (Eq. (2.6)). Figure 2.2 depicts the conversion of polar voltage to real and imaginary voltages.

$$V_{re} = V \cdot \cos\theta \tag{2.5}$$

$$V_{im} = V \cdot \sin\theta \tag{2.6}$$

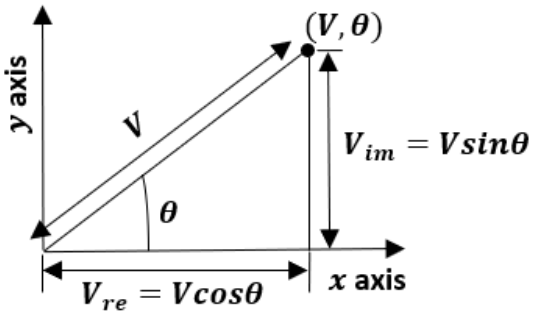


Figure 2.2: Conversion from polar voltage to real and imaginary voltages

Voltage and current phasors can be measured with phasor measurements units. In order to measure the exact phase shift between two signals at different locations, the PMUs need to be clock synchronized. The synchronization is usually done with global positioning system (GPS).

2.2 WAMS Structure and Topologies

Most WAMSs have a hierarchical structure (see in Fig. 2.3), and consist of intelligent electronic devices (IEDs), PMUs, phasor data concentrators (PDCs), super PDCs, phasor gateways (PGWs), and communication facilities to transfer data between these components and a control center (CC) [136]. PMU measurements are time-stamped at the source using the GPS to ensure clock synchronization.

Regional or organizational PDCs gather data from different PMUs, sort the data according to the timestamps, create a combined record and forward the combined records up in the hierarchy.

Distributed PMUs allow accurate clock-synchronized measurements of voltage and current phasors (amplitudes, phase angles) and frequencies. The sensor data from PMUs provide situational awareness in the grid, and are used as input for control decisions. The measurement values are processed and decisions regarding appropriate grid control actions are made in the CC. As a consequence, utilities are affected by the decisions in the CC.

Mostly data flow is upwards in the hierarchy from PMUs to the CC, but commands (e.g., for device configuration), requests (e.g., requesting data formats or device information) or software updates require communication in the reverse direction. PMU messages are transferred using TCP (transmission control protocol) or UDP (user datagram protocol) over IP or can also be transmitted directly over Ethernet or other available transport means [169]. In addition to the measurement data reported from PMUs to PDCs, also configuration files with data interpretation settings can be reported to PDCs. Furthermore, PDCs can send command files to PMUs to request information [136]. All these files have a common structure.

WAMS infrastructures can contain various combinations of components, communication equipment, applications, visualization tools and many more [80]. The Fig. 2.3 shows different variants of WAMS topologies.

PMUs report data to PDCs or directly to the controller. PDCs check and aggregate PMU records and then forward them to super PDCs or to the CC. Additionally PDCs may already calculate some values from the reported data [69]. That means PDCs need to have access to credentials for decrypting PMU records and also need to be able to sign PDC records.

Local storage, data verification and application functions are usually available in PDCs. Another possible level of hierarchy is super data concentrators (SDCs), also called super

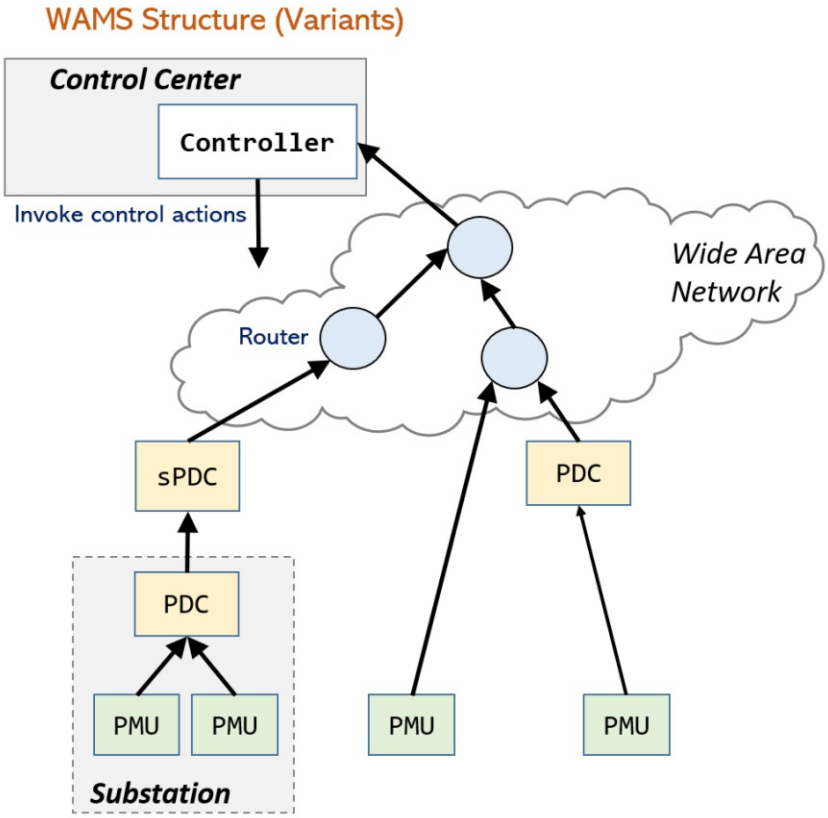


Figure 2.3: WAMS Architecture (source Paudel et al. [131])

PDCs. Super PDCs have functions that are identical to regional PDCs. Further, the data storage facility in the super PDCs can store the data associated with time-tags and a stream of near real-time data that can be used for the applications in the entire system [136],[169].

Furthermore, it is possible to deploy PGWs. PGWs are introduced by the North American SynchroPhasor Initiative (NASPI)[1] as a concept to interconnect multiple organizations. PDCs or PMUs report their data to the PGW. A PGW then communicates the data to other PGWs by a publish-subscribe system. PGWs can support Quality of Service functions and serve as security gateway between organizations. At the top level, a CC is connected via a WAN and controls all activities regarding monitoring, protection, and control. Since PMUs may also directly report to a CC, all hierarchy levels (PDC, Super PDC, PGW) are optional. We call all systems on the way from PMU to CC (PDC, Super PDC, PGW, core and access routers) intermediate systems.

[1] https://www.naspi.org/

2.3 WAMS Communication Standards

In a WAMS, different communication protocols from different standards are used. Figure 2.4 shows an overview of the most important standards used for end-to-end communication and the following paragraphs describe them briefly.

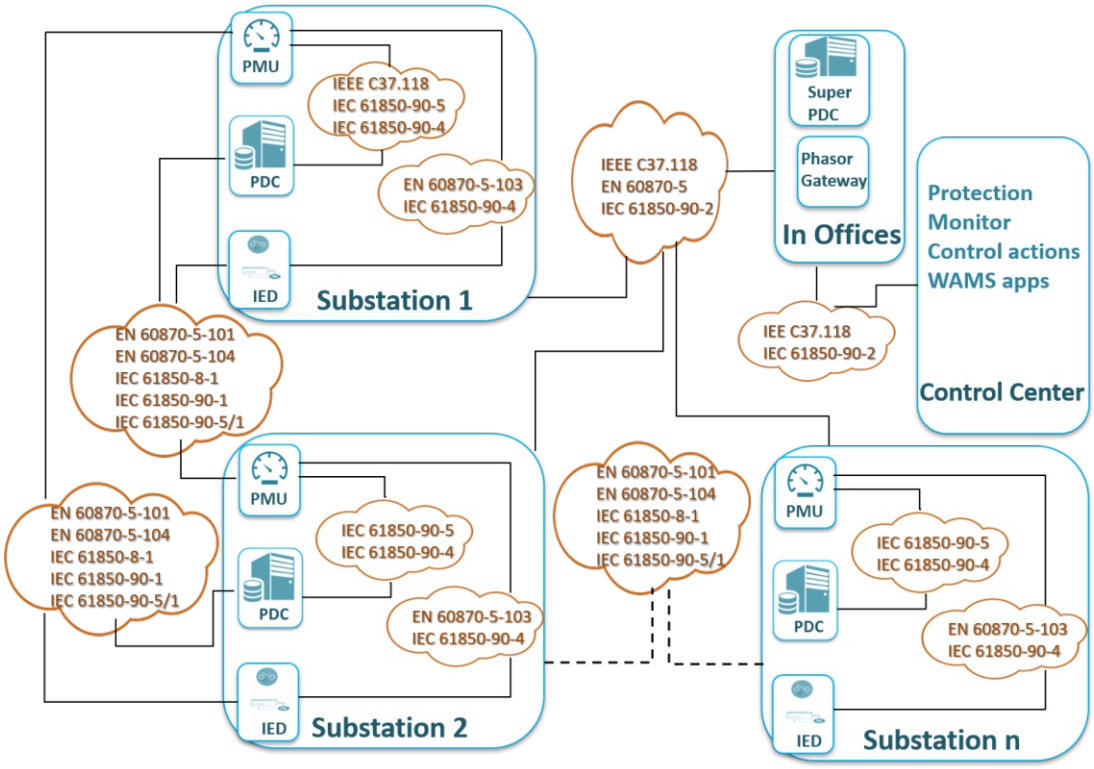


Figure 2.4: WAMS Communication Protocols (source Paudel et al. [129]).

IEEE C37.118 is used for phasor measurements communication in power systems. IEEE C37.118.1 [70] specifies measurements of a synchrophasor, and IEEE C37.118.2 [71] describes a protocol for the real time transfer of phasor data. It defines data messages, configuration messages, header messages and command messages that are required for communication [184].

The standard IEC 61850 [63] is a communication protocol that facilitates utility automation, including protection and control [37]. Originally, the standard was developed for IEDs in substations. Nevertheless, now the standard includes various communication features [10, 9]. Further, an architecture of electric power systems and data models that are used for communication are defined by the standard [10, 9]. Abstract data models defined in this standard are mapped to a variety of protocols. For example, some models are mapped to generic object oriented events (GOOSE), generic substation events (GSE),

and sampled measured values (SMV) [66]. It supports the sending of real-time data and supervisory control functions using manufacturing message specification (MMS) via TCP/IP and transmission of GOOSE via Ethernet in substation LANs.

IEC 61850-90-1 [67] provides guidelines of using IEC 61850 for the communication between substations. Similarly, EN IEC 61850-90-1/5 and IEC 61850-8-1 provide guidelines for communication between PMUs and PDCs within substations. IEC 61850-90-4 provides guidelines for communication inside a substation. Standard IEC 61850-90-2 [68] covers the communication within substations and the CCs using IEC 61850 standard [30].

Standard EN 60870-5-103 [61] provides guidelines for connecting PMUs/IEDs inside a substation. EN 60870-5-101 [60] provides transmission procedures between substations. Similarly, EN 60870-5-104 [62] is an extension of standard EN 60870-5-101 and provides guidelines between PMUs and data concentrators between substations [30]. An overview of the use of these protocols is presented in Figure 2.4.

The standard IEC 62351 has been developed for securing the communication protocols that are defined in IEC 60870-5 and the IEC 61850 series of standards [64]. IEC 62351-6 [65] defines the security of IEC 61850 profiles by specifying messages, procedures, and algorithms for securing the operations of all protocols that are derived from the standard IEC 61850. IEC 61850 provides reliable communication in substations. For instance, it supports intrusion detection before/after accessing networks, reduction of handshake duration between devices [48]. The specification applies at least to the protocols IEC 61850-8-1, IEC 61850-9-2 and IEC 61850-6. It also provides security for profiles not based on TCP/IP, e.g., GOOSE, GSSE (generic substation status event) and SMV. The IEC 61850 profile using MMS over TCP/IP uses IEC 62351-3 and IEC 62351-4.

PGWs support IEEE C37.118 for phasor data traffic (e.g., traffic to and from PDCs, super PDCs and PGWs), but it is not enough for additional control and administrative traffic beyond the PGW [32].

2.4 Safety Limits

In this section, we present foundational information related to safety and security of a power system. Safety limits according to standards and guidelines are presented below.

Table 2.1 illustrates operating conditions of voltage, frequency, phase angles, projected situations and corresponding control actions for a 50Hz power system. The numerical values of voltage, frequency, phase angles and the control actions in this table are based on our literature survey.

Abnormal voltage and frequency conditions can cause failures like line tripping, generation tripping etc. For example, under voltage or under frequency can cause generation trip, and over load or loss of synchronization can cause line tripping. Control actions can be triggered against the failures, proactive control actions can prevent failures, whereas reactive control actions overcome the failures after they occurred [172].

Table 2.1: Measurements, events, situation and control actions; Mea. = measurement; Ref. = references; Freq. = frequency; Imp. = impedance.

Mea.	Event	Situation	Control action	Ref.
Voltage (v)	v <0.9 p.u.	Under voltage	Active power control, reactive power control, shed load	[1] [51] [172]
	0.9 p.u. <v <1.1 p.u.	Normal operation	Continue operation	
	v >1.1 p.u.	Over voltage	Active power control, reactive power control	
Freq. (f)	f <47.5 Hz	Under frequency	Disconnect power generation, partition network	[153] [172]
	f <49 Hz	Low frequency	Load shedding	
	49.8 <f <49.98 Hz 50.02 <f <50.2 Hz	Normal operation	Continue operation	
	f >51.5 Hz	Over frequency	Disconnect power generation, blackout	
Power (P), phase angles $(\theta 1, \theta 2)$	P $>(v1 - v2)/Imp.$ $(\theta 1 - \theta 2) > 90°$	Line outage, generation trip, load change, power oscillation	Load shedding, partition network	[161] [151] [166] [172]
	$(\theta 1 - \theta 2)$ <90°	Normal operation	Continue operation	

Various standards and guidelines are developed for dealing with problems on high voltage (HV), medium voltage (MV) and low voltage (LV) networks. The standards cover different power and voltage levels. German standards VDE-AR-N 4105:2011 [6] and BDEW-2008 [3] address LV and MV/HV, respectively. Similarly, IEC 61727-2004 [2] focus on the PV systems network, and IEEE 1547 [1] was developed for primary and secondary distribution voltages. Standards also specify a range for normal and abnormal behavior of measurements (e.g., voltage, phase angle, frequency etc.) and define their critical thresholds for protecting the power system.

The voltage magnitude in a network can be affected by power injection, e.g. due to distributed generation. Protection schemes are applied to maintain secure operation in a system. Over voltage and under voltage protection schemes ensure appropriate voltage levels before applying control actions. For example, according to standard IEEE 1547 [1] voltage from 88% to 110% are considered as normal operation, greater or less than this range requires protection actions, otherwise a network or a subsystem should

be disconnected within the specified time. Some events can exist while transmitting power across a network. Existence of events like load change, generation trip, line outage etc. can change the voltage phase across the network. Thus the phase angle difference between two points of a network correlates to the power being transferred from one point to the other of the grid [166]. The phase angle difference within a power system is the difference between the measured angles at two points at the same time instant and to the same reference [119].

CHAPTER 3

State of the Art

Notice of adoption from previous publications in Chapter 3
Parts of the contents of this chapter have been published in the following papers:

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. 4th International Symposium for ICS and SCADA Cyber Security Research, *2016*

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids, *CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM*

[131] S. Paudel, P. Smith, and T. Zseby. Data Attacks in Wide Area Monitoring System. Symposium on Innovative Smart Grid Cybersecurity Solutions, *2017*

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
The attacks on smart grids, FDI attacks on WAMSs, modeling attacks, challenges and security issues, and existing approaches against security issues described in this chapter are based on the work done in [130], [131] and [132]. Attack detection using

state estimation described in this chapter is based on the work done in [129]. A part of stealthy attacks described in this chapter is based on the work done in [133].

S. Paudel performed the theoretical considerations together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, we present an analysis of related work on attacks on smart grids, attacks on wide area monitoring systems, modeling attacks, security issues, existing approaches, bad data detection, stealthy attacks and attack detection techniques based on state estimation. Attacks to major components of wide area monitoring systems can be detected by analyzing available information in the system. In detection approaches, we focus on how the existing work on estimated states, aggregated information (data or events) and anomaly detection can help detecting attacks to key components of the wide area monitoring systems.

3.1 Attacks on Smart Grids

Smart grids (SGs) are cyber-physical systems. Therefore, we need to consider security in both the physical and the cyber domains. Standards, guidelines and other existing works address vulnerabilities, threats, attacks, security requirements, and intrusion detection in SGs.

The north american electric reliability corporation (NERC)[1] critical infrastructure protection (CIP) provides security requirements for bulk power system. The national institute of standards and technology (NIST)[2] provides documents on cybersecurity for SGs. Guidelines provided by NIST in [123] discuss security requirements for secure architecture and interfaces, ensuring reliable functionality and maintaining information confidentiality and other security measures. Potential vulnerabilities, security problems of SGs are discussed together with cybersecurity requirements of reliable and scalable operation [124].

ATT&CK for industry control systems (ICSs) [110] present techniques which can be considered by adversaries for compromising ICSs. Tactics used in the techniques, software used for compromising assets of ICSs and impacts in control system addressed by ATT&CK help in better understanding of adversary behaviour. Guidance for securing ICSs (e.g., supervisory control and data acquisition (SCADA), distributed control system (DCS), programmable logic controllers (PLCs)) are provided in [160]. In addition, it points out threats and vulnerabilities of the ICSs, and provides countermeasures for mitigating the risks.

[1]https://www.nerc.com
[2]https://www.nist.gov/

3.2 FDI Attacks on WAMS

In wide area monitoring systems (WAMSs), false data injection (FDI) attacks modify measurement or control data (either the original readings from phasor measurement units (PMUs) or aggregated events from PDCs or super PDCs). In WAMS, the processing of falsified data estimates incorrect states of the power system and can cause wrong decisions, such as triggering protection elements when they are not needed or suppressing a vital protective action. For example, the system may believe that it has secure voltage in overloaded branches and vice versa [43]. This can cause delay, e.g., for load shedding or grid reconfiguration. Advanced persistent threats (APTs) can be created that combine different attack techniques. Information may be first gathered in a passive attack to learn system state and vulnerabilities. Then an active attack can cause major damage to the system.

Attackers aim to compromise key components in a WAMS. Here, we point out some attack scenarios in WAMS. PMU measurements can be modified by compromising the PMU itself, PDC, super PDC, PGW or routers during transmission. PMU data modification attacks by compromising PMUs are presented in [43, 99, 83, 39, 126, 92]. Attack scenarios of a PDC and super PDC are presented in [164, 126, 122]. Similarly, attacks on routers are shown in [126]. These attack scenarios and applicable detection techniques will be discussed in Sec. 3.8.

If attackers inject falsified information it could lead to implausible states in the power system and therefore raise suspicion. State estimation (SE) methods usually consider the case when wrong measurement data is received directly from PMUs, but the original measurement data can also be modified in PDCs or on routers on the path, as described in Sec. 5.1.2.2.

3.3 Modeling Attacks

Attack trees [152] are common models to represent complex attacks. An attack tree contains a root, branches, several intermediate nodes and leaf nodes. The root represents the ultimate goal of an attack, different branches shows different possibilities of reaching the root node. Different possibilities could be reached by combining all branches or only by following one branch. This depends on the type of relationship (AND/OR) while branching the node. For an AND relationship, all sub goals must be reached; meanwhile, for an OR relationship, reaching at least one sub goal is enough to reach the higher goal. Each intermediate node is a sub goal of the attack, whereas leaf nodes represent the start points. Moore et al. [116] propose an attack modeling method by describing format and semantics of the attack trees; a straight solid line is used between the branches of a node if they have AND relation and a curve solid line is used if they have OR relation. Figure 3.1 shows an example attack tree that uses the AND/OR relationship while branching node. In the example attack tree, the ultimate goal can be achieved by achieving two sub

goals which are shown by AND relationship between branch 1 and branch 2. A sub goal in branch 2 can be achieved by either achieving the sub goal in branch 3 or achieving the sub goal in branch 4.

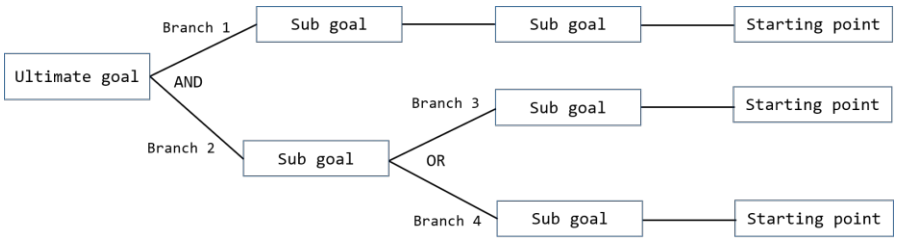


Figure 3.1: An attack tree that uses the branches' AND/OR relationship.

Attack trees provide a valuable overview of the pre-requisites and sub goal relations for attacks that are based on the combination of multiple different actions. They help to analyze attack goals and potential chains of actions. They assist in assessing the likelihood and costs of specific sub goals and attack branches and support detection and prevention of complex combined attacks.

Attack trees provide a suitable method for considering attacks in the physical as well as in the cyber domain. In [28], the authors investigate vulnerabilities in MODBUS-based SCADA systems using attack trees. MODBUS [111] is a serial communication protocol that is used for transmitting information between intelligent electronic devices (IEDs) using PLCs. There is also a TCP/IP version of MODBUS. The mostly used element is a client-server command, which is used in MODBUS networks to send MODBUS messages from a master to a slave. The protocol is used to connect a supervisory computer with a remote terminal unit (RTU) in SCADA systems. The authors present attack trees for gaining access to the SCADA system. Furthermore, they provide an estimated level of technical difficulty, severity of impact, probability of detection and underlying critical vulnerabilities for the sub-goals. Closely related to this work, a technical report from the EU-funded SPARKS project [59] presents different attack patterns for compromising components, communications, and functions for cyber-physical systems in the SG. The authors identify attacks that can be performed by physical or cyber means, locally or remotely.

Chen et al. [35] investigate the use of Petri nets [143] for modeling attack trees for attacks to SGs. A Petri net model is a directed graph with states, transitions and directed arcs that are used to model concurrent processes. In the graphs, states are drawn as circles, transitions are drawn as boxes or bars [135, 118]. The authors in [35] propose a hierarchical method to construct large Petri nets. Several Petri nets created separately from various different domain experts can be integrated to create a large Petri net. Authors also describe the usage of the model by constructing a cyber-physical attack on smart meters. This example integrates both cyber and physical actions created by domain experts and their integration in a single Petri net model. They also present how

a low level detailed cyber attack and a detailed physical attack can be integrated in a high level Petri net model. Additionally, authors do a validation of the method using an example of the smart meter attack in a Python program. This is a good example of modeling cyber and physical attacks, but the Petri nets method is mainly of advantage if we want to model concurrency.

3.4 Challenges and Security Issues

WAMSs improve situational awareness in the SG but security is a major concern in the system. Zseby et al. [183] study recent approaches for the WAMS communication and point out the security challenges that need to be addressed. The authors describe communication scenarios using PMUs and PDCs using unicast and multicast functions in wide area networks (WANs). Sensor data or PMU data is transmitted from different geographical locations over WANs. Latency and security are more challenging in a WAN than in a LAN. The authors also discuss security challenges in WAMS. Attackers can try to inject, modify or deny messages, e.g., measurement values in WAN.

In [98] Lui et al. present an overview of relevant security and privacy issues, and discuss the potential research fields in SG based on a literature survey. Advanced metering infrastructure (AMI), SCADA, Communication Protocols and Standards are pointed out as key components of SGs. SG security has several challenges considering these components, e.g., protocols communication requirements, designing and implementing technologies and protocols without considering cyber security. Authors also point out security issues as i) Device issues (e.g., smart meter, customer interface), ii) Networking issues (network, wireless network, sensor network issues), iii) Dispatching and management issue (asset management, cipher key management, real time operation management issues), and iv) Anomaly detection issue (e.g., temporal information such as timestamping log files, data service issues). Additionally, authors also mention demand response issues, and protocol and standards issues.

3.5 Existing Approaches Against Security Issues

An overview of attack entry points and existing detection methods to data integrity attacks is provided in [129]. The paper summarizes different methods to deal with inconsistencies in sensor data. One common method is to use SE based on sensor data to deduce the state of the grid and to check the plausibility of collected measurements. In [43], authors discuss PMU data modification attacks during transmission from PMUs to PDC and present detection technique by using SE. The authors calculate a state vector V_i, where i is the number of PMUs in a network. Measurement values from all PMUs except from the i^{th} PMU is used to calculate the vector and same estimation process is repeated for each of the i PMUs. The authors calculate the euclidean distance between

each state vector and the average vector. The average vector is calculated using the average state vector. The state vector that has the largest distance to the average vectors is compared to the predefined threshold for finding the deviation. If the deviation is greater than the threshold value, then the particular PMU is considered as attacked.

Similarly, a detection scheme using dynamic SE in a power grid is developed in [164]. In a system, parameters and knowledge are always static and values that updated continuously are dynamic. Authors use the knowledge, parameters and dynamic values to verify a real-time depiction of a nominal system in the literature. They estimate unknown inputs, detect malfunctions, cyber attacks and disturbances. After that they identify attack locations and faulty channels in the system. Attacked components are reconfigured after the diagnosis. Ensuring the power system's observability, it is restored to the nominal state and begins operations again.

Although there are various detection techniques already developed, new and unexpected attacks are launched due to vulnerabilities in technology. Preventing attacks and protecting communication networks against these attacks is a challenging task [13, 15]. For instance, packet drop attacks are possible in a communication network, but packet loss can happen due to congestion or due to an attacker.

A real-time mechanism to detect packet drop attacks is proposed by Pal et al. [127]. The authors develop a classifier which distinguishes the causes of a packet drop as either due to congestion or an attacker. If PMU packets are dropped, the packets are classified as due to an attacker and exceed the predefined threshold then an attack detection alarm is generated. Similarly, a security analysis tool for AMI misconfiguration is proposed by Rahman et al. [140]. The authors create a model representing the global behavior of an AMI configuration, and compliance with security constraints applicable to AMI configuration. It detects misconfiguration by verifying the constraint violation.

Soule et al. [159] compare four methods to analyze residuals for computer network anomaly detection: (i) by setting a threshold based on the residual's behaviour (ii) by comparing a local and global variance; (iii) by analysing wavelet; and (iv) by using a generalized likelihood ratio test. Pignati et al. [139] propose an algorithm that uses pre-fit residuals to detect bad data in real-time PMU measurements. Bad data is assumed to occur for benign reasons, such as communication network and timing errors. A Kalman filter is used for SE, and pre-fit residuals are used to detect when an anomaly exists in the observed data. A dynamic detection threshold is defined based on a confidence level. If an anomaly is detected, the observed bad data is replaced. How the values are replaced depends on whether the bad data was caused by power system fast dynamics (e.g., line faults) or other root causes. Changes in a power system's topology, e.g., caused by faults or switching, can have an effect on SE. In their work, Møller et al. [113] review the bad data detection algorithm proposed by Pignati et al. [139]. They propose a method to detect branch errors by checking a bias in branch flows. The normalized bias branch flow is calculated based on normalized measurement innovation for wide area measurements. Detecting a branch error in the pre-estimation phase prevents severe power system impacts, such as overloading branches.

Some previous work also addresses the detectability of attacks. Barreto et al. [20] present undetectable cyber attacks on packet-based time synchronization protocols using a "delay box". The resulting delay attacks exploit vulnerabilities of the protocol and remain undetectable by passing the clock-servo algorithm inside the targeted slave clock. Similarly, delay attacks to linear SE that manipulate time references are presented in [19]. The attacks remain undetectable for classical methods like the χ^2 test, largest normalized residual tests, and bypass the bad data detection algorithm. In order to cope with the dynamic evolution of cyber threats and system configurations the authors in [18] propose an online anomaly detection algorithm for detecting anomalies in measurements. They perform simulations using an IEEE 14 bus power system and demonstrate a good balance of minimum attack magnitude and thresholds to improve the detection performance. Dan et al. [42] present some protection schemes against stealthy attacks on state estimators of power systems. The authors compute a security index for successful stealthy attacks and use the index for quantifying the security of encrypted devices and measurements.

3.6 Bad Data Detection

Data sources can have errors due to various reasons (e.g., sensor failures). Therefore, many grid operators implement BDD methods to check for failures in the measurements that could influence SE. For this, residuals from the SE are often used. If the residuals are too high, it is inferred that the data is not correct [171]. Residual-based bad data detection (BDD) methods are described in detail in Sec. 7, here we mainly provide closely related work.

Post-fit residuals are the difference between estimation and observation, and pre-fit residuals are the difference between prediction and observation. Generally, the traditional BDD methods use post-fit residuals for checking bad measurements. For instance, linear weighted least squares (LWLS) method has only post-fit residuals. Usually the traditional BDD methods use some methods for instance L2-norm of the residuals, cumulative chi-square distribution [86] for deriving the threshold for BD detection. The authors in [22] define a BDD threshold based on the L2-norm of measurement residuals without noise and compare the L2-norm of the residuals to the defined threshold for detecting the bad measurements.

A BDD method based on the pre-fit residuals of SE using Kalman filter is presented in [139]. This approach uses a dynamic threshold for detecting bad data, and compares pre-fit residuals to the dynamic threshold. We adopt this BDD method in our attacks scenarios (attacks generated using our attack model).

3.7 Stealthy Attacks

A stealthy attack is an attack that can circumvent the detection. Most of the existing works (e.g., [22]) assume that an attacker needs to know complete network information for constructing stealthy FDIs attacks. But in contrast to this statement some of the existing works [141, 79] show the necessary information to craft a stealthy attack can be derived from incomplete information of the system (e.g., online grid topology, offline grid topology, market data, power flow measurement etc.). Thus launching an FDI attack needs to implement some techniques such that attackers inject errors on measurements by keeping residuals under the threshold or by exploiting tolerated measurement error in SE. We implement FDI attacks in a way they exploit the SE process such that the pre-fit residuals are under the thresholds [132].

Undetected attacks which successfully circumvent the detection are classified as stealthy attacks. Dan et al. [42] present stealthy attacks against state estimators. The difficulties of performing stealthy attacks against measurements are defined by Sanberg et al. [148] as security indices, and an efficient computation of the security indices is presented in [42].

These attacks are known as minimal stealthy and optimal stealthy, depending on the compromised measurements while being stealthy.

3.7.1 Optimal and minimal Stealthy Attacks

Minimal stealthy attacks are stealthy attacks that manipulate only a minimum of the measurements. Dan et al. [42] describe the minimum number of measurements to be falsified for performing a stealthy attack. If an attacker can perform an attack from a substation then the attacker potentially can manipulate all measurements from that substation. In this case, an attacker can manipulate the optimal number of measurements to be stealthy. It means an attacker can increase the number of manipulated measurements until the attack remains stealthy. The authors in [42] address minimal stealthy attacks. Similarly, optimal stealthy attacks on CPS are presented in [168]. Here, the difference between minimal and optimal stealthy attacks is in the first case an attacker manipulates the minimum number of measurements while being undetectable, and in latter case an attacker manipulates the possible number of measurements while remaining undetectable.

3.8 Detection of FDI Attacks

Various mechanisms have been proposed to detect data injection attacks in SG systems. We focus on data injection attacks on WAMS. Attackers can compromise key components of a WAMS like PMUs, PDC, super PDC, PGW, core routers and access routers by having physical access or remote access. Further details on the possibilities of accessing the components and the consequences of the attack scenarios will be presented in Sec. 5.1.2.2.

The attack scenarios describe data integrity attacks by compromising PMUs, PDC, super PDC, PGWs or routers in WAMS. Here we classify the existing approaches and investigate to which extent they can help to detect or mitigate the data integrity attacks. We define three categories depending on the suitability of the methods:

- *Category 1:* Techniques that directly help to detect attacks at least in some parts of the scenarios in WAMS.
- *Category 2:* Techniques that can be modified to be applied to our scenarios in WAMS.
- *Category 3:* Techniques that do not help in our scenarios in WAMS.

We map each of the scenarios to the techniques using signs ✓ for category 1, ∼ for category 2 and ✗ for category 3 in Table 3.1.

Table 3.1: Mapping of the scenarios S_1 to S_6 as described in Sec. 5.1.2.2 to the existing techniques; signs ✓ for category 1, ∼ for category 2 and ✗ for category 3 (source Paudel et al. [129]).

Detection Techniques	**PMUs** (S1)	**PDCs** (S2)	**super PDCs** (S3)	**PGWs** (S4)	**Access routers** (S5)	**Core routers** (S6)
State estimation	✓[43] ✓[99] ✓[83] ✓[39] ∼[164] ✓[126]	∼[43] ∼[99] ∼[83] ∼[39] ✓[164] ✓[126]	∼[43] ∼[99] ∼[83] ∼[39] ✓[164] ✓[126]	∼[43] ∼[99] ∼[83] ∼[39] ∼[164] ∼ [126]	∼[43] ∼[99] ∼[83] ∼[39] ∼[164] ✓[126]	∼[43] ∼[99] ∼[83] ∼[39] ∼[164] ✓[126]
Aggregation	✗	∼[81] ✓[122]	∼[81] ✓[122]	∼[81]	✗	✗
Anomaly detection	∼[162] ✓[92]	∼[162]	∼[162] ∼[140]	∼[140]	∼[127]	∼[127]

Table 3.1 illustrates the applicability of relevant existing works for detecting the attack scenarios on PMUs, PDCs, super PDCs, PGWs, core routers and control routers. Three categories of existing detection methods appear in the first column. In columns 2-7 (PMUs, PDCs, super PDCs, PGWs, core routers and control routers) it shows the mapping of the techniques to the relevant scenarios. The existing works are described in the following subsections.

3.8.1 State estimation

State estimation methods usually help to detect attacks on intermediate systems. We denote this with a ∼ to indicate that the solution can be applied even if not originally

developed for the specific scenario. Nevertheless, if an attacker compromises a PDC and can modify data from multiple PMUs it is easier to alter data so it still looks consistent for SE. This is similar to the situation with a set of colluding attacks from multiple compromised devices.

Dehghani et al. [43] discuss attacks by altering PMU data during transmission in PMUs or PDCs. The authors develop an approach based on static SE (SSE) algorithm to detect integrity attacks in PMU networks. The PMU network consists of i number of PMUs in the network. The authors calculate a state vector V_i using measurements from all PMUs except data from the i^{th} PMU and then repeat this for all i PMUs. Then they use the average of the state vectors to calculate the Euclidean distance between each state vector and the average vector. They then select the state vector that has the largest distance to the average vector and compare it to a predefined threshold for deviation. If it exceeds the threshold, they assume that the measurement values from that particular PMU have been altered by an attacker. Applying this algorithm in PMU networks, we can detect compromised PMU frames in Scenario 1 (PMU compromised) in Sec. 5.1.2.2, but only if the modifications are large enough to cause a large deviation. Also setting appropriate thresholds for such systems is not trivial. The method was developed to detect attacks on PMU measurement values. It may be applied to detect attacks in intermediate systems as explained in Sec. 5.1.2.2, but with access to multiple PMU values in intermediate systems it can be easier to make values to appear consistent.

A detection scheme using dynamic SE (DSE) has been developed by [164]. A real-time depiction of the nominal system is verified based on the knowledge and parameters of a power system model and real-time PMU measurements. Knowledge and parameters are static, whereas measurements are dynamic as values are updated continuously. This step verifies measurement values with the system model. Then the unknown power system parameters and unknown inputs are estimated using real time PMU data and the system model. As a third step, malfunctions, cyber attacks and disturbances are detected by estimating attack vectors and using an attack detection filter. The filter detects compromised nodes and compromised measurements. Fourth, attack locations and faulty channels are identified. Fifth, the attacked components are diagnosed and reconfigured ensuring observability of the power system. After ensuring the observability of the power system, it is brought back to the nominal state and starts operation, otherwise it keeps on diagnosing and reconfiguring the system. The method has been developed for the National Electric Sector Cybersecurity Organization Resource (NESCOR)[3] scenario for attacks on PDCs [8], but can be applied to super PDCs also to detect direct attacks on the PMU data modified at the PMU. This would also work if PMU data is changed on PGWs and routers.

Pal et al. [126] assume that nominal transmission line parameters to which the PMUs are connected are known. The authors then use the measured PMU data to estimate transmission line parameters (bus voltages, current and phase angles). If the deviation

[3] https://smartgrid.epri.com/NESCOR.aspx

between measured and nominal data exceeds a threshold, a data modification alarm is generated. The method is developed for data manipulation attacks on PMU data. The data can be modified in the PMU itself or on the way to the control center in PDCs, super PDCs or routers. The authors do not mention PGWs, but the method can also be applied if an attacker modifies original measurement values in a PGW.

Liu et al. [99] show how malicious attackers can craft a coordinated stealthy attack that bypasses classical bad data detection in SE based on a DC power flow model. They show that attack vectors exist, even if attackers have access only to selected measurements and limited resources. In [83] the use of some highly secured observation points as trusted references is proposed based on the same model. Those trusted anchors make it harder for attackers to find suitable values for stealthy attacks. In [39] both approaches are discussed and a distributed algorithm is proposed to detect coordinated data injection attacks. The algorithm is defined for general coordinated attacks in the wide area system.

The methods proposed in [99], [83] and [39] are suitable to mitigate data injection attacks on PMUs or intermediate systems. As described in Sec. 5.1.2.2 we assume that routers may be also able to modify the data. Therefore the methods are also suitable against attacks on routers.

3.8.2 Aggregation

For detection systems based on aggregation, we distinguish between two methods: Data aggregation that deals with aggregating the measured data itself, and event aggregation that aggregates the events that were derived by inspecting measurements from one or multiple observation points.

3.8.2.1 Data Aggregation

Several components in a WAMS perform data aggregation. Aggregating measurement data at certain points of a system helps to analyze the situation in the overall system, without the need to store and transmit a vast amount of fine-grain information. Data aggregation can also have a smoothing effect that reduces the impact of wrong data in a larger dataset. Nevertheless, aggregation systems might be compromised.

In [122] a data aggregation scheme for smart metering reports is proposed that can cope with malicious aggregating gateways. Their goal is to maintain non-repudiation and integrity under the assumption that the gateways have been compromised. In their scenario, the smart meters and the control center can be trusted and just an intermediate aggregator on the path is compromised. In this scheme the aggregator does not own credentials itself, but rather uses homomorphic authenticators to combine authenticated messages from multiple records into an own authenticated aggregated record, but without knowing the secret. In contrast to other schemes that assume that the gateway just

eavesdrop on the data but otherwise follows protocol, the proposed solution does also work if the aggregator does not comply to protocol operations.

The authors also show that their technique has less computational and communication overhead, compared to existing techniques. The scheme can be applied to WAMS scenarios to prevent malicious activities in aggregating devices such as PDCs or super PDCs. But since PDCs mainly combine records, homomorphic operations on the data may not even be necessary.

3.8.2.2 Event Aggregation

Kim et al. [81] present a security events aggregation system to provide situation analysis. This system collects security events from sensors and aggregates the data periodically or on demand. Event aggregation techniques are widely used for identifying correlated activities based on the frequency of information. The assumption is that several suspicious events indicate a problem, whereas a single outlier might be just a false positive.

We can modify and adjust event aggregation systems to aggregate events in PDCs, super PDCs, PGWs and CCs.

3.8.3 Anomaly detection

It is challenging to protect communication networks from new and unforeseen attacks, as new vulnerabilities and sophisticated attacks are introduced every day [13, 15]. Anomaly detection techniques help to detect such attacks and provide hints to identify the cause and origin of incidents.

Pal et al. [127] propose a real-time mechanism for detecting packet drop attacks (on sensitive synchrophasor data) over the Internet. Packet loss can be due to congestion or due to an attacker. The authors build a classifier to distinguish both cases. In a given time interval, if dropped PMU packets are classified as attack drops and the number exceeds a threshold, then an alarm is generated. This is applicable to attacks on routers.

Sun et al. [162] propose a cyber physical monitoring system to detect smart meter bad data injection attacks. The authors use Snort [38] to analyze the traffic flow, and in addition perform energy measurements in the physical system. Energy measurements are verified against the physical topology and energy conservation laws. Alerts from cyber network and physical systems are fused to detect attacks. This system checks the injection energy by combining the energy consumption, total transmission loss and measurement error. The threshold of total transmission and measurement error is defined as 5% of the injected energy. If the total transmission and measurement error exceeds the threshold then an alarm is triggered and it records IP address, date and time. The method is targeted at smart meters, but by defining threshold values of measurement error, and checking the difference between the generated values from PMUs and received values in a PDC, we can detect compromised PMUs.

Rahman et al. [140] propose a security analysis tool for detecting misconfigurations in advanced metering infrastructures (AMI). They create a formal model representing the global behavior of AMI configuration, compliance with security constraints and verify the potential security threats violating the constraints. The method is targeted at smart meters, but can be modified to be applied to PGWs and super PDCs to detect misconfiguration.

Kwon et al. [92] propose a behavior-based Intrusion Detection System (IDS) for the IEC 61850 protocol by using statistical analysis of classical network features and metrics based on the protocol specification. The authors combine static features (e.g., protocol consistency), dynamic features (e.g., frequency and distribution of GOOSE message) and generic features (e.g., bits and packets per seconds) from the communication network. They define three metrics for i) generic network features, ii) GOOSE behavior-based usage pattern and iii) MMS protocol-based commands as input for the anomaly detection. The authors implement the IDS in a substation and demonstrate that the system detects attack scenarios successfully. We can apply a similar combined intrusion detection technique in substation or intra-substation communication, in order to detect anomalous transmission of PMU data using IEC 61850.

3.9 Summary

In this chapter, we presented the investigation of attacks on SGs, attack modeling techniques; challenges, security issues and attacks on WAMSs; and existing approaches for attack detection.

In a first step, we investigated the potential vulnerabilities, security problems, challenges of SGs and adversary techniques in the literature. It provides insights into i) the effective cybersecurity measures, ii) the sources of the threats, iii) attack techniques and impacts of attacks to control systems. This existing work helps me to understand the vulnerabilities in cyber-physical systems and assess the importance and impact of different attack types. Our study finds that there is there's only some research and there are still many open issues on SG WAMS security. We therefore set the objective of this research to strengthen the security of the WAMS.

In a second step, potential attacks on different components of SG WAMS are investigated. It showed that data modification attacks (e.g. FDI attacks) on sensor measurements at different components of WAMS could impact in the control systems of the power grid. But literature review shows that existing body of research rarely focus on data modification attacks against WAMS. Thus in contrast to existing research, we perform a threat analysis particularly for WAMS architecture. To this end, we develop attack vectors, attack scenarios and attack trees in order to derive vulnerabilities and attack scenarios in WAMS. Further, we develop generic attack trees (e.g., for compromising a device) and develop specific attack trees (e.g., for causing a blackout, manipulating sensor data).

In a third step, we study models to represent attacks. Based on our knowledge, existing body of work on attack modeling represents one attack type with a model; none of them generate multiple FDI attack types with a model. We bridge the gap by developing a FDI attack model for generating different attack types.

In a fourth step, existing approaches for detecting bad data and attacks are studied. It provides the insights into the nature of potential attacks and their detection methods. In the literature, detection methods are classified in two categories; bad data and attack detection methods. Bad data can be detected using residuals of SE methods. In order to apply bad data detection methods in our use case, we implement SSE and DSE methods namely linear weighted least squares and Kalman filters. We then develop the promising methods from literature, L2-norm and normalized residual-based methods. In addition, an existing plain residual-based anomaly detection method in literature [139] which was previously used for simulated data is reproduced and adopted to our use case with real data. Further, we design FDI attacks that circumvent detection of the adopted method. We investigated anomaly detection methods and find that existing anomaly detection approaches barely consider critical infrastructure setup (e.g., limited computation resources) of SG. This motivates us to develop an anomaly detection model considering the critical infrastructure setup. For this, we applied selected lightweight statistical methods; median absolute deviation (MAD), Kullback-Leibler divergence (KLD) and Cumulative sum (CUSUM). We further investigate methods for improving detection performance. In contrast to existing work, we apply the combination of lightweight statistical methods in SGs. We use weighted voting scheme in [101] previously used for combining machine learning methods to combine the statistical methods.

In the final step, we investigate methods for correcting bad (anomalous) measurement. Models in the literature use historical data for recovering inconsistent system state using previous consistent (normal) state. Similarly, we develop a model for estimating current system, replacing anomalous measurement by predicted value of DKF in attack types. In contrast to existing work, we replace anomalous measurement by predicted value of DKF in attack types and show how voltage estimation integrity is preserved based on the anomaly detection results of the statistical methods. Our approach covers the future work in [85]).

CHAPTER 4

State Estimation for Power Systems

Notice of adoption from previous publications in Chapter 4
Parts of the contents of this chapter have been published in the following papers:

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
Discrete Kalman filter and a part of experimental setup described in this chapter are based on the work done in [132]. Weighted least squares described in this chapter are based on the work done in [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, we use a simplified state estimation model (derived from [50]) for voltage estimation based on one phase on a single link. We describe the two SE approaches, linear weighted least squares and Kalman filter that we use for our experiments. We apply linear weighted least squares and Kalman filter to data from real power system measurements

and validate the methods. In addition, we implement an example of the Kalman filtering and modify the Kalman filter model for adopting it to our context. Then we apply the two state estimation methods (linear weighted least squares and Kalman filter) for estimating states in our use case.

State estimation (SE) [16, 114] is used for system monitoring and estimating the unknown system states of a power grid by evaluating the measurement and the power flow models. Staff members in a control center (CC) or an operator reasons about potential problems in the power grid based on the SE output. If anomaly detection thresholds are exceeded then actions are taken against the problems and the effects of the problems. Static SE [155, 154] relies on a single set of measurements all taken at one snapshot in time, whereas dynamic SE [109, 181] covers the evolution of the state over consecutive measurement instants and provides accurate dynamic states of the system.

A power flow model uses a set of equations representing the energy flow on transmission lines of a power grid. An AC power flow model considers real power and reactive power formulated by nonlinear equations, which is computationally expensive and may not converge to a solution [100]. So power engineers often use a linearized model called a DC model for approximating the AC power flow model. Our research considers the linear DC model.

We denote the n states $x_1, x_2, ..., x_n$ (n is the number of sates) and m measurements $z_1, z_2, ...z_m$ from the m meters. A system state of n is represented as state vector by Eq. (4.1)

$$\boldsymbol{x} = (x_1, x_2, ..., x_n)^T \tag{4.1}$$

Similarly, a measurement set from the m meters is represented as a measurement vector by Eq. (4.2)

$$\boldsymbol{z} = (z_1, z_2, ..., z_m)^T \tag{4.2}$$

where $m \geq n$. The measurement error $\boldsymbol{v}$ is expressed in Eq. (4.3)

$$\boldsymbol{v} = (v_1, v_2, ..., v_m)^T \tag{4.3}$$

The measurements set depends on the state which is based on a function $\boldsymbol{h}(\boldsymbol{x})$ and the measurement error $\boldsymbol{v}$. It is represented by Eq. (4.4)

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \tag{4.4}$$

where $\boldsymbol{h}(\boldsymbol{x}) = (h_1(x), ..., h_m(x))^T$.

By considering the DC model, the relationship between measurements and states is represented as a linear relation, which is expressed in Eq. (4.5).

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \tag{4.5}$$

where $\boldsymbol{H}$ is a $m \times n$ matrix. It depends on the measurements and state variables. Due to the linear relation it is called a linear model.

State estimations of variables (e.g, voltage, current [138], phase angle [42],) can be done based on meter/sensor measurements (e.g., from a PMU). Multiple measurements from different sensors/meters can be used for estimating states.

Existing works (e.g., [139, 137, 150, 77]) provide approaches for linear SE based on three phases. If the phases are mutually coupled, SE depends on three phases. We do not have any information about the mutual coupling between the phases. In a similar manner to [176], we make the assumption that the phases are independent of each other, so that we can estimate the states of each phase separately.

We use PMU measured voltage-phasors for estimating the system states. Using rectangular coordinates allow us to apply linear SE [11]. Thus, the one phase system true state $\boldsymbol{x}$ is represented by real voltage and imaginary voltage Eq. (4.6).

$$\boldsymbol{x} = [V_{re_{true}}, V_{im_{true}}]^T \tag{4.6}$$

With the linear SE, the relationship between the measurements $\boldsymbol{z}$ and state $\boldsymbol{x}$ can be represented as Eq. (4.7).

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \tag{4.7}$$

where $\boldsymbol{H}$ is an identity matrix, and $\boldsymbol{v}$ is a measurement noise.

Using this simple model we apply the two SE methods, which have been compared in [150]: linear weighted least squares (LWLS) and discrete Kalman filter (DKF).

4.1 Weighted Least Squares

We can express the measurement from a sensor $\boldsymbol{z}_k$ at time step k by the true state $\boldsymbol{x}_k$ and the measurement noise $\boldsymbol{v}_k$ and the matrix $\boldsymbol{H}$ describes how the state is related to the measurements.

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.8}$$

We assume the measurement noise to be Gaussian $p(v) \sim N(0, \boldsymbol{R})$ with covariance matrix $\boldsymbol{R}$ and consider the measurement noise covariance matrix $\boldsymbol{R}$ as time-invariant [176].

SE using LWLS minimizes the objective function [50] represented by Eq. (4.9).

$$J(x) = \sum_{j=1}^{N} \frac{(z_j - \sum_{r=1}^{S} H_{jr} x_r)^2}{R_{jj}} \tag{4.9}$$

where N is the number of measurements, S is the number of states and R_{jj} is the variance of the j^{th} measurement.

With LWLS the estimated state $\hat{\boldsymbol{x}}_{LWLS,k}$ at time step k is calculated from the measurement $\boldsymbol{z}_k$, the matrix $\boldsymbol{H}$, the noise covariance matrix $\boldsymbol{R}$ and the gain matrix $\boldsymbol{G}$ as follows in Eq. (4.10) as shown in [50].

$$\hat{\boldsymbol{x}}_{LWLS,k} = \boldsymbol{G}^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{z}_k \tag{4.10}$$

with

$$\boldsymbol{G} = \boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H} \tag{4.11}$$

4.2 Kalman Filter

Table 4.1: Notation used in Kalman Filtering

Notation	Description
$\boldsymbol{A}$	State-transition model
$\boldsymbol{B}$	Control input model
$\boldsymbol{H}$	Observation model
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{u}$	Control input
$\boldsymbol{v}$	Measurement noise
$\boldsymbol{v}_k$	Measurement noise (time-variant)
$\boldsymbol{w}$	Process noise
$\boldsymbol{w}_k$	Process noise (time-variant)
$\boldsymbol{Q}_k$	Process noise covariance matrix (time-variant)
$\boldsymbol{R}$	Measurement noise covariance matrix
$\boldsymbol{P}_{k\|k-1}$	Predicted process covariance matrix (time-variant)
$\boldsymbol{P}_{k\|k}$	Process covariance matrix (time-variant)
$\boldsymbol{z}_k$	Actual measurement (time-variant)
$\boldsymbol{z}$	Actual measurement
$\boldsymbol{z}_v$	Observed measurement
$\boldsymbol{z}_e$	Estimated measurement
$\boldsymbol{z}_{vk}$	Observed measurement (time-variant)
$\boldsymbol{y}_k$	Pre-fit residual (time-variant)
$\boldsymbol{y}_{k/k}$	Post-fit residual (time-variant)
$\boldsymbol{x}_k$	Real state (time-variant)
$\hat{\boldsymbol{x}}_{k\|k-1}$	Predicted state (time-variant)
$\hat{\boldsymbol{x}}_{k\|k}$	Estimated state (time-variant)
$\boldsymbol{L}_k$	Kalman gain (time-variant)
γ	Decision level (in DKF)

Kalman filters are widely used for SE in different domains. Kalman filters estimate a system state $\boldsymbol{x} \in \mathbb{R}^m$ based on the previous state and additional variables. $\boldsymbol{x}_k$ represents

the power system state at the current time step k. In the Kalman filter model, the true system state at time k is represented by the linear equation

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k \tag{4.12}$$

where $\boldsymbol{A}$ is a $m \times m$ matrix that links a system state at time step k to the previous state [170]. In our case (estimating the voltage state from voltage measurements), $\boldsymbol{A}$ is an identity matrix [176].

$\boldsymbol{u}_k \in \mathbb{R}^l$ represents a set of control variables at time step k, and $\boldsymbol{B}$ is a $m \times l$ matrix that relates the system state to control variables at time step k. $\boldsymbol{w}_k \in \mathbb{R}^m$ represents the process noise at time step k, which is assumed to be Gaussian white noise $p(w) \sim N(0, \boldsymbol{Q}_k)$ with covariance matrix $\boldsymbol{Q}_k$.

Normally, the true state in a system is not observable, but one can perform measurements that are influenced by measurement noise $\boldsymbol{v}$. Like in Eq. (4.8), a measurement from a sensor at time k is represented by $\boldsymbol{z}_k$ in Eq. (4.13), wherein $\boldsymbol{v}_k$ is the measurement noise and the matrix $\boldsymbol{H}$ describes how the state is related to the measurements.

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.13}$$

All of the notations used in this section (Kalman filtering) are illustrated in Tab. 4.1. Types of Kalman filters (e.g., DKF [170], extended Kalman filter - EKF [73, 12] are used for estimating states of different systems. For instance DKF is used for linear state estimation (LSE), EKF is used for solving non-linear problems. In our work, we only use DKF as LSE has a lower computational complexity than non-linear SE.

4.2.1 Discrete State Kalman Filter

In a steady state Kalman filter model, the noise covariances (process noise and measurement noise) do not change over time. A case study in [105] illustrates the Kalman filter steady state design. A state-space model uses state variables to describe a system. It describes a system by a set of first order differential equations. A discrete plant as expressed in [105] is a state-space system. The state-space system model has process noise $\boldsymbol{w}$, measurement noise $\boldsymbol{v}$ and a control signal $\boldsymbol{u}$ as inputs, and real measurement $\boldsymbol{z}$ and measured signal $\boldsymbol{z}_v$ as outputs. Details of the system and its parameters are in Appendix A.2.

The model assumes the discrete plant with an additive Gaussian measurement and process noise as an input to the model. The process noise vector $\boldsymbol{w}$ is chosen from a normal distribution with covariance $\boldsymbol{Q}$ and the measurement noise vector $\boldsymbol{v}$ is chosen from a normal distribution with covariance $\boldsymbol{R}$.

A DKF model (Kalman state estimator) [52, 89] is designed by combining the discrete plant and the Kalman function [104].

DKFs are used for SE when measurements occur at discrete times. Here, we use LSE using DKFs [27] to estimate power system voltage states, based on PMU measurements. Thus, the voltage states have a linear dependency on the PMU measurements. The SE has two stages, named *prediction* and *estimation*. Figure 4.1 depicts the process for estimating states using a DKF. Table 4.1 summarizes the notation that we use for the Kalman filter.

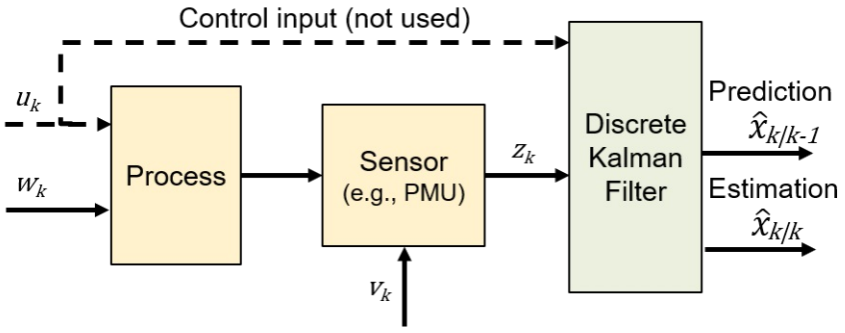


Figure 4.1: DKF Model for measuring and estimating states (source Paudel et al. [132]).

In our scenario, we do not have any control input [176]. Therefore, the model is reduced to the influence of the previous state and the process noise [138]:

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{4.14}$$

$\boldsymbol{Q}_k$ is the process noise covariance matrix. We assume that $\boldsymbol{Q}_k$ is time-variant and Gaussian, i.e., the covariance matrix of the process noise changes over time and use a heuristic method to calculate $\boldsymbol{Q}_k$ [177]. The heuristic method depends on the estimation error which varies over time.

Here we recall a measurement from a sensor at time k is represented by $\boldsymbol{z}_k$ in Eq. (4.13). In our case, measurements are taken by PMUs.

We assume the measurement noise also to be Gaussian $p(v) \sim N(0, \boldsymbol{R})$ with covariance matrix $\boldsymbol{R}$ and consider the measurement noise covariance matrix $\boldsymbol{R}$ as time-invariant [176]. This means the noise factor is random, but the distribution of the noise does not change over time. In a similar manner as in [176], we assume measurements in a phase are treated separately and project uncertainty of conversion from polar to rectangular coordinates with known $\boldsymbol{R}$. The matrix $\boldsymbol{H}$ is related to the real and imaginary parts of the measurements. If a measurement is taken from the same power system bus and phase, as in our scenario, $\boldsymbol{H}$ is an identity matrix [176].

4.2.1.1 Prediction and Estimation

In the *prediction* step, an *a priori* prediction of the current state is determined, based on the previous estimated state ($\hat{\boldsymbol{x}}_{k-1|k-1}$). This is also called the "time-update" step,

because an update is performed that is looking forward in time prospective. The prediction at time k is determined using Eq. (4.15).

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_k \hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{w}_k \tag{4.15}$$

The predicted process covariance matrix depends on the previous process covariance matrix ($\boldsymbol{P}_{k-1|k-1}$) and the current process noise covariance matrix ($\boldsymbol{Q}_k$), where $\boldsymbol{P}_{k-1|k-1}$ is a $m \times m$ matrix:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1|k-1} + \boldsymbol{Q}_k \tag{4.16}$$

Meanwhile, for the *estimation* step, an *a posteriori* estimate of the current state is performed, based on the predicted state and the observed measurements. This is also called the "measurement-update" or "correction" step, because the prediction is corrected with real measurements.

SE is based on the predicted state ($\hat{\boldsymbol{x}}_{k|k-1}$) and the observation vector ($\boldsymbol{z}_k$), represented by Eq. (4.17).

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.17}$$

Here, the Kalman gain (KG) $\boldsymbol{L}_k$ describes the relative weight of the measurements and the current estimated state, i.e., it represents how trustworthy are the measurements.

If the KG is high, the filter puts more weight on the most recent measurements and therefore follows them more responsively. If the KG is low, the filter follows the model predictions more closely and puts less trust in the measurements. The KG is between zero and one and can be used to measure the performance of the filter.

Eq. (4.18) shows the calculation of the Kalman gain $\boldsymbol{L}_k$ for time step k.

$$\boldsymbol{L}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{4.18}$$

$\boldsymbol{L}_k$ is a $m \times n$ matrix that is chosen to minimize previous estimate error covariances. If the measurement covariance $\boldsymbol{R}$ is small, there is more trust in the measurement.

The process covariance matrix is updated based on the KG ($\boldsymbol{L}_k$) and the predicted process covariance matrix ($\boldsymbol{P}_{k|k-1}$), represented by Eq. (4.19), where $\boldsymbol{I}$ is an identity matrix.

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{L}_k\boldsymbol{H})\boldsymbol{P}_{k|k-1} \tag{4.19}$$

In this work, we use the LSE method that is proposed by Pignati et al. [139] and Møller et al. [113]. The main reason for using LSE is that it has a lower computational complexity than non-linear SE. For the measurement error a standard deviations of real voltage and imaginary voltage of 0.001 p.u. are assumed as suggested in [150, 149]. The elements of process noise ($\boldsymbol{Q}_k$) are initialized to 0.01, and we calculate the elements of $\boldsymbol{Q}_k$ at each step, as proposed by Method 2 in [177]. The filter we use for SE is partially steady state as only the measurement noise covariance matrix $\boldsymbol{R}$ is time-invariant.

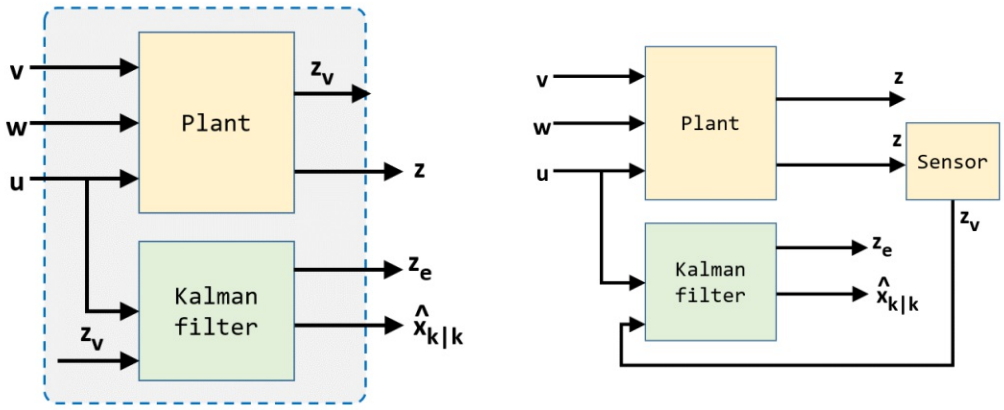
4.2.2 Kalman Filter Example

In this section, an example is implemented for validating the concept of filtering.

A Matlab command represented in Eq. (4.20) returns a state space model $kalmf$, Kalman gain $\boldsymbol{L}$, innovation gain $\boldsymbol{M}$ (chosen in order to minimize estimation error covariance) and steady state error covariance $\boldsymbol{P}$. $Kalmf$ model has two outputs, estimated plant output $\boldsymbol{z}_e$ and estimated state $\hat{\boldsymbol{x}}_{k|k}$. We only take into consideration the estimated state $\hat{\boldsymbol{x}}_{k|k}$ and discard $\boldsymbol{z}_e$.

$$[kalmf, \boldsymbol{M}, \boldsymbol{P}, \boldsymbol{L}] = kalman(Plant, \boldsymbol{Q}, \boldsymbol{R}) \tag{4.20}$$

A parallel connection between the plant and the Kalman filter are created using Matlab command $parallel(sys1, sys2)$ [106]. For a parallel connection both of the connecting models should be either continuous or discrete. In our case we consider both of them as discrete models. Here a system is formed connecting the plant and the Kalman filter in parallel. The steady state Kalman filter model is shown in Fig. 4.2. Sub-figure 4.2a shows the original model, parallel connection, inputs to the model and outputs from the model. As shown in this sub-figure, in the original model (an example) we do not have access to the observed measurement z_v, which is used as input to the Kalman filter. The model only provides access to the original measurement z (without the noise component). Measurement noise $\boldsymbol{v}$ (input to plant) and actual measurement $\boldsymbol{z}$ are combined internally and fed to Kalman filter using a positive feedback but the value is not exported. The dotted line shows one could not access or see the values.



(a) Original

(b) Modified

Figure 4.2: Steady state Kalman filter.

As an attacker we want to manipulate $\boldsymbol{z}_v$. For the manipulation we need to access the sensor or $\boldsymbol{z}_v$. So we modify the model such that we can access the measured value ($\boldsymbol{z}_v$). Sub-figure 4.2b shows how we split up the sensor from the plant, so that we get $\boldsymbol{z}_v$.

A sensor which has sensor noise ($\boldsymbol{v}$) is connected to the output of the plant. In other words the sensor connects the plant's output ($\boldsymbol{z}$) to the filter input ($\boldsymbol{z}_v$) with a positive feedback [103]. The estimated value of $\boldsymbol{z}_v$ by the filter is $\boldsymbol{z}_e$ and the estimated state based on $\boldsymbol{z}_v$ is $\hat{\boldsymbol{x}}_{k|k}$; $\boldsymbol{z}_e = \boldsymbol{H} \cdot \hat{\boldsymbol{x}}_{k|k}$ where $\boldsymbol{H}$ relates states to measurement. From the connection we can see in the figure $\boldsymbol{z}_v$ is the output of the plant and also an input to the Kalman filter.

We adopt this model to our use case in the power system. Modifications will be discussed in Sec. 4.2.2.1.

4.2.2.1 Model Modification

Aim of modification: The original model described above in this section has inputs $(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v})$ and outputs $(\boldsymbol{z}, \boldsymbol{z}_e, \hat{\boldsymbol{x}}_{k|k})$. As we use PMU measured values for our experiment, as an attacker we want to manipulate the measured response $\boldsymbol{z}_v$ which is the input to the Kalman filter. Therefore, we want to have access to the measured response in the above model. In addition we want to check innovation (residuals) $\boldsymbol{y}_k$, Kalman gain $\boldsymbol{L}_k$ for all iterations.

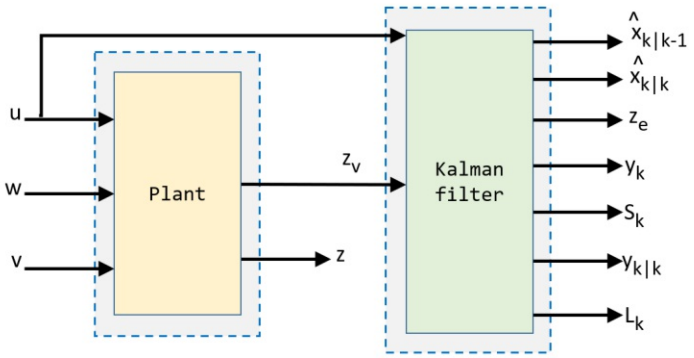


Figure 4.3: Modified model with all output values.

Modification: We modify the model such that $\boldsymbol{z}_v$ is assigned as input to the Kalman filter. We simulate the plant and Kalman filter separately. At first we simulate the plant, which has two outputs: true response and measured response. We assign the measured response from the plant as input to Kalman filter. Using the estimated output and estimated state from the filter we calculate $\boldsymbol{y}_k$ and $\boldsymbol{L}_k$. Figure 4.3 shows modified version of the Kalman filter.

4.2.2.2 Experiment

Input signal is a sinusoidal signal $u = sin(t/5)$ where t increases from 0 to 100 ($t = [0 : 100]$ discrete time steps). We set process noise covariance $\boldsymbol{Q}$ to 1. Similarly, measurement

noise covariance $\boldsymbol{R}$ is set to 1. Process noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$ are randomly selected noise of Gaussian distribution.

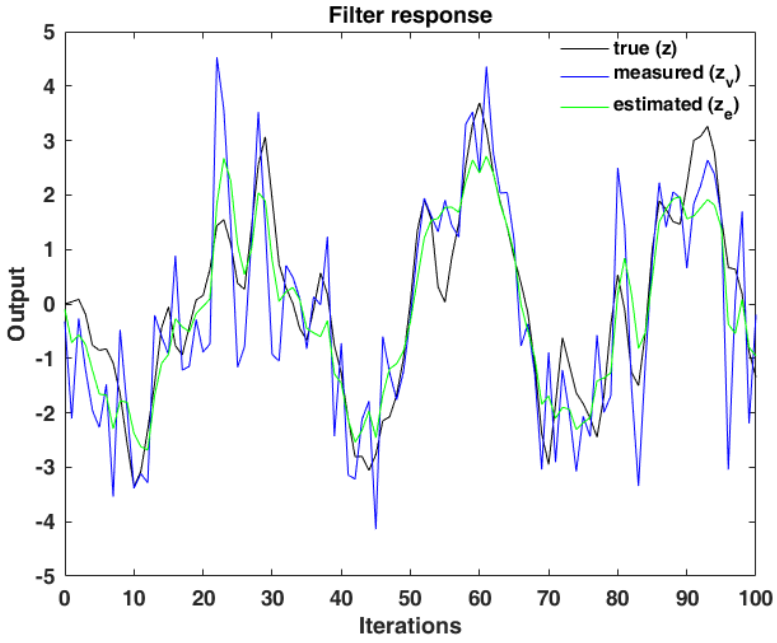


Figure 4.4: True signal, measured signal and estimated signal using the Kalman filter.

The measured signal is a combination of true signal and sensor noise (an additive Gaussian noise). Figure 4.4 shows true, measured and estimated signal of the steady state Kalman filter.

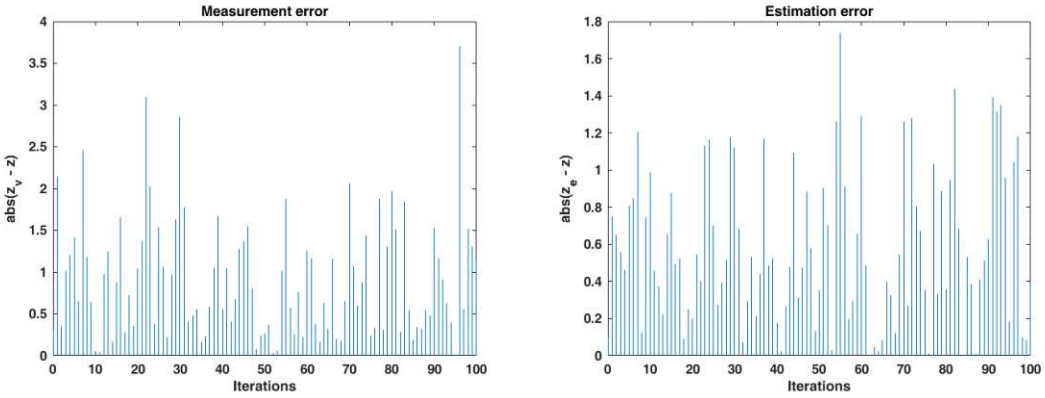
Sensor connected to the model can cause measurement error ($\boldsymbol{z}_v - \boldsymbol{z}$) and estimation error ($\boldsymbol{z}_e - \boldsymbol{z}$). The measurement error due to the sensor noise is shown in sub-figure 4.5a and estimation error is shown in sub-figure 4.5b.

Thus the filtering process filters out some noise such that the estimation error is less than the measurement error. It indicates that after the filter process, the signal gets closer to real signal than the measured signal. Similarly, pre-fit residuals and post-fit residuals are shown in Fig 4.6. Pre-fit residuals in sub-figure 4.6a is greater than post-fit residuals in sub-figure 4.6b.

4.3 Use Case

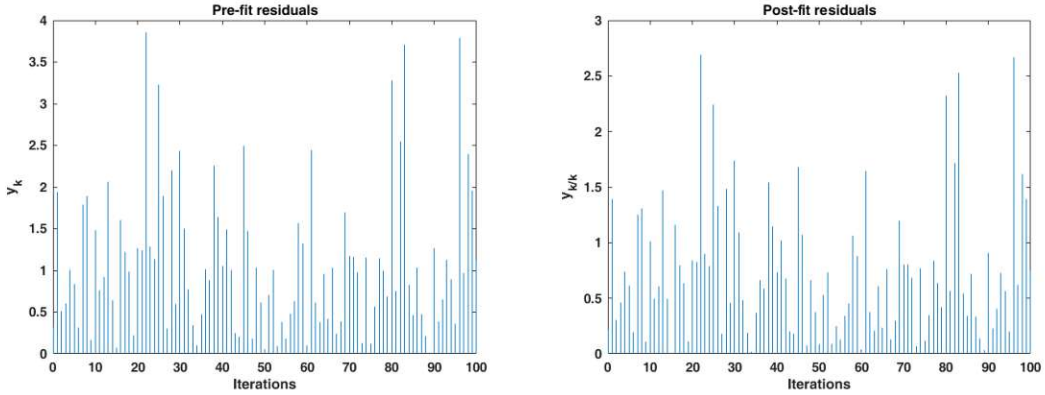
4.3.1 Power System Measurements

A measurement model connects the measurements to the system state via a matrix $\boldsymbol{H}$ which we name as measurement matrix [176]. In this section, we present the measurement



(a) Measurement error (b) Estimation error

Figure 4.5: Measurement error and estimation error in normal operation.



(a) Pre-fit residuals (b) Post-fit residuals

Figure 4.6: Pre-fit residuals and post-fit residuals in normal operation.

model and the measurement matrix used in our use case.

4.3.1.1 Measurement Model

A bus is a vertical line where components (e.g., loads, generators) are connected, and a path between two buses is a branch. A measurement model [176, 149] (represented by Eq.(4.5)) links the sets of measurements measured in buses, branches of a power system to the system's state variables. Let $\boldsymbol{S}$ be a set of buses and $\boldsymbol{N}$ be the set of state variables of a one-phase-power-system. Then the network state is represented as $\boldsymbol{x} \in \mathbb{R}^n$ where $n = 2s$, $s = |S|$, and $n = |N|$. The state $\boldsymbol{x}$ at all buses is represented as Eq. (4.21)

$$\boldsymbol{x} = [V_1, .., V_i, .., V_n]^T \tag{4.21}$$

We consider only one bus ($n = 2$) in the following, so the state vector $\boldsymbol{x}$ at the bus is reduced to Eq. (4.22)

$$\boldsymbol{x} = [V_1, V_2]^T \tag{4.22}$$

Let M be the set of phasor-measurements including voltage-phasors and current-phasors, then measurements set $\boldsymbol{z} \in \mathbb{R}^m$ where $m = |M|$ is represented as Eq. 4.23.

$$\boldsymbol{z} = \begin{bmatrix} z_V \\ z_I \end{bmatrix} \tag{4.23}$$

where $\boldsymbol{z}_V$ is the set of voltage-phasors and $\boldsymbol{z}_I$ is the set of current-phasors.

Current measurements contain a set of current injection and current flow in a system. Current injection in a node or a bus, and current flow in a branch can be calculated using the complex current-phasors. Sets of voltage, current injection and current flow measurements are based on the sets of buses and branches. Thus Eq. 4.23 can be rewritten as Eq. (4.24).

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_V \\ \boldsymbol{z}_{I_{inj}} \\ \boldsymbol{z}_{I_{flow}} \end{bmatrix} \tag{4.24}$$

where $\boldsymbol{z}_{I_{inj}}$ is the set of current injection phasors and $\boldsymbol{z}_{I_{flow}}$ is the set of current flow phasors represented in Eq. (4.25).

$$\begin{aligned} \boldsymbol{z}_V &= [V_1, ..., V_i, ..., V_n] \\ \boldsymbol{z}_{I_{inj}} &= [I_1, ..., I_i, ..., I_n] \\ \boldsymbol{z}_{I_{flow}} &= [I_1, ..., I_i, ..., I_n] \end{aligned} \tag{4.25}$$

PMUs provide synchrophasor measurements, and they can be expressed in rectangular coordinates. Thus SE is linear when we express state (Eq. 4.22) and measurements (Eq. 4.25) in rectangular coordinates [167, 78, 76]. States (Eq. 4.22 in rectangular coordinates are represented as Eq. (4.26).

$$\boldsymbol{x} = [V_{1,re}, ..., V_{n,re}, V_{1,im}, ..., V_{n,im}]^T \tag{4.26}$$

Similarly, measurements (Eq. 4.25) in rectangular coordinates are represented as Eq. (4.27).

$$\begin{aligned} \boldsymbol{z}_V &= [V_{1,re}, ..., V_{n,re}, V_{1,im}, ..., V_{n,im}] \\ \boldsymbol{z}_{I_{inj}} &= [I_{1,re}, ..., I_{n,re}, I_{1,im}, ..., I_{n,im}] \\ \boldsymbol{z}_{I_{flow}} &= [I_{1,re}, ..., I_{n,re}, I_{1,im}, ..., I_{n,im}] \end{aligned} \tag{4.27}$$

4.3.1.2 Measurement Matrix

The linear measurement model linearly relates the measurements to the state variables. Therefore matrix $\boldsymbol{H}$ links measurements to states. Matrix $\boldsymbol{H}$ is time-invariant because it

depends on the network topology, and electric parameters of the power system. It needs recalculation only if there is a change in the network topology.

$\boldsymbol{H}$ is related to voltage-phasors, current injection and current flow phasors. Therefore, $\boldsymbol{H}$ is splitted in three parts as represented by Eq. (4.28).

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_V \\ \boldsymbol{H}_{I_{inj}} \\ \boldsymbol{H}_{I_{flow}} \end{bmatrix} \tag{4.28}$$

H for voltage-phasor measurements Elements of $\boldsymbol{H}_V$ are either ones or zeros, and they link real or imaginary part of the voltage measurement to real or imaginary part of the state variable. $\boldsymbol{H}_V$ is defined as

$$\boldsymbol{H}_V = \begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix} \tag{4.29}$$

where h_1 links real part to real part of voltage to real part of state, h_2 links real part of the voltage to imaginary part of state, h_3 links imaginary part of voltage to real part of state and h_4 links imaginary part of voltage to imaginary part of the state.

$\boldsymbol{H}_v$ for a bus is represented by Eq.(4.30). Derivation of $\boldsymbol{H}_v$ for multiple buses is shown in Appendix A.3.1.1.

$$\boldsymbol{H}_V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.30}$$

H for current-injection-phasor measurements Elements of $\boldsymbol{H}_{I_{inj}}$ are the elements of the admittance matrix, and they link current-injection measurement to the voltage measurement. Admittance matrix for s buses network can be defined as a $\boldsymbol{Y}_{s\times s}$ matrix, and it is represented by Eq. (4.31) [14, 76].

$$\boldsymbol{Y} = \begin{bmatrix} Y_{11} & ... & Y_{1s} \\ ... & ... & .... \\ Y_{s1} & ... & Y_{ss} \end{bmatrix} \tag{4.31}$$

Real and imaginary parts of the admittance matrix can be written as Eq. (4.32).

$$\boldsymbol{Y} = G + jB \tag{4.32}$$

where G is the real part and B is the imaginary part of $\boldsymbol{Y}$.

$\boldsymbol{H}_{I_{inj}}$ for one bus is represented by Eq. (4.33). Derivation of $\boldsymbol{H}_{I_{inj}}$ for multiple buses is shown in Appendix A.3.1.2.

$$\boldsymbol{H}_{I_{inj}} = \begin{bmatrix} G & -B \\ B & G \end{bmatrix} \tag{4.33}$$

H for current-flow-phasor measurements Elements of $\boldsymbol{H}_{I_{flow}}$ link current-flow measurement to the voltage measurement. Thus elements of $\boldsymbol{H}_{I_{flow}}$ are the functions of state variables. Considering a two port π model of transmission lines as shown in Fig. 4.7, we can derive the elements of $\boldsymbol{H}_{I_{flow}}$. A two port π model has two buses l and h at two ends of a transmission line. This transmission line is called a branch between the buses l and h. The model has π-longitudinal admittance (also called as series admittance) and π-transverse admittance (also called as shunt admittance).

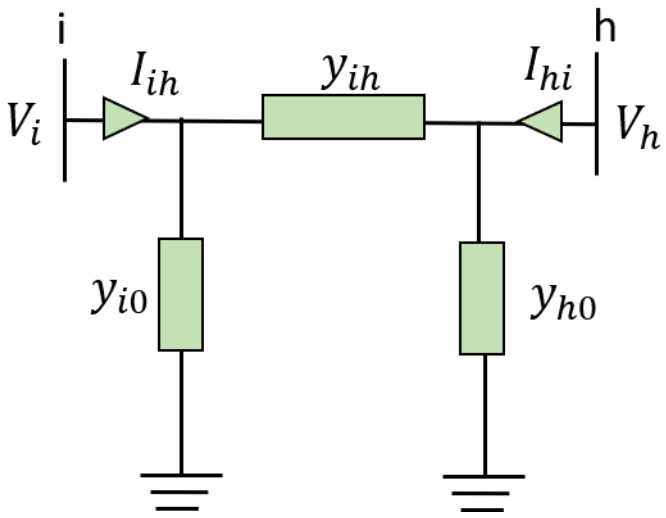


Figure 4.7: 2 port π model of a transmission line (adapted from source Abur et al. [14]).

The π-longitudinal admittance is the inverse of the π-longitudinal impedance z_{lh} of the transmission line (or the branch). The impedance in rectangular coordinates is represented by Eq. (4.34)

$$z_{lh} = r_{lh} + jx_{lh} \tag{4.34}$$

where r_{lh} is resistance and x_{lh} is the reactance. Thus the π-longitudinal admittance is represented as Eq. (4.35)

$$y_{lh} = \frac{1}{z_{lh}} \tag{4.35}$$

The π-transverse admittance y_{l0} from the side of bus l in rectangular coordinates is represented by Eq. (4.36)

$$y_{l0} = g_{l0} + jb_{l0} \tag{4.36}$$

where g_{l0} is conductance and b_{l0} is susceptance from the side of the bus l.

Similarly, π-transverse admittance matrix y_{h0} from the side of the bus h in rectangular coordinates is represented by Eq. (4.37)

$$y_{h0} = g_{h0} + jb_{h0} \tag{4.37}$$

where g_{h0} is conductance and b_{h0} is susceptance from the side of the bus h.

Current flow in a branch from the two ends l and h of the branch are different. We denote the current flow phasor in the branch from bus l to h by I_{lh}, and the current flow phasor from bus h to bus l by I_{hl}. Calculation of I_{Ih} using voltage phasors (or using state variables) is represented by Eq. (4.38) [176].

$$I_{lh} = y_{lh}(V_l - V_h) + y_{l0}V_l \tag{4.38}$$

where V_l is the voltage at bus l and V_h is the voltage at bus h.

Real part of the current flow phasor I_{lh} is represented by Eq. (4.39)

$$I_{lh,re} = g_{lh}(V_{l,re} - V_{h,re}) - b_{ih}(V_{l,im} - V_{h,im}) + g_{l0}V_{l,re} - b_{l0}V_{l,im} \tag{4.39}$$

Imaginary part of the current flow phasor I_{lh} is represented by Eq. (4.40)

$$I_{lh,im} = g_{lh}(V_{l,im} - V_{h,im}) + b_{lh}(V_{l,re} - V_{h,re}) + g_{l0}V_{l,im} + b_{l0}V_{l,re} \tag{4.40}$$

Elements of $\boldsymbol{H}$ that relate current flow measurements to the state variables are defined as a matrix $\boldsymbol{H}_{I_{flow}}$ in Eq. (4.41). Each element of $\boldsymbol{H}$ is defined for relating real part (re) and imaginary part (im) of the measurements and states variables. In our use case we have one bus, so the matrix $\boldsymbol{H}_{I_{flow}}$ for a bus is represented as Eq. (4.41) (see Appendix A.3.1 for multiple buses).

$$\boldsymbol{H}_{I_{flow}} = \begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix} \tag{4.41}$$

where

$$h_1{}^{re}_{re} = g_{lh} + g_{l0} \tag{4.42}$$

$$h_2{}^{re}_{im} = -(b_{lh} + b_{l0}) \tag{4.43}$$

$$h_3{}^{im}_{re} = b_{lh} + b_{l0} \tag{4.44}$$

$$h_4{}^{im}_{im} = g_{lh} + g_{l0} \tag{4.45}$$

The superscripts in (4.42) to (4.45) mean the real part (re), the imaginary part (im) of the measurements; and the subscripts mean the real part (re) and the imaginary part (im) of the state variables.

4.3.2 Power System Use Case

We use PMUs measurements of a smart grid network in EPFL campus. Electric parameters of lines used in the network are resistance $R = 0.159\ \Omega/km$, reactance $X = 0.113\ \Omega/km$. It is a short lines network as all of the lines are less than a kilometer. We do not have information about the shunt capacitance of the network, and according to

Reta-Hernandez [144], for transmission lines less than 80 km effect of shunt capacitance is negligible. In this case we consider only the available information of resistance and reactance of the lines. We also do not have any information of mutual coupling of the phases, therefore we make as assumption that phases are independent to each other [176].

We consider a model which has PMUs deployed in each of the nodes (buses) and have a branch between the nodes. Current injection in a node is measured by a PMU installed in the node. Current injection to the node and current flow from the side of the node have different signs due to their directions. If there is only one branch between two nodes (as depicted in Fig. 4.8) under the above circumstances (shunt admittance is negligible), then using Kirchhoff's law we can say current injection in a node equals the current flow from the side of the node, only the difference is the positive/negative sign. Here we do not distinguish between flow or injection and use the term current for representing the current magnitude. Thus we develop a one-phase SE model intended to be deployed at each node/bus. It uses PMU measurements at each node for estimating the states.

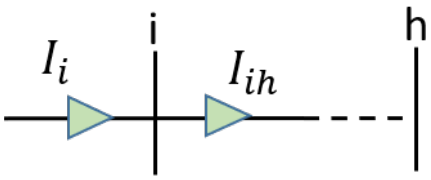


Figure 4.8: Current injection to a bus and current flow from the side of the bus in a branch.

PMU measurements at each node is represented by Eq. (4.46).

$$\boldsymbol{z} = \begin{bmatrix} V \\ I \end{bmatrix} \tag{4.46}$$

where $\boldsymbol{z}$ is a set of voltage V and current I measurements.

Separation of real and imaginary parts of the measurement $\boldsymbol{z}$ in Eq. (4.46) is as follows:

$$\boldsymbol{z}_{re} = \begin{bmatrix} V_{re} \\ I_{re} \end{bmatrix} \tag{4.47}$$

$$\boldsymbol{z}_{im} = \begin{bmatrix} V_{im} \\ I_{im} \end{bmatrix} \tag{4.48}$$

We consider the voltage phasor for states, which we express with real and imaginary part. Thus, state at each node in rectangular coordinates is represented by Eq. 4.49.

$$\boldsymbol{x} = \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} \tag{4.49}$$

As PMU measurements are the complex phasors of voltage and current. Separation of real and imaginary parts of the system state $\boldsymbol{x}$ in Eq. (4.49) makes SE less complex. Separation of the state's real and imaginary parts is as follows:

$$x_{re} = \begin{bmatrix} V_{re} \end{bmatrix} \tag{4.50}$$

$$x_{im} = \begin{bmatrix} V_{im} \end{bmatrix} \tag{4.51}$$

Relationship between the measurements and the system state can be written as Eq. (4.52).

$$\boldsymbol{z} = \begin{bmatrix} H_V \\ H_I \end{bmatrix} \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} + \boldsymbol{v} \tag{4.52}$$

where $\boldsymbol{H}_V$ is the identity matrix for voltage measuremetns and $\boldsymbol{H}_I$ is the admittance matrix for current measurement as shown in Equations (4.30) and (4.33) respectively.

Here again we recall the SE using both LWLS and DKF. Using LWLS method, estimated state at time step k is represented by Eq. (4.53) [100].

$$\hat{\boldsymbol{x}}_{LWLS,k} = \boldsymbol{G}^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{z}_k \tag{4.53}$$

where $\boldsymbol{G}$ is a gain matrix.

Similarly, using DKF method, estimated state at time step k is represented by Eq. (4.54).

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.54}$$

where $\hat{\boldsymbol{x}}_{k|k-1}$ is the predicted state at time step k and $\boldsymbol{L}_k$ is kalman gain at time step k.

The states can be estimated in two ways i) using only voltage measurements or ii) using both voltage and current measurements. We present these two cases in the following sections.

4.3.2.1 Experimental setup

EPFL campus PMU network [47] is part of the electrical distribution network. A 20 kV active distribution network (ADN) connects PMUs via a communication network. The PMUs in the network are intended to meet the requirements of IEEE Std C37.118.1-2011 [5] and IEEE Std C37.118.1a-2014 [7] for the synchrophasor measurements of power systems. Some adjustments to the PMUs have been made for them to be used in an ADN; therefore, the PMUs that are described in [145] are used. In this setting, UDP datagrams are encapsulated according to IEEE Std C37.118.2-2011 [4] and communicated over a secured communication network. A detail description of the system architecture and characteristics of the PMUs is presented in [138]. The base voltage of the PMU network is 11547.0054 kV.

Polar voltages are derived from PMU provided voltage magnitudes and base voltage of the network. Series impedance is calculated using resistance and reactance. First, we calculate admittance using the impedance then calculate conductance (G) and suceptance (B) using the admittance. According to closely related existing work [176] we define standard deviations of real voltage, imaginary voltage, real current and imaginary current current as 0.001 p.u. We keep the phase angle fixed by the first observed phase angle in order to be able to apply the residual-based detection method, as explained in Section 6.2.

Let n be the number of states using m number of measurements at time step k. Table 4.2 shows matrices and their dimensions in SE using DKF and LWLS. In our scenario, true system state $\boldsymbol{x} = [V_{re}, V_{im}]^T$. Thus the number of states is $n = 2$. We initialize the true state $\boldsymbol{x}$ with the observed state and separate SE in two cases i) using only voltage measurements i.e., $m = 2$, $\boldsymbol{z} = \boldsymbol{z}_V$ where $\boldsymbol{z}_V = [V_{re}, V_{im}]^T$ and ii) using both voltage and current measurements i.e., $m = 4$ and $\boldsymbol{z} = [\boldsymbol{z}_V, \boldsymbol{z}_I]^T$ where $[\boldsymbol{z}_V, \boldsymbol{z}_I]^T = [V_{re}, V_{im}, I_{re}, I_{im}]^T$.

Table 4.2: Matrices of Kalman Filter and their dimensions.

Notation	**Dimension**	**Description**
x_k	$n \times 1$	True state (time-variant)
z_k	$m\times 1$	Measurement (time-variant)
L_k	n $\times m$	Kalman gain (time-variant)
$P_{k/k-1}$	$n \times n$	Predicted process covariance (time-variant)
$P_{k/k}$	$n \times n$	Process covariance (time-variant)
Q_k	$n \times n$	Process noise covariance (time-variant)
A	$n \times n$	State transition
H	$m \times n$	Observation model
R	$m \times m$	Measurement noise covariance
$\hat{x}_{k/k-1}$	$n \times 1$	Predicted state (time-variant)
$\hat{x}_{k/k}$	$n \times 1$	Estimated state (time-variant)
G	$n \times m$	LWLS gain
$\hat{x}_{LWLS,k}$	$n \times 1$	LWLS Estimated state

In the attack scenario, real voltage and imaginary voltage are calculated from manipulated polar voltage whereas real current and imaginary current are calculated from actual (non-manipulated) polar voltage. Voltage measurements under attack are $\boldsymbol{z}_V = [V_{re,m}, V_{im,m}]^T$ and current measurements are $\boldsymbol{z}_I = [I_{re}, I_{im}]^T$. The measurements are fed to the estimator. Here non-manipulated measurements are illustrated without any additional subscripts.

In Kalman filtering, we initialize process noise $\boldsymbol{Q}_k$ as diag$(0.01^2, 0.01^2)$ where standard deviations of estimated real state and imaginary states are assigned as 0.01.

4.3.2.2 Using only voltage measurements

Two measurements (real voltage and imaginary voltage) are used for estimating two states (real state and imaginary state). In this setup, we have a 2×2 identity matrix $\boldsymbol{H}$ and a 2×2 measurement covariance matrix $\boldsymbol{R}$, as shown below

$$\boldsymbol{R} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \tag{4.55}$$

where σ_1^2 is variance of real voltage and σ_2^2 variance of imaginary voltage.

Voltage measurements using the measurement model are calculated as

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} = \begin{bmatrix} 1 \cdot V_{re} \\ 1 \cdot V_{im} \end{bmatrix} \tag{4.56}$$

Thus voltage measurements are just the same as the state represented as

$$z_{V,re} = V_{re} \tag{4.57}$$

$$z_{V,im} = V_{im} \tag{4.58}$$

Using LWLS, the objection function in Eq. (4.9) (see Sec. 4.1) can be simplified as

$$J(x_V) = \frac{(V_{re} - (1 \cdot V_{re} + 0 \cdot V_{im}))^2}{{\sigma^2}_{V_{re}}} + \frac{(V_{im} - (0 \cdot V_{re} + 1 \cdot V_{im}))^2}{{\sigma^2}_{V_{im}}} \tag{4.59}$$

Thus the objective function in Eq. (4.59) can be rewritten as

$$J(x_V) = \frac{(V_{re} - 1 \cdot V_{re})^2}{{\sigma^2}_{V_{re}}} + \frac{(V_{im} - 1 \cdot V_{im})^2}{{\sigma^2}_{V_{im}}} \tag{4.60}$$

It shows that the estimated value equals the observed value and residuals are zero. In other words $J(x_V) = 0$.

Using DKF, the estimated value at time step k shown in Eq. (4.61) is based on the predicted value ($\hat{x}_{k|k-1}$) and measurement (z_k), thus values of pre-fit residual ($z_k - \hat{x}_{k|k-1}$) and post-fit residual ($z_k - \hat{x}_{k|k}$) using DKF are non-zero.

$$\hat{\boldsymbol{x}}_{k|k} = H\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.61}$$

Estimated states of DKF and LWLS using only the voltage measurements are presented in the following sections. First, we show estimated states in normal operation, then illustrate the estimation under attack.

Estimation Results Here we keep the phase angle constant with the first observed phase for SE. Actual polar voltage is visualized in Fig.4.9 and first phase angle is 0.4340 radian.

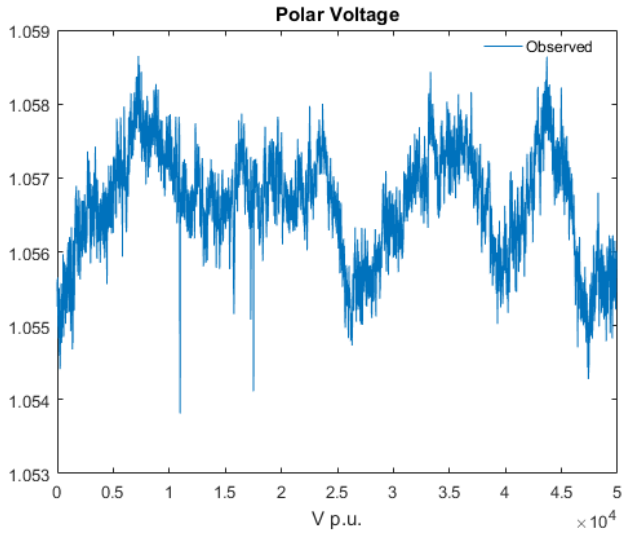


Figure 4.9: Visualization of actual polar voltage.

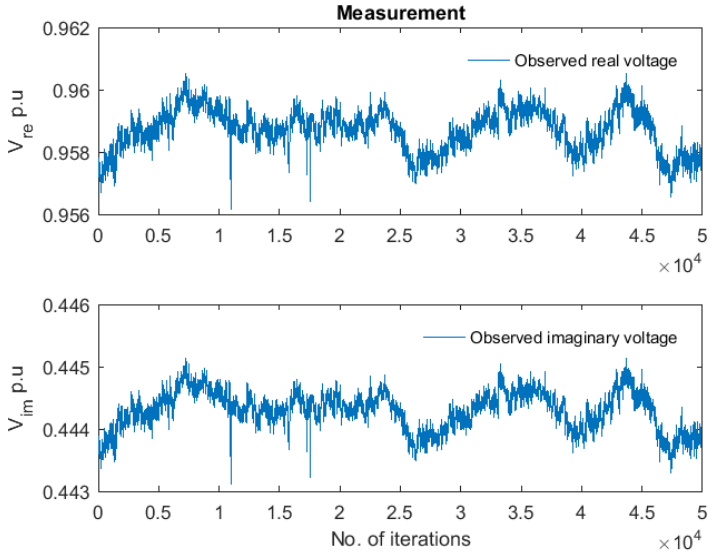
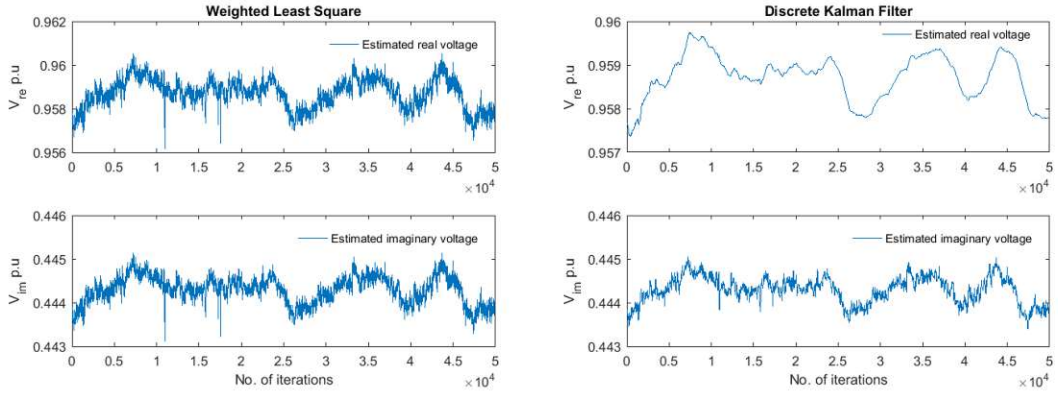


Figure 4.10: Visualization of measured real voltage and imaginary voltage.

Real voltage and imaginary voltage are calculated using the polar voltage and the constant phase angle. Figure 4.10 shows observed real voltage and imaginary voltage in normal operation. From the figures 4.9 and 4.10, one can see polar voltage, real voltage and

imaginary voltage have same pattern but different magnitudes. The two voltage signals shown in Fig. 4.10 are fed to the estimator, where two estimation methods (LWLS and DKF) are deployed.

We compare estimated states using the methods (LWLS and DKF) in normal operation, which are visualized in Fig. 4.11. Sub-figure 4.11a shows estimated real and imaginary voltage using LWLS. The estimated real and imaginary voltage using LWLS is the same as the observed voltages signals. Similarly, sub-figure 4.11b shows estimated real voltage and imaginary voltage using DKF. One can see from the sub-figure 4.11b that estimation process filters out noise and smooths the real voltage and the imaginary voltage signals. From the upper part of the sub-figure 4.11b, we can see the estimation process filters out more noise and smooths the real voltage signal than in the imaginary voltage (see lower part of the sub-figure 4.11b). It is because DKF puts less trust in real voltage measurements and follows the model predictions, and puts more trust on imaginary voltage measurements and follows the measurements more responsively. This can be explained using Kalman gain. From Fig. 4.12, we can see that Kalman gain of real voltage is lower than imaginary voltage because real voltage has higher variation than imaginary voltage (see Fig. 6.3 in Chapter 6). This results less trust in real voltage measurements (follows prediction model) and more trust on imaginary voltage measurements (follows the measurements more responsively). This can be seen in the sub-figure 4.11b that DKF follows prediction model so that estimated real voltage is less noisy and the estimated imaginary voltage is close to the measurements as it follows the measurements.



(a) Estimated states using LWLS

(b) Estimated states using DKF

Figure 4.11: Estimated real voltage and imaginary voltage in normal operation.

Using DKF, residuals are categorized in two categories, pre-fit residuals and post-fit residuals but this is not the case using LWLS because there is just one estimation step in LWLS. Thus we analyse and compare pre-fit residuals and post-fit residuals of DKF and compare them to residuals of LWSE. Residuals of DKF and LWLS in normal operation are visualized in Fig. 4.13. Sub-figure 4.13a shows pre-fit residuals and sub-figure 4.13b shows post-fit residuals of real and imaginary voltage using DKF. From the sub-figures, one can

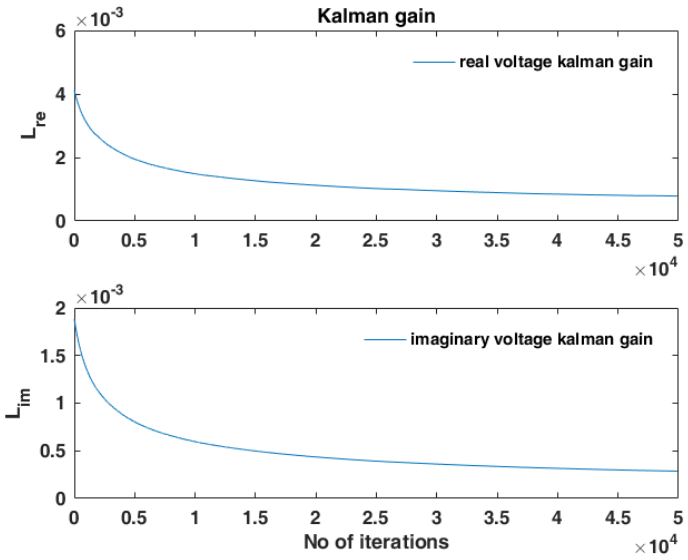
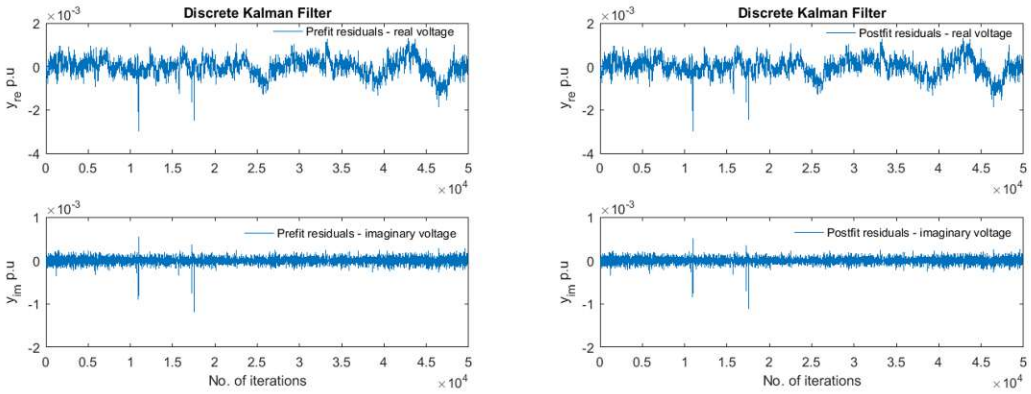


Figure 4.12: Visualization of Kalman gain of real voltage and imaginary voltage in normal operation.

see the pre-fit residuals and post-fit residuals are close to each other. But the residuals look different for real and imaginary voltage as the estimation process follows prediction model in real voltage and follows measurements in imaginary voltage. Magnitudes of the residuals of LWLS aligns to the theory presented in the previous section and have zero values.



(a) Pre-fit residuals of DKF

(b) Post-fit residuals of DKF

Figure 4.13: Residuals of real voltage and imaginary voltage in normal operation.

In the case of SE using only voltage measurements, as expected residuals of LWLS is zero but both pre-fit residuals and post-fit residuals using DKF are non-zero. In addition,

as expected residuals of DKF follow the pattern of voltage signal, for instance residuals fluctuates if voltage signal fluctuates and so on.

4.3.2.3 Using both voltage and current measurements

Here we use four measured values (real voltage, imaginary voltage, real current and imaginary current) for SE. Two states (real and imaginary) are estimated using the measurements.

Calculation of voltage measurements are shown in section 4.3.2.2, we recall representation of real voltage and imaginary voltage as follows

$$\boldsymbol{z}_V = \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} \tag{4.62}$$

Now we show calculation of current measurements using the measurement model. For the current measurements, we name matrix H as H_I. H_I is a 2×2 matrix as shown below

$$\boldsymbol{H}_I = \begin{bmatrix} G & -B \\ B & G \end{bmatrix} \tag{4.63}$$

where G is real part and B is imaginary part of the admittance $\boldsymbol{H}$.

Current measurements using the measurement model are calculated as below

$$\boldsymbol{z}_I = \boldsymbol{H}_I \boldsymbol{x} = \begin{bmatrix} G & -B \\ B & G \end{bmatrix} \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} = \begin{bmatrix} G \cdot V_{re} - B \cdot V_{re} \\ B \cdot V_{re} + G \cdot V_{im} \end{bmatrix} \tag{4.64}$$

Thus real current and imaginary current are represented as

$$I_{re} = G \cdot V_{re} - B \cdot V_{im} \tag{4.65}$$

$$I_{im} = B \cdot V_{re} + G \cdot V_{im} \tag{4.66}$$

Thus measurements are represented as

$$\boldsymbol{z} = \begin{bmatrix} V_{re} \\ V_{im} \\ I_{re} \\ I_{im} \end{bmatrix} \tag{4.67}$$

Measurement covariance matrix R is represented as

$$\boldsymbol{R} = \begin{bmatrix} \sigma^2_{V,re} & 0 & 0 & 0 \\ 0 & \sigma^2_{V,im} & 0 & 0 \\ 0 & 0 & \sigma^2_{I,re} & 0 \\ 0 & 0 & 0 & \sigma^2_{I,im} \end{bmatrix} \tag{4.68}$$

where $\sigma^2_{V,re}$ is variance of real voltage, $\sigma^2_{V,im}$ is variance of imaginary voltage, $\sigma^2_{I,re}$ is variance of real current and $\sigma^2_{I,im}$ is variance of imaginary current.

Matrix $\boldsymbol{H}$ from voltage and current measurements is

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ G & -B \\ B & G \end{bmatrix} \tag{4.69}$$

With LWLS the estimated state $\hat{\boldsymbol{x}}_{LWLS,k}$ at time step k is calculated from voltage and current measurements $\boldsymbol{z}_k$, the matrix $\boldsymbol{H}$, the noise covariance matrix $\boldsymbol{R}$ and gain matrix $\boldsymbol{G}$ as shown in Eq. (4.70) [50].

$$\hat{\boldsymbol{x}}_{LWLS,k} = \boldsymbol{G}^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{z}_k \tag{4.70}$$

Objective function of LWLS (represented in Eq. (4.9)) for current measurement can be simplified as

$$J(x_I) = \frac{(I_{re} - (G \cdot V_{re} - B \cdot V_{im}))^2}{\sigma^2{}_{I_{re}}} + \frac{(I_{im} - (B \cdot V_{re} + G \cdot V_{im}))^2}{\sigma^2{}_{I_{im}}} \tag{4.71}$$

Thus the function in Eq. (4.71) can be rewritten as

$$J(x_I) = \frac{(I_{re} - G \cdot V_{re} + B \cdot V_{im})^2}{\sigma^2{}_{I_{re}}} + \frac{(I_{im} - B \cdot V_{im} - G \cdot V_{im})^2}{\sigma^2{}_{I_{im}}} \tag{4.72}$$

The objective function using both voltage and current measurements is

$$J(x_{V,I}) = J(x_V) + J(x_I) \tag{4.73}$$

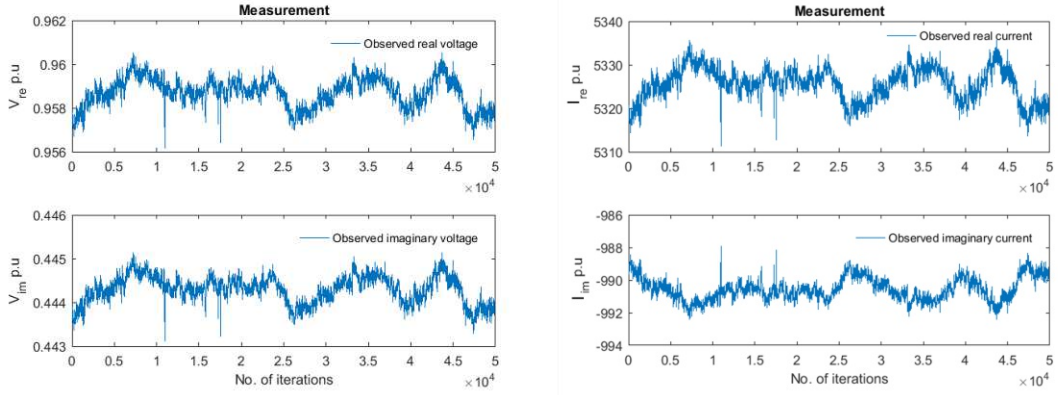
$J(x_V) = 0$ but $J(x_I)$ is non zero. Thus in this case, residuals of LWLS are non-zero.

Using DKF, estimated value at time step k shown in Eq. (4.74) is based on the predicted value ($\hat{x}_{k|k-1}$) and measurement ($\boldsymbol{z}_k$), thus in this case also values of pre-fit residual ($\boldsymbol{z}_k - \hat{\boldsymbol{x}}_{k|k-1}$) and post-fit residual ($\boldsymbol{z}_k - \hat{\boldsymbol{x}}_{k|k}$) are non-zero.

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.74}$$

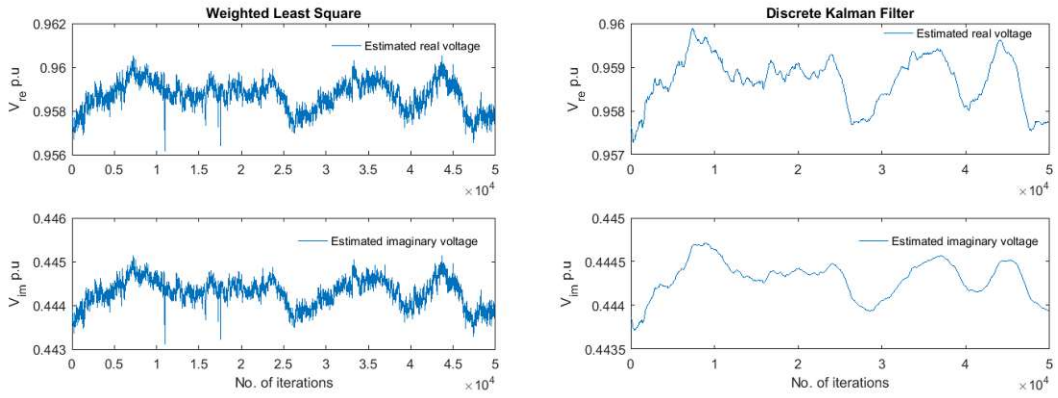
Estimated states using DKF and LWLS will be presented in the following sections.

Estimation Results We use polar voltage shown in Fig. 4.9 and calculate voltage measurements as mentioned in Sec. 4.3.2.2. Current measurements are calculated using the real voltage and imaginary voltage derived from the actual signal. Observed voltage and current measurements in normal operation is shown in Fig. 4.14. Sub-figure 4.14a visualizes observed real and imaginary voltages, and sub-figure 4.14b visualizes observed real and imaginary currents.



(a) Real and imaginary voltage (b) Real and imaginary current

Figure 4.14: Observed voltage and current measurements in normal operation.



(a) LWLS (b) DKF

Figure 4.15: Estimated states using LWLS and DKF in normal operation.

Figure 4.15 shows estimated real and imaginary voltage using LWLS and DKF. From the sub-figures 4.15a and 4.15b, one can see that estimation using DKF smooths out the signals. It is due to less trust on the recent measurements which can be explained using Kalman gain. Kalman gain of real voltage is lower than imaginary voltage From Fig. 4.16, one can see Kalman gain is low in normal operation because of high variation.

Estimation using LWLS is quite close to the original real and imaginary voltage signals. It can be seen by taking a closer look in the residuals.

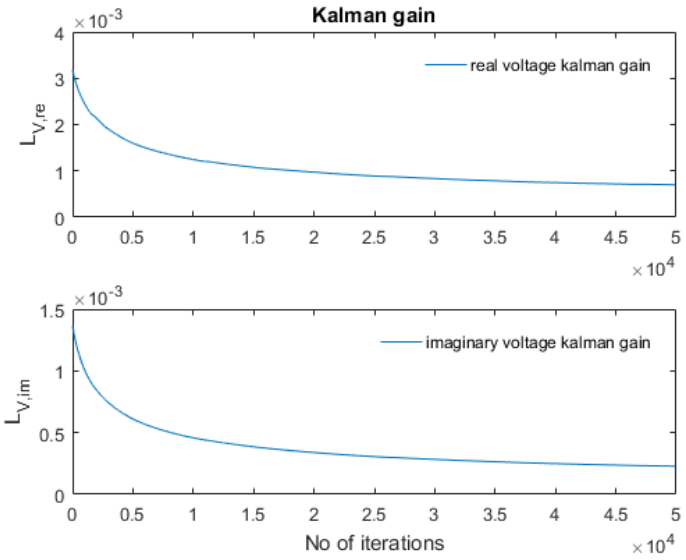
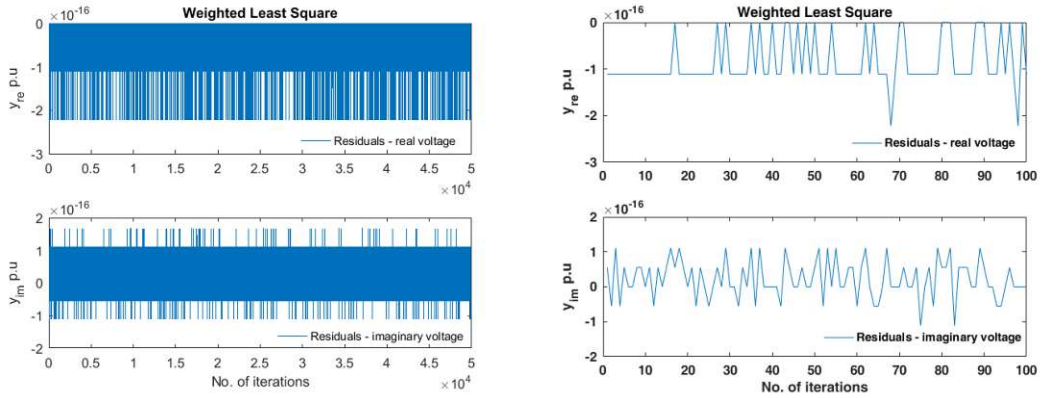


Figure 4.16: Kalman gain in normal operation.

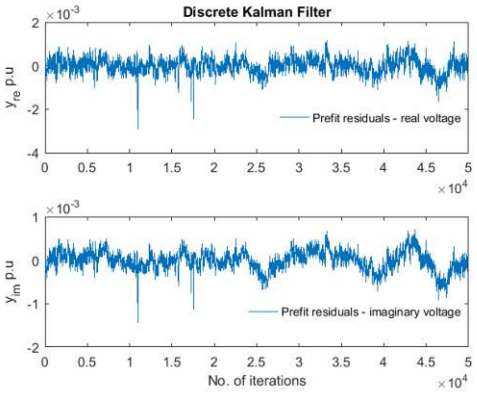


(a) Residuals of LWLS

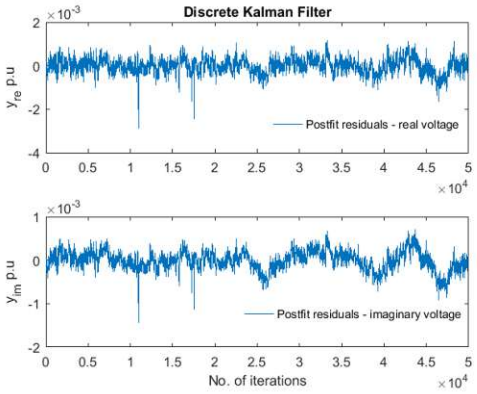
(b) Residuals of LWLS (zoomed in)

Figure 4.17: Residuals of real voltage and imaginary voltage using LWLS in normal operation.

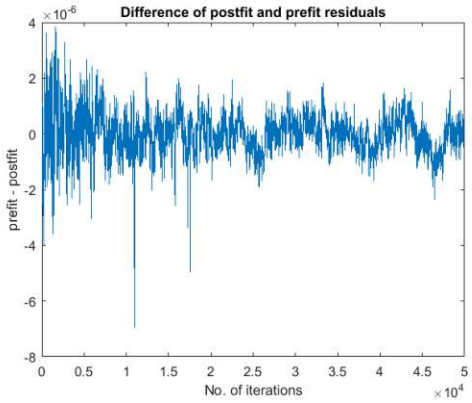
From Fig. 4.17, we can see residuals in LWLS are close to zero. Residuals of real and imaginary voltage in LWLS are shown in sub-figure 4.17a. (Post-fit) Residuals (observed values - estimated values) of LWLS for first 100 data points are shown in sub-figure 4.17b, this figure shows that estimated values are close to observed values. The estimated values of real voltage are always greater than the measured (observed) values (because



(a) Prefit residuals of DKF



(b) Postfit residuals of DKF



(c) DKF (difference of prefit and postfit residuals of real votlage)

Figure 4.18: Residuals of real voltage and imaginary voltage using DKF in normal operation.

the signals of real voltage and real current are same, so estimated values are greater), so the residuals of real voltage are always negative, but this is not the case in imaginary voltage (because the signals of imaginary voltage and imaginary current are different so estimated values are smaller).

Figure 4.18 shows residuals of DKF. Sub-figures 4.18a and 4.18b visualize pre-fit residuals and post-fit residuals of real voltage and imaginary voltage respectively. From sub-figure 4.18c we can see there is very small difference in pre-fit and post-fit residuals of real voltage. Thus pre-fit and post-fit residuals are similar in normal operation.

4.4 Summary

In this chapter, we presented the SE model, accompanied by two SE approaches (i.e., LWLS and DKF). These two approaches represent two classes of methods for SE: LWLS is a static SE method and DKF is a dynamic SE method. The results analysis showed that dynamic SE is better than static SE because the residuals from dynamic SE are non zero and also the changes are visible in the residuals.

In the first step, we used an example to demonstrate SE. Concept validation of Kalman filtering was demonstrated using the Kalman filter example, which was later modified and adopted in order to validate results from our use case.

In the second step, our use case of a power system was presented. A measurement matrix was used to describe the relationship between the system-states and the measured values (voltage phasors, current-injection phasors and current-flow phasors). The SE using the approaches WLS and DKF was presented for two cases i) using only voltage measurements, and ii) using both voltage and current measurements. The experimental results of the SE were presented for the cases.

CHAPTER 5

Attacker Model

Notice of adoption from previous publications in Chapter 5
Parts of the contents of this chapter have been published in the following papers:

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. 4th International Symposium for ICS and SCADA Cyber Security Research*, 2016*

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM*

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*, pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
Introduction of this Chapter is based on the work done in [132]. The threat analysis described in this chapter is based on the work done in [129] and [130], the attack vectors is based on the work done in [129] and the attack trees is based on [130]. The attack model described in this chapter is based on the work done in [133]. The false data injection attacks described in this chapter is based on the work done in [132] and [133].

S. Paudel implemented the model and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, we present an overview of attackers motivation, a threat analysis describing potential attack vectors, measurement manipulation attacks and potential consequences of the attacks. We then introduce false data injection attacks and present an attack model with a generation of attack types using this model.

Attacks on computer networks can be classified in two categories, namely (i) active attacks and (ii) passive attacks. In an active attack, the adversary manipulates data or equipment in the network. Data injection, data modification, packet drop attacks etc. are examples of active attacks. Passive attacks are related to gathering critical information in the network and learning the characteristics of the network or data being transferred, for example, by sniffing or eavesdropping. Passive attacks are often used to gather information for attack preparation. Multiple attack components can be combined to create a more advanced type of attack called an advanced persistent threat (APT), in which for instance first information is exfiltrated in a passive attack to learn the system state and its vulnerabilities, and then an active attack is launched to cause major damage to the system. An APT combines different attack methodologies, intrusion technologies, techniques and tools to compromise interconnected information and their target [34]. APTs usually implement several stages or sub goals before achieving an ultimate goal.

Phasor measurement units (PMUs) report high-frequency real-time voltage, frequency, phase angle, and many more measurements. Amongst other applications, they can be used to support real-time situation awareness, enabling operators to rapidly identify problems, such as line faults, and take informed corrective actions. Measurements from PMUs can be input to a state estimator, e.g., based on weighted least squares or Kalman filters, to mitigate measurement errors and provide estimates about the state of regions of a power grid that are not being directly observed. It is expected that PMUs and state estimation will play an increasingly important operational role in power systems monitoring and control.

Meanwhile, there have been several recent cases in which an attacker – by cyber means – has successfully caused operational impact [93, 94, 75]. Perhaps the most notable example of this form of attack being the incident in the Ukraine in December 2015, in which a major power blackout was caused [94]. In almost all of these cases, the attacker is thought to have implemented a series of attack steps – a so-called *kill chain* – using relatively advanced and stealthy techniques over an extended period of several months.

Our research questions about attacks on a wide area monitoring system (WAMS) read:

- **RQ 1.1**: How can an attacker cause false data injection attacks in a WAMS?
  Rationale: Components used for critical functions and making decisions in a WAMS use sensor data (e.g., PMU data). Attackers target to compromise the components used for critical functions and making decisions. We want to identify potential attacks against a WAMS by investigating major possibilities of how an attacker can launch severe attacks against critical components.

- **RQ 1.2**: How can multiple different false data injection attack forms be expressed in one comprehensive attack model?
  Rationale: An attacker can generate different attack types. For instance, an attacker aims to appear normal, aims to hide a faulty system state and so on. We investigate the possibilities of generating different attack types. Consequently, we assume we could generate different attack types using different attack parameters. It is of advantage to have a general model that covers different attack forms. With this theoretical considerations the model can be formulated in a more general way.

In this context, a potentially attractive target for an attacker is the measurement and state estimation infrastructure, which leverages PMUs, that is used to inform control decisions. The goal of an attacker would be to manipulate measurements, such that they do not reflect the *actual* system state, so that an operator – or an automated system acting on their behalf – performs incorrect control decisions. For example, an attacker could manipulate measurements to appear normal, to hide a faulty system state that should be mitigated (and vice versa). Such an attack could result in the severe power systems effects that have been explored by previous work [113]. For the attacker ideally, this data manipulation attack should be implemented in a way that it is hard to detect, i.e., the attack should be *stealthy*. In previous work, we already have explored the different ways how an attacker can use several attacks steps to compromise WAMSs, which includes PMUs, using cyber and physical means [129]. For establishing an attack model, we perform the following steps

- We first look at the attack vectors that can be used to attack a WAMS architecture.
- We then provide descriptions of different attack scenarios.
- Based on the scenarios, we generate attack trees to show different options how an attacker can reach his goal and discuss which strategies an attacker may choose.

5.1 Threat Analysis

A WAMS has many components for critical functions and making decisions. Attackers launch complex sophisticated attacks targeting WAMS. In this section, we will consider all the devices, their interfaces, hardware and software in WAMS and will investigate on major possibilities how an attacker can launch severe attacks using these components.

For this, we will use attack trees and will present generic and specific attack models on WAMS by representing them in graphical and in simple text lines tree format.

5.1.1 Attack Vectors

Each device in a WAMS is a combination of hardware and software. For the attack model, we distinguish between a) the physical device itself, b) the software running on the device and c) the communication components. We explain the influencing factors of the different components for attacks below.

5.1.1.1 Physical Device

Physical devices are connected to each of the phasor components. PMUs are the physical devices that measure the electrical waves on an electricity grid. Different devices are used for hosting PMUs, PDCs, super PDCs, or control processes. Routers are used for interconnection of different components and devices in the Smart Grid network. The level of protection for the devices differs based on the functionality. PMUs are at different locations in the power grid, for instance in substations. Therefore, some protection of the physical device can be assumed. PDCs collect data from multiple PMUs and may be even more protected. PMUs and PDCs can also have some tampering detection methods installed to prevent unauthorized opening of the chassis. Since a direct attack to the CC allows an attacker to directly invoke control decisions, it is considered as the most severe case. The CC is therefore assumed to be the best protected element. Intermediate network devices such as routers also typically reside in physically protected server rooms.

5.1.1.2 Software

Software is used for different functionality, providing the operating system, monitoring tasks, and taking decisions. A software can have different components for example, a component for dealing with input data, a component for computing core functionality and processing data, and a component for sending data to other components as output. Services are used to communicate between the components. Therefore, software used to produce such services can have intermediate services and/or public services.

Software used in WAMS are not free from vulnerabilities. Therefore, exploiting such vulnerabilities attackers can gain access to WAMS components. Nevertheless, we assume that the amount of different software and services running on WAMS devices is comparatively small compared to classical Internet devices.

5.1.1.3 Communication

Physical devices have interfaces to connect and communicate with other components in WAMS. Each device has multiple interfaces. This is important for the attack model, because the interfaces can be used for injecting or exfiltration of data as part of an attack.

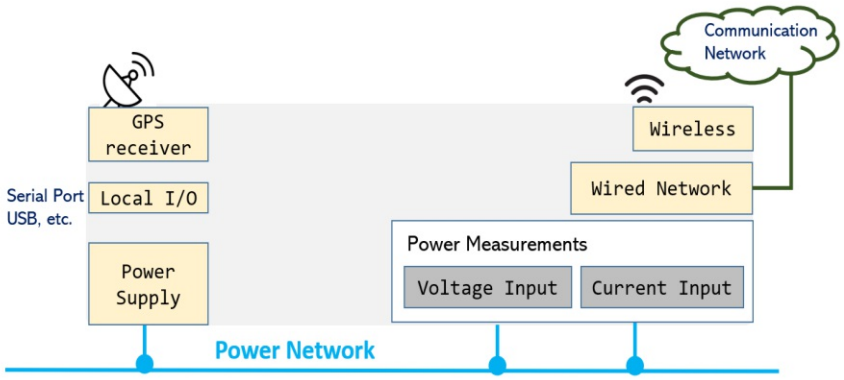


Figure 5.1: PMU interfaces (source Paudel et al. [130]).

In Fig. 5.1 the interfaces of a PMU device are shown. The device is connected to the power network in order to measure voltage and current phasors. For transmitting the measurement data it is connected to a communication network using wired or wireless technologies. A PMU can have local interfaces, such as a serial or USB port to connect media or other devices locally and it has a GPS receiver for receiving clock synchronization signals. All those interfaces can be used as elements in an attack and therefore need to be considered in the attack trees.

The selection of communication protocols depends on the type of communication. For example, for configuration information and commands a reliable data transmission is preferable, whereas for the transmission of PMU measurement records a low latency is more important.

For WAMS communication different communication protocols are used from different standards. For example, standard IEC 61850 [63] is a communication protocol that facilitates utility automation including protection and control [37]. Although initially it was developed for the IEDs within substations, now it covers various communication features [10, 9]. Additionally, it defines an architecture, the data models used for the communication within electric power systems [10, 9] (see also Chapter 2).

5.1.2 Scenarios

WAMSs use synchrophasor technology and different devices to generate, receive and utilize synchrophasor data [156]. Such devices are vulnerable to various security threats. Attackers can compromise devices by leveraging the vulnerabilities in the system. The impact of failures and attack scenarios in a WAMS is dependent on the data used for monitoring, protection and control [121]. For example, in a wide area network if a monitoring application is also used for protection and making control decisions then failures in such applications can cause higher impact than a failure in an application used only for monitoring.

PMUs, IEDs, PDCs, super PDCs, PGWs and various network components are connected to support communication between the devices and applications that are used in a WAMS. A number of attack scenarios are presented in Sec. 5.1.2.2, which examine these impacts.

In this study, our focus is on data integrity attacks that modify measurement data, either the original readings from PMUs or aggregated data (or just events) from PDCs, Super PDCs or PGWs. After processing falsified data, the WAMS may estimates inaccurate states of the power system. The impact can be wrong decisions such as triggering protection elements if not needed or suppress a vital protective action. For example, due to modified measurement data the system may believe that overloaded branches have secure voltage and vice versa [43]. It can also cause delay in taking actions, e.g., for load shedding or grid reconfiguration. Cascading failures across utilities and can be caused due to the system delays in other utilities[121] and also to equipment damage.

5.1.2.1 Assumptions

Our threat model mainly considers the problem of compromised machines in a WAMS: PMUs, PDCs, Super PDCs, PGWs and routers in the path. We consider intrusions that concern both physical power systems, as well as communication networks. For example, intrusions in PMU devices (physical components) and in their embedded software (cyber part). We suppose that software and hardware of WAMS components, as any other systems, are not free from vulnerabilities and assume that an attacker gets access to a WAMS component using any kind of exploit. Furthermore, attackers may have physical access to systems in the field. We concentrate only on data integrity attacks on the measurement data itself. So, we consider only the data flow from the sensors (PMUs) towards the data collection (CC), and do not consider data integrity attacks on the control data that is sent to IEDs.

In general, we assume that if a device is compromised that the attacker has access to all data including cryptographic keys on the system[1]. That means the attacker can generate valid message authentication codes or digital signatures for the measurement data and also can encrypt and decrypt data with the appropriate keys. But this applies only for

[1] In well-secured systems keys may be stored in a separated trusted platform.

the end-to-end communication. Devices on the path that are not configured to access or modify the data (e.g., access or core routers) do not have access to appropriate keys. If measurement data is integrity protected and encrypted, attacks on such intermediate devices such as routers are limited to data dropping or duplication attacks. Routers may also delay data in a way that they arrive too late to contribute to control applications. Attacks to routing protocols or those directed to the CC are out of scope of this work. Also attacks on the clock synchronization system, required in a WAMS, are not considered in this work.

5.1.2.2 Attack Scenarios

The National Electric Sector Cybersecurity Organization Resource (NESCOR)[2] has investigated cybersecurity failure scenarios that result in a failure to maintain the confidentiality, integrity and availability (CIA) of cyber assets, which have a negative impact on generation, transmission, and delivery of power. In [121] they provide failure scenarios and prioritize them. Further, they developed detailed information for the scenarios with the highest priority. We study the NESCOR failure scenarios related to wide area monitoring, taking them as basis to derive six failure scenarios that are related to PMUs, PDCs, super PDCs, PGWs, access routers and core routers in a WAMS. We also describe the WAMS's response to the scenarios and their impact on the system. Key components and attack points of a WAMS are shown in Fig. 5.2.

Scenario 1 - PMU compromised In this scenario an attacker gets access to a PMU and forges PMU frames with wrong data. The frames with wrong information are then sent to a PDC or CC. Such an attack is, for instance, described in [43]. If we have PGWs in the network, then frames can also be sent directly to a PGW. We separate this scenario into three cases, based on how a PMU reports falsified data up in the hierarchy.

- *Case 1:* PMUs have a PDC as data aggregation point. A local or regional PDC aggregates the falsified data and sends it up in the hierarchy.
- *Case 2:* PMUs are connected directly to a PGW. Falsified data are sent directly to a PGW, which shares information with other PGWs.
- *Case 3:* PMUs are connected directly to a CC. Falsified data are directly sent to a CC. Due to the lack of an aggregation step, the severity of damage in this case could be higher than in the other cases.

Scenario 2 - PDC compromised Although access control and connection authentication between a PDC and a PMU are already considered in some protocols [63], PDCs

[2] https://smartgrid.epri.com/NESCOR.aspx

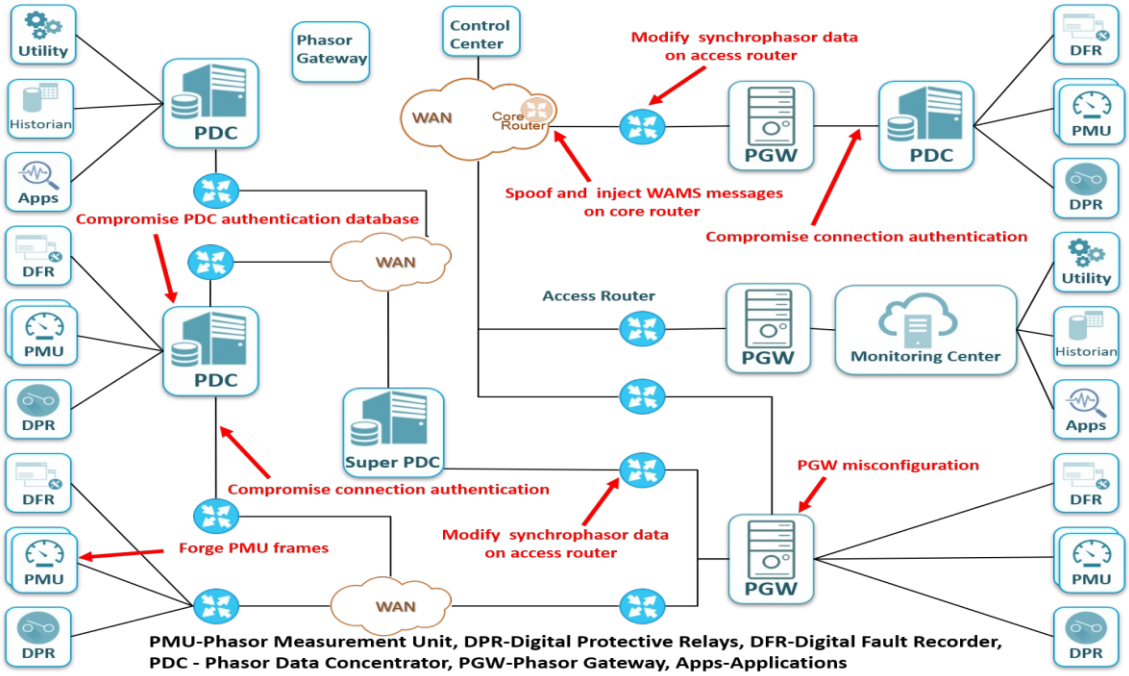


Figure 5.2: WAMS with key components, compromised points and attacks (source Paudel et al. [129]).

may be compromised due to a backdoor or an attack to an authentication database. An attacker can get access to the database in the PDC and modify or steal the information that allows malicious introduction of false measurement data [121]. A PDC can be connected to other regional PDCs, super PDC, PGW or directly to the CC. We have four cases depending on how a PDC sends false measurement data to other components in a WAMS.

- *Case 1:* The PDC sends false measurements to another PDC, which then processes the false measurement values.
- *Case 2:* The PDC sends false measurements to a Super PDC.
- *Case 3:* The PDC sends false measurement values directly to a PGW. The PGW shares the false information to other PGWs.
- *Case 4:* The PDC sends false measurement values directly to a CC. The CC uses the falsified information as inputs in the applications.

Scenario 3 - super PDC compromised If a super PDC is compromised, it may send wrong information about all its connected devices (PDCs and PMUs), which are reporting to the Super PDC according to the hierarchical topology. In addition, a super PDC may

be misconfigured to not recognize other super PDCs, regional PDCs or PMUs in the network, or just send incomplete measurement data up in the WAMS hierarchy. Super PDCs may report to other super PDCs, PGWs or CC. We have three cases depending on how a super PDC sends false measurements data to other components in a WAMS.

- *Case 1:* The super PDC sends false measurements to another super PDC, which then processes the false measurement values.
- *Case 2:* The super PDC sends false measurements to the next level in hierarchy represented by a PGW. The PGW directly shares information to other PGWs.
- *Case 3:* The super PDC sends false measurement values directly to a CC. The falsified information is used as inputs in the WAMS applications.

Super PDCs are above in the hierarchy and therefore probably better protected. So we assume that compromising a super PDC is harder than compromising a PDC.

Scenario 4 - PGW compromised If a PGW is compromised, it can not only falsify the collected data, but may also refuse to share synchrophasor measurement data with other PGWs. Since PGWs provide security isolation of trusted internal systems to the external ones, and create a trusted gateway-to-gateway connection [120], PGWs should present only a smaller attack surface. So if all proposed PGW's security measures are implemented, it should be harder to launch attacks against a PGW.

Scenario 5 - access router compromised An attacker that gains control of access routers on the path from a PMU to a CC can drop or duplicate synchrophasor packets on the access routers that belong to the PMU communication. If the data is not integrity protected (e.g., by a message authentication code or digital signature), an attacker on a router can act as man in the middle and modify the data or inject own data packets. Routers may also delay data in a way that they arrive too late to contribute to control applications. If the data is not encrypted, an attacker on a router can read the PMU data, which may include configuration and location information. This does not change the measurement data, but such information may be useful for attack preparation.

Scenario 6 - core router compromised Attacks on core routers can have the same impact as those on access routers. But core routers handle many more data flows from many different locations, so data from many PMUs, PDCs or PGWs may be affected if an attacker gains control over devices in the core. Nevertheless, core routers are usually better protected than access routers.

For our analysis we assume that (core and access) routers are able to modify data, either because credentials have been compromised from end systems or because data is not end-to-end integrity protected. Furthermore, PDCs mainly aggregate and reorder the

records received from multiple PMUs, and do not perform any operation on the sensor data itself. So an attack on PDCs or super PDCs usually does not change the original measurement data (nor any derived values or events). As a consequence, mitigation strategies based on sensor data plausibility analysis can also help against attacks in intermediate systems.

We also assume that colluding and coordinated attacks are quite likely. Devices deployed in smart grids are often equal or similar regarding hardware, software and configuration (e.g., multiple PMUs from one vendor). Therefore, it is quite possible that an attacker can gain access to multiple systems at the same time or that attackers collude. Attacks on intermediate systems (PDCs, routers) can enable an attacker to launch a coordinated attack, even if he has only access to one system.

Impacts of all above mentioned attacks can be a failure to take actions when needed, improper synchronous closing, leading to equipment damage, a line trip leading to cascading failures and many more [121].

5.1.3 Attack Trees

Attack trees [152] are common models to represent complex attacks. An attack tree contains a root, branches, several intermediate nodes and leaf nodes. The root represents the ultimate goal of an attack, different branches shows different possibilities of reaching the root node. Different possibilities could be reached by combining all branches or only by following one branch. This depends on the type of a relationship AND/OR while branching the node. For an AND relationship, all sub goals must be reached; meanwhile, for an OR relationship, reaching at least one sub goal is enough to reach the higher goal. Each intermediate node is a sub goal of the attack, whereas leaf nodes represent the start points. Moore et al. [116] propose an attack modeling method by describing format and semantics of the attack trees. Authors use a straight line between the branches of a node if they have AND relation and use curve lines if they have OR relation. Both of the cases use solid lines. Attack trees provide a valuable overview of the pre-requisites and sub goal relations for attacks that are based on the combination of multiple different actions. They help to analyze attack goals and potential chains of actions. They assist in assessing the likelihood and costs of specific sub goals and attack branches and support detection and prevention of complex combined attacks.

These models help to assess which branches are easier to achieve for attackers, and provide strategic guidance for the deployment of suitable countermeasures. Therefore, attack models for WAMS environments provide useful insights to improve wide area monitoring security.

5.1.3.1 Assumptions

In cyber physical systems cyber attacks can have implications on physical components. Both aspects, direct physical attacks and cyber attacks with physical impact, need to be covered in our models. Furthermore, attacks have different visibility, depending on the methods used and also the costs for achieving sub goals are different.

We argue that cyber attacks to the power grid are much easier and have several advantages compared to physical attacks. We especially assume the following advantages:

- Remote access: A physical attack requires the physical presence of one or more persons. Attackers may need to travel and need to access the premisses. All this requires resources and detailed information on the physical target, which may need costly reconnaissance at the premises. Physical presence can lead to an easier detection and identification of the attackers (e.g., video supervision, intrusion alarms). In contrast, cyber attacks can be performed remotely. This is especially of advantage if attacks are launched from other countries.
- Deniability: In cyber attacks, it is easier to conceal traces and deny any involvement in attacks. This is a critical advantage especially in espionage and cyber war.
- Scalability: Cyber attacks can target multiple systems simultaneously and can be launched by a single person. For physical attacks coordination and logistics are required for colluding attackers.
- Safety: Destroying equipment, cutting transmission lines or breaking physical protection methods contain a personal risk for the attacker to get injured. So the personal barrier to launch a physical attack is higher then for cyber attacks. This also makes it easier (and inexpensive) for criminals to recruit persons for cyber attacks than for physical attacks.

We therefore assume that the leaf nodes in the attack trees that can be achieved by cyber means will be preferred by attackers and therefore have a higher likelihood. APTs can also be a combination of physical and cyber attacks. For instance a spy might gather some information about the premises, before a cyber attack is launched or a cyber attack can be used to disable video surveillance to assist in a physical attack. We clearly distinguish physical and cyber leaf nodes in our attack trees to provide an enhanced assessment of specific branches based on the attack types (cyber or physical) involved to achieve sub goals.

For the assessment of sub goals and the likelihood of specific branches in attack trees, we make the following assumptions about the attackers preferences:

- Cyber over physical: Cyber attacks have many advantages over physical attacks. We provide several reasons for this above. Therefore, we assume that cyber attacks are more attractive to an adversary than physical attacks.

- Stealthiness: Although it is obvious that something happened when a blackout or major disruptions occur, we assume that the attacker(s) prefer stealthy methods (that are harder to detect) during the attacking process .
- Cost reduction: We assume that attackers prefer methods that are inexpensive regarding resources. So if a sub goal can be achieved by different means, the inexpensive method is likely chosen.

However, despite these general assumptions, assigning specific costs (and likelihoods) to the attacks is infrastructure-specific. The architecture of a system, installed countermeasures, its vulnerabilities, technical difficulty of an attack, and an attack's cost are interrelated [28]. Therefore, we concentrate on attack techniques – assigning, cost, feasibility, and likelihood is out of scope of this work.

5.1.3.2 Generic Attack Tree

Attackers use various methodologies, techniques and tools to compromise communication, software or hardware of devices to reach an ultimate goal of an attack. Compromising a communication protocol requires different techniques than those used to compromise hardware or software on a device.

Compromising a field device In this section, we describe generic techniques that attackers can use to compromise a device. Since compromising different devices in the WAMS architecture occur as sub goals in our attack trees and similar means can be used to compromise different devices, we use this generic tree as one building block for the specific trees for WAMS attacks.

We define "compromise device" as gaining full access to a machine and its data, including login credentials, root access, the ability to run arbitrary software, as well as gaining access to keys for establishing secure connections, encryption and signing messages (e.g., PMU records). Figure 5.3 shows a generic tree for compromising a device.

The root node shows the final goal, in our case compromising the devices. Connected child nodes represent sub goals that are required to achieve the parent goal. In accordance with [116], we mark child connections with one additional straight line to denote that all of the subgoals are required to achieve the parent goal (AND connection). If any of the subgoals is sufficient to achieve the parent goal, we use two curved lines (OR connections). Leaf nodes marked with a star require compromising a (different) device and therefore can be considered as root nodes for another "compromise device" attack tree, with the same structure that then needs to be attached there.

A device can be compromised locally and remotely. We define a local attack as a condition in which the attacker has direct physical access to the device. Therefore, all local attacks require physical access to the installation. An attacker can have physical access in many

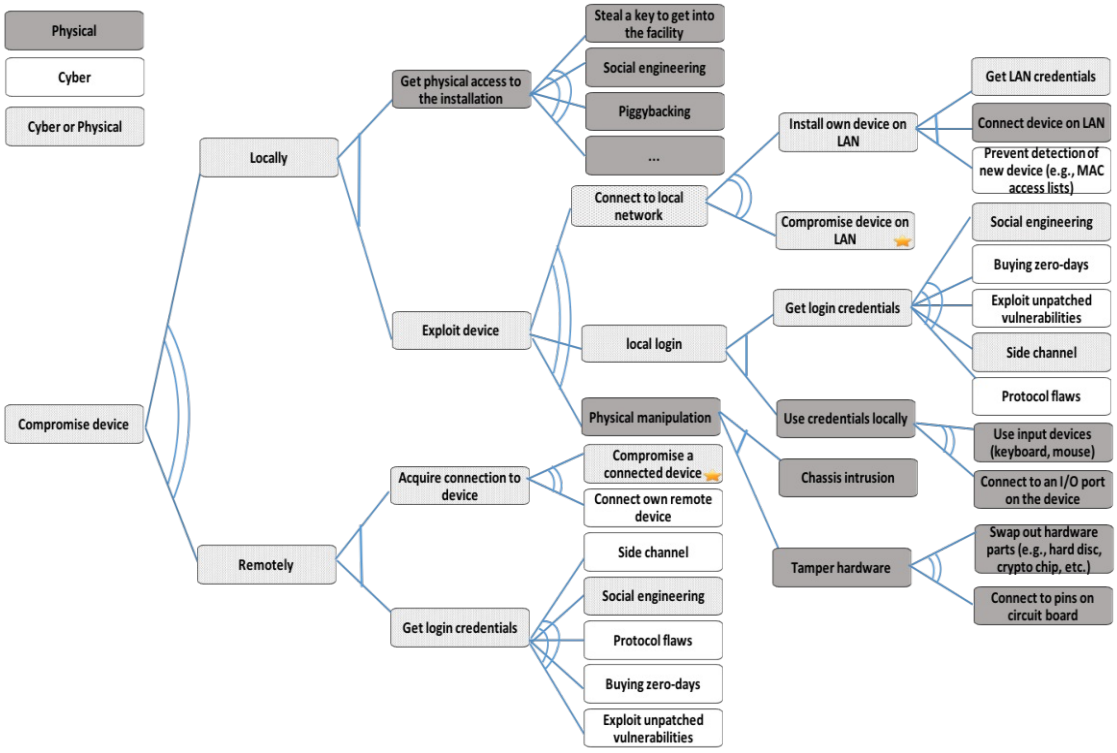


Figure 5.3: Generic attack tree for compromising a device (source Paudel et al. [130])

ways, for example, by stealing a key to get inside, social engineering by convincing someone who has access, piggybacking into a facility, etc. An attacker then has three options to exploit a device. He either can physically tamper with the device. For this he needs to open the chassis, circumventing any chassis intrusion detection, if present.

Subsequently, he can either swap out hardware devices like the hard disc or a crypto chip, or connect to pins on the board to read or modify data. A second option is that he can login locally. For this he needs the login credentials. He can get the credentials either by social engineering, buying zero day vulnerabilities, exploiting unpatched vulnerabilities, use some form of side channel attack, or exploit protocol flaws. After getting the login credentials, the attacker can use the credentials by using input devices like keyboard, mouse or connection of I/O port on the device. A third option is that he can connect to the local network that the device is connected to. This can happen in two ways: either he can compromise a device on the LAN or install his own device on the LAN. Installing a device on the LAN requires some steps, at first he needs to get LAN credentials, connect the device to the LAN and prevent detecting the installed new device, for example, by preventing that the MAC is denied by MAC access list.

Remote attacks are a condition in which an attacker can compromise a device remotely, connecting via the Internet. All remote attacks require login credentials to acquire a

connection to a device remotely. The attacker can get login credentials either from a side channel attack, social engineering, using protocol flaws, buying zero-days, or by exploiting unpatched vulnerabilities. Then he can connect the device either by compromising a connected device or connecting his own remote device.

To summarise, we present the compromise device attack tree as text, showing all the AND/OR relationships.

```
Attack Goal: Compromise device
OR
1. Locally
        AND
        1.1. Get physical access to the installation
                OR
                1.1.1. Steal a key to get into the facility
                1.1.2. Social engineering
                1.1.3. Piggybacking
        1.2. Exploit device
                OR
                1.2.1. Connect to local network
                        OR
                        1.2.1.1. Install own device on LAN
                                AND
                                1.2.1.1.1. Get LAN credentials
                                1.2.1.1.2. Connect device on LAN
                                1.2.1.1.3. Prevent detection of new device
                                        (e.g., MAC access lists)
                        1.2.1.2. Compromise device on LAN
                1.2.2. Local login
                        AND
                        1.2.2.1. Get login credentials
                                OR
                                1.2.2.1.1. Social Engineering
                                1.2.2.1.2. Buying zero-days
                                1.2.2.1.3. Exploit unpatched vulnerabilities
                                1.2.2.1.4. Side channel
                                1.2.2.1.5. Protocol flaws
                        1.2.2.2. Use credentials locally
                                OR
                                1.2.2.2.1. Use input device (keyboard, mouse)
                                1.2.2.2.2. Connect to an I/O port on the device
                1.2.3. Physical manipulation
                        AND
                        1.2.3.1. Chassis intrusion
                        1.2.3.2. Tamper hardware
                                OR
                                1.2.3.2.1. Swap out hardware parts
                                        (e.g., hard disc, crypto chip, etc.)
                                1.2.3.2.1. Connect to pins on circuit board
2. Remotely
        AND
        2.1. Acquire connection to device
                OR
                2.1.1. Compromise a connected device
                2.1.2. Connect own remote device
        2.2. Get login credentials
                OR
                2.2.1. Side channel
                2.2.2. Social engineering
                2.2.3. Protocol flaws
```

```
53                  2.2.4. Buying zero-days
54                  2.2.5. Exploit unpatched vulnerabilities
```

5.1.3.3 Specific Attack Trees

In this section, we introduce two specific attack trees for WAMS scenarios:

- Blackout tree: This tree shows different opportunities to cause a blackout in the power grid. It can be considered as a high level goal of the attacker, and multiple subtrees can be defined to achieve some of the sub goals shown in the tree.
- Manipulate input data: This tree shows the subgoals that must be achieved to manipulate input data to a control algorithm. Here we show as an example how to manipulate the phase angle data that is sent to a controller. This tree can be seen as one subtree attached to the blackout tree, at the leaf node "send wrong input values to controller".

In the figures, we mark all sub goals that require to compromise a specific device with a star. The sub goals for compromising a device are described in the generic compromise device tree.

Blackout Tree In the blackout tree (Fig. 5.4), the root goal is to cause a power blackout. This can be done by destroying critical equipment or disconnecting parts of the grid. Both goals can be achieved by physical or cyber means. Disconnecting parts of the grid can be done by physical means, by cutting transmission lines or physically tampering with Circuit Breakers (CB). Disconnections can also be invoked by cyber means, by sending a trip command to a CB. Triggering trip commands is achievable using three different approaches: a) compromising an IED, which then sends the trip command; b) invoking incorrect control decisions in the controller, which are then forwarded to the IED; or c) modifying controller commands on the path from the controller to the IED.

Option a) requires an attacker to compromise an IED, which is an achievable option. But tampering with IEDs might be easily detected by comparing IED actions to the commands that were sent by the controller. For option c), it is necessary to get access to credentials to modify controller commands (i.e., decryption and signing credentials), and then compromise a device on the path or install an own device to launch a man-in-the-middle (MITM) attack.

As shown in the generic "compromise device" attack tree, there are different ways to gain access to credentials, such as using side channels, social engineering or buying them on the market. If those options are not available, it may require attackers to compromise the controller itself, which then opens the possibility for direct tampering with controller commands (described as sub goals under option b)), and does not require modification

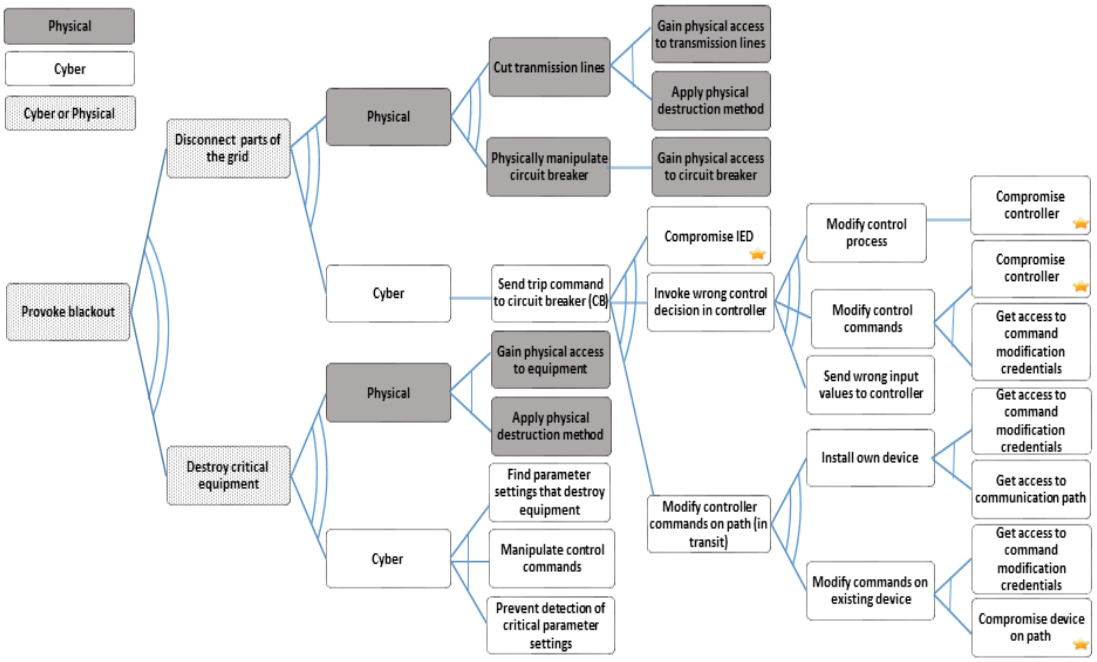


Figure 5.4: Attack tree for causing a power blackout (source Paudel et al. [130]).

of commands on the path. In addition, running a man-in-the middle (MiTM) attack requires to install fast processes for modifying data in transit to not cause an unusually high delay. Since many grid processes are time critical, and latencies are kept to a minimum, any additional delay may be detectable. Another difficulty with tampering with data on the path, is that it may raise suspicions if commands that arrive at the IED differ from those that were sent by the controller. In our attack tree, we assume that the attacker gains access to the credentials, but it would be simple to implement some additional checking mechanism to compare commands issued by the IED with those sent by the controller.

Due to the reasons described above we argue that option b), invoking wrong control decisions in the controller, is a more attractive option for the attacker. Here again there are three options to accomplish this. In order to trigger a control process to send incorrect commands, one can influence the control process or modify control commands on the controller. Both options require to compromise the controller. Due to their central role, we assume that controllers are much more protected (physically and in cyber space) than other devices, and therefore consider it as more costly to attack them directly.

An alternative is to indirectly influence controller decisions by sending wrong input values to the controller, which then invoke wrong control decisions. Sensors (such as PMUs) are typically deployed in the field, and are therefore more easily accessible than a controller in a CC. Therefore, we consider it as a reasonable, and the most attractive scenario, for

attackers to manipulate input data to invoke wrong control decisions. We provide a more detailed attack tree for this scenario in the next section.

The second major option is the destruction of equipment. This can be done by physical means, by just accessing and destroying devices or it can be done using cyber means. For a destruction using cyber attacks, an attacker needs to find parameter settings that lead to conditions that in the short or long term cause malfunctions in the devices. For example, the slow destruction of centrifuges that were manipulated by the Stuxnet malware. Nevertheless, it is not always possible to invoke parameter settings that can cause malfunctions or degeneration, because most devices have local protection mechanisms to prevent critical conditions. Furthermore, an attacker needs to hide the new parameter settings and, to achieve this, may need to alter measurement reports from supervision functions (as done in the Stuxnet attack). The third sub goal "manipulate control commands" can be achieved by the same child nodes as shown for the "invoke wrong control decisions" for tripping a circuit breaker.

We present the provoke blackout attack tree also as text showing all the AND/OR relationships. A node having only one branch is just listed next to it without AND/OR condition.

```
1  Attack Goal: Provote blackout
2  OR
3  1. Disconnect parts of the grid
4          OR
5          1.1. Physical
6                  OR
7                  1.1.1. Cut transmission lines
8                  1.1.2. Physically manipulate circuit breaker
9                         1.1.2.1. Gain physical access to circuit breaker
10         1.2. Cyber
11                 1.2.1. Send trip command to circuit breaker (CB)
12                        OR
13                        1.2.1.1. Compromise IED
14                        1.2.1.2. Invoke wrong control decision in controller
15                                OR
16                                1.2.1.2.1. Modify control process
17                                        2.1.2.1.1. Compromise controller
18                                1.2.1.2.2. Modify control commands
19                                        AND
20                                        1.2.1.2.2.1. Compromise controller
21                                        1.2.1.2.2.2. Get access to command
22                                                modification credentials
23                                1.2.1.2.3. Send wrong input values to controller
24                        1.2.1.3. Modify controller commands on path (in transit)
25                                OR
26                                1.2.1.3.1. Install own device
27                                        AND
28                                        1.2.1.3.1.1. Get access to command
29                                                modification credentials
30                                        1.2.1.3.1.2. Get access to communication
31                                                path
32                                1.2.1.3.2. Modify command on existing device
33                                        AND
```

```
                                                1.2.1.3.2.1. Get access to command
                                                        modification credentials
                                                1.2.1.3.2.2. Compromise device on path
2. Destroy critical equipment
        OR
         2.1. Physical
                AND
                 2.1.1. Gain physical access to equipment
                 2.1.2. Apply physical destruction method
         2.2. Cyber
                AND
                 2.2.1. Find parameter settings that destroy eqipment
                 2.2.2. Manipulate control commands
                 2.2.3. Prevent detection of critical parameter settings
```

Manipulate Input Data Tree In this section, we concentrate on data manipulation attacks. The final goal is to influence a control decision by manipulating input data to a controller or decision making process in a control loop. Figure 5.5 shows the attack tree for manipulating input data. The most important measurement values for grid stability are usually: i) the phase angle between measured sinusoidal signals at different locations; ii) the frequency; and iii) the voltage.

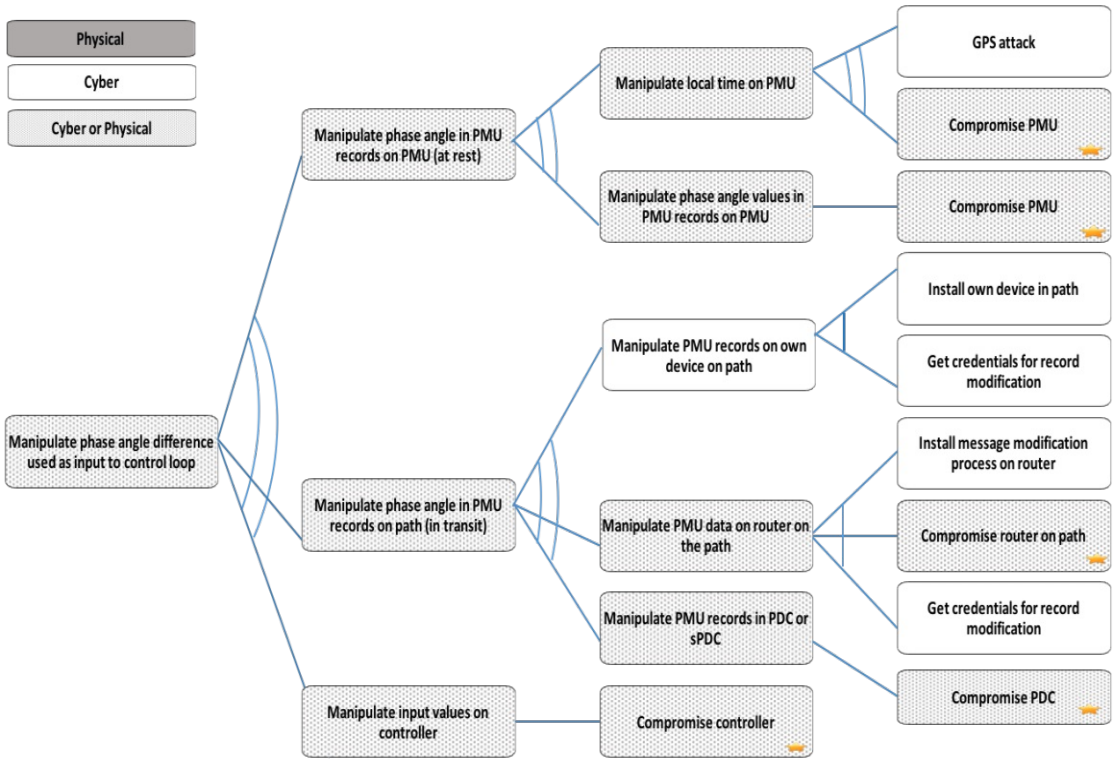


Figure 5.5: Attack tree for manipulating the phase angle (source Paudel et al. [130]).

Figure 5.5 shows an attack tree for manipulating the phase angle between measurements

at two different locations. The phase angle is calculated from measurements at two PMUs placed at different locations in the power grid. One possibility is to modify the PMU records on the PMU directly. For this, it is necessary to compromise a PMU device (including gaining access to credentials for encrypting and signing records). For the goal "compromise PMU", we can use the compromise device tree defined in Sec. 5.1.3.2. The specifics of the PMU are the specific interfaces that are described in Fig. 5.1. The connected devices for the PMU are shown in Fig. 2.3.

Another possibility to modify the phase angle in a PMU record is to change the local time on the PMU. Clock settings can be changed locally if the PMU is compromised. Nevertheless, it is also possible to modify the clock without acessing the PMU by tampering with the GPS signal, which the PMU requires as input [74].

If the attacker cannot get access to the PMU, it is also possible to modify data on the path from the PMU to the CC. PMU data is usually send to a PDC. So it is also possible that the attacker gets access to a PDC and modify the PMU records there. However, to achieve this it is a pre-requisite to have access to the credentials required to encrypt and sign PMU records. PDCs combine multiple PMU records into an aggregated record, and may need to be able to decrypt the records. Therefore, decryption credentials may be accessible on the PDC itself. The PDC may send data to a super PDC, to a phasor gateway, or directly to a CC [129]. In order to provide the network connectivity on the path, there may be also multiple routers or other network devices. All these devices provide additional entry points for an attacker so that the attacker can modify the PMU data. But always with the pre-requisite that credentials for modifying the records or the protocol for sending records does not provide authentication or integrity checks.

For instance, routers are not required to decrypt PMU data, and therefore have no access to the credentials. Furthermore, if a modification to data is implemented on a router, it has to be ensured that no unusual delay is caused by the modification. If PMU data arrives with too high delay, the values may be discarded or it may be detected.

We also present the manipulate phase angle attack tree as text, showing all the AND/OR relationship.

```
1  Attack Goal: Manipulate phase angle difference used in input to control loop
2  OR
3  1. Manipulate phase angle in PMU records on PMU (at rest)
4          OR
5          1.1. Manipulate local time on PMU
6                  OR
7                  1.1.1. GPS attack
8                  1.1.2. Compromise PMU
9          1.2. Manipulate phase angle values in PMU records on PMU
10                 1.2.1. Compromise PMU
11 2. Manipulate phase angle in PMU records on path (in transit)
12         OR
13         2.1. Manipulate PMU records on own device on path
14                 AND
15                 2.1.1. Install own device in path
16                 2.1.2. Get credentials for record modification
17         2.2. Manipulate PMU data on router on the path
18                 OR
```

```
                2.2.1. Install message modification process on router
                2.2.2. Compromise router on path
                2.2.3. Get credentials for record modification
        2.3. Manipulate PMU records in PDC or sPDC
                2.3.1. Compromise PDC
3. Manipulate input values on controller
        3.1. Compromise controller
```

5.2 False Data Injection Attack

An attacker can compromise meters in a substation or hack the computers storing the meter measurements and inject malicious data. These malicious measurements can affect the SE and the resulting wrong information can reduce the situation awareness (SA) of the operators, which helps the attacker reaching the malicious goals. Such data attacks are named as false data injection (FDI) attacks.

We make an assumption that an attacker has the required knowledge of the power system (e.g., measurements, topology) for constructing the FDI attacks. An attacker can also gain access to the power network model, some specific areas e.g., substations, meters/sensors and other devices; and use the resources for constructing the FDI attacks.

Depending on the knowledge and resources, attackers can construct FDI attacks targeting functional or operational components (e.g., states, topology, load) of a power system. In [182] authors mention that FDI attacks can be of 3 types i) state attacks [100, 88] ii) topology attacks [82, 171] and ii) load redistribution attacks [175]. Our focus is the FDI attacks on SE as described in [100, 88].

FDI attacks on SE can construct attack vectors targeting to have different impact e.g., lead to wrong estimation of states and with this mislead the CC by to make wrong decisions. Wrong information about states in a CC can trigger wrong control actions. Consequences due to the FDI attacks on SE are discussed in [97].

FDI attacks against SE are presented by Liu et al. [100]. Authors make an assumption that an attacker gets access to power system configuration information and manipulates the meters. Additionally, authors in [100, 44] show that an attacker can inject arbitrary errors into state variables not detected by bad data detection algorithms. Liu et al. [100] show that an attacker can systematically and efficiently construct attack vectors that arbitrarily change the SE in two scenarios i) the attacker can manipulate only the meters, ii) the attacker is constrained to the specific meters.

Here we adopt the definition of FDI attacks from Liu et al. [100]. Let $\boldsymbol{z} = (z_1, ..., z_m)^T$ be a vector of original measurements, then the observation which may contain malicious measurements is defined as [100]

$$\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a} \tag{5.1}$$

where $\boldsymbol{a} = (a_1, ..., a_m)^T$ is an attack vector, and elements of $\boldsymbol{a}$ are the malicious measurements. If an element a_i is non zero then attacker has manipulated i^{th} meter and replace z_i by $z_i + a_i$.

FDI attacks can be detected if a system (states) is observable. But if an attacker is aware of configuration of a power system and injected malicious measurements then it can mislead SE without being detected by BDD techniques. In addition, an attacker can increase the impact of FDI attacks without triggering alarms (or being detected) by utilizing the tolerance of measurement errors in SE.

5.3 Attack Model

In this section, we introduce a model for FDI attacks. Similar to [100, 39, 24, 112], we focus on FDI attacks of a power system. In our model we address several capabilities (randomizing the signal, adding constant offset, adding incremental constant or random offsets to the signal) of an attacker in a form of a single model.

We show different methods to manipulate measurements assuming that the attacker does not want the changes to be detected. We use attacks on voltage measurements as an example. Similar attacks can be performed on other measurement values, such as frequency, current or phase angle.

We assume that an attacker does not want to exceed any safety limits to cause immediate action but rather wants to poison the measurement data. This poisoning can aim at influencing historic data for planning or post-incident analysis, influencing the state estimation or also preparing an attack, e.g., by poisoning the "new normal", such that the manipulated data points are taken as reference for the subsequent time step.

In our example, the attacker is manipulating only the polar voltage measurement values, as follows. Since we only manipulate voltage, the equation can be expressed with scalars instead of vectors.

$$z_{k,a} := z_k + a_k \tag{5.2}$$

Here z_k is the k^{th} voltage measurement value, a_k is the attack component (that the attacker adds to the measurements) and $z_{k,a}$ is the manipulated measurement value at k. In order to define different attacks, we describe the attack component as a combination of a random component r_k, with $r \sim \mathcal{N}(\mu, \sigma^2)$, a linear increasing component $s \cdot k + c$ with a slope s and a constant offset c.

$$a_k := \begin{cases} r_k + s \cdot k + c & \text{during attack} \\ 0 & \text{else} \end{cases} \tag{5.3}$$

By varying the attack parameters, we implement different types of injection attacks. In our experiments, we consider four general types of attacks:

1. *constant offset (CO)*, where $c > 0$ and all other parameters are zero. In our experiments an attacker adds an offset such that the first manipulated voltage reaches an additional 75% of the nominal voltage, and then keeps the offset constant for all observations.

2. *random offset (RO)*, where we add a random component r and a constant c.

3. *incremental constant offset (ICO)*, where we linearly increase the offset with $s \cdot k + c$

4. *incremental random offset (IRO)*, where we add a random and a linear component $r_k + s \cdot k + c$

5. *incremental random offset with more noise (IROMN)*, where we add a random and a linear component $r_k + s \cdot k + c$. Here the random component is higher than in the incremental random offset attack.

6. *incremental constant offset with high slope (ICOHS)*, where we linearly increase the offset with $s \cdot k + c$. Here the linearly increasing offset is higher than in the incremental constant offset attack.

Table 5.1 illustrates the types of attacks and shows the values we used.

Table 5.1: Attack parameters and attack types

Type	**Random**	**Slope**	**Constant**
Constant offset (CO)	$r = 0$	$s = 0$	$c = 0.075$
Random offset (RO)	$r \sim \mathcal{N}(0, 4 \cdot 10^{-6})$	$s = 0$	$c = 0$
Incremental constant offset (ICO)	$r = 0$	$s = 1.96 \cdot 10^{-7}$	$c = 0$
Incremental random offset (IRO)	$r \sim \mathcal{N}(0, 1.6 \cdot 10^{-7})$	$s = 1.96 \cdot 10^{-7}$	$c = 0$
IRO with more noise (IROMN)	$r \sim \mathcal{N}(0, 4 \cdot 10^{-6})$	$s = 1.96 \cdot 10^{-7}$	$c = 0$
ICO with high slope (ICOHS)	$r = 0$	$s = 4.33 \cdot 10^{-7}$	$c = 0$

The actual measurement in rectangular form is represented as Eq. (5.4).

$$\boldsymbol{z}_k := \begin{bmatrix} V_{k,re} \\ V_{k,im} \end{bmatrix} \tag{5.4}$$

The attack on the polar voltage affects the measurements in rectangular form. The manipulated real and imaginary voltage can be expressed as Eq. (5.5).

$$\begin{bmatrix} {V^*}_{k,re} \\ {V^*}_{k,im} \end{bmatrix} := \begin{bmatrix} V_{k,re} \\ V_{k,im} \end{bmatrix} + \begin{bmatrix} c_{k,1} \\ c_{k,2} \end{bmatrix} \tag{5.5}$$

where $c_{k,1}$ is the resulting offset in real voltage and $c_{k,2}$ is the resulting offset in imaginary voltage at time step k that is caused by adding the different signals from the different attacks to the true voltage values.

We craft the attacks so that the attacks remain stealthy as they can be seen in Chapter 7. Also in the incremental offset attacks, the offset values are arbitarily selected so that the manipulated signal would have approximately 45 degree.

By varying and combining the attack parameters, we implemented four additional attacks that are variations of the basic attacks in Tab. 5.1. In our experiments, we consider four types of attacks:

1. *Small deviation (SD)*, where $c = 0.006$ and all other parameters are zero. In order to make this attack similar to an attack in [139], we inject only a small offset in the signal. The offset value that we selected is greater than the value used in [139] and the attack starting data point is different than in [139].

2. *Random signal with changing variance (RSCV)*, where we add different random components in different intervals. For instance, we add $r \sim \mathcal{N}(0, 4 \cdot 10^{-4})$ from data point 1,000 to data point 1,500, $r \sim \mathcal{N}(0, 5 \cdot 10^{-4})$ from data point 1,500 to data point 2,000 and so on. We select the random values in order to circumvent detection of residuals based method in Sec. 7.1.1.

3. *Incremental constant offset stepwise (ICOS)*, where we add many constants in an increasing order in different intervals. For instance, we add $c = 0.002$ from data point 1,000 to data point 1,500, $c = 0.003$ from data point 1,500 to data point 2,000 and so on. We select the offsets values so that residuals remain below the threshold in both real and imaginary voltages.

4. *Incremental random offset with changing variance (IROCV)*, where we add different random components and different constant components in different intervals. For instance, $c = 0.0005$ and $r \sim \mathcal{N}(0, 3.8 \cdot 10^{-4})$ are added from data point 1,000 to data point 1,500, $c = 0.0009$ and $r \sim \mathcal{N}(0, 9 \cdot 10^{-5})$ are added from data point 1,500 to data point 2,000 and so on. We select the offsets values so that the attacks are able to circumvent detection of residuals based method in Sec. 7.1.1 in both real and imaginary voltage.

Table 5.2 illustrates the types of attacks with extension and shows the values we used. The extended attacks are constructed combining different values of attack parameters.

Table 5.2: Extension of attack parameters and attack types

Type	Random	Slope	Constant
Small deviation (SD)	$r = 0$	$s = 0$	$c = 0.006$
Random signal with changing variance (RSCV)	$r \sim \mathcal{N}(0,\, 4 \cdot 10^{-4})$ $r \sim \mathcal{N}(0,\, 4.5 \cdot 10^{-4})$ $r \sim \mathcal{N}(0,\, 5 \cdot 10^{-6})$ $r \sim \mathcal{N}(0,\, 5.4 \cdot 10^{-4})$	$s = 0$	$c = 0$
Incremental constant offset stepwise (ICOS)	$r = 0$	$s = 0$	$c = 0.002$ $c = 0.0025$ $c = 0.003$ $c = 0.053$
Incremental random offset with changing variance (IROCV)	$r \sim \mathcal{N}(0,\, 3.8 \cdot 10^{-4})$ $r \sim \mathcal{N}(0,\, 4 \cdot 10^{-4})$ $r \sim \mathcal{N}(0,\, 9 \cdot 10^{-4})$	$s = 0$	$c = 0.0005$ $c = 0.0009$ $c = 0.001$

5.4 Summary

In this chapter, we presented the threat analysis on WAMS and an attack model for false data injection.

We first presented an analysis of attack vectors on the physical devices, software and communication networks. Then we discussed the attack scenarios against the components of WAMS and their consequences. The attack scenarios of PMUs, PDCs, super PDCs, PGW and routers in the WAMS and the possibilities of transforming falsified data in the control center highlighted the attack surfaces and the difficulties of launching such attacks. Additionally, we presented a generic attack tree for compromising a device and specific attack trees for i) causing a blackout and ii) manipulating phase angle. The attack trees demonstrated the different opportunities to gain local or remote access to the different components of WAMS. The above mentioned threat analysis, analysis of attack vectors, attack scenarios and attack trees supports answering **RQ 1.1** (How can an attacker cause FDI attacks in a WAMS?) and a part of **RQ 1.2** (How can multiple different false data injection attack forms be expressed in one comprehensive attack model?).

As we already expected for our reasoning **RQ 1.1**, potential attacks are identified by investigating major possibilities how an attacker can launch severe attacks using components for critical functions and making decisions.

In the second step, we introduced FDI attacks. A model for injecting false data attacks is provided. Typical attack parameters and the types of attacks were discussed with specific attack parametrers values. To this end, we developed an attack model that generates types of FDI attacks namely, CO, RO, ICO, IRO, IROMN and ICOHS. Further, we generate attacks SD, RSCV, IROCV and IROS by extending attack parameters and their

values. We used the attack model to show how the multiple different FDI attack forms could be expressed using one comprehensive attack model. The generation of attack types using the attack model supported answering **RQ 1.2**.

As we already expected for our reasoning **RQ 1.2**, the generation of attacks using different attack parameters and extention of the attack parameters showed that an attacker model could generate different attack types using different attack parameters.

CHAPTER 6

PMU Data Analysis

Notice of adoption from previous publications in Chapter 6
Parts of the contents of this chapter have been published in the following papers:

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
The introduction of EPFL network and phase angle and frequency in this chapter is based on the work done in [132]. The selected data in this chapter is based on the work done in [132]. The preprocessing of data in this chapter is based on the work done in [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, we provide an overview of real power system network PMU data along with the analysis of the PMU data. First, we present a voltage, phase angle and frequency analysis and show how we selected data for the experiments. Then we present training and test data, and show data preprocessing.

For our experiments, data is used from the EPFL[1] campus PMU network [47], wherein a smart grid infrastructure has been deployed as part of the electrical distribution network. A 20 kV active distribution network (ADN) connects PMUs via a communication network. The PMUs in the network are intended to meet the requirements of IEEE Std C37.118.1-2011 [5] and IEEE Std C37.118.1a-2014 [7] for synchrophasor measurements in power systems. The PMUs that are described in [145] are deployed in the network. In this setting, for the transmission of measurement data, the PMU records are transmitted using the IEEE Std C37.118.2-2011 [4] standartd with UDP. and communicated over a secured communication network. A detail description of the system architecture and characteristics of the PMUs is presented in [138]. The base voltage of the PMU network is 11, 547.0054 kV.

Here we show an initial analysis of the measured PMU data. The analysis helps us to understand the nature of voltage, phase angle and frequency, and selecting representative datasets for our experiment. The data analysis is presented in the following sections.

6.1 Voltage

Our analysis starts with analyzing voltage values. First we check whether voltage values lie within the safety limits. Further we check missing values and analyze weekly, daily, hourly patterns. Comparison of weekly, daily, hourly basis voltage profiles helps us to better understand the nature of the power system data. For our experiment, we use PMU data from March and April of 2016[2]. All voltage magnitudes in this work are assumed to be per unit (p.u.).

A study of voltage statistical properties, for instance, minimum, maximum, mean, standard deviation makes our analysis supports in selecting a representative dataset for our experiment. The analysis is done for 24 hours of a day for two months March and April, and at different times of a day. For instance, here we briefly show a comparison of voltage profiles at different times of a day in Tab. 6.1. From the table we can see minimum, maximum, mean and standard deviation of voltage do not vary much among the days of months and times of day. Mean and median are very similar, difference of means and medians are approximate 2×10^{-3}. Based on our analysis, we have chosen a representative voltage profile from 02:00-03:00 of UTC (Coordinated Universal Time) because it is in the night and therefore we assume it is more stable. We observe the data pattern and find it is more stable at night. The local time where the grid located is UTC+01:00.

Weekly voltage profiles for several weeks look similar, voltage variation on weekdays and weekends for different weeks look similar. Variation on voltage magnitudes also depends on time of the day. Here we present detail results of a day (1^{st} March 2016) and show voltage statistical properties at different times.

[1] https://www.epfl.ch/en/
[2] http://nanotera-stg2.epfl.ch/

Table 6.1: Voltage analysis at different times of the day, DP: data points, Med: median, STD: standard deviation.

Day	Time UTC	Total DP	Voltage				
			Min	Max	Mean	Med	STD
01.03-14.04	46 full days	$198{,}720\times10^3$	0.951	1.077	1.053	1.054	8×10^{-3}
01.03	08:00-09:00	180,000	1.025	1.060	1.050	1.050	6×10^{-3}
01.03	12:00-13:00	180,000	1.025	1.065	1.060	1.060	6.5×10^{-3}
01.03	18:00-19:00	180,000	1.048	1.068	1.064	1.064	5×10^{-3}
01.03	21:00-22:00	180,000	1.040	1.070	1.068	1.068	6×10^{-3}

6.2 Voltage and Phase Angle

Here we show rectangular coordinates (real voltage and imaginary voltage) from polar coordinate (polar voltage, phase angle) in two cases i) time invariant phase angle (i.e. by fixing the phase angle by the first angle in the data set) and ii) time variant phase angle (i.e. by considering the different phase angles). Some models (e.g., in [139]) we use from literature assume a fixed phase angle as they use simulations. As we use the model, for some experiments we also fix the phase angle by the first observed phase angle in the first case (time invariant phase angle).

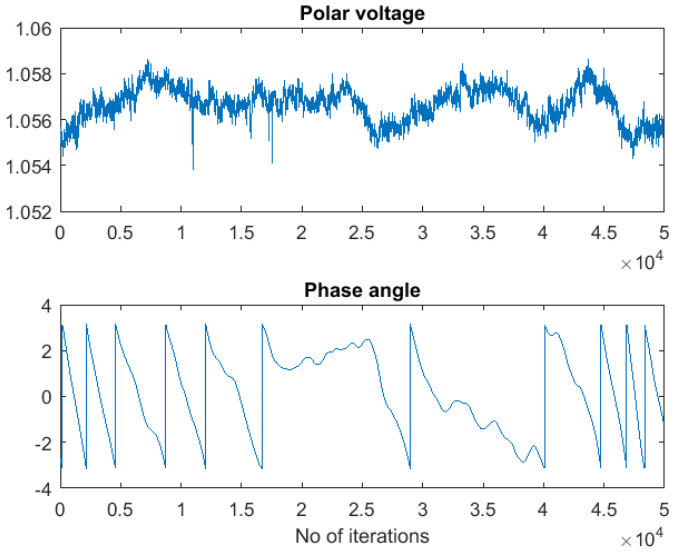
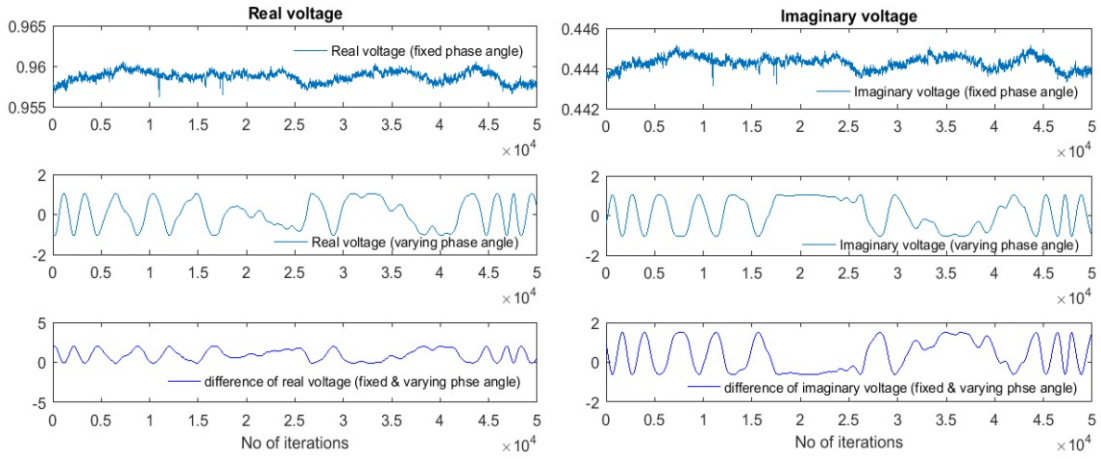


Figure 6.1: Voltage and phase angle change over time. Upper part: observed polar voltage. Lower part: observed phase angle.

Here we aim for analyzing the difference between the voltages in the above mentioned cases. Figure 6.1 shows the observed polar voltage and phase angle, the upper part

visualizes the voltage signal and the lower part visualizes the phase angle.

Figure 6.2 shows the calculated real voltage and imaginary voltage in the cases of constant and varying phase angles. In sub-figure 6.2a, the upper part visualizes the real voltage with fixed phase angle, the middle part visualizes real voltage with varying phase angle, and the lower part depicts the difference between the voltage signals in upper and middle subplots. The real and imaginary part for the varying phase angle are the signals that occur in reality as derived from the original data set. If we fix the phase angle one can see that the real and imaginary voltage differ a lot from the original data. Nevertheless, we decided to do some experiments based on the modified data, in order to analyze methods that only work with a fixed phase angle (see Chapter 4). Similarly Fig. 6.2b shows for imaginary voltage, the difference between constant phase angle and time-variant phase imaginary voltage is significant.



(a) Real voltage (b) Imaginary voltage

Figure 6.2: Observed real and imaginary voltage of time variant and invariant (fixed by first phase angle).

From the figures, one can see that there is a close relation between the real and imaginary voltages and the phase angles. Changes in the magnitudes of real and imaginary voltages depends on the changes in the phase angle.

In the case with fixed phase angle, from the Fig. 6.2 we can see that variance of real voltage is larger than variance of imaginary voltage. As we fix the phase angle to -0.344 radian (first phase angle) the variation of polar voltage has much higher impact on the variation of imaginary voltage.

As expected, from Fig. 6.3, we can see that with small phase angle (smaller phase angle than in Fig. 2.2 of Chapter 2) the conversion of polar voltage to real and imaginary voltage has high influence in real voltage. Thus it causes the variation of real voltage is higher than in imaginary voltage.

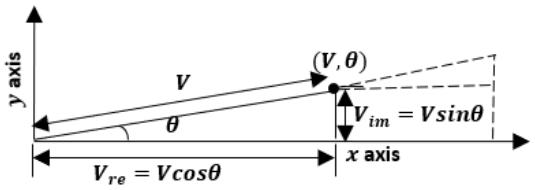
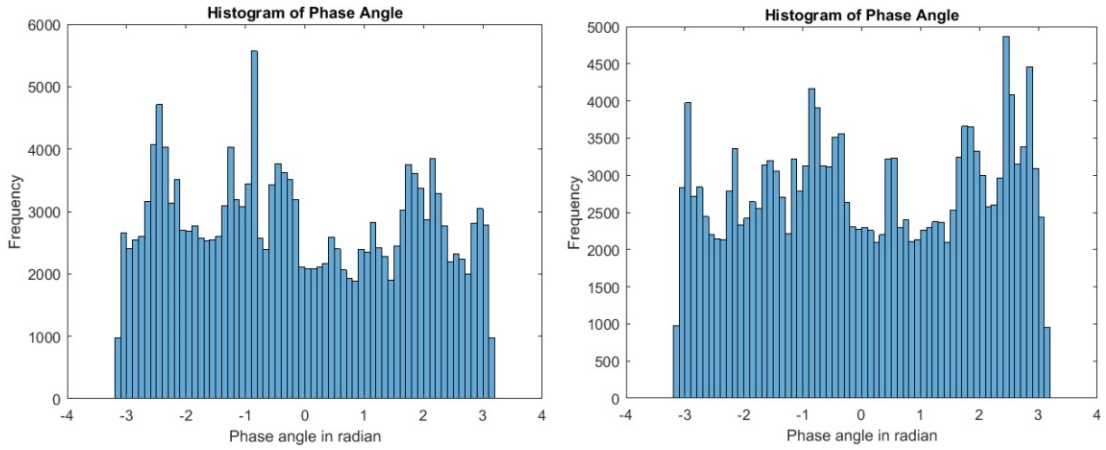


Figure 6.3: Conversion from polar voltage to real and imaginary voltages (small phase angle).

In the first case (with fixed phase angle), if an attacker manipulates voltage then the signal changes significantly which can trigger an alarm but in second case (with changing phase angle) if an attacker manipulates voltage while the signal is in lower peak or while phase angle changes slowly then there is less chance of triggering an alarm. If an attacker manipulates voltages by adding offset to the signal then there is less possibility of detecting the offset. Figure 6.4 shows histograms of phase angles from training and test data.



(a) Histogram of training data phase angles.

(b) Histogram of test data phase angles.

Figure 6.4: Histograms of PMU measured phase angles from training and test data.

6.3 Phase Angle and Frequency

A real power system's frequency usually varies over time around its nominal value (50 hertz (Hz) in Europe). Frequency changes cause changes in the phase angle. We make an assumption that the system is in a quasi steady state. Under our assumption the transition matrix A is an identity matrix.

We assume a 50 Hz power system. Our data shows that the frequency is in the range of 49 Hz in most of the time, that is around the nominal frequency 50 Hz. Figure 6.5 depicts the frequency and how the phase angle changes due to the frequency variations.

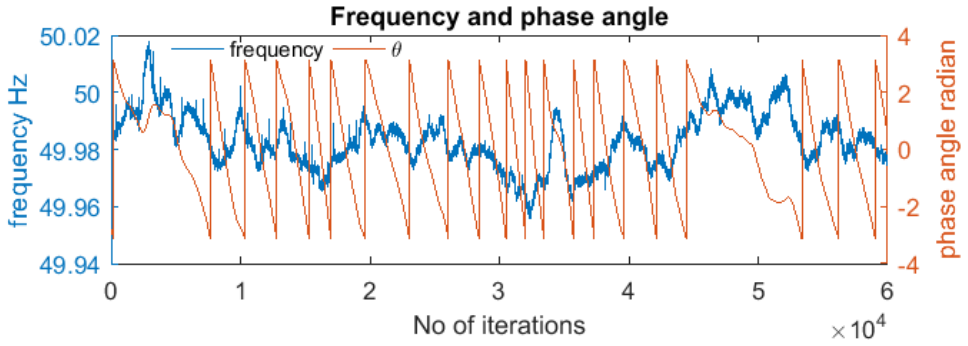


Figure 6.5: Phase angle and frequency change over time (source Paudel et al. [132]).

6.4 Selected Data

In order to build reference data for our experiments, we use data from different days. We always select the same hour (02:00-03:00 time of UTC) per day in order to compare similar behaviour.

We first separate the data into three portions: historical, training and test data. Table 6.2 shows information of data separation.

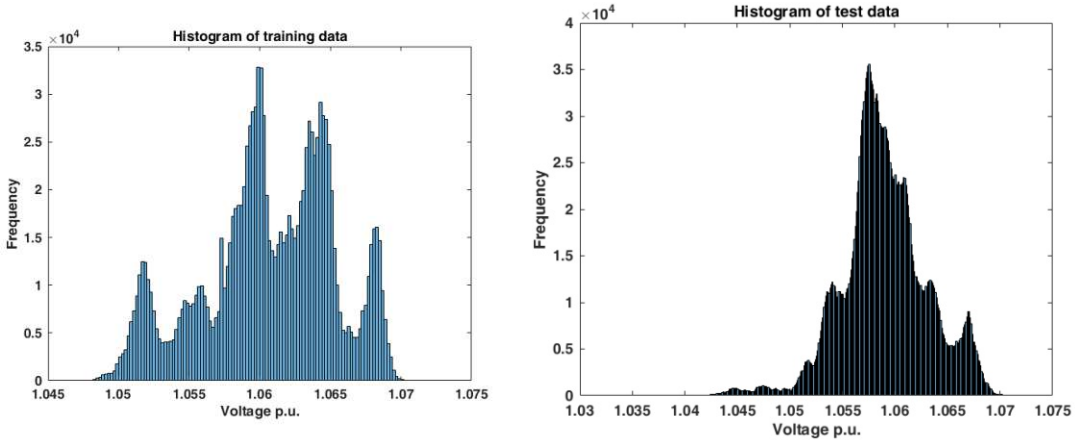
Table 6.2: Separation of datasets, time of UTC.

Dataset	**Day**	**Time**
Historic	01.03.2016-24.03.2016	02:00-03:00 (24×1h)
Training	25.03.2016-31.03.2016	02:00-03:00 (7×1h)
Test	01.04.2016-14.04.2016	02:00-03:00 (14×1h)

The historical data is only used for building a reference histogram for the distribution-based KLD method, and for this we take data from three weeks. Derivation of the reference histogram will be presented in the experimental setup of KLD in Sec. 9.2.2.

The training data is used as a reference for setting thresholds. For this, we take data from 7 different days, always taking one hour from each day. The test data is used to test our algorithms. For this, we use data from 14 different days, considering the same hour per day. In order to test the algorithms, we inject different attacks starting at same data point on all test data sets, so that we have normal and anomalous data points (for detail see Sec. 6.5).

Figure 6.6 shows histograms of all 7 days training data and all 14 days test data. The histogram of training data (shown in sub-figure 6.6a) has several peaks and is not normal distributed. But the Fig. 6.6 shows the histogram of test data (shown in sub-figure 6.6b) looks a bit closer to normal than the training data, but it still differs from a normal distribution. In order to see how much the distributions differ, quantile-quantile plots of training and test data are shown in Fig. A.7 of Appendix A.6.

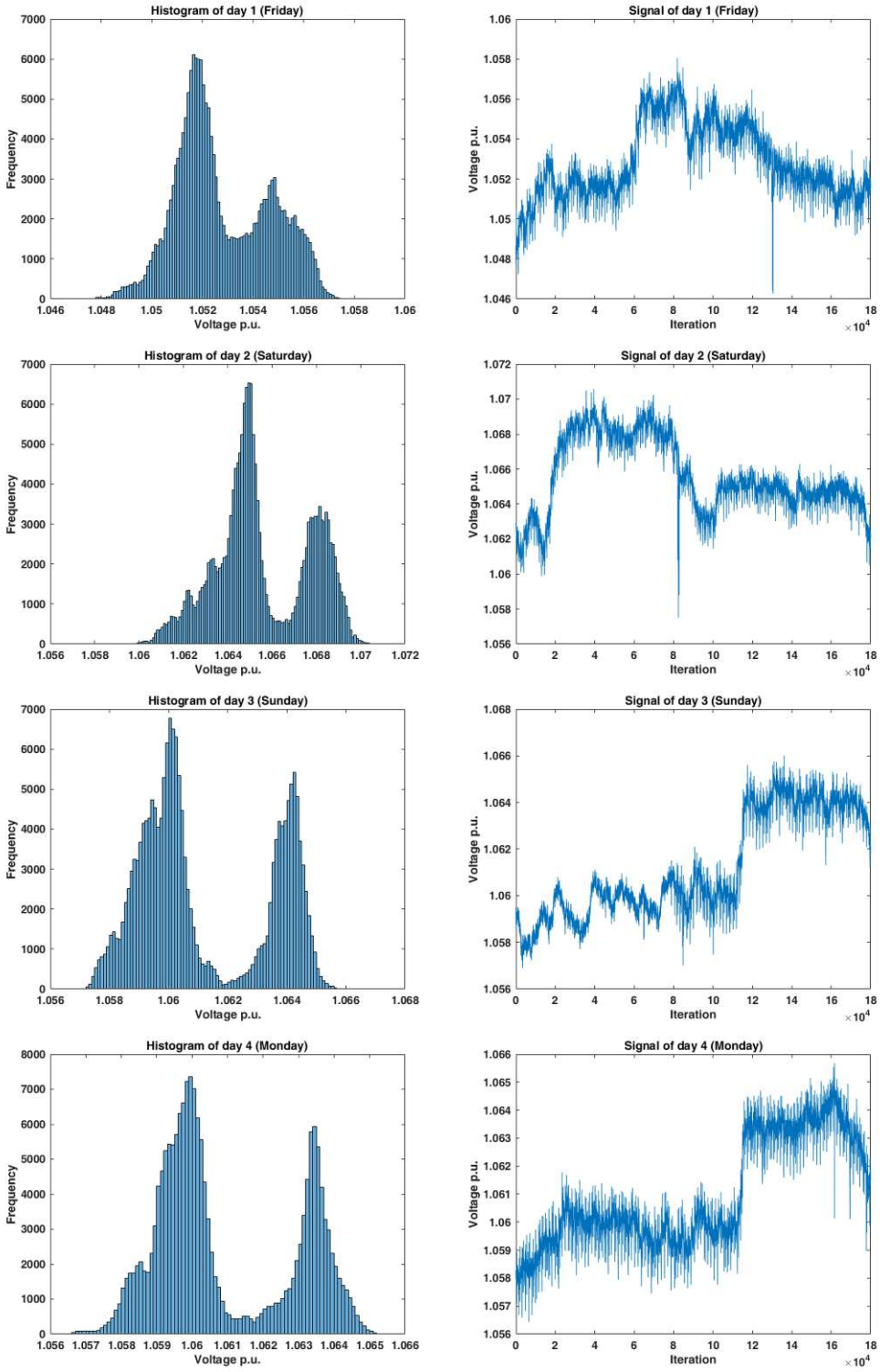


(a) Training data

(b) Test data

Figure 6.6: Histograms of all 7 days training data and all 14 days test data.

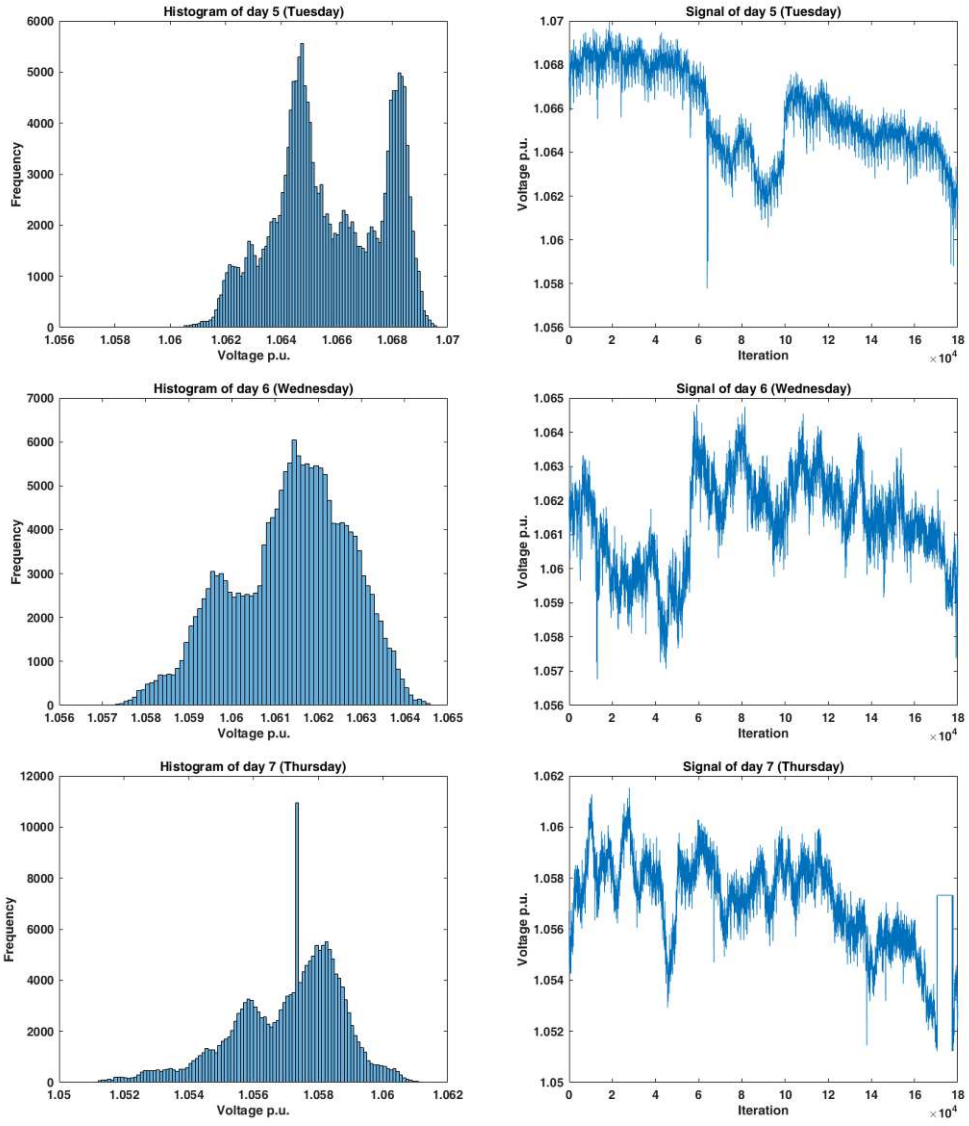
Figure 6.7 shows histograms and signals of the day 1 (25.03 Friday), day 2 (26.03 Saturday), day 3 (27.03 Sunday), and day 4 (28.03 Monday) of the training data. One can see from the Fig. 6.7a, data from the days are different from a normal distribution and further, can clearly see a shift in the mean. So it could be that due to the change of the mean in the signal we here have two distributions that overlap. In order to further analyse the distribution we generated quantile-quantile (Q-Q) plots that can be found in Appendix A.6. Q-Q plots of the day 1 to day 7 of the training data are shown in Fig. A.8 of the Appendix A.6.



(a) Histogram (b) Signal

Figure 6.7: Day 1, day 2, day 3 and day 4 of training data.

Similarly, Fig. 6.8 shows histograms and signals from day 5 (29.03 Tuesday), day 6 (30.03 Wednesday) and day 7 (31.03 Thursday) of the training data. The signals look a bit different in the way that we do not see a clear jump to a different mean. At the end of the day 7 the signal (see sub-figure 6.8b), labelled benign anomalies which were then substituted by us with the median (see sec. 6.5). The substitution causes a high peak in the histogram of the day 7 shown in sub-figure 6.8a.

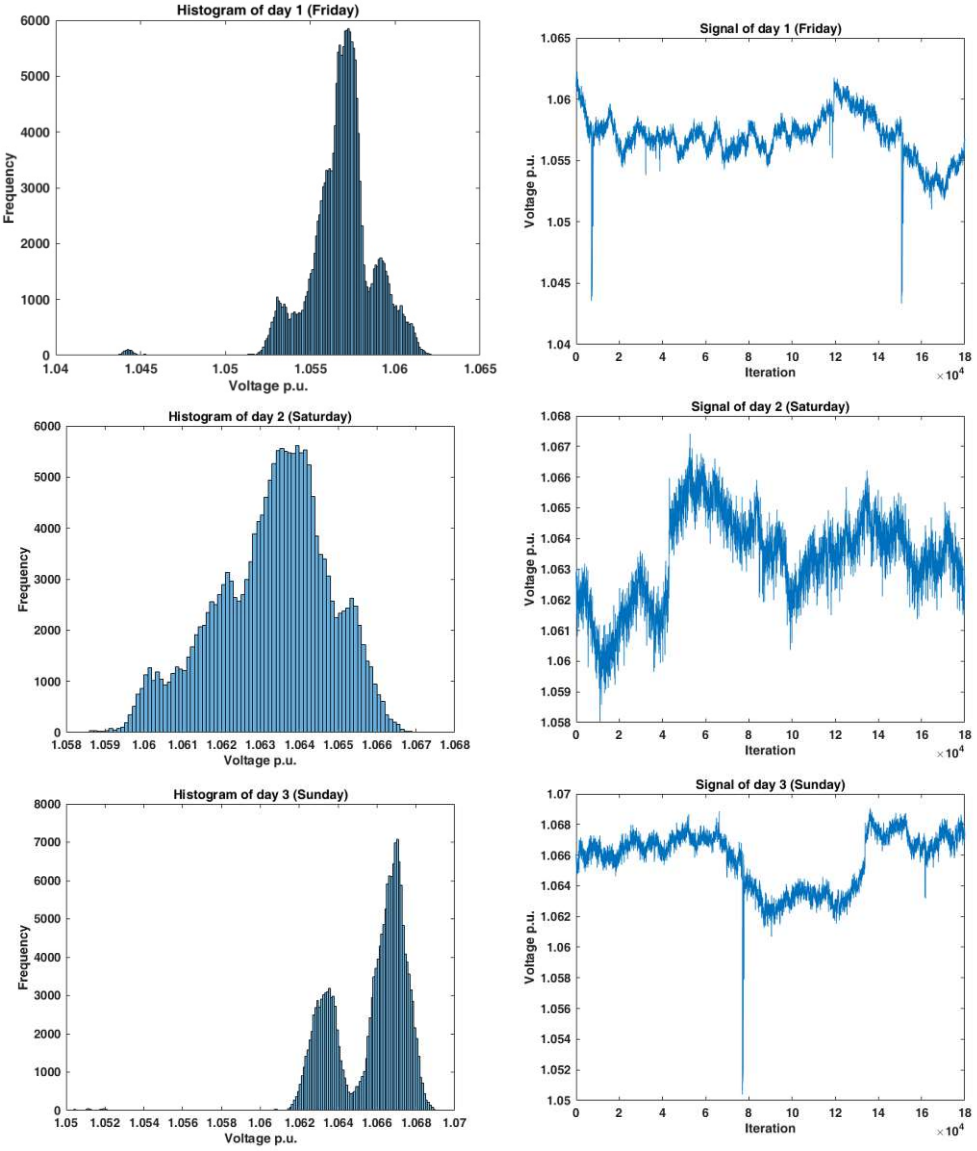


(a) Histogram (b) Signal

Figure 6.8: Days 5, 6 and 7 of training data.

Figure 6.9 shows histograms and signals of the day 1 (01.04 Friday), day 2 (02.04 Saturday)

and day 3 (03.04 Tuesday) of the test data. One can see from the Fig. 6.9a, data from the day 1 and day 2 seem to have only one main peak but are far from normal distributed. Day 3 has two distributions which could be the change in the mean in the signal. Q-Q plots of the day 1 to day 14 of the test data are shown in Fig. A.9 of Appendix A.6.

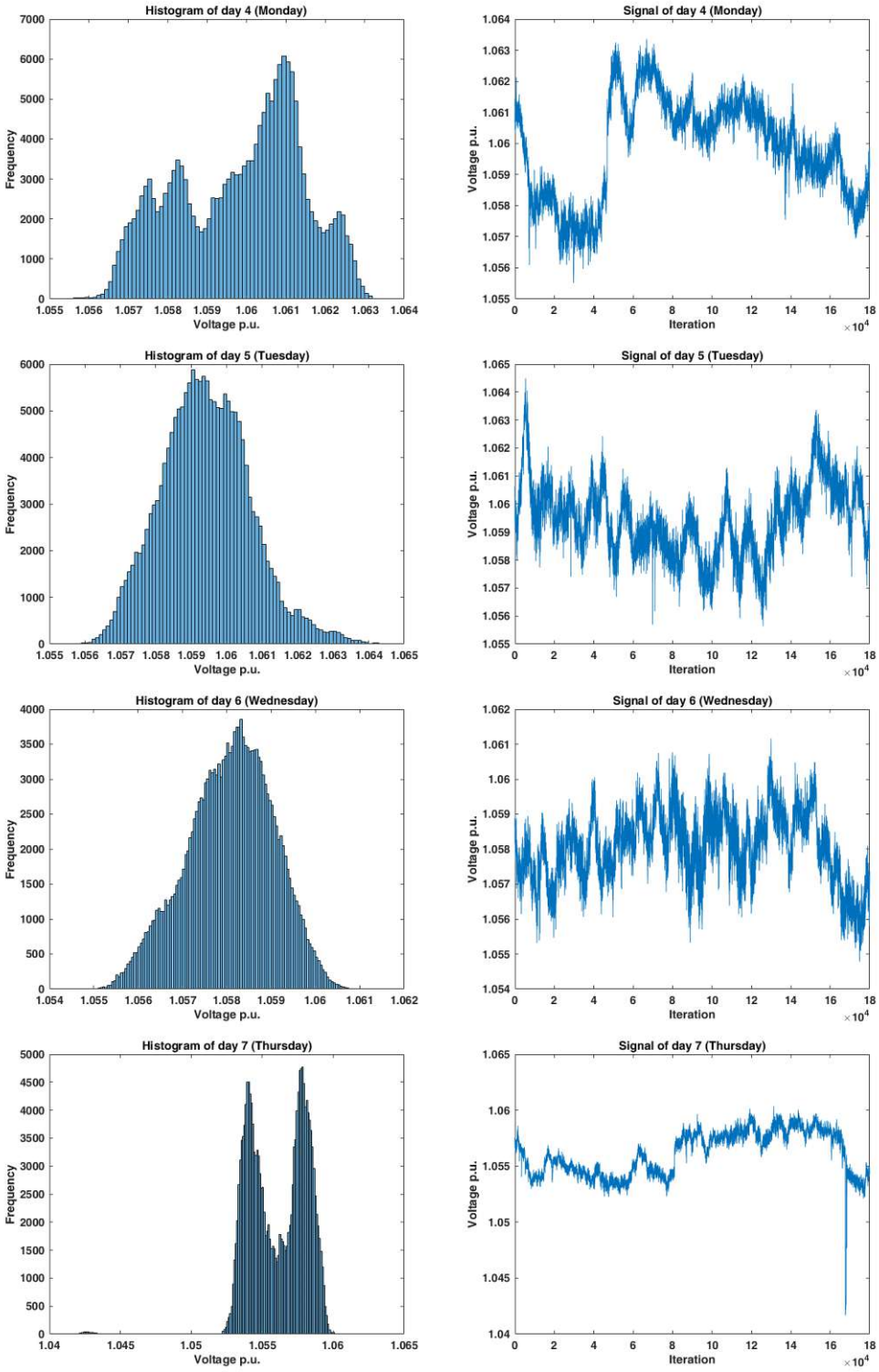


(a) Histogram

(b) Signal

Figure 6.9: Days 1, 2 and 3 of test data.

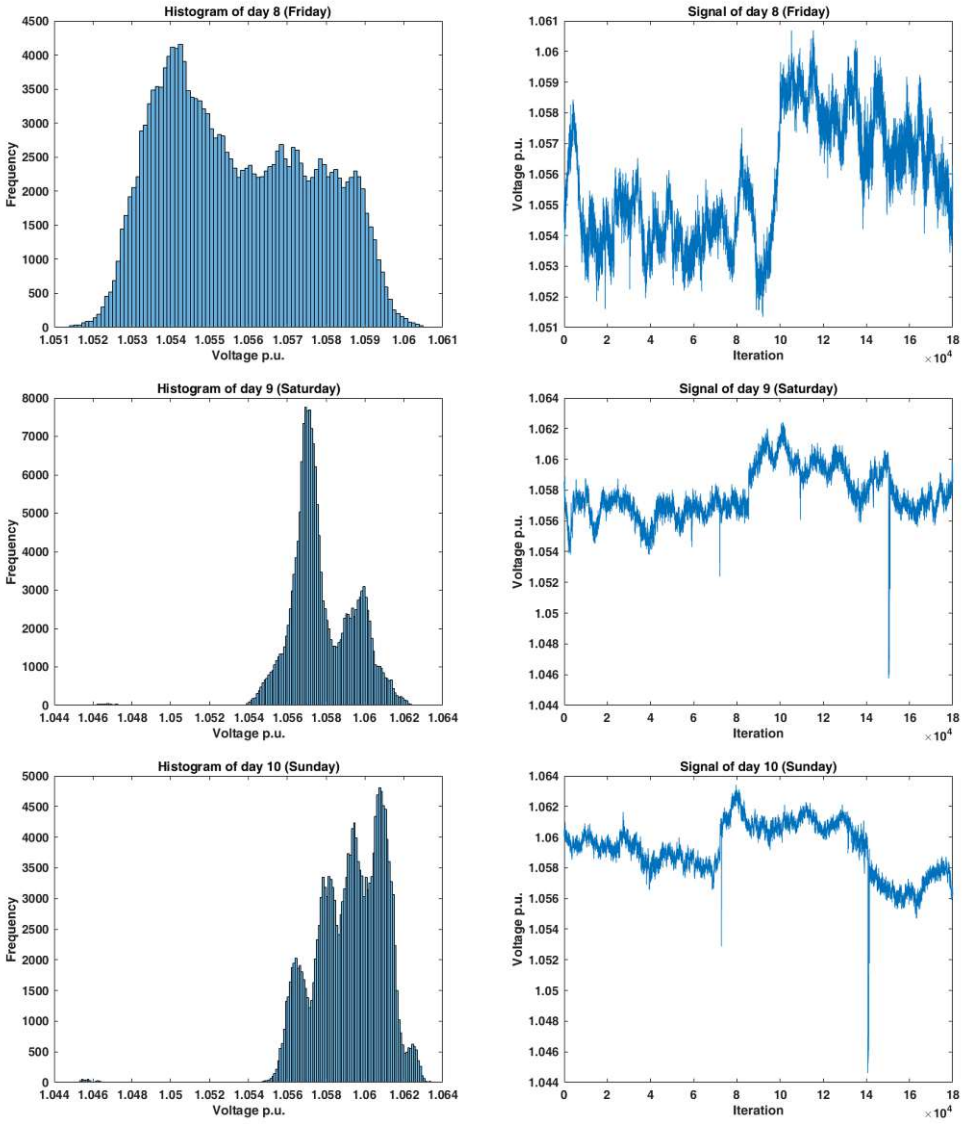
Figure 6.10 shows histograms and signals for day 4 (04.04 Wednesday), day 5 (05.05 Thursday), day 6 (06.05 Friday), and day 7 (07.04 Saturday). From sub-figure 6.10a, one can see data from days 4, 5, and 6 have a single main peak and day 7 has two peaks.



(a) Histogram (b) Signal

Figure 6.10: Days 4, 5, 6 and 7 of test data.

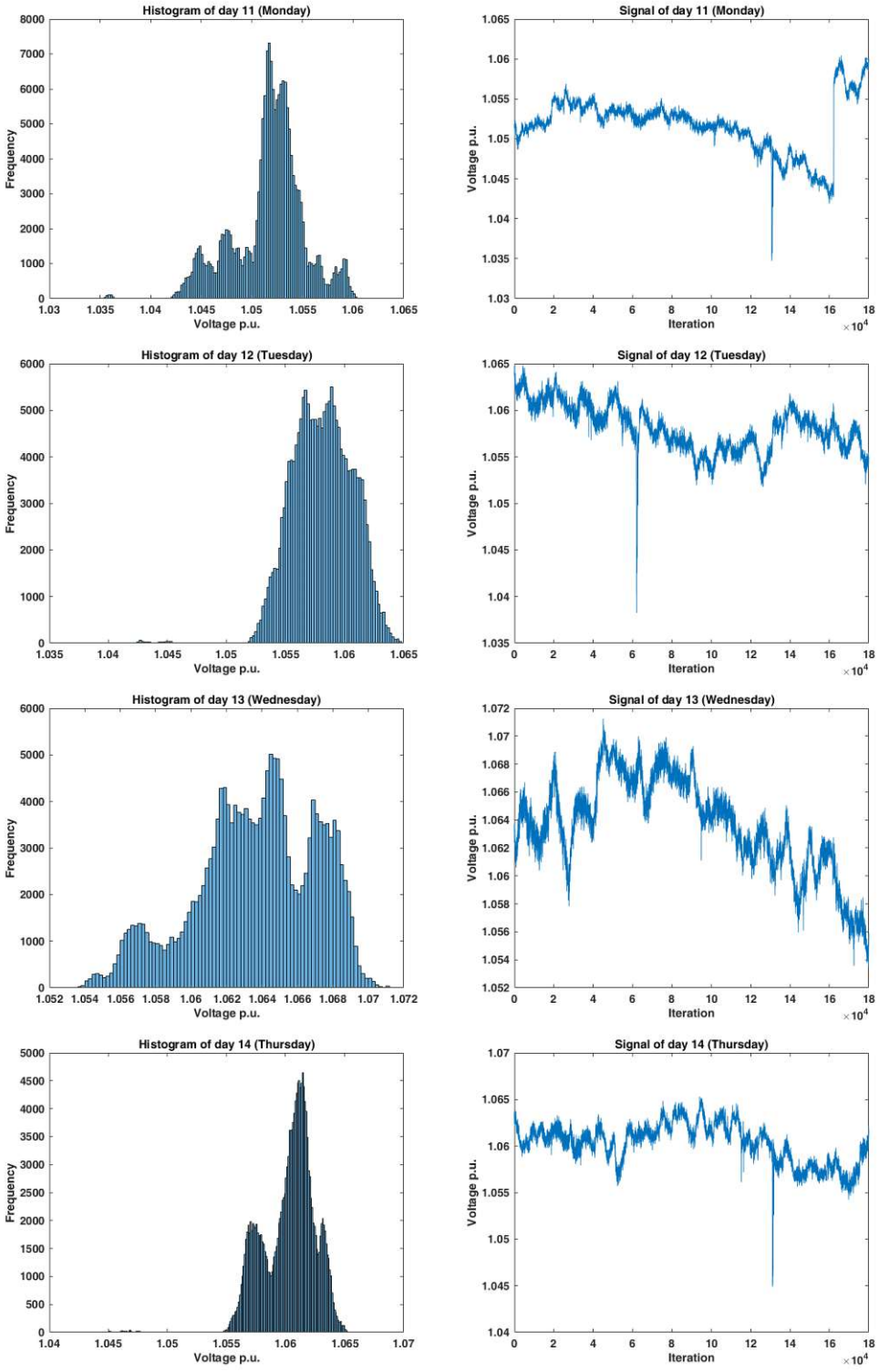
Figure 6.11 shows histograms and signals for day 8 (08.04 Sunday), day 9 (09.04 Monday) and day 10 (10.04 Tuesday).



(a) Histogram (b) Signal

Figure 6.11: Days 8, 9 and 10 of test data

Figure 6.12 shows histograms and signals for day 8 (08.04 Sunday), day 9 (09.04 Monday) and day 10 (10.04 Tuesday). Sub-figure 6.12a shows data from the days 11, 12, 13 and 14 look a bit closer to normal than others.



(a) Histogram (b) Signal

Figure 6.12: Days 11, 12, 13 and 14 of test data.

6.5 Preprocessing

We work with data from a real power system and first perform some pre-processing. Preprocessing consists of a) converting the data to rectangular coordinates b) splitting the data in historic, training and test sets c) labelling the data and d) defining thresholds for the detection.

Converting the data to rectangular coordinates: PMU measurements are observed as a voltage V and phase angle θ, for each of the (three) phases, represented in polar coordinates. We convert the polar coordinates to rectangular coordinates to obtain the real V_{re} and imaginary V_{im} voltages because for state estimation we need the real and imaginary voltage for state estimation. The conversion for the i^{th} measurement from a phase is represented by Eq. (6.1) and (6.2) [149].

$$V_{i,re} = V_i \cdot \cos\theta_i \tag{6.1}$$

$$V_{i,im} = V_i \cdot \sin\theta_i \tag{6.2}$$

Splitting the data: The measurement granularity is always 50 measurement values/sec. The selected data (see in Sec. 6.4) is split into historic, training and test data. Table 6.3 shows detailed information about the data sets used in our experiment, including the total number of data points, number of benign anomalies (BAs) (after the labeling step), number of injected malicious anomalies (MAs). Only the data in the interval of 1 hour is used.

Table 6.3: Data sets used for the experiment, showing number of all data points (Total DP), benign anomalies (BA), malicious anomalies (MA), substituted (subs.) (source Paudel et al. [133])

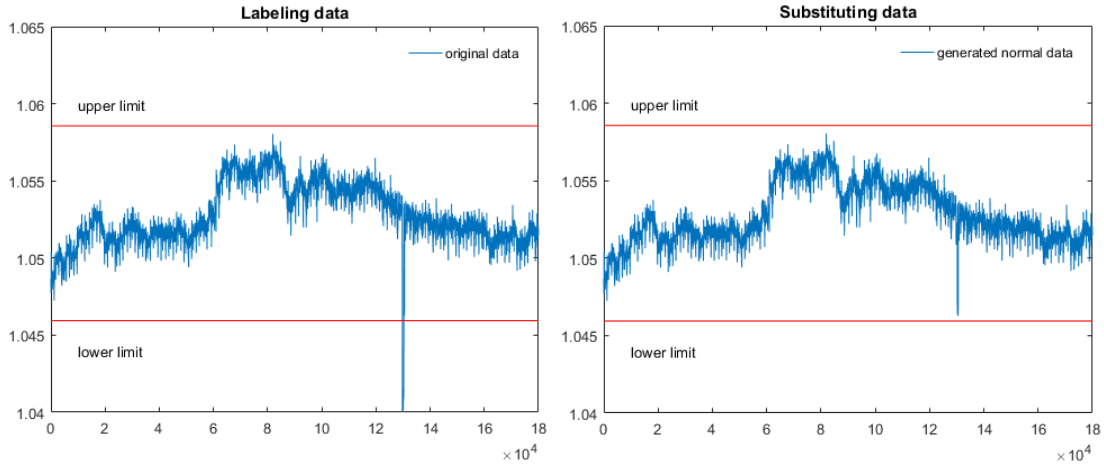
Data	Duration	Time	Total DP	BAs	MAs
Historical	24 days	2am-3am	24 * 180,000	not labeled	0
Training (cleaned)	7 days	2am-3am	7 * 180,000	0 (8,796 subs.)	0
Test	14 days	2am-3am	14 * 180,000	7,727	0
Manipulated test	14 days	2am-3am	14 * 180,000	7,727	1673,087

Labelling the data: Data is labelled with the MAD method to distinguish between normal and benign anomaly (BA) data in the original data. Then all modified values are labeled as malicious anomaly (MA). We consider an attack to be detected as soon as at least one data point is detected as an anomaly.

Labels in training data are needed to find any non-malicious anomalies (BAs). In our case, those BAs are substituted by the median to form a clean set of reference data. And

labels in test data help to assess the detection performance, e.g. to identify whether the detection is due to benign or malicious anomalies. We inspect the data for significant non-malicious anomalies (extremely high values) and found some high values but not with a long duration. In order to decide about anomalies, existing work [139]) use pre-fit residuals based method with a decision level 3. They use simulated data and simulate the data with a Gaussian noises.

For our experiment, as we use real data that vary and also also noisier than the simulated data, we use the MAD method with a decision level 3.5 to mark all data outside the interval $median - 3.5 \cdot MAD < x_i < median + 3.5 \cdot MAD$ (median of 1 hour dataset) as BAs and all data within the interval as normal data points. In the training data, we then replace the data outside of the interval (the BAs) by the median to get a data set with only normal data points. The training data then is used to set suitable thresholds in a way that all normal data points are below the threshold. Figure 6.13 shows labeling data and generating normal data. Sub-figure 6.13a shows labeling polar voltage using MAD interval, and sub-figure 6.13b shows generated polar voltage signal after substituting BAs. In the data set shown there are 462 BAs detected between data points 129797 and 130259. They were then substituted by the median 1.0523 p.u. and the 462 values remain under the threshold. So we still see some unusually low values in the figure.



(a) Original data.

(b) Data with substituted benign anomalies.

Figure 6.13: Labeling with MAD interval and substituting benign anomalies.

Treatment of BAs: In the test data we keep the benign anomalies and in addition inject the four different attack types described in Sec. 5.3 and label those data points (which are not labeled as benign anomalies) as malicious anomalies. If a benign anomaly is detected as an anomaly then it is counted as a true positive. We add the attack in each of the 14 test data files, which each contain 1 hour of measurement data. All attacks start at the $60,001^{st}$ measurement. We assume that the attacker is just running some program to add an attack component and does not check the values before. So therefore

we manipulate the data by always adding the component $a(k)$ to the original value.

We calculate confidence intervals based on the confidence levels of the predicted values of real voltage and imaginary voltage for the plain pre-fit residuals based method described in [139]. The confidence levels of the predicted values are indicated by the measurement innovation covariance matrix.

Defining thresholds for the detection: The cleaned training data is used to define thresholds for the methods L2-norm, normalized residuals, MAD, KLD and CUSUM. For the plain residuals, we use a different method and calculate the threshold at time step k based on the assumed decision level and the standard deviation of innovation at the time step.

For the L2-norm, we use the sequence of L2-norms of the pre-fit residuals from the real voltage and imaginary voltage, and the median absolute deviation to check which decision level we need to set so that all normal data points lie within the interval. Thus, we use a level of decision that covers the L2-norm of pre-fit residuals from real voltage and imaginary voltage of the training data (without benign anomalies) and define a threshold.

For the normalized residuals, we use a level of decision that covers normalized pre-fit residuals of real voltage and imaginary voltage from the training data (without benign anomalies) and define a threshold. For the residual-based detection, we work with rectangular coordinates and a fixed phase angle. But for the lightweight statistical detection methods, we use the original voltage in polar coordinates to set and check the thresholds.

For the MAD, we use a level of decision that covers generated normal training data and define a threshold. For the KLD, we use a level of decision that covers KLD sequence from the cleaned training data and define a threshold. For the CUSUM, we use maximum allowed variation in mean and standard deviation from the cleaned training data, and the desired maximum probability of accepted false alarms to define a threshold. A detailed description about the different parameters, the anomaly detection methods and the thresholds is presented in Sec. 7.1.1.

Figure 6.14 shows the data processing steps. It visualizes the steps for defining a reference histogram, defining thresholds for different anomaly detection methods and the application of the defined thresholds in attack scenarios.

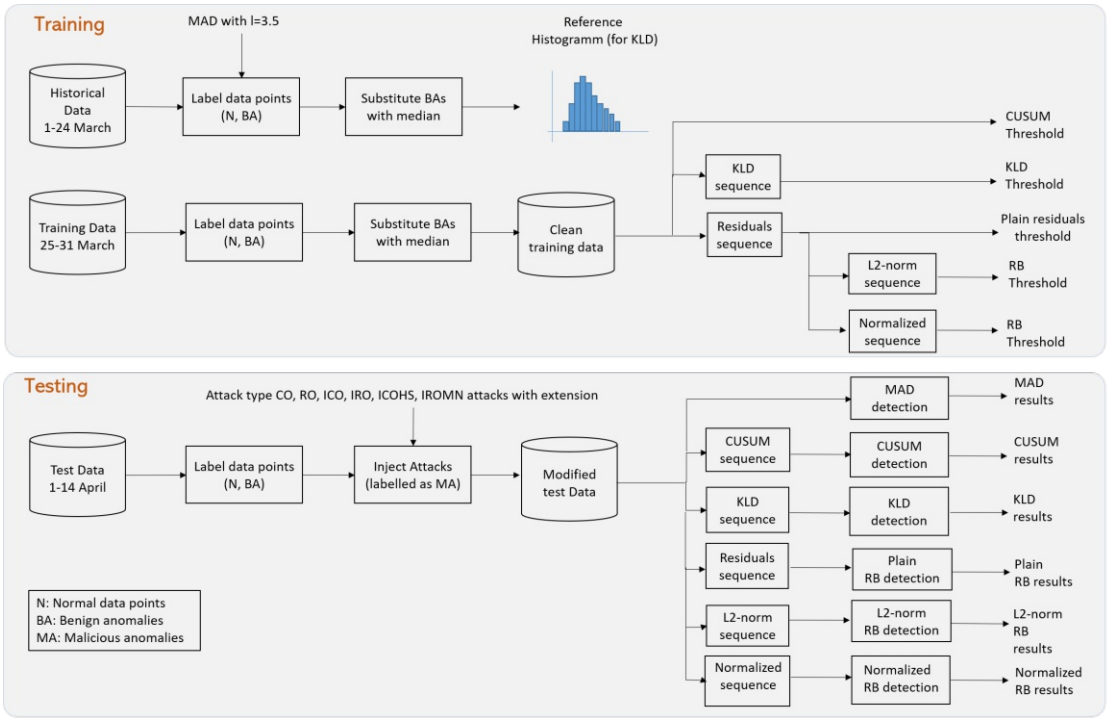


Figure 6.14: Overview of data processing (source Paudel et al. [133]).

6.6 Summary

In this chapter, we provided an overview of the data from the EPFL PMU network and analysed the data set to investigate the relationships between the parameters.

Voltage analysis helped us in the selection of "time per day" in order to select data for our experiment. Further we showed the statistical properties of the data at different times and days. The relationship of the voltage and the phase angle was also clearly shown using this data set. Additionally, the relationship of phase angle and frequency was investigated.

Then the selection of historic, training and test data was provided. Presentation and explanation of histograms and signals of the training and the test data clarified the distribution and outliers of the selected data.

Finally, we provided an overview of the training and testing phases of our experiment. We described the dependencies and the methods used to set parameters and thresholds. Further, details on how data is refined and used in the experiments is provided.

CHAPTER 7

Residual-Based Bad Data Detection Methods

Notice of adoption from previous publications in Chapter 7
Parts of the contents of this chapter have been published in the following papers:

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
The plain pre-fit residual-based bad data detection in this chapter is based on the work done in [132]. The L2-norm and normalized residual-based bad data detection in this chapter is based on the work done in [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, in a first step, we review an anomaly detection method that uses measurement pre-fit residuals of linear Kalman filters to detect unusual measurement values in power grid systems and present two bad data detection methods. To this end, we describe classical residual-based bad data detection methods using L2-norm and normalized

residuals. In a second step, we present the experimental setup of residual-based bad data detection methods. In the experimental setup, training data usage for setting thresholds for the methods is shown with an overview of methods and their thresholds. Then we present results from the experiment. For this, first we present anomaly detection with the plain pre-fit residual-based method in Sec. 7.3.1), second we show undetected attacks by the plain pre-fit residual-based method in Sec. 7.3.2 and last experimental results using L2-norm and normalized residual-based methods in Sec. 7.3.3. We present experimental results of attacks introduced in Tab. 5.2 of Chapter 5 using the plain pre-fit residual-based method and show that the attacks are not detected by the plain pre-fit residual-based method. Similarly, we present experimental results of attacks introduced in Tab. 5.1 of Chapter 5 using L2-norm and normalized residual-based.

Since data sources can have errors, many grid operators implement bad data detection (BDD) methods to check for errors in the measurements that would influence state estimation (SE). For this, usually residuals from the SE are used. Residuals show the difference between predictions and observations (pre-fit residuals) or the difference between estimation and observation (post-fit residuals) [42, 40]. If those residuals get too high, the BDD would be triggered. The residual-based (RB) approach has the advantage that residuals can be simply calculated as a by-product of SE [42, 171].

It seems likely, that BDD also would detect actively manipulated data. Therefore, our research question about using residual-based bad data detection methods reads:

- **RQ 2.1**: To what extent can residual-based bad data detection methods detect different FDI attacks?

Here we assume an attack is detected if at least one of the injected malicious data point is detected. We want to investigate different methods proposed in the literature, we divide the research question **RQ 2.1** into the following sub-research questions:

- **RQ 2.1.1**: Can the plain pre-fit residual-based method proposed in [139] detect the injected attacks in our data set?
  Rationale: The method proposed in [139] uses plain pre-fit residuals from SE using Kalman filter to detect bad data. Therefore, we assume that the method proposed in [139] can also be used to detect the attacks introduced in Tab. 5.2 of Chapter 5 and maybe also other attack types in our data. One difference to our approach is, that [139] uses data from a simulation whereas we use data from real measurements. Therefore, we investigate two variants: our data set but with a fixed phase angle (as in the simulated data) and our data set with a varying phase angle (as in our original data).
- **RQ 2.1.2**: Can attackers avoid being detected if plain pre-fit residuals are used for detection?

Rationale: Anomaly detection methods use pre-fit residuals from SE. But SE adjusts to measurements values. Therefore also the pre-fit residuals change if the measurement values change. Therefore, we would like to check if an attacker who is aware of the pre-fit based detection method can manipulate the measurements in a way that the manipulation does not cause significant changes in the residuals and therefore the attack remains undetected.

- **RQ 2.1.3**: Can the L2-norm residual-based method using LWLS proposed in [100], which is based on LWLS SE, detect our injected attacks in our data set also if we use residuals from DKF?
  Rationale: In [100] the L2-norm of the residuals from LWLS SE is used for BDD. Since BDD detects deviations from the expected measurement sequence, we assume it can also detect the manipulated measurements from our attacks. In [100] LWLS is used for SE and for calculating the residuals. Since DKF can be used as an alternative for SE, we check both: the detection with residuals from LWLS (as used in [100]) and in addition the detection based on residuals from DKF SE.

- **RQ 2.1.4**: Can the normalized residual-based method proposed in [14] using DKF detect our injected attacks in our data set?
  Rationale: In [14] the normalized residual-based method is used for BDD. The normalized residual-based method proposed in [14] use residuals from SE using DKF. We assume the normalized residual-based method can detect the attacks that are generated by our model. Therefore, we check the detection of the injected attacks in our data set using the normalized residual-based method if we use residuals from DKF.

In order to show if BDD works on our attacks, we use three BDD methods based on the pre-fit residuals of a linear Kalman filter. We use plain pre-fit residuals, L2-norm and normalized residuals for detecting bad data. Plain pre-fit residuals use a dynamic threshold (based on the confidence level of the prediction) to detect bad data, whereas L2-norm and normalized residuals methods are based on pre-defined thresholds.

Table 7.1 shows the intention of using the residual-based methods and the data used for the experiment. Details on parameter settings for the experiment are presented in Sec. 7.2.

Table 7.1: Overview of residual-based methods.

Methods	Data*	Goals	Sections
Plain pre-fit residuals	Test data (01.04.2016)	- to answer **RQ 2.1.1** - to answer **RQ 2.1.2**	7.1.1 7.2.1 7.3.1 7.3.2
L2-norm	Training data (22.03 - 31.03) Test data (01.04.2016 - 14.04.2016)	- to answer **RQ 2.1.3**	7.1.2 7.2.2 7.3.3
Normalized residuals	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	- to answer **RQ 2.1.4**	7.1.3 7.2.3 7.3.3

* For training and test data of plain pre-fit residuals, L2-norm and normalized residuals, for each day one hour at 00:00-01:00 UTC is used.

7.1 Theoretical Background

7.1.1 Plain Pre-Fit Residuals

Using a DKF, anomalous measurements can be detected using either *pre-fit* [139, 113] or *post-fit residuals* [159, 36, 41]. The former tracks prediction errors and the latter estimation errors. In this work, we make use of a pre-fit residual-based approach to anomaly detection that has been proposed by Pignati et al. [139]. This existing work has shown good detection performance for anomalies with several benign root causes. We adopt the anomaly detection method and apply it in our use cases.

The SE process consists of two stages, as described in Sec. 4.2.1. We do not consider any control input. The true system state at time k with no control input is represented by the linear Eq.(4.14). Similarly, a measurement from a sensor $\boldsymbol{z}_k$ and the predicted state ($\hat{\boldsymbol{x}}_{k-1|k-1}$) at time step k are represented in the equations (4.13) and (4.15) respectively in Sec. 4.2.1.

Anomalies are assumed when the DKF is unable to accurately predict the measured values, so that there is a significant difference between the predicted and the measured value. Pre-fit residuals ($\boldsymbol{y}_k$) (or *innovation*) are determined using Eq. (7.1), where $\boldsymbol{z}_k$ is the observed measurement vector at iteration k, $\boldsymbol{H}$ is the observation model (i.e., a matrix that shows how the state is related to the measurements) and $\hat{\boldsymbol{x}}_{k|k-1}$ is the predicted state, predicted from the previous state, and $\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}$ therefore expresses the measurement vector that one would expect from the predicted state. Figure 7.1 depicts the steps of the anomaly detection method using the pre-fit residuals (notations are

shown in Tab. 4.1 of Chapter 4).

$$\boldsymbol{y}_k = \boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} \tag{7.1}$$

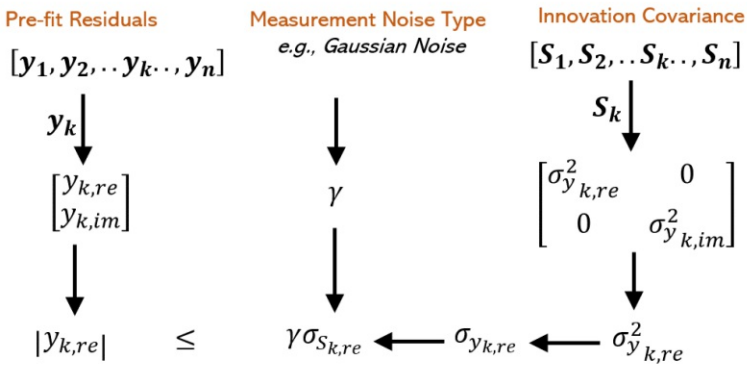


Figure 7.1: Anomaly detection using pre-fit residuals, innovation covariance and measurement noise (only real voltage shown) (source Paudel et al. [132]).

The innovation covariance ($\boldsymbol{S}_k$) is based on the past and current iterations of the DKF, and is determined using Eq. (7.2), where $\boldsymbol{P}_{k|k-1}$ is the predicted process covariance matrix.

$$\boldsymbol{S}_k = \boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T + \boldsymbol{R} \tag{7.2}$$

$\boldsymbol{S}_k$ changes in response to sudden changes in system state. In addition, the Eq. (7.2) with the vectors and the matrices can be represented as Eq. (7.3).

$$\boldsymbol{S}_k = \boldsymbol{H}\begin{bmatrix} \sigma^2_{P_{k|k-1,re}} & 0 \\ 0 & \sigma^2_{P_{k|k-1,im}} \end{bmatrix}\boldsymbol{H}^T + \begin{bmatrix} \sigma^2_{re} & 0 \\ 0 & \sigma^2_{im} \end{bmatrix} \tag{7.3}$$

where $\sigma^2_{P_{k|k-1,re}}$ is the variance of real part process noise, $\sigma^2_{P_{k|k-1,im}}$ is the variance of imaginary part process noise, σ^2_{re} is the variance of the real part measurement noise and σ^2_{im} is the variance of the imaginary part measurement noise. Derivation of the variance of the real part σ^2_{re} and imaginary part σ^2_{im} are shown in Appendix A.12. The predicted process covariance matrix $\boldsymbol{P}_{k|k-1}$ depends on the previous process covariance matrix ($\boldsymbol{P}_{k-1|k-1}$) and the current process noise covariance matrix ($\boldsymbol{Q}_k$) (see Eq. (4.16) in Sec. 4.2.1).

Calculation of $\boldsymbol{S}_k$ using $\boldsymbol{P}_{k|k-1}$ and $\boldsymbol{R}$ from Eq. (7.2) is represented by Eq. (7.4). We assume $\boldsymbol{H}$ as an identity matrix (see Chapter 4).

$$\boldsymbol{S}_k = \begin{bmatrix} P_{k|k-1,re} + \sigma^2_{re} & 0 \\ 0 & P_{k|k-1,im} + \sigma^2_{im} \end{bmatrix} \tag{7.4}$$

where σ_{re}^2 is the variance of real voltage and σ_{im}^2 is the variance of imaginary voltage.

To calculate $\boldsymbol{S}_k$, the prediction error covariance matrix ($\boldsymbol{P}_{k|k-1}$) and the measurement error covariance matrix ($\boldsymbol{R}$) are used. The prediction error covariance matrix uses the process noise covariance matrix. Therefore, innovation gain depends on the process noise and measurement noise covariance matrix. In our scenario, $\boldsymbol{y}_k$ is represented by Eq. (7.1) and (7.5) and $\boldsymbol{S}_k$ is represented by Eq. (7.6).

$$\boldsymbol{y}_k = \begin{bmatrix} y_{k,re} \\ y_{k,im} \end{bmatrix} \tag{7.5}$$

$$\boldsymbol{S}_k = \begin{bmatrix} \sigma^2_{y_{k,re}} & 0 \\ 0 & \sigma^2_{y_{k,im}} \end{bmatrix} \tag{7.6}$$

$\boldsymbol{S}_k$ gives the measurement innovation covariance matrix. The first diagonal element of $\boldsymbol{S}$ represents the variance of the innovation for the real voltage and the second diagonal element represents the variance of the innovation for the imaginary voltage variance.

A confidence interval is calculated based on the confidence level of the predicted values, which is indicated by $\boldsymbol{S}_k$. Each of the elements in $\boldsymbol{y}_k$ satisfy Eq. (7.7) in normal operation (shown for real voltage). We use the following method proposed in [139] to detect anomalies:

$$\mid \boldsymbol{y}_{k,re} \mid \leq \gamma \sigma_{y_{k,re}} \tag{7.7}$$

where $\mid \boldsymbol{y}_{k,re} \mid$ is the magnitude of the real voltage pre-fit residual at time step k, γ is a confidence level, $\sigma^2_{\boldsymbol{y}_{k,re}}$ equals the variance of the innovations up to time step k; and $\sigma_{\boldsymbol{y}_{k,re}} = \sqrt{\sigma^2_{\boldsymbol{y}_{k,re}}}$. Similarly, for measured real voltage $z_{k,re}$ at time step k since residual $\boldsymbol{y}_k = \boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}$, the Eq. (7.7) can be also expressed based on the measurement and the predicted state. An anomaly is detected if the voltage $z_{k,re}$ exceeds one of the thresholds shown in Eq. (7.8) and Eq. (7.9).

$$\boldsymbol{z}_{k,re} \leq \hat{\boldsymbol{x}}_{k|k-1,re} + \gamma \sigma_{y_{k,re}} \tag{7.8}$$

$$\boldsymbol{z}_{k,re} \geq \hat{\boldsymbol{x}}_{k|k-1,re} - \gamma \sigma_{y_{k,re}} \tag{7.9}$$

where $\hat{\boldsymbol{x}}_{k|k-1,re}$ is a predicted state at time step k. It is also done for the imaginary voltage. A data point is considered as an anomalous if it is detected as an anomaly in real or imaginary voltage.

7.1.1.1 Evading Offset Attacks

Here we show how an attacker can evade the plain pre-fit residual-based detection using the attacker model presented in Sec. 5.3.

We consider that an attacker is manipulating the voltage measurement data in the polar coordinates, as it is sent from the PMUs. Suppose that an offset is added in the k^{th}

measurement of the polar voltage magnitude. The real and imaginary voltages after manipulation are given by Eq. (7.10) and (7.11).

$$V_{k,re,a} = (V_k + \text{offset}_{\text{polar}}) \cdot \cos\theta \tag{7.10}$$

$$V_{k,im,a} = (V_k + \text{offset}_{\text{polar}}) \cdot \sin\theta \tag{7.11}$$

The resulting offsets in the real and imaginary voltages depend on the phase angle, and are shown in Eq. (7.12) and (7.13).

$$\text{offset}_{k,re} = V_{k,re,a} - V_{k,re} \tag{7.12}$$

$$\text{offset}_{k,im} = V_{k,im,a} + V_{k,im,} \tag{7.13}$$

The manipulated measurement vector $\boldsymbol{z}_{k,a}$ is represented by Eq. (7.14), where offset represents the vector with the offset in real $\text{offset}_{re,k}$ and imaginary voltage $\text{offset}_{im,k}$ shown in Eq. (7.15).

$$\boldsymbol{z}_{k,a} = \boldsymbol{z}_k + \textbf{offset} \tag{7.14}$$

$$\textbf{offset} = (\text{offset}_{k,re}, \text{offset}_{k,im})^T \tag{7.15}$$

Similarly, the manipulated measurements affect the pre-fit residuals $\boldsymbol{y}_{k,a}$ and the measurement innovation covariance matrix $\boldsymbol{S}_k$, as shown in Eq. (7.16) and (7.17), respectively.

$$\boldsymbol{y}_{k,a} = \boldsymbol{z}_{k,a} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} \tag{7.16}$$

$$\boldsymbol{S}_k = \boldsymbol{z}_{k,a} - \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R} \tag{7.17}$$

Considering the observed PMU measurements, the anomaly detection method tests whether each of the elements in the pre-fit residuals vector – given by Eq. (7.16) – satisfy the condition that is represented by Eq. (7.7). We calculate the maximum undetectable offset by combining equations (7.14), (7.16), (7.17) and (7.7). The result for the real part is given in Eq. (7.18) as shown in [139] and used in [132].

$$\begin{aligned} \mid y_{k,re,a} \mid &\leq \gamma\sigma_{y_{k,re}} \\ \mid z_{k,re,a} - \boldsymbol{H}\hat{x}_{k|k-1,re} \mid &\leq \gamma\sigma_{y_{k,re}} \\ \mid z_{k,re} + \text{offset}_{\text{k,re}} - \boldsymbol{H}\hat{x}_{k|k-1,re} \mid &\leq \gamma\sigma_{y_{k,re}} \\ \mid \text{offset}_{\text{k,re}} + y_{k,re} \mid &\leq \gamma\sigma_{y_{k,re}} \end{aligned} \tag{7.18}$$

This means an attack remains undetected if an attacker can add an offset in the polar voltage in a way, such that the resulting offsets in the real and imaginary voltages fulfil Eq. (7.18) (i.e., the offset has to be small enough not to exceed the typical variations in the residuals).

To achieve this, the attacker needs to have knowledge about the power system, the SE method, and the anomaly detection system. For instance, knowledge about the power

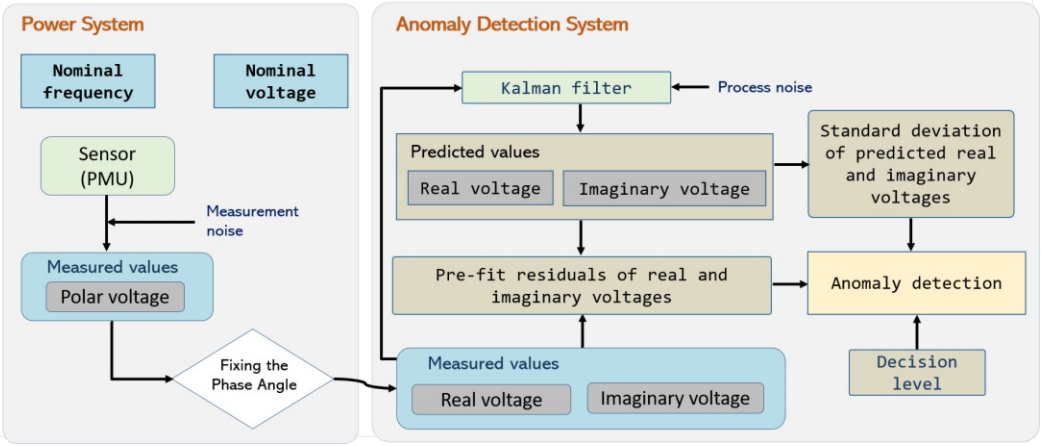


Figure 7.2: An attack detection system, showing major information that is needed for detection (source Paudel et al. [132]).

system infrastructure, nominal values (e.g, voltage, frequency) and SE is required to adjust to the expected innovation covariance matrix $\boldsymbol{S}_k$, which depends on the measurement noise covariance matrix $\boldsymbol{R}$, process noise covariance matrix $\boldsymbol{Q}_k$ and the predicted process covariance matrix $\boldsymbol{P}_k$. In addition, an attacker who wants to circumvent detection, needs to know the factor γ that is used in the anomaly detection system to adjust the threshold. Figure 7.2 depicts an attack detection system. It shows the information that is available from the power system and the anomaly detection system. It is not that easy to gain access to the information of a power system. As pre-fit residuals depend on the predicted value and the observation, the attacker can add a maximum offset remaining within the interval and be undetectable. Such stealthy data attacks can lead to incorrect SE and trigger wrong control actions.

7.1.2 L2-norm of residuals

Instead of plain pre-fit residuals, BDD can also use L2-norm residuals [100]. Here we check the residuals' length calculated using a standard method L2-norm. Generally, the L2-norm residuals method is applied in weighted least squares based SE. In contrast to existing work, here we also make use of the L2-norm pre-fit residuals of Kalman filters based SE to detect false data injection attacks.

The L2-norm of the residual is defined as $\|\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}\|$. The L2-norm is defined as shown in Eq. (7.19) [100].

$$\|y\| = \sqrt{y_{re}^2 + y_{im}^2)} \tag{7.19}$$

Usually a BDD alarm is triggered if $\|y\| > t$ where t is a pre-defined threshold.

7.1.3 Normalized residuals

Instead of plain pre-fit residuals and L2-norm, we can use normalized residuals [14, 87] for BDD. The normalized residuals method works well for independent and non-correlated measurements [100, 53, 115], i.e. a measurement is not interacting with other measurement at time k. We make use of normalized residuals in our use case as in our use case the covariance of real and imaginary voltage is zero.

Normalized residual for k^{th} measurement is calculated as expressed in Eq. (7.20).

$$y_{k,norm} = \frac{|y_k|}{\sqrt{S_{k,k}}} \tag{7.20}$$

where $\sqrt{S_{k,k}}$ is a diagonal element of residual covariance matrix $\boldsymbol{S}_k$, i.e. it is the standard deviation of the residuals up to time step k.

The normalized residual $y_{k,norm}$ is compared to the pre-defined threshold t for checking bad measurement. If $y_{k,norm} > t$ then the k^{th} measurement is detected as a bad data.

7.2 Experimental Setup

7.2.1 Plain Pre-fit Residuals Based Method

For our experiments, we have implemented a DKF and the pre-fit residual-based anomaly detection method (described in Sec. 7.1.1) in MATLAB. To determine the correctness of our implementation, we repeated the experiments that were conducted in [139], but in contrast to using simulation-based data, we use measurement data from the EPFL campus network (see Chapter 6). We conduct an experiment with the small deviation (SD) attack (introduced in Tab. 5.2 of Chapter 5) – a similar experiment was conducted by Pignati et al. in [139]. This is a small deviation type attack described in Sec. 5.3. The offset is introduced to the measured data from iteration 1,500 to 18,000.

7.2.1.1 Detection with Varying Phase Angle

In the original approach described in [139], it is assumed that the network frequency is fixed, e.g., constant at 50Hz. By using simulated data in [139] this condition could easily be established. However, in the true measured data from the EPFL campus, the network frequency changes over time. This is normal behavior in deployed systems, as frequency changes in response to shifting generation and load profiles. Due to the small frequency changes in the measurement data also the phase angle changes over time, as shown by Fig. 6.5 in Sec. 6.3. This is an important difference for the anomaly detection method. The detection method is based on data in the rectangular notation and the

network frequency has a direct influence on the phase angle, and consequently on the real and imaginary voltages that are used for the anomaly detection.

In [139, 146] the confidence interval is defined using a decision level 3 for Gaussian noises. The data we considered for our experiment is from a real power system which is noisy and varies over time so that it results in high standard deviation. Therefore, for our experiment, we define the confidence interval of innovation at decision level 2. Thus, the threshold for a time step k is defined as 2 times the standard deviation of innovation at the time step.

We experiment with the method with a varying phase angle. The results are shown in Sec. 7.3.1.1. From our experiment, we see that in this case the anomaly detection does not work and therefore conclude that the anomaly detection method proposed in [139] only works if there are no frequency changes in the network and therefore the phase angle remains constant. So for real PMU data with varying phase angle the approach does not provide a satisfactory detection performance.

If the detection method proposed in [139] is applied in a network without any modifications, small frequency changes in the system would be sufficient to confuse the detection method.

7.2.1.2 Detection with Fixed Phase Angle

In order to provide comparability with the results in [139], in a similar manner to [137, 50] we fix the phase angle in the measured data with the first observed value. This is a small but important change and due to the modification in the phase angle the new real and imaginary voltage do not reflect the original real and imaginary voltage in the data. Using this modification the method proposed in [139] performs as expected and our implementation of the anomaly detection method is able to successfully detect the offset that is introduced to the measurements (see results in Sec. 7.3.1.2).

In a similar manner to Sec. 7.2.1.1, we assume decision level 2 (see Tab. 7.4). Thus, the threshold for a time step k is 2 times the standard deviation of innovation at the time step. The experiments described in Sec. 7.2.2 and Sec. 7.2.3 were conducted with the modified data (fixed phase angle).

7.2.2 L2-Norm Residuals

The L2-norm method is described in Sec. 7.1.2. In order to set a threshold, we calculate L2-norm of residuals (pre-fit residuals) from the training data without BAs. Values of the L2-norm residuals in training data are then considered as the normal behavior.

In order to set a threshold, we apply the MAD method to the L2-norm residuals of real and imaginary voltages. First MAD is calculated from the L2-norm residuals of real and imaginary voltages, then we calculate the decision level that covers normal behavior based on the maximum L2-norm residual in the training data and the MAD. An interval that

covers all L2-norm residuals of real and imaginary voltages is calculated. It could simply calculate the minimum and maximum value from the timeseries of the L2-norm residuals and use as boundaries, but we make use of the MAD method to calculate the interval because with this we put the boundaries a bit larger than the minimum and maximum values of L2-norm and get a safety margin. We expect high deviations as the variance of the data is quite large, this might be due to high variations in the given time interval. Table 7.2 depicts intervals for different decision levels, the upper boundaries of real and imaginary voltage L2-norm residuals values in a decision level that covers normal behavior are shown in bold numbers. In the table, normal behaviors of both real voltage and imaginary voltage are covered in decision level 7. We use the MAD of the L2 Norm e.g. MAD_{L2}. The interval using the MAD is $median - 7 \cdot MAD < x_i < median + 7 \cdot MAD$, it results in $-0.49 < x_i < 0.67$ for real voltage and $-0.95 < x_i < 1.41$ for imaginary voltage. As the interval at this level covers the normal behavior, we define the greater L2-norm value among the real and imaginary voltages as a threshold. We could also use different thresholds for real and imaginary voltage but we use same threshold for both in a similar manner to existing works using the method (e.g., in [100, 22]). The authors in [100] use chi-square distribution of squares of L2-norm residuals for defining a threshold where a significance level can be chosen based on the system noise. A smaller value of significance level can result in a smaller threshold. Authors in [100] set L2-norm residuals of the actual measurement (without injected bad data) as a threshold. Residuals are very small if there are not any changes. As we aim for detecting an anomaly caused due to the changes, we consider the upper boundary of imaginary voltage for defining a threshold. Therefore the defined threshold for L2-norm is 1.41.

Table 7.2: Upper and lower boundaries of L2-norm residuals of training data for different decision levels, DL = decision level, UB = upper boundary, LB = lower boundary.

DL	**Real voltage**		**Imaginary voltage**	
	UB	**LB**	**UB**	**LB**
3.0	0.34	−0.16	0.73	−0.28
4.0	0.42	−0.24	0.90	−0.45
5.0	0.50	−0.33	1.07	−0.62
6.0	0.59	−0.41	1.24	−0.79
7.0	**0.67**	−0.49	**1.41**	−0.96

7.2.3 Normalized Residuals

The normalized residuals method is described in Sec. 7.1.3. In order to set a threshold of the level of decision, we use the training data without BAs. We calculate the normalized residuals of real and imaginary voltage from the training data. The normalized residuals (pre-fit residuals) from the training data represents the normal behavior.

Table 7.3: Upper and lower boundaries of normalized residuals of training data for different decision levels, DL = decision level, UB = upper boundary, LB = lower boundary.

DL	Real voltage		Imaginary voltage	
	UB	**LB**	**UB**	**LB**
3.0	1.54	−1.01	2.05	−0.99
4.0	1.96	−1.44	2.55	−1.50
5.0	2.39	−1.86	3.06	−2.00
6.0	2.81	−2.29	3.56	−2.51
7.0	3.24	−2.71	4.07	−3.02
8.0	3.66	−3.14	4.58	−3.52
9.0	4.09	−3.56	5.08	−4.03
10.0	4.51	−3.99	5.59	−4.53
11.0	4.94	−4.41	6.09	−5.04
12.0	5.36	−4.84	6.60	−5.55
13.0	5.79	−5.26	7.11	−6.05
14.0	6.21	−5.69	7.61	−6.56
15.0	6.64	−6.11	8.12	−7.07
16.0	7.06	−6.54	8.63	−7.57
17.0	7.49	−6.96	9.13	−8.08
18.0	7.91	−7.39	9.64	−8.58
19.0	8.34	−7.81	10.14	−9.09
20.0	**8.76**	−8.24	**10.70**	−9.60

MAD (represented by Eq. (9.2)) is applied to the normalized residuals of real-and-imaginary voltages. In a similar manner to Sec. 7.2.2, first MAD is calculated from the normalized residuals of real and imaginary voltages, then different decision levels are tested for defining an interval that covers normal behavior. An interval that covers all normalized residuals of real and imaginary voltages is calculated. Table 7.3 depicts intervals for different decision levels, the upper boundaries of real and imaginary voltage normalized residuals values in a decision level that covers normal behavior are shown in bold numbers. We select an interval of a decision level that covers the normal behaviors of both real voltage and imaginary voltage. Decision level 20 covers the normal behavior. We use the MAD of the normalized residuals. The interval using the MAD is $median - 20 \cdot MAD < x_i < median + 20 \cdot MAD$, it results in $-8.24 < x_i < 8.76$ for real voltage and $-9.60 < x_i < 10.7$ for imaginary voltage. A maximum value among real and imaginary voltages at the interval is defined as a threshold. Thus in a similar manner to Sec. 7.2.2, we define threshold as 10.7.

Table 7.4 shows an overview of residual-based methods, parameter settings, thresholds and injected attacks. We recall that the plain pre-fit residual-based method has a dynamic threshold, so threshold at time step k is different from time step k+1, and the threshold of real voltage is different than threshold of imaginary voltage at each time step. We could

Table 7.4: Overview of residual-based methods parameters setting, thresholds and injected attacks; Exp. = experiment; DL = decision level; $\sigma_{y_{k,re}}$ = stdev of real voltage innovation; $\sigma_{y_{k,im}}$ = stdev of imaginary voltage innovation; CO = constant offset; RO = random offset; ICO = incremental constant offset; IRO = incremental random offset; IROMN = incremental random offset with more noise; ICOHS = incremental constant offset with high slope.

Exp.	Methods	Data*	Param. setting	Threshold	Injected attacks	Sec.
7.1	Plain pre-fit residuals	Test data (01.04.2016)	DL = 2	$2 \cdot \sigma_{y_{k,re}}$ $2 \cdot \sigma_{y_{k,im}}$	SD RSCV IROCV ICOS	7.3.1.2 7.3.2
7.2	L2-norm	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	DL = 7	1.41 p.u.	CO, RO ICO, IRO IROMN, ICOHS	7.3.3.1
7.3	Normalized residuals	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	DL = 20	10.70 p.u.	CO, RO ICO, IRO IROMN, ICOHS	7.3.3.1

* For test data of plain pre-fit residuals, one hour at 00:00-01:00 UTC is used.
* For all the given days of training and test data of L2-norm and normalized residuals, one hour at 00:00-01:00 UTC is used.

also use different thresholds for real and imaginary voltage but we use same threshold for both in a similar manner to existing works using the method (e.g.,in [87]). The threshold for real voltage depends on $\sigma_{y_{k,re}}$ - the standard deviation of real voltage innovation, and the threshold for imaginary voltage depends on $\sigma_{y_{k,im}}$. For L2-norm and normalized residual-based methods real voltage and imaginary voltage have a static threshold defined from the training data.

7.3 Results

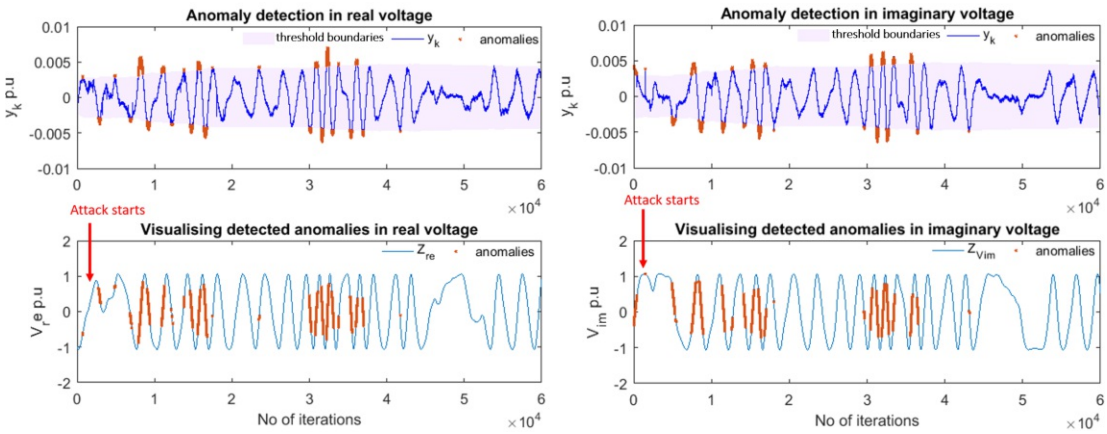
7.3.1 Plain Pre-fit Residual-Based Detection

In order to repeat the experiments conducted in [139], we conduct experiment 7.1 where we also defined a small deviation (SD) attack (introduced in Tab. 5.2 of Chapter 5) and

analyze if the attack can be detected. We apply the plain pre-fit residual-based method to SD attack in two cases: i) varying phase angle and ii) fixed phase angle. Instead of using data from a simulation (as in [139]) we use the manipulated EPFL data set.

7.3.1.1 Results for Varying Phase Angle

Figure 7.3 shows the results of applying the anomaly detection method described in [139] and in Sec. 7.1.1 on the measured data (with keeping the varying phase angle of the original data). As an attack we inject a 0.006 p.u. offset between data points 1500 to 18,000. It can be seen that the anomaly detection method does not perform well at all. More specifically, the method generates over four thousand false positives, i.e., it detected anomalies in the non-manipulated portion of the data, and fails to detect over ten thousand anomalous points (false negatives). As already presented in Fig. 6.5 and Fig. 7.3, there is a clear relationship between frequency, phase angle and the anomalies that are detected in the real and imaginary voltages – during periods when the phase angle is relatively steep (e.g., between 3,000 and 4,000 data points), the detection method identifies the voltages as anomalous.



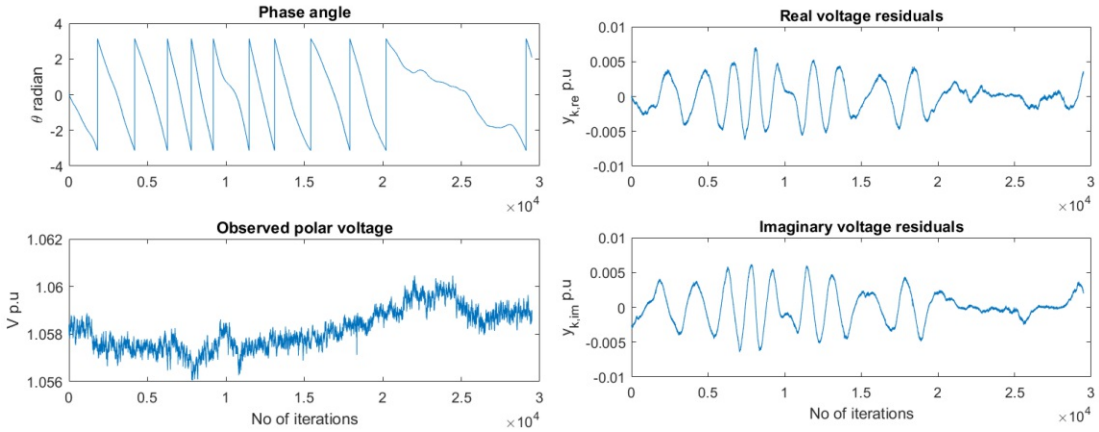
(a) Real voltage (upper figure - residuals, lower figure - voltage)

(b) Imaginary voltage (upper figure - residuals, lower figure - voltage)

Figure 7.3: Anomalies that have been detected in the real and imaginary voltage when an offset of 0.006 p.u. is introduced at data point 1,500, (with changing phase angles) (source Paudel et al. [132]).

Figure 7.3a visualizes the detected anomalies in real voltage. When steepness in the phase angle is high then the difference between consecutive measurements is also high. Thus it results in high pre-fit residual (see upper part of Fig. 7.3a).

Figure 7.4 shows steepness of the phase angle and residuals of real voltage and imaginary voltage increase at the same time. From the figure we can see up to data points 2,000 the



(a) Phase angle (upper part) and observed polar voltage (lower part)

(b) Residuals (upper - real voltage, lower - imaginary voltage)

Figure 7.4: Phase angle together with the residuals and the polar voltage (steepness zoomed in between data points 24,245 and 53,745 of Fig. 7.3).

phase angle steepness and residuals are in same proportion (both of them have a stable period), and between data points 2,000 and 2,900 the phase angle steepness decreases and the residuals also decrease, and thus the periods of the signal increases and decreases again. Therefore, the chance that residuals exceed the threshold and raise an alarm is high when steepness of phase angle is high. If the phase angle is changing fast, then the real and imaginary voltage signal is also changing fast and is therefore harder to predict with the linear model. Therefore the value predicted with a linear model differs from the measurements the residuals also get a high value.

Detected as	**Labeled as**	
	Non-malicious	Malicious
Non-malicious	41,500 (TN)	13,805 (FN)
Malicious	1,999 (FP)	2,696 (TP)

Table 7.5: Confusion matrix of real voltage

Table 7.5 depicts the confusion matrix for real voltage. It shows that the method detects 4,695 anomalies, among them 1,999 non-malicious points and 2,696 malicious points are detected as attacks.

Similarly also in imaginary voltage high steepness of phase angle results in high difference between consecutive measurements. This ends up with high pre-fit residuals. Thus the chance that residuals exceed the threshold and raise an alarm are related to the phase angle changes. If the phase angle changes faster, the difference between predicted state and the observed measurement increases.

The confusion matrix for imaginary voltage is shown in Tab. 7.6 which shows that the

method detects 5,220 anomalies, among them 2,236 are non-malicious points and 2,984 malicious points are detected as malicious.

Detected as	**Labeled as**	
	Non-malicious	Malicious
Non-malicious	41,263 (TN)	13,517 (FN)
Malicious	2,236 (FP)	2,984 (TP)

Table 7.6: Confusion matrix of imaginary voltage

Here, we visualize anomalies of real and imaginary voltages together in polar voltage. Figure 7.5 shows the visualization in the polar voltage. From the experiment, it is clear in the figure that the anomaly detection method proposed in [139] only works well when phase angle steepness is normal during an attack. In Fig. 7.5, the attack starts at data point 1,500 and ends at 18,000; around 30,000 data points polar voltage magnitudes are normal but phase angle steepness is very high. So, these points are detected as anomalies.

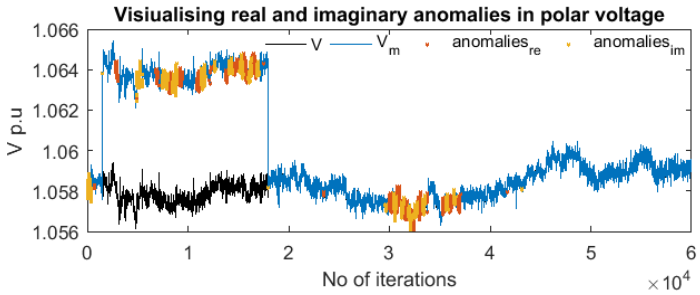


Figure 7.5: Visualisation of detected anomalies in polar voltage - attack starts at data point 1,500 and ends at 18,000 (red points are anomalies detected in real voltage, orange points are anomalies detected in imaginary voltage).

The confusion matrix of the joint results is shown in Tab. 7.7. In the previous paragraphs, it is mentioned that 4,713 anomalies are detected in real voltage and 5,223 anomalies are detected in imaginary voltage but some of the points are detected as anomalies in both cases. So, the method detects total 9,730 points as anomalies. In the joint results, it detects 4,209 non-malicious and 5,521 malicious points as attacks.

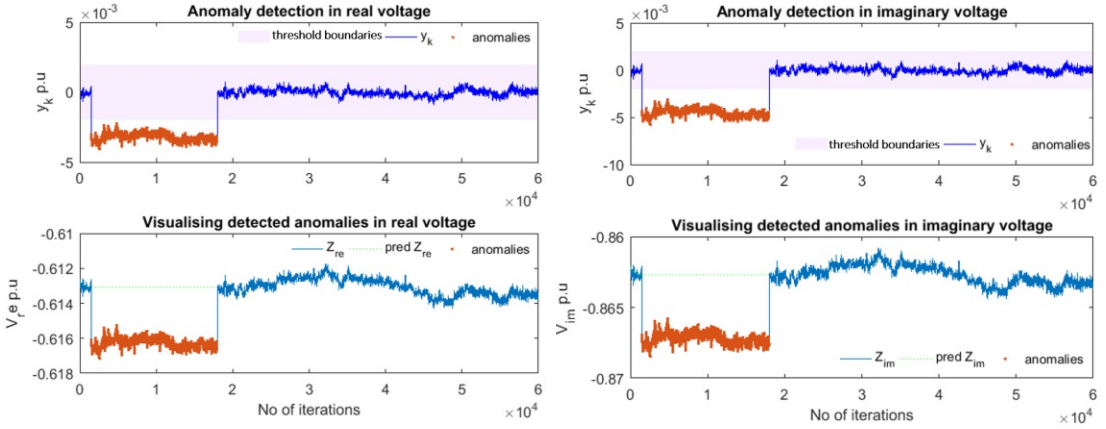
Detected as	**Labeled as**	
	Non-malicious	Malicious
Non-malicious	39,290 (TN)	10,980 (FN)
Malicious	4,209 (FP)	5,521 (TP)

Table 7.7: Confusion matrix of polar voltage

From the detection results, we conclude that the anomaly detection method proposed in [139] does not work with our data. Our assumption is that this is because in our data the frequency and therefore also the phase angle varies. Therefore, we also conducted additional experiments with a fixed phase angle.

7.3.1.2 Results for Fixed Phase Angle

In order to check the method with similar conditions proposed in [139], we fixed the phase angle to -0.3443 radian which is the first phase angle (the phase angle reported in the first PMU message of the test data set) and repeated the experiments. Results can be seen in Fig. 7.6. With the phase angle fixed, the method detects anomalies successfully. In Fig. 7.6a, when the attack starts pre-fit residuals are out of the threshold boundaries in the upper part of the figure. These are pointed out by orange stars. The detected anomalies are visualized in real voltage in the lower part of the figure where z_{re} is the observed measurement for the real voltage and pred z_{re} is the prediction. . Green dots represent replaced values of BD during the attack (proposed by the BDD method in [139]).



(a) Real voltage

(b) Imaginary voltage

Figure 7.6: Anomalies that have been detected in the real and imaginary voltage when an offset of 0.006 p.u. is introduced, (with constant phase angles) (source Paudel et al. [132])

Real voltage confusion matrix in Tab. 7.8 shows that the TP is equal to the number of total malicious points and the FP is 0. Thus the method detects all malicious points as attacks. TN equals to total non-malicious points, so none of the non-malicious points are detected as attacks.

Figure 7.6b visualizes pre-fit residuals and anomalies in imaginary voltage. Resulting offset in imaginary voltage causes pre-fit residuals cross the threshold boundaries. Points

Detected as	Labeled as	
	Non-malicious	Malicious
Non-malicious	43,499 (TN)	0 (FN)
Malicious	0 (FP)	16,501 (TP)

Table 7.8: Confusion matrix of real voltage.

detected as anomalies are pointed out by orange stars in the figure. Green dots during attack represent the replacement of BD.

Imaginary voltage confusion matrix in Tab. 7.9 shows that it hits all of the malicious points. TP equals total malicious points and FN is 0. Thus, it also detects all of the malicious points as attacks in imaginary voltage.

Detected as	Labeled as	
	Non-malicious	Malicious
Non-malicious	43,499 (TN)	0 (FN)
Malicious	0 (FP)	16,501 (TP)

Table 7.9: Confusion matrix of imaginary voltage.

Here, we present the offset and anomaly detection due to the offset in polar voltage. During the attack offset 0.006 is clearly visible in polar voltage as shown in Fig. 7.7. All of the malicious points are detected as anomalies in both real and imaginary voltage. So in the visualization, all anomalies in real voltage overlap with anomalies in imaginary voltage.

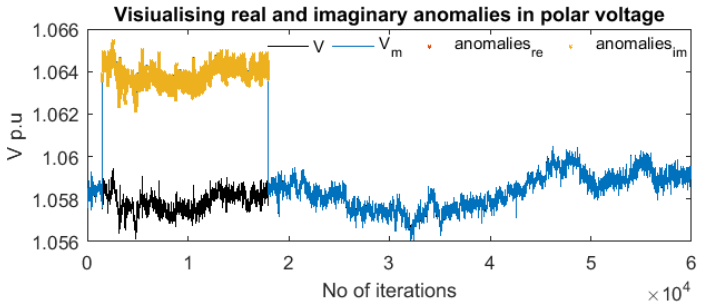


Figure 7.7: Anomaly detection and visualisation in polar voltage.

Confusion matrix in Tab. 7.10 shows joint anomaly detection of real and imaginary voltages. Joint anomaly detection of real and imaginary voltages hits 100% of the anomalies as it hits all the attacks in real and imaginary voltages.

As expected the method works well if the phase angle is fixed. We therefore conclude that the anomaly detection method proposed in [139] only works if the frequency does

Detected as	Labeled as	
	Non-malicious	Malicious
Non-malicious	43,499 (TN)	0 (FN)
Malicious	0 (FP)	16,501 (TP)

Table 7.10: Confusion matrix of polar voltage.

not change and therefore the phase angle remains constant.

7.3.1.3 Results Findings for Plain Pre-fit Residuals

From the results analysis, we conclude the following findings:

- F 2.1.1: With the fixed phase angle, the method proposed in [139] works as expected and detects all attacks, but if the frequency and therefore the phase angle varies the method does not work. Since in a normal power grid, the frequency typically varies, it is unrealistic, that the phase angle is fixed. Therefore, the method proposed in [139] is not well applicable in real power grid scenarios.
- F 2.1.2: With knowledge about the system an attacker can craft an attack such that it cannot be detected in the residuals, i.e., an attacker crafts an attack making sure the residuals remain under the threshold (see Sec. 7.1.1). Nevertheless, the changes have to stay below the normal variations in the pre-fit residuals and therefore usually only small changes may be possible. Relations between the offsets in real and imaginary voltages, and the details are described in Sec. 7.1.1.1.

In the following experiments on residual-based detection, we continue to use the data set with the fixed phase angle.

7.3.2 Undetected Attacks using Plain Pre-fit Residuals

In this section the results of experiment 7.1 are discussed where we present three types of undetected attacks (introduced in Tab.5.2 of Chapter 5), which meet the conditions of detecting offset attacks on real and imaginary voltage that are presented in Sec. 7.1.1.1 (the conditions mean residual y_k at time step k satisfy $\mid y_k \mid \leq \gamma\sigma_{y_k}$) and test with the pre-fit residual-based anomaly detection system in Sec. 7.1.1 with fixed phase angle. In all cases, we arbitrarily choose the 745^{th} data point as attack start point in the data and continue to manipulate the voltage measurements until the end of the measurement series. This results in 17201 manipulated data points. Figure 7.8 depicts an example of the original voltage signal from April 01, 00:00-01:00 of UTC and its histogram. The histogram shows the voltage magnitude in the original signal ranges from 1.0565 p.u. to 1.0605 p.u. The voltage manipulation attacks change the voltage magnitudes and

the range of the values as we will show in the subsequent sections. Figure 7.9 shows actual signals of real and imaginary voltage, and their dynamic threshold upper and lower boundaries. From the figure, one can see how the threshold is adjusted based on the signal.

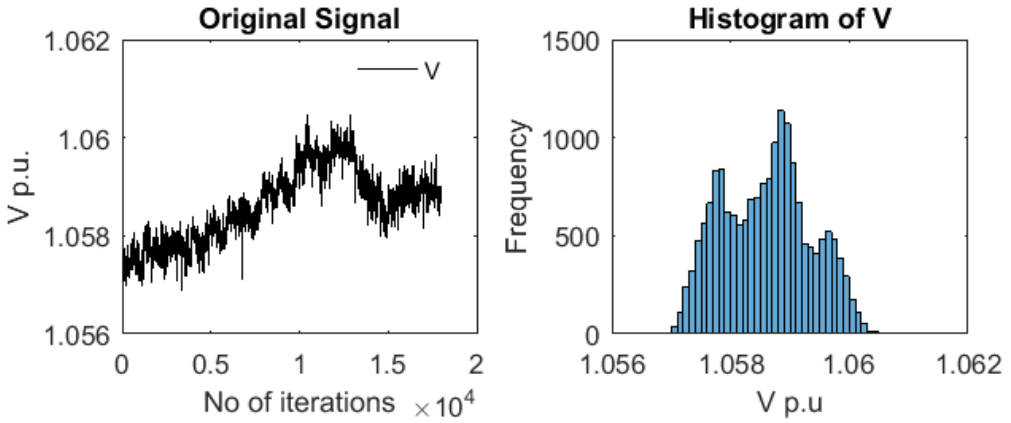
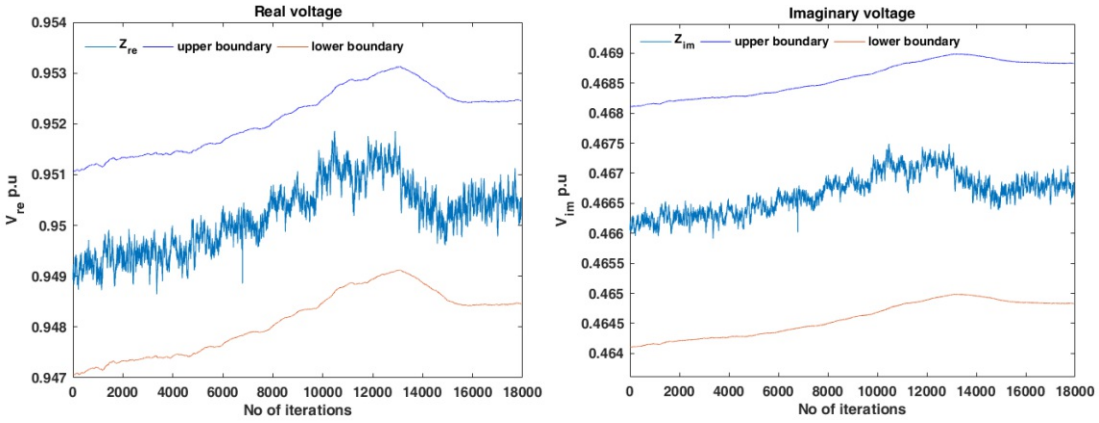


Figure 7.8: EPFL data: original signal and histogram (source Paudel et al. [132]).



(a) Real voltage

(b) Imaginary voltage

Figure 7.9: Actual signal and dynamic threshold boundaries of real and imaginary voltage (the threshold boundaries of the voltages are shown in Eq. (7.8) and Eq. (7.9)).

7.3.2.1 Randomize Signal with Changing Variance (RSCV)

Figure 7.10 shows the added signal (in polar voltage) in RSCV attack. The randomized signal (actual + added signal) and its histogram is shown in Fig. 7.11. In this (RSCV) attack, the ranges (different intervals) and random offsets values are selected based on

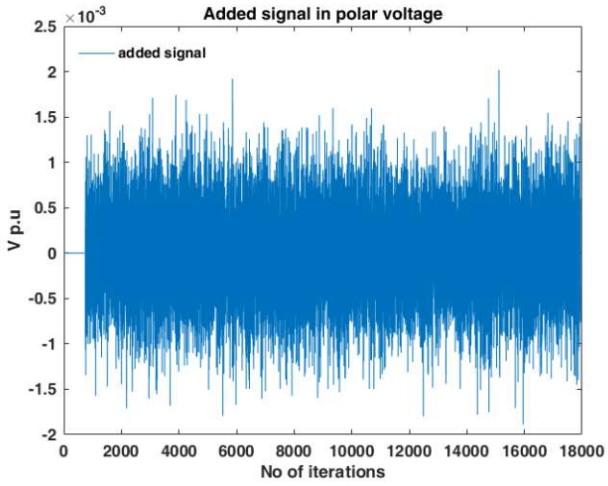


Figure 7.10: Added random signal in RSCV attack (the variations in the stdev is too small to be visible).

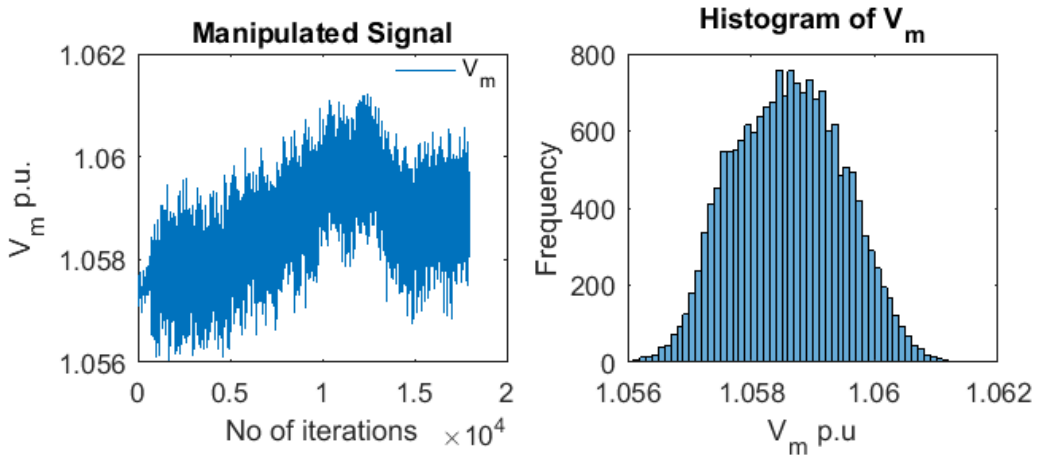
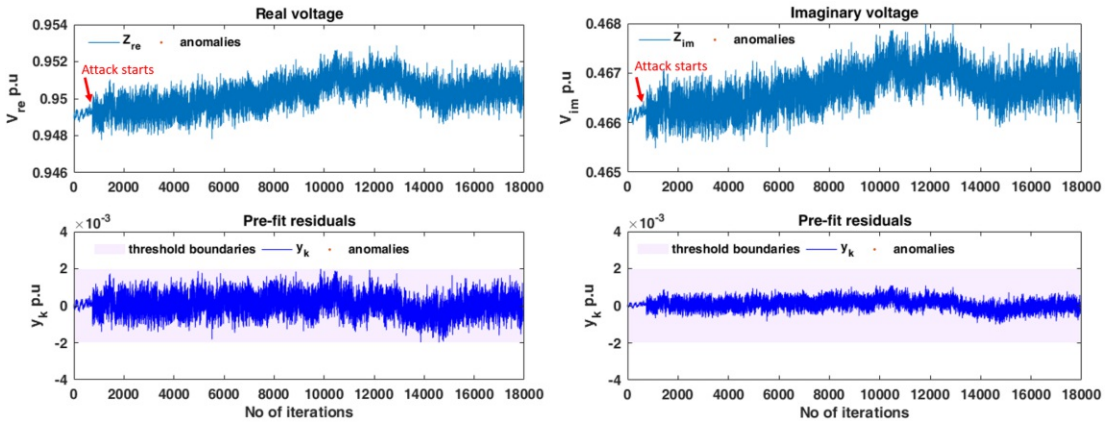


Figure 7.11: Randomized signal and histogram (source Paudel et al. [132])

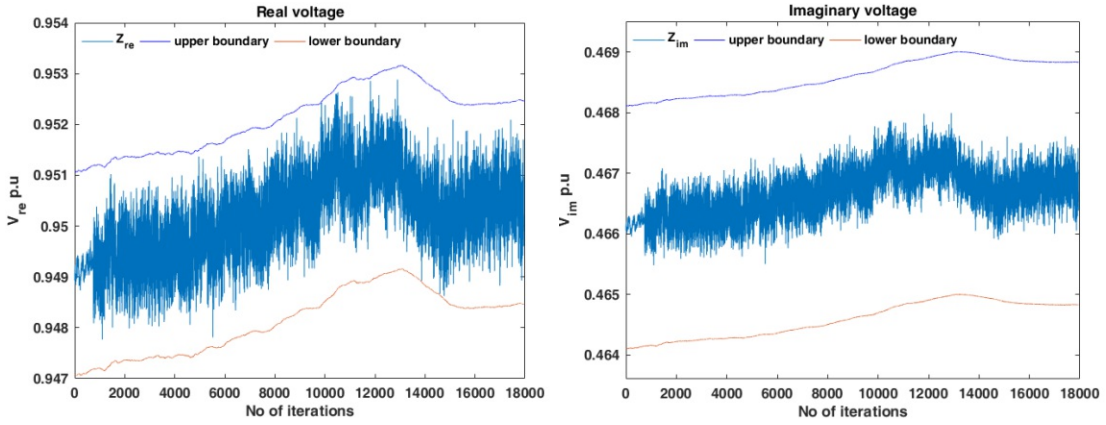
the signal and the expected residuals so that the residuals stay within the threshold boundaries and remain undetected: random offsets from the range $r \sim \mathcal{N}(0,\, 4 \cdot 10^{-4})$ are added from steps 745 to 1,944; the range $r \sim \mathcal{N}(0,\, 5 \cdot 10^{-4})$ is added from step 1,945 to step 8,744; the range $r \sim \mathcal{N}(0,\, 4.5 \cdot 10^{-4})$ is added from steps 8,745 to 9,044; the range $r \sim \mathcal{N}(0,\, 5.4 \cdot 10^{-4})$ from steps 9,045 to 9,644; and randomization varies for intervals till the end. The ranges from which the random values are selected are chosen so that residuals stay with the threshold boundaries and the attack remains undetected.

As can be seen in Fig. 7.12a (bottom), once the signal starts to be manipulated, the pre-fit residuals are clearly affected. This is caused by consecutive (manipulated) measurements



(a) Real voltage (b) Imaginary voltage

Figure 7.12: Upper part: real voltage and imaginary voltage. Lower part: pre-fit residuals and threshold boundaries of real voltage and imaginary voltage pre-fit residuals (the dynamic threshold boundaries of the residuals (shown in Eq. 7.7) have a small variation). No anomalies detected (source Paudel et al. [132]).



(a) Real voltage (b) Imaginary voltage

Figure 7.13: Manipulated signal and dynamic threshold boundaries of real and imaginary voltage (the threshold boundaries in the voltage are shown in Eq. (7.8) and Eq. (7.9)).

having a high difference, which is an abnormal behavior. However, an anomaly is not detected using the threshold presented in Tab. 7.4 and satisfying the condition that residuals stay below the variance (see Sec. 7.1.1.1) to be undetected in both real voltage and imaginary voltage by adding offsets and maintaining residuals below threshold. The dynamic thresholds for the residuals vary but variations cannot be seen because the variation is too small. Figure 7.12a (bottom) shows that the pre-fit residuals for the real voltage stay within the threshold boundaries. The effect of randomization in

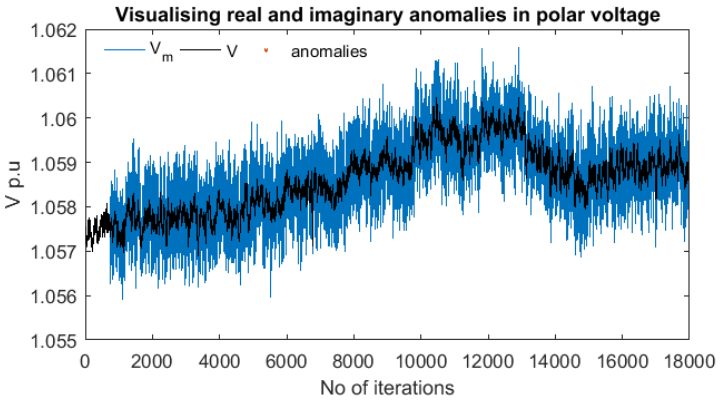


Figure 7.14: Voltage represented in polar coordinates for RSCV attack. No anomalies detected (source Paudel et al. [132]).

real or imaginary voltage depends on the phase angle (in our case constant). In our case the randomization has a higher effect in the real voltage than in the imaginary voltage. Figure 7.12b shows the randomized imaginary voltage and the corresponding pre-fit residuals. Similarly, Fig. 7.13 shows manipulated signals of real and imaginary voltage and the dynamic thresholds (upper and lower boundaries) of the voltages which are calculated based on the predicted states. The threshold boundaries in the voltage are shown in Eq. (7.8) and Eq. (7.9)) which are calculated based on the predicted state, measurement, confidence level and standard deviation of the residuals. From this figure, we can see how the thresholds adjust to the manipulated signal because with the manipulation of the signal also the variation of the residuals changes so that manipulated signal is always within in the boundaries. The original and randomized polar voltages are shown in Fig. 7.14. The original signal is clearly different from the manipulated signal, but the manipulation is not detectable by the anomaly detection system because the variance of the residuals is updated and therefore the residuals values do not exceed the threshold boundaries.

7.3.2.2 Incremental Constant Offset Stepwise (ICOS)

In this ICOS attack, an increasing offset is added in polar voltage starting at the 745^{th} data point. Figure 7.15 shows added signal in ICOS attack. The offset keeps on increasing, remaining undetectable in both real and imaginary voltages. A constant offset of 0.002 is added from step 745 to 944, 0.0025 from step 945 to 1,044, 0.0030 from step 1,045 to 1,244 and so on. The attack ends with a 0.045 offset. During the attack, the maximum offset such that the attack is not detected (taken from Eq. (7.18)) is added ensuring the pre-fit residuals remain within the threshold boundaries. The manipulated signal with its histogram is shown in Fig. 7.16. The histogram shows that after manipulation, the voltage range is shifted and continues up to 1.105 p.u. The polar voltage manipulation

attack results in real and imaginary voltage manipulation.

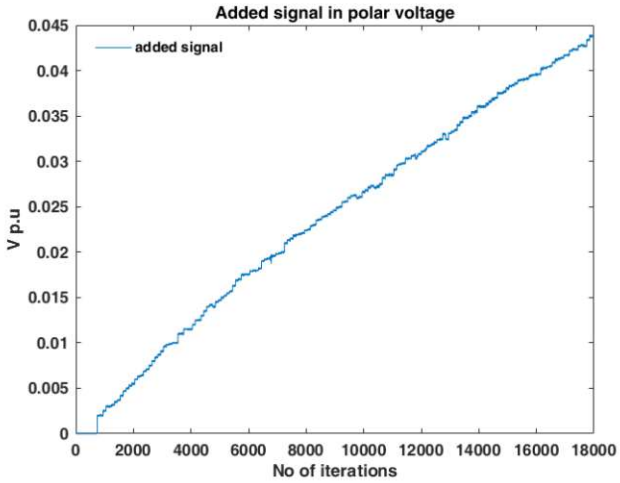


Figure 7.15: Added signal in ICOS attack.

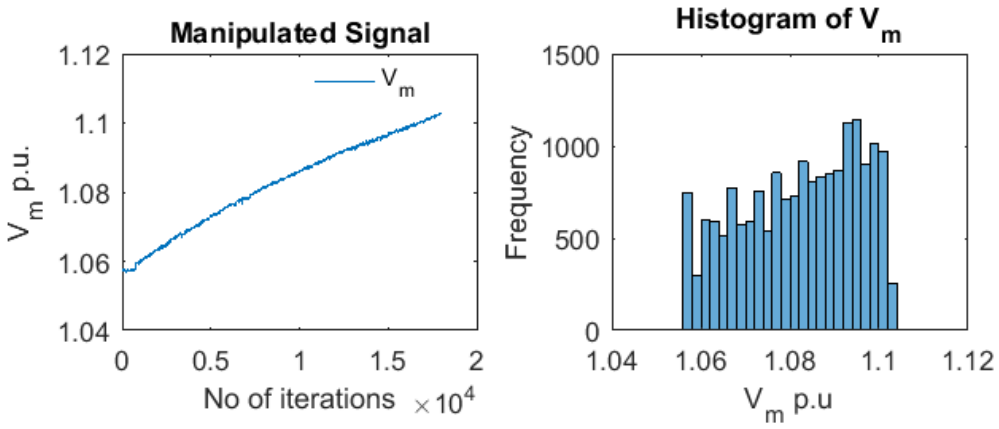
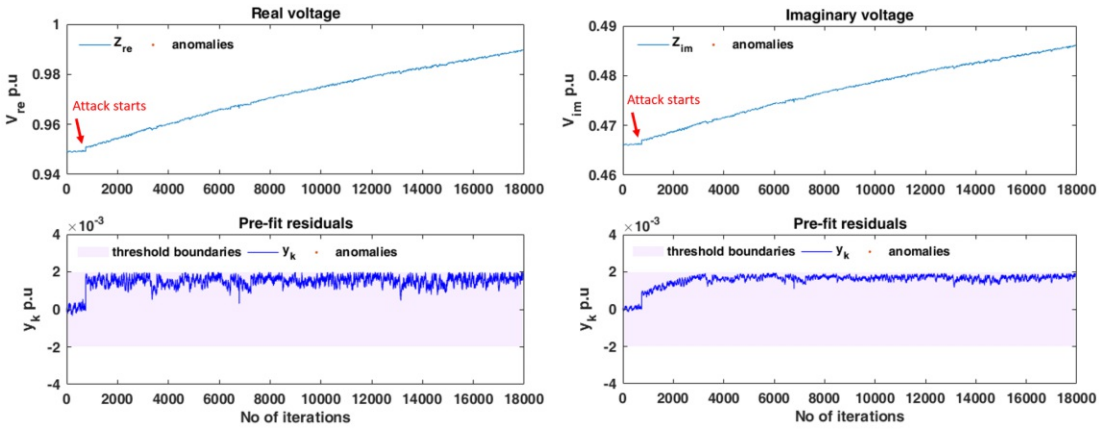


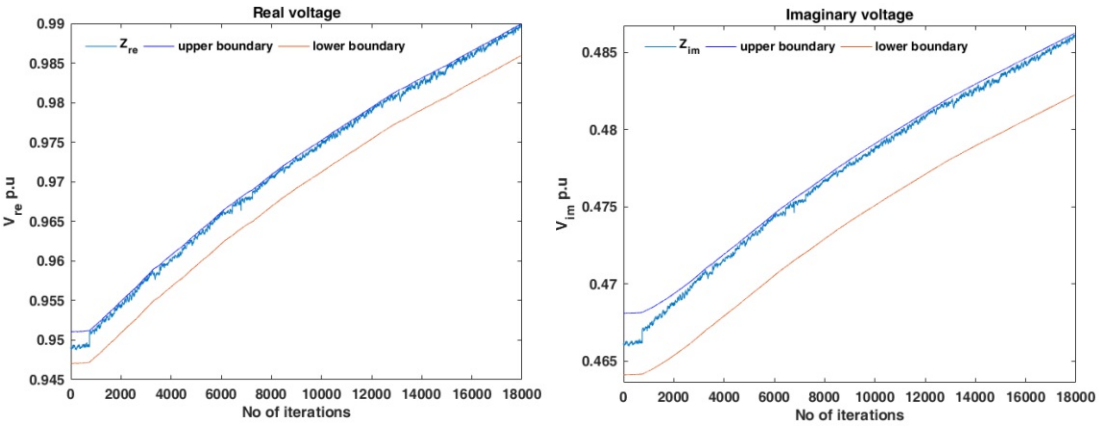
Figure 7.16: Incremental offset-manipulated signal and histogram (source Paudel et al. [132]).

Figure 7.17a (bottom) shows that pre-fit residuals, due to the resulting offsets in the real voltage, do not cross the threshold boundaries whilst an attacker keeps on increasing the offset. Meanwhile, Fig. 7.17b shows the attack is undetectable in the imaginary voltage, as pre-fit residuals do not cross the boundaries of the threshold.



(a) Real voltage (b) Imaginary voltage

Figure 7.17: Upper part: real voltage and imaginary voltage. Lower part: pre-fit residuals and threshold boundaries of real voltage and imaginary voltage pre-fit residuals (changes in the dynamic threshold boundaries are very small). No anomalies detected (source Paudel et al. [132]).



(a) Real voltage (b) Imaginary voltage

Figure 7.18: Manipulated signal and dynamic threshold boundaries of real and imaginary voltage (the threshold boundaries are shown in Eq. (7.8) and Eq. (7.9)).

Similarly, Fig. 7.18 shows the manipulated signals of real and imaginary voltage remain within the threshold boundaries. Figure 7.19 visualizes the undetected offsets in polar voltage. Since the resulting offsets in the real and imaginary voltages remain undetected, the attack remains undetected in the polar voltage as well. The first experiment with the stealthy attack ends with 0.053 p.u. offset. We further increased the stealthy attack up to 30,000 data points, so up to an offset of 0.63 p.u. in polar voltage which was also not detected. By slowly increasing the offset we create a condition in which the manipulated

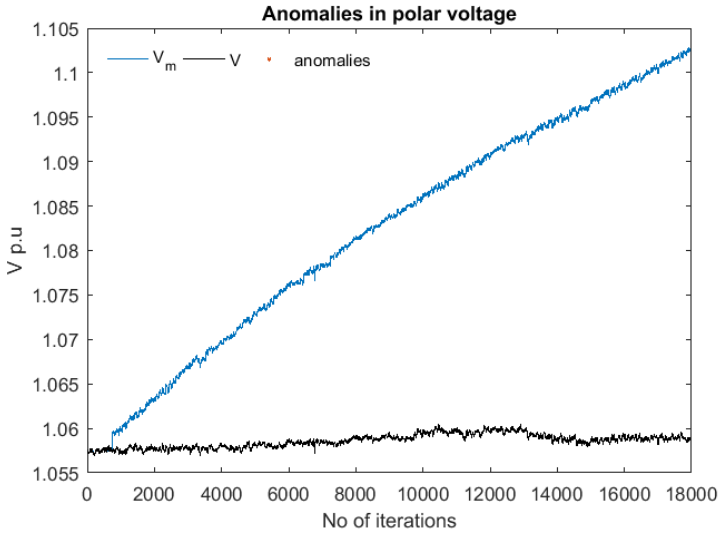


Figure 7.19: Voltage represented in polar coordinates for incremental offset. No anomalies detected (source Paudel et al. [132]).

voltage is 1.121 p.u., while the original voltage is 1.058 p.u. Thus if we increase the offsets slow enough then with the higher offsets the attack can still remain undetected.

7.3.2.3 Incremental Random Offset with Changing Variance (IROCV)

In this scenario (IROCV attack), an attacker adds an increasing offset to the polar voltage measurements, along with a random component that similarly changes over time. The aim of this attack is to hide potential over- or under-voltage situations whilst remaining undetected. Specifically, an offset of 0.0005 plus a random value in the range $r \sim \mathcal{N}(0,\, 3.8 \cdot 10^{-4})$ is added from steps 745 to 1,944. Thereafter, 0.0009 plus a random value $r \sim \mathcal{N}(0,\, 9{\cdot}10^{-5})$ are added from 1,945 to 2,244 steps; 0.0009 plus $r \sim \mathcal{N}(0,\, 4{\cdot}10^{-4})$ are added from steps 2,245 to 2,944; 0.001 plus a random value $r \sim \mathcal{N}(0,\, 4 \cdot 10^{-4})$ are added from 2,945 to 3,244 steps, and so on. The attack ends with an incremental offset of 0.0136.

Figure 7.20 shows added signal in IROCV attack. Figure 7.21 shows how an attacker manipulates the voltage measurements. The histogram in Fig. 7.21 shows that the voltage range is from 1.057 p.u. to 1.072 p.u. Most of the voltage values are greater than the maximum voltage value in the original signal. There are two peaks reaching 1,200 in this range.

As shown in Fig. 7.22a (bottom), as in the previous attack, when the attack is started at the 745^{th} step the real voltage pre-fit residuals are clearly affected. Spikes in the pre-fit residuals of the imaginary voltage are visible once the attack takes place, as shown in Fig. 7.22b. The attack noise is visible by observing the voltage behaviour. From Fig.

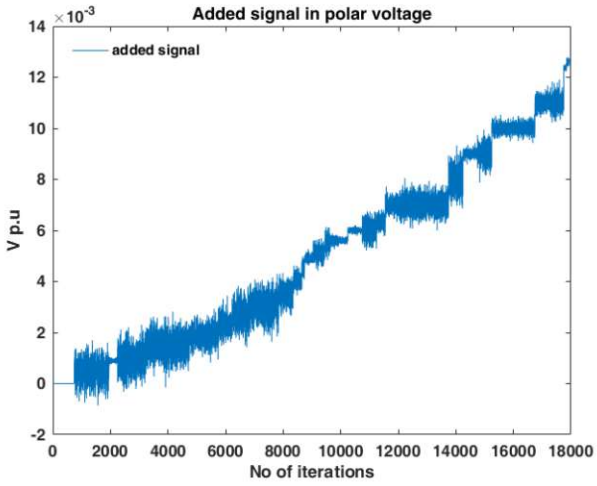


Figure 7.20: Added signal in IROCV attack.

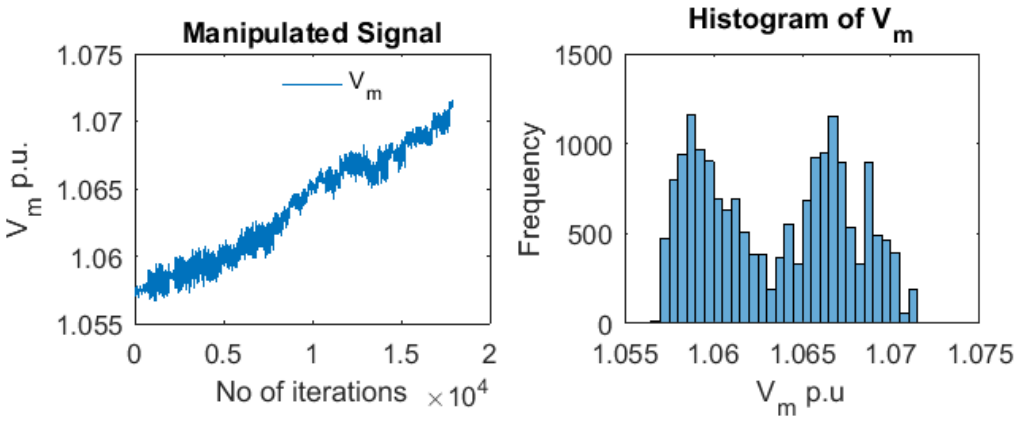
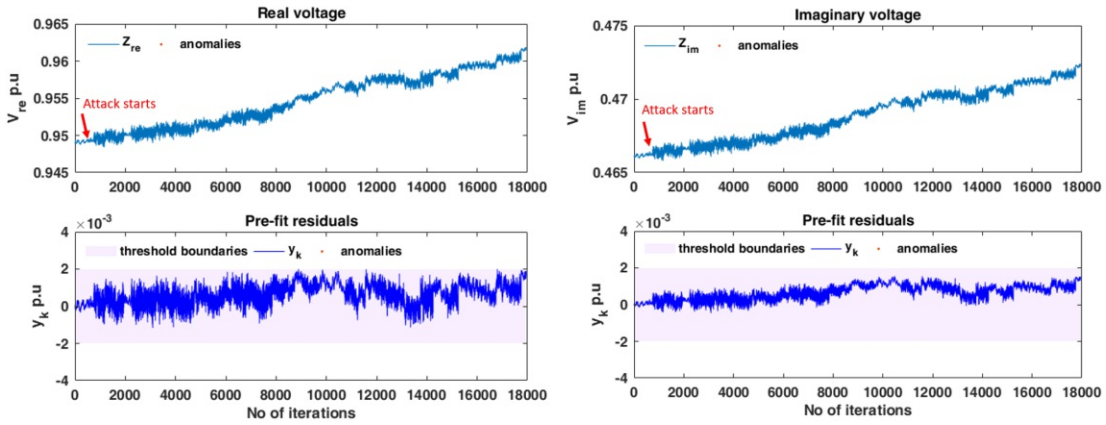


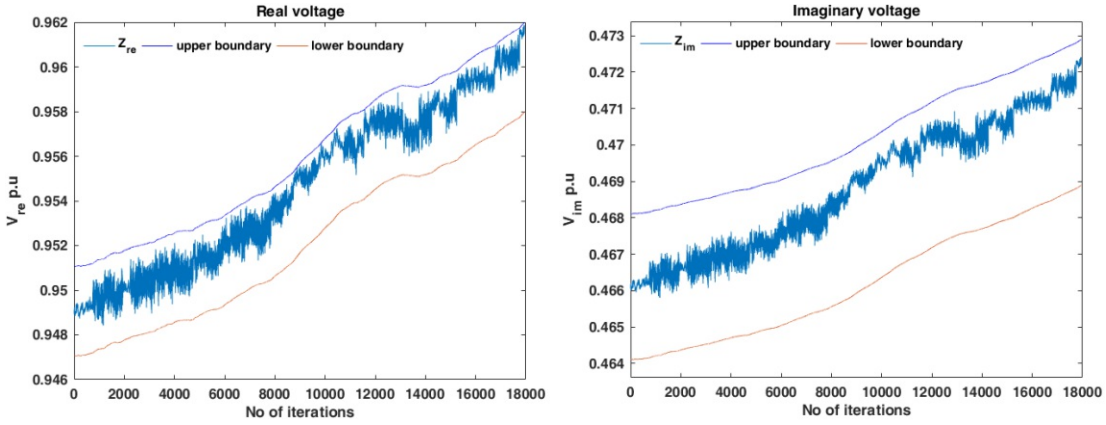
Figure 7.21: Random offset-manipulated signal and its histogram (source Paudel et al. [132]).

7.23, one can see the thresholds of manipulated real and imaginary voltage are adjusted so that the manipulated signals are within the upper and lower boundaries. Figure 7.24 shows how the observed voltage is different from the actual voltage. In this way, the attacker hides in the attack noise and is able to insert offsets. This is possible due the randomness (measurement noise) because the Kalman filter follows the prediction model more precisely. Thus it will be undetected if small enough incremental random noise is added in the signal. The anomaly detection system does not detect this abnormal behavior and the attack remains undetectable.



(a) Real voltage (b) Imaginary voltage

Figure 7.22: Upper part: real voltage and-imaginary voltage. Lower part: pre-fit residuals and threshold boundaries of real voltage and imaginary voltage pre-fit residuals (changes in the dynamic threshold boundaries are very small). No anomalies detected (source Paudel et al. [132]).



(a) Real voltage (b) Imaginary voltage

Figure 7.23: Manipulated signal and dynamic threshold boundaries of real and imaginary voltage (the threshold boundaries are shown in Eq. (7.8) and Eq. (7.9)).

7.3.2.4 Discussion about Undetected Attacks

We presented attacks that were especially crafted to avoid detection and the anomaly detection method in Sec. 7.1.1 does not detect any of the attacks in the real and imaginary voltages, so attacks also remain undetected in the polar form.

The confusion matrix of the undetected attacks (RSCV, ICOS and IROCV) for real, imaginary, and polar voltages is shown in Tab. 7.11. The confusion matrix looks the

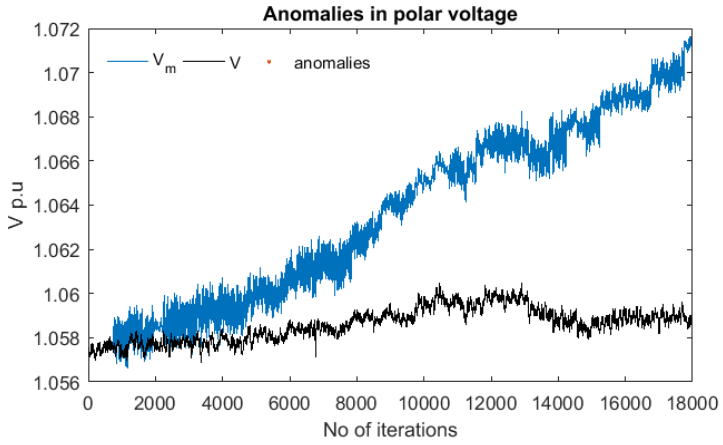


Figure 7.24: Voltage represented in polar coordinates for incremental random offset. No anomalies detected (source Paudel et al. [132]).

same for all attacks, because the attack always starts at the same data point, so there are the same number of malicious anomalies and in all three cases no anomaly has been detected. Thus, here the number of false negatives (FN) equals the total number of malicious measurement values that were injected, and the number of true positives (TP) equals 0 – i.e., no attack behavior was detected. In order to work with the method proposed in [139], we had to fix the phase angle. Therefore, real and imaginary voltages look much different from the original measured real and imaginary voltages.

Table 7.11: Confusion matrix for the undetected attacks (RSCV, ICOS and IROCV) (source Paudel et al. [132]). It is same for all attacks as the attack always starts at the same data point and remains undetected.

Detected as	**Labeled as**	
	Non-malicious	Malicious
Non-malicious	744 (TN)	17,201 (FN)
Malicious	0 (FP)	0 (TP)

We conclude the following findings from the results:

- F 2.1.3: If there are slow changes (slowly increasing offsets) then due to the dynamic threshold that increases when adding the new manipulated values, the dynamic threshold will get larger so that the attack will not be detected. It is because they slowly increase and the changes over a specific time interval remain small but influences the variance of the innovation.
- F 2.1.4: The delta from the original signal can grow large if the attacker slowly increases step by step such that residuals stay below the threshold. We have small attacks but if we increase offset stepwise (e.g., in ICO stepwise) so the anomaly

detection system will not detect the attack. Nevertheless, alarms could be raised if the safety threshold is exceeded.

- F 2.1.5: For the IROCV attack, the pre-fit residuals have a changing variance which indicates abnormal behavior. However, this is not the case in the ICOS. We suggest that the ICOS attack is perhaps the most challenging to detect using pre-fit residuals.

Evidence from these experiments suggest that alternative methods to detect the attacks need to be used. For instance, changes in the histograms can be used to observe characteristics that make the attacks detectable. Furthermore, observing the evolution of pre-fit residual over time could yield insights about a potential attack. For instance, in Fig. 7.12, residuals before starting the attack are much smaller than after starting the attack.

7.3.3 L2-norm and Normalized Pre-fit Residual-Based Detection

In this section the results of experiments 7.2 and 7.3 are discussed where we present detection with L2-norm and normalized residuals. As shown in Tab. 7.4, we investigate the detection performance of L2-norm and normalized residuals for the different attacks (introduced in Tab. 5.1 of Chapter 5): CO, RO, ICO, IRO, IROMN and ICOHS. Then as an example, we visualize anomaly detection results for the first test data (April 01, 02:00-03:00 of UTC). Figure 7.25 visualizes voltage signal from the first data set. The actual voltage signal (polar voltage) has a voltage fluctuation which cause voltage drops between data points 7,093 and 7,815, and between data points 150,735 and 171,043. We label these high voltage drops as benign anomalies.

Here also we continue with a fixed phase angle. As described in Sec. 6.2, we set the phase angle to a constant value of the first observed phase angle (-0.3443 radian) before converting from polar to rectangular coordinates. Figure 7.26 depicts the actual voltage, estimated voltage, pre-fit residuals, L2-norm and normalized residuals in normal operation of the voltage signal in Fig. 7.25. From the figure 7.26, we can see that the L2-norm shows some steps but normalized shows a big peak at the benign anomalies.

The six types of attacks described in Tab. 5.1 of Sec. 5.3 are simulated on the voltage signal visualized in the Fig. 7.25. All attacks start at $60,001^{st}$ data point and continue until the end of the hour. None of the attacks are detected by the L2-norm residuals method with the pre-defined threshold 1.41 derived from the training data.

As L2-norm detects neither the attacks (MAs) nor BAs with this threshold, as the L2-norm continues increasing the attacks could be detected if they continue further. Here we present the detection performance metrics, distinguished detection rates of benign and malicious anomalies, and the detection delay of the different attacks only for the normalized residual-based method.

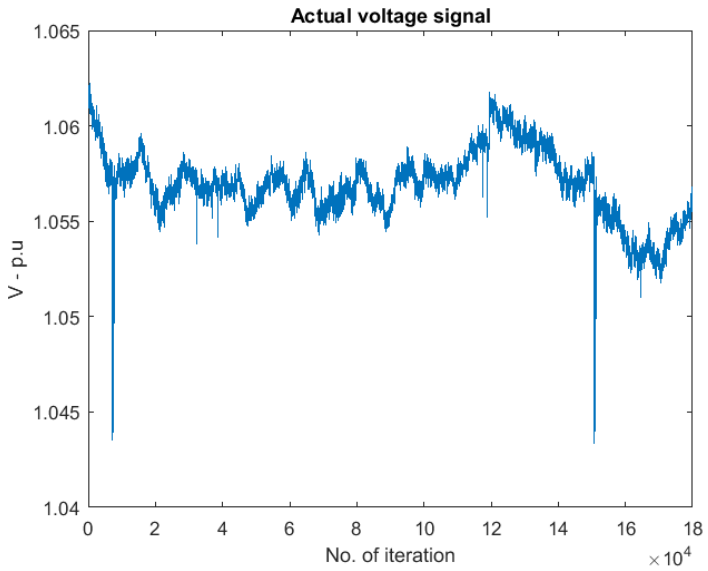
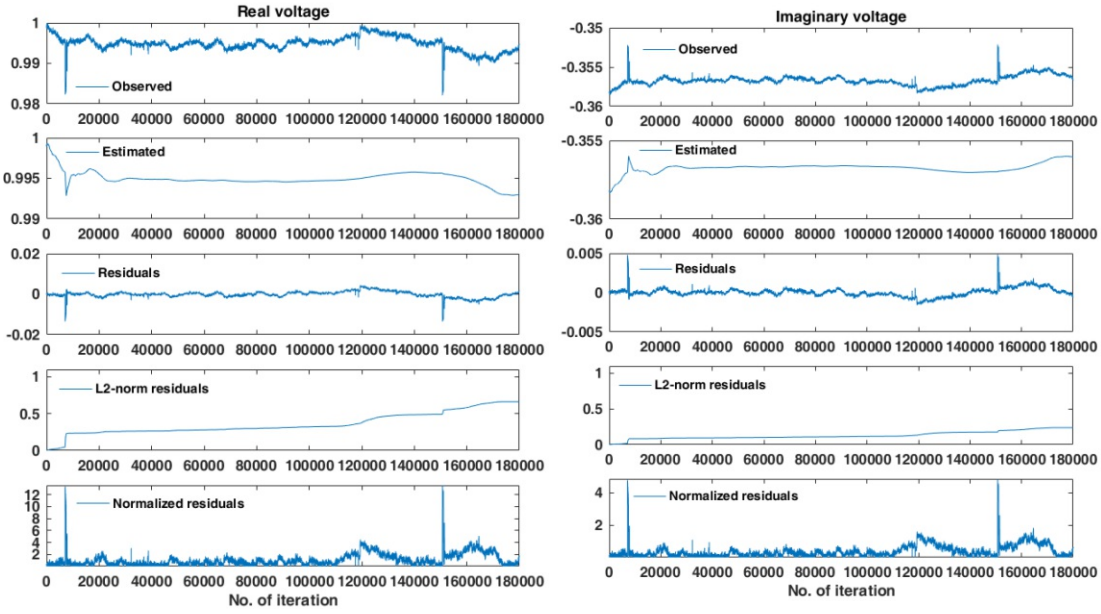


Figure 7.25: Actual voltage signal (polar voltage) for an hour (April 01, 02:00-03:00 of UTC).



(a) Real voltage

(b) Imaginary voltage

Figure 7.26: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in normal operation (y-axis of observed signal, estimated signal and plain residuals are in p.u.).

We present detection based on single data points. For the calculation of the performance metrics we use the original labels, which are normal or anomalous (BA or MA), and compare it with the labels provided by the detection methods. From this we derive: a) true positives (TP), i.e. BA or MA correctly identified as an anomaly, b) true negatives (TN), i.e. how many normal data points are correctly classified as normal, c) false positives (FP), i.e. how many normal data points are classified as anomalies and d) false negatives (FN), i.e. how many anomalies we miss (anomalies classified as normal), From the TP, TN, FP, FN we then calculate accuracy, recall, false positive rate (FPR) and precision. The minimum, maximum and average detection performance metrics for data points-based approach of the 14 test data sets are shown in Tab. 7.12.

Table 7.12: Anomaly detection performance of normalized residual-based method (BAs and MAs not separated). The values shown are the minimum, maximum and average anomaly detection performance metrics of the 14 test data sets. Min/max for FPR is always (0/0)%, and min/max for precision is always (100/100)% for all attack types (source Paudel et al. [133]).

Method	**Attack**	**Accuracy average (min/max)**	**Recall average (min/max)**	**FPR average**	**Precision average**
Normalized residuals (polar voltage)	CO	94.54% (33.30/100)%	91.81% (0/100)%	0%	100%
	RO	48.06% (33.14/98.86)%	22.18% (0/98.29)%	0%	100%
	ICO	44.96% (33.27/98.86)%	17.53% (0/98.29)%	0%	100%
	IRO	47.43% (33.27/98.86)%	21.23% (0/98.29)%	0%	100%
	IROMN	52.15% (33.14/99.20)%	28.31% (0/98.80)%	0%	100%
	ICOHS	48.01% (33.27/98.86)%	22.10% (0/98.29)%	0%	100%

Table 7.13 shows the different detected attack types for the residual-based methods. In the Tab. 7.13, detected attacks means that at least one malicious data point was detected as an anomaly. The table depicts the number of test datasets in which the attacks are detected, and the average detected data points on 14 test data sets. Malicious and benign data points on the 14 test data sets are shown in Tab. 6.3 of Chapter 6. Detected data points on each data sets using normalized residuals are shown in Tab. A.1 of Appendix A.10. One can see that no anomalies are detected with L2-norm and only some with the normalized residuals.

Malicious data points are detected only for the constant offset attack. In attack type RO, IROMN the detected anomalies are all from benign anomalies, even some of the benign

Table 7.13: Detected attacks (at least one malicious data point was detected as an anomaly) out of the 14 injected attacks using L2-norm and normalized residual-based methods (rounded average values are shown for the detected data points).

Methods	**Attacks (average detected data points)**					
	CO	RO	ICO	IRO	IROMN	ICOHS
Normalized residuals	13(2,453)	5(65)	7(565)	8(544)	6(56)	8(5,487)
L2-norm residuals	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)

anomalies are missed once the attack starts, and in attack type ICO, ICOHS, IRO only the benign anomalies are detected. From the variation of recall, one can see that in some data sets no anomalies (neither BAs nor MAs) are detected. CO attack is not detected in 1 test data set, RO attack is not detected in 9 test data sets, ICO attack is not detected in 7 test data sets, IRO attack is not detected in 6 test data sets, IROMN attack is not detected in 8 test data sets and ICOHS attack is not detected in 6 test data sets.

For the normalized residual-based case we showed the detection results for the voltage expressed in polar coordinates. Since we use a linear Kalman filter the actual detection is based on the detection of anomalies in the imaginary and real part of the voltage (see [132] for details). We then just combine all anomalies detected in either the real or in the imaginary voltage.

In order to analyze the detection performance for the manipulated data points, we check how many of the detected anomalies belong to benign (BAs) or malicious anomalies (MAs). If one data point was BA and MA, it was labeled only as BA; we have only 7,727 BAs and 1673,087 MAs in the manipulated test data sets (see Tab. 6.3 in Sec. 6.5). From the analysis using normalized residuals, we find that most of the detected anomalies are BAs. Since we use the same data as a basis for injecting different attacks, the BAs are the same in the basis data sets and therefore also remain BAs when the four attacks are inserted. For attack types RO, ICO, ICOHS, IRO and IROMN the detection is only due to benign anomalies. For attack type CO, malicious data points are detected but some benign anomalies were missed. Therefore, we can say that the spikes caused by BAs are detected with the normalized residuals, but the attacks (MAs) often remain undetected. Table 7.14 shows detection rates of benign anomalies and malicious anomalies in polar voltage.

Table 7.15 in addition shows detection delay of the normalized residual-based method. It considers detection delay of the anomalous data points. Minimum and maximum detection delay among 14 test data sets are in first and second columns respectively. Average detection delay of the 14 test data sets is in the third column. Detection delay varies from 1^{st} anomalous data point to $17,004^{th}$ data point in constant offset attack. In attack type RO, ICO, ICOHS and IRO, minimum detection delay is the same. This is caused by a BA on day 12 that is located at data point 61,957 and therefore falls in the

Table 7.14: Average detection rates of benign and malicious anomalies by normalized residual-based method. TPRB = TPR benign, TPRM = TPR malicious (source Paudel et al. [133]).

Method	Attack	TPRB	TPRM
Normalized residuals (polar voltage)	CO	58.40%	1.89%
	RO	39.67%	7.30×10^{-6}%
	ICO	56.93%	0.36%
	IRO	57.06%	0.34%
	IROMN	37.86%	1.37×10^{-5}%
	ICOHS	51.32%	4.48%

Table 7.15: Minimum, maximum and average anomaly detection delay of normalized residual-based method (source Paudel et al. [133]).

Method	Attack	Detection Delay		
		min	max	average
Normalized residuals (polar voltage)	CO	1	17,004	9,786.86
	RO	2,056	17,858	93,399.14
	ICO	2,056	10,7675	98,972.14
	IRO	2,056	10,7675	94,536.79
	IROMN	1,440	70,670	86,047.00
	ICOHS	2,056	10,7675	93,487.36

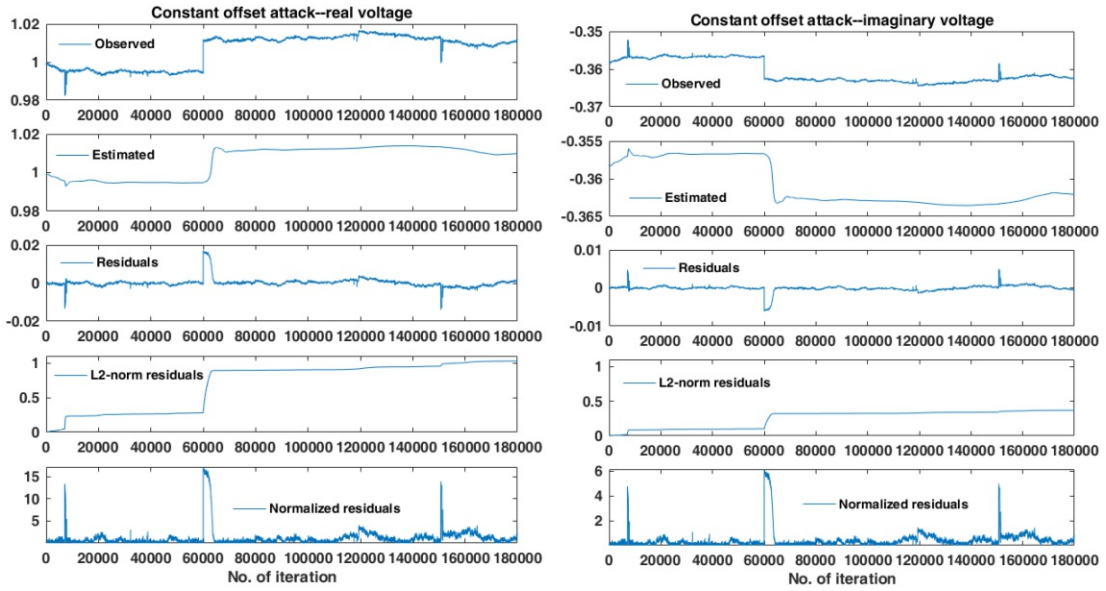
period of the attack exactly at 2056 data points after the attack start. It is clear from the table where BAs are located and the anomalous data points are detected. In attack type IROMN detection delay varies from $1,440^{th}$ anomalous data point to $70,670^{th}$ data point. From the table, we also can deduce that all attacks, except the CO attack, are detected only because of BAs that exceed the threshold and not because of the attack itself. It may be that the BA itself is large enough to trigger the detection or that it exceeds the threshold only because the attack is added to an already high value.

In the following we describe details about the detection of the different attacks.

7.3.3.1 Detailed Detection Results per Attack

Constant Offset Attack The normalized residual-based method detects only the constant offset attack only in real voltage. Figure 7.27 depicts manipulated and estimated real voltage and imaginary voltage signals, their residuals, L2-norm residuals and normalized residuals during constant offset attack. In the sub-figure 7.27a, we can see constant offset attack has high impact in real voltage such that there is significant change (near 6) in real voltage. A jump in residuals between data points 150,735 and 171,043 is due to BAs. The attack causes significant change in residuals once it starts. This change

causes L2-norm residuals increases till end whereas normalized residuals have significant change at attack start only. BAs cause high jump in normalized residuals whereas it has less impact in L2-norm. Thus normalized residual-based BDD method detects BAs.



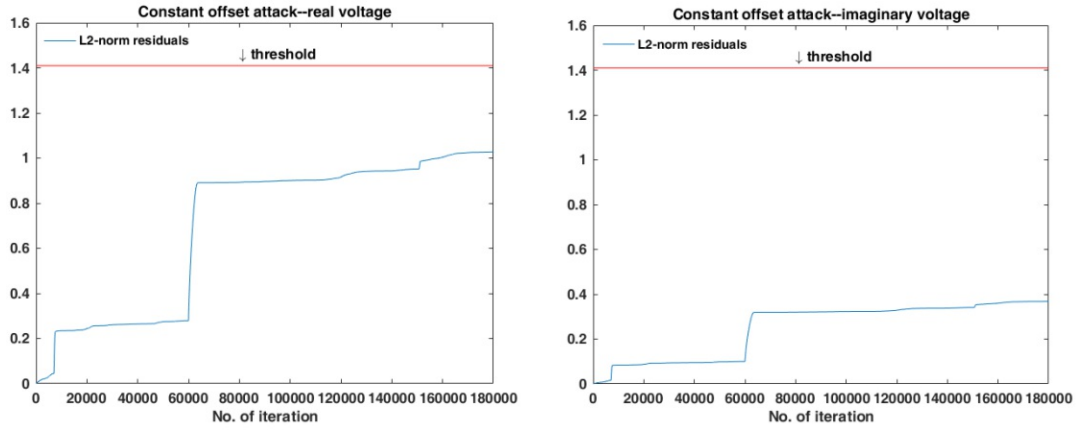
(a) Real voltage (b) Imaginary voltage

Figure 7.27: Observed voltage, estimated voltage, residuals, L2-norm and normalized residuals in constant offset attack (y-axis of observed signal, estimated signal and plain residuals are in p.u.) (source Paudel et al. [133]).

Similarly, sub-figure 7.27b shows manipulated and estimated imaginary voltage signals, their residuals, L2-norm residuals and normalized residuals. Residual changes significantly in imaginary voltage. As in the real voltage signal, first jump (near 6) is due to the attack and a second jump (between data points 150,735 and 171,043) in imaginary voltage is due to the BAs.

Constant offset attack is not detected by L2-norm but detected by normalized residual-based method (see Tab. 7.13). Figure 7.28 shows L2-norm residuals for CO attack. From this figure, we can clearly see a jump in the L2-norm residuals but the attack is not detected. The CO attack could be detected in real voltage if it continues further but the pre-defined threshold is very high for detecting the CO attack in imaginary voltage.

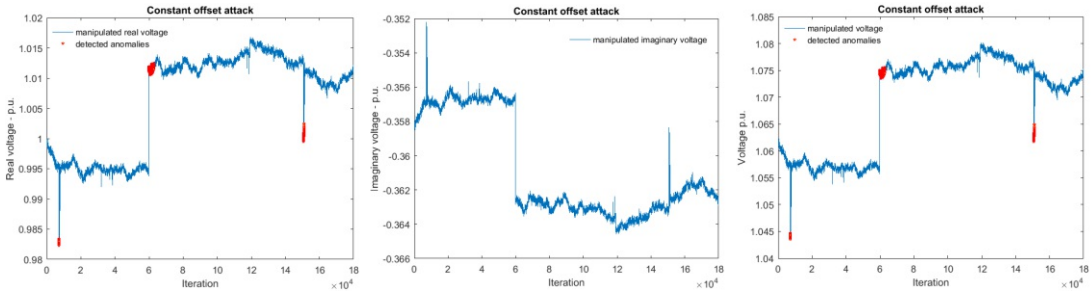
The normalized residual-based method detects the BAs and also some MAs when the attack starts and the voltage changes abruptly in the attack and therefore in the given example detects some anomalous data points. After the detection of some anomalies the algorithm adapts the prediction and therefore it does not detect the subsequent malicious data points as such. On average, it detects only 2,453 data points on all 14 test data sets (see Tab. 7.13).



(a) Real voltage (b) Imaginary voltage

Figure 7.28: L2-norm residuals in constant offset attack (the change is visible but the selected threshold is too high).

The results for the real, imaginary and polar voltage for the first test data set are shown in Fig. 7.29. It depicts that anomaly is missed in imaginary voltage.

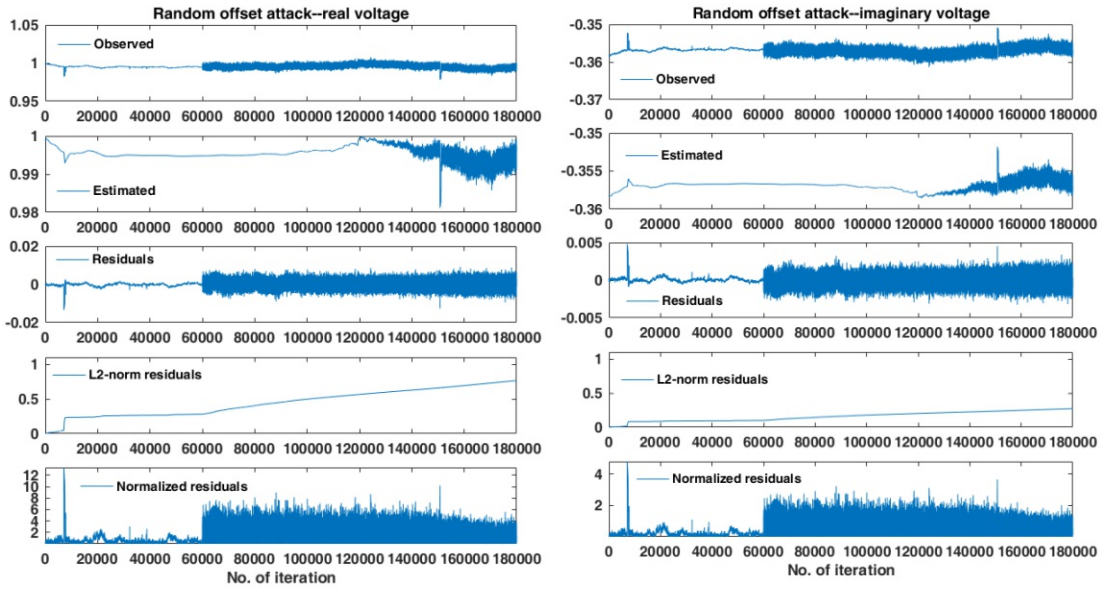


(a) Real voltage (b) Imaginary voltage (c) Polar voltage

Figure 7.29: Visualization of detected anomalies in constant offset attack using normalized residuals (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).

Random Offset Attack Figure 7.30 depicts manipulated and estimated voltage signals, L2-norm residuals and normalized residuals in real voltage and imaginary voltage during random offset attack. Sub-figure 7.30a shows effect of the attack in real voltage, residuals, L2-norm and normalized residuals in real voltage. Similarly, sub-figure 7.30b shows about imaginary voltage.

L2-norm residuals does not detect random offset attack but normalized residuals trigger some alarms only due to BAs (see Tab. 7.13). Normalized residual-based method does not detect malicious points, it detects only the benign anomalies (which cause high jumps in



(a) Real voltage

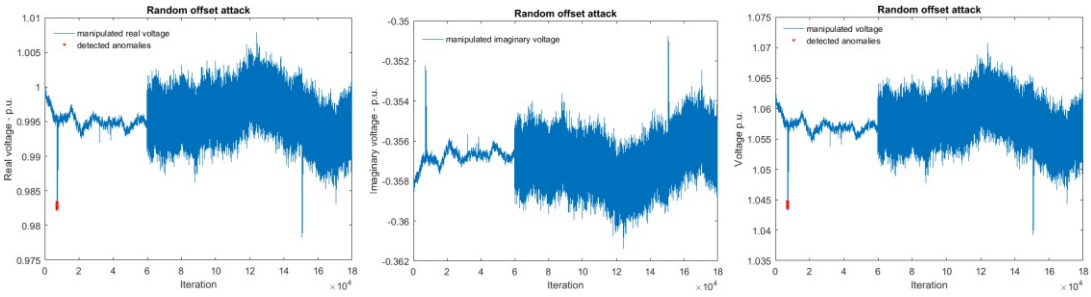
(b) Imaginary voltage

Figure 7.30: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in random offset attack (y-axis of observed signal, estimated signal and plain residuals are in p.u.).

the signal). In some data there are some fluctuations in the signal which triggers alarms after the starting point of attacks. In these cases an operator does not know whether the alarm is due to benign or malicious anomalies. It can be seen in Fig. 7.31 that the attack is not identified and some of the benign anomalies are identified as anomalies. It detects only the first benign anomalies (between data points 7,093 and 7,815) and then when noise is added from attack it does not detect the benign anomalies (between data points 150,735 and 171,043). The normalized residual-based method fails to detect several benign anomalies, therefore the overall performance with 14 test data sets is rather low (i.e only 65 data points on average see Tab. 7.13).

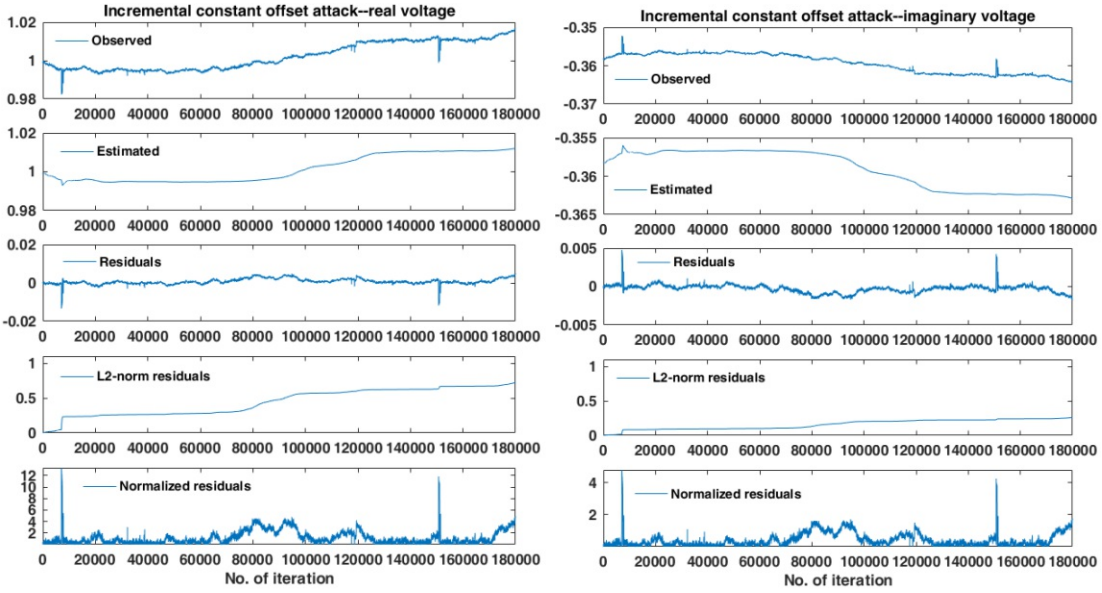
Incremental Constant Offset Attack Figure 7.32 depicts manipulated and estimated voltage signals, L2-norm residuals and normalized residuals in real voltage and imaginary voltage during incremental constant offset attack. Sub-figure 7.32a shows effect of the attack in real voltage, residuals, L2-norm and normalized residuals in real voltage. Similarly, sub-figure 7.32b shows about imaginary voltage.

L2-norm does not identify anomalous points but normalized residuals identify only some benign anomalies during incremental constant offset attack (see Tab. 7.13). Normalized residual-based approach detects attacks of type ICO in 7 data sets and on average only 565 data points from the 14 test data sets. The detection is only due to the presence of



(a) Real voltage (b) Imaginary voltage (c) Polar voltage

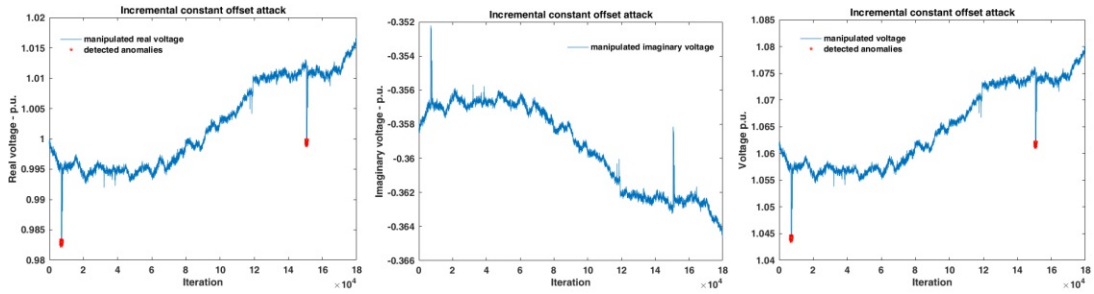
Figure 7.31: Visualization of detected anomalies in random offset attack using normalized residual based method (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).



(a) Real voltage (b) Imaginary voltage

Figure 7.32: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in incremental constant offset attack (y-axis of observed signal, estimated signal and plain residuals are in p.u.).

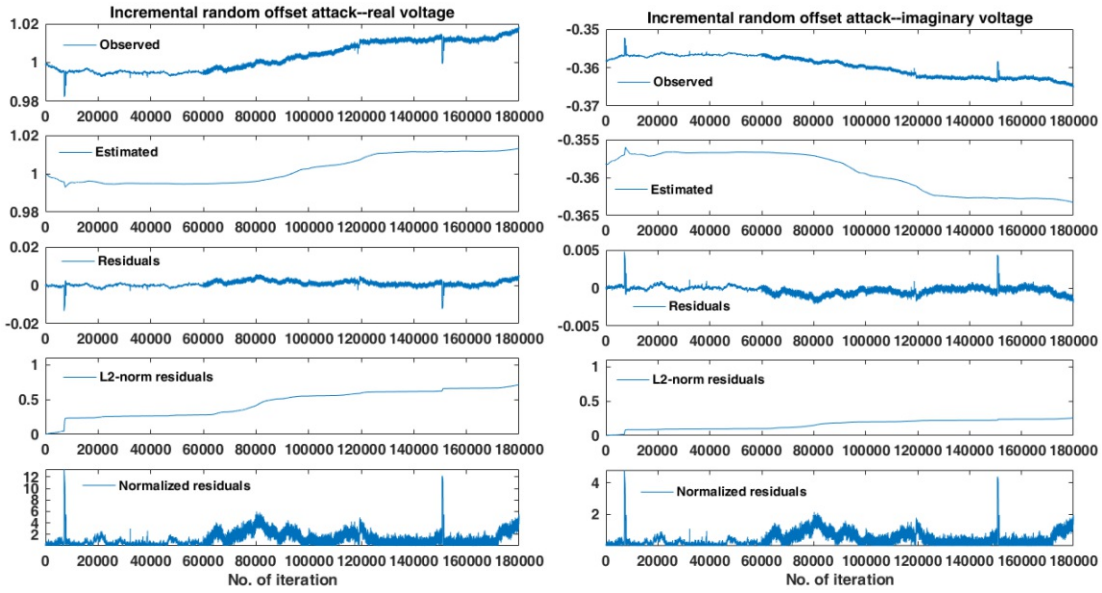
the benign anomalies before or after the attack starting points. It can be clearly seen in Fig. 7.33 that only the high fluctuation in the signal is identified as an anomaly.



(a) Real voltage (b) Imaginary voltage (c) Polar voltage

Figure 7.33: Visualization of detected anomalies in incremental constant offset attack using normalized residual-based (shown on April 01, 02:00-03:00).

Incremental Random Offset Attack Figure 7.34 depicts manipulated and estimated voltage signals, L2-norm residuals and normalized residuals in real voltage and imaginary voltage during incremental random offset attack. Sub-figure 7.34a shows effect of the attack in real voltage, residuals, L2-norm and normalized residuals in real voltage. Similarly, sub-figure 7.34b shows about imaginary voltage.

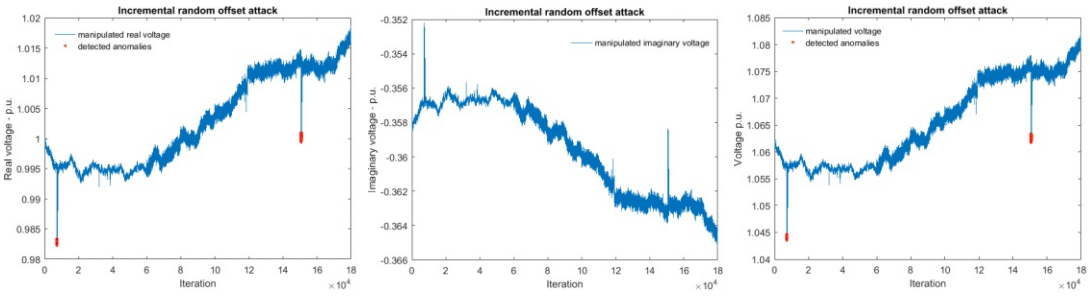


(a) Real voltage (b) Imaginary voltage

Figure 7.34: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in incremental random offset attack (y-axis of observed signal, estimated signal and plain residuals are in p.u.).

L2-norm does not identify anomalous points but normalized residuals identify only some

anomalies during incremental random offset attack (as shown in Tab. 7.13). Normalized residual-based also does not detect attacks of type IRO in all of the 14 data sets. Similar to other cases, the detection is due to the benign anomalies before or after the attack starting points. On average, it detects only 544 data points on all 14 test data sets (see Tab. 7.13). Figure 7.35 clearly depicts the attack is not identified and only the high fluctuation in voltage signal is identified as anomalous.

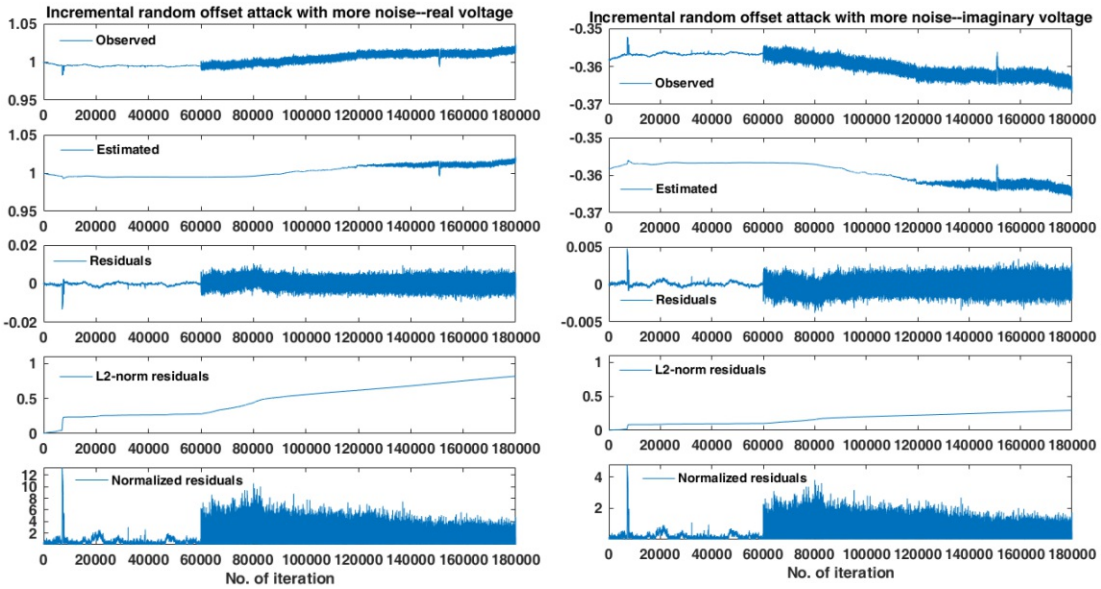


(a) Real voltage (b) Imaginary voltage (c) Polar voltage

Figure 7.35: Visualization of detected anomalies in incremental random offset attack using normalized residual-based (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).

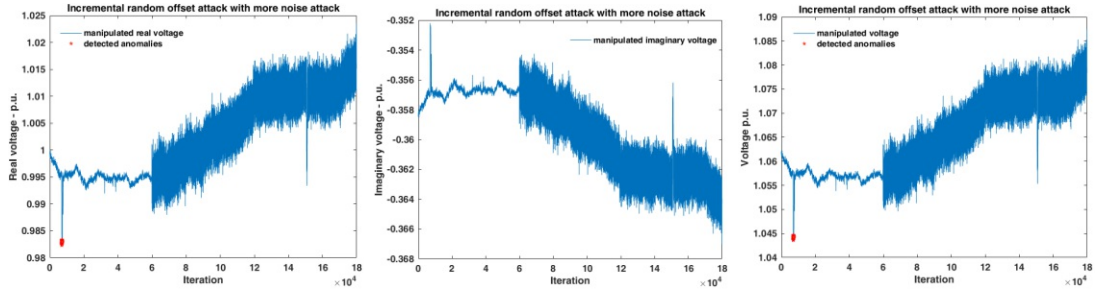
Incremental Random Offset Attack with More Noise Here we show the effect of adding more noise in IRO attack. Figure 7.36 depicts manipulated and estimated voltage signals, L2-norm residuals and normalized residuals in real voltage and imaginary voltage during incremental random offset attack with more noise. Sub-figure 7.36a shows effect of the attack in real voltage, residuals, L2-norm and normalized residuals in real voltage. Similarly, sub-figure 7.36b shows about imaginary voltage.

L2-norm does not identify anomalous points but normalized residuals identify only benign anomalies during incremental random offset attack with more noise (as shown in Tab. 7.13). Normalized residual-based method also does not detect attack type IROMN in all of the 14 data sets. Similar to other cases, the detection is due to the benign anomalies before or after the attack starting points. Figure 7.37 clearly depicts the attack is not identified and only the high fluctuation before starting the attack in voltage signal is identified as anomalous, the fluctuation after starting the attack is not detected. On average, it detects only 56 data points (as shown in Tab. 7.13). In comparison to Fig.7.35, one can see addition of more noise reduces performance of detection.



(a) Real voltage (b) Imaginary voltage

Figure 7.36: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in incremental random offset attack with more noise (y-axis of observed signal, estimated signal and plain residuals are in p.u.).

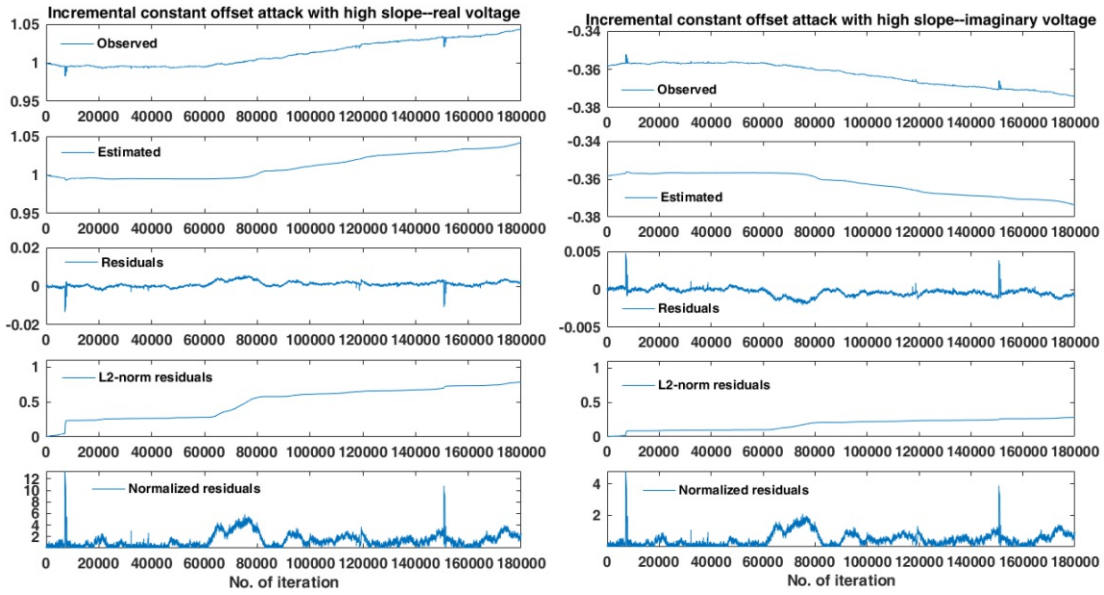


(a) Real voltage (b) Imaginary voltage (c) Polar voltage

Figure 7.37: Visualization of detected anomalies in incremental random offset attack with more noise using normalized residual-based (shown on April 01, 02:00-03:00).

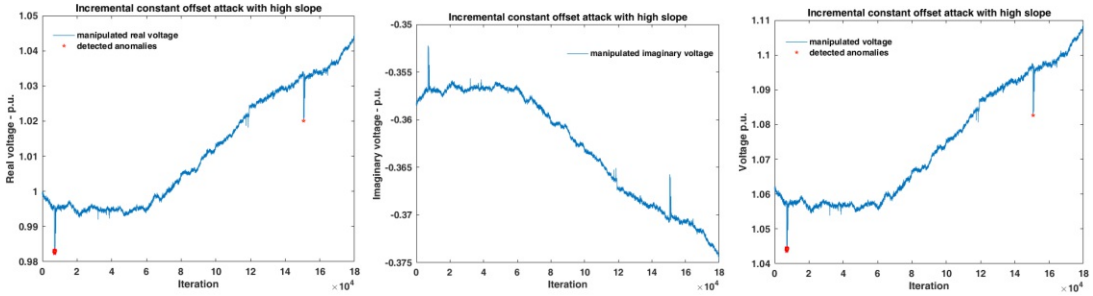
Incremental Constant Offset Attack with High Slope Here we show the effect of increasing slope in ICO attack. Figure 7.38 depicts manipulated and estimated voltage signals, L2-norm residuals and normalized residuals in real voltage and imaginary voltage during incremental constant offset attack with high slope. Sub-figure 7.38a shows effect of the attack in real voltage, residuals, L2-norm and normalized residuals in real voltage. Similarly, sub-figure 7.38b shows about imaginary voltage.

L2-norm does not identify anomalous points but normalized residuals identify only benign



(a) Real voltage (b) Imaginary voltage

Figure 7.38: Observed voltage, estimated voltage, residuals, L2-norm residuals and normalized residuals in incremental constant offset attack with high slope (y-axis of observed signal, estimated signal and plain residuals are in p.u.).



(a) Real voltage (b) Imaginary voltage (c) Polar voltage

Figure 7.39: Visualization of detected anomalies in incremental constant offset attack with high slope using normalized residual-based (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).

anomalies during incremental constant offset attack with high slope (as shown in Tab. 7.13). Normalized residual-based method does not detect attacks of type ICOHS in all of the 14 data sets. The detection is only due to the presence of the benign anomalies before or after the attack starting points. It can be clearly seen in Fig. 7.39 that only the high fluctuation in the signal is identified as an anomaly. On average, it detects only 5,487 data points (as shown in Tab. 7.13). In comparison to Fig. 7.33, one can see

increasing the slope also does not trigger alarms.

7.3.3.2 Results Findings for L2-Norm and normamlized residuals

Plain pre-fit uses a different threshold but this can be used by the attacker to adjust the threshold. L2-norm residuals thresholds are based on training data and cannot be adjusted by the attacker. Similarly, normalized residuals threshold is also based on training data. We can conclude the following findings from the results:

- F 2.1.6: L2-norm residual-based method detects none of the attacks and also no benign anomalies. This might be due to the high threshold as we set a safety margin while defining the threshold but the attacks could be detected if they continue further.
- F 2.1.7: Normalized residual-based method detects only one of the attacks (attack type CO). The normalized residual-based method does not work for all data points but detects at least 1 data point in many manipulated test data sets. The method only detects BAs. Either the BA was already large enough to trigger an alarm or the data points exceeded the threshold after the attack was added. In this case, the attack only became visible if added to an already untypically high (or low) data value.
- F 2.1.8: The normalized residual-based method detects changes quickly but also adapts to the changes quickly so after a short time the residuals get smaller and then remain within the threshold. So after some anomalies occurred, it does not detect subsequent anomalies for long.
- F 2.1.9: With the normalized residual-based method attackers may be able to train the detection system to adapt to the changes similar to the plain pre-fit residuals. Attack types RO, ICO, ICOHS, IRO and IROMN are not detected by the normalized residual-based method as the detection is due to the BAs in actual data.

7.4 Summary

In this chapter, we presented residual-based methods for detecting bad data.

The advantage of using residual-based detection methods is that residuals are available as a side product from SE. We made use of BDD methods based on plain pre-fit residuals, L2-norm and normalized residuals.

We described the relation of anomaly detection with varying phase angle, with fixed phase angle and how we concluded using fixed phase angle with residual-based methods for

our experiment. We showed the setup for the experiment with the plain pre-fit residuals, described the parameters for setting the thresholds for L2-norm and normalized residuals.

Finally, we provided the anomaly detection results using the residual-based BDD methods (plain pre-fit residuals, L2-norm and normalized residuals). The following major results findings from the plain pre-fit residual-based method supported answering our research questions **RQ 2.1.1** (Can the plain pre-fit residual-based method proposed in [139] detect the injected attacks in our data set?) and **RQ 2.1.2** (Can attackers avoid being detected if plain pre-fit residuals are used for detection?).

- The plain pre-fit residual-based method did not detect any of the injected attacks on our real data. The method detected our attacks with similar attack parameters as used in [139] only when we fixed the phase angle in our real data.
- The plain pre-fit residual-based method detected only the SD type attack among the attacks that have been introduced in Tab. 5.2 of Chapter 5 as the SD attack has a quick high increase in offsets whereas the undetected attacks have slowly increasing offsets.

As we already expected for our reasoning **RQ 2.1.1**, the evidence from experiments showed that we can detect the injected attacks in our data set using the plain pre-fit residuals but only if the phase angle is fixed. Similarly, as we already expected for our reasoning **RQ 2.1.2**, the evidence from the experiments showed that attackers can avoid plain pre-fit residual-based detection because the plain pre-fit residual-based method did not detect the attacks that have slowly increasing offsets.

The following major results findings from the L2-norm residual-based method supported answering our research questions **RQ 2.1.3** (Can the L2-norm residual-based method using LWLS proposed in [100], which is based on LWLS SE, detect our injected attacks in our data set also if we use residuals from DKF?).

- The L2-norm residual-based method did not detect any of the attacks that have been introduced in Tab. 5.1 of Chapter 5.
- There was a significant change in the L2 norm, when the attacks started, but the threshold was not exceeded. The results showed the attacks were too small to be detected by the L2-norm method using the defined threshold based on the training data.

For our reasoning **RQ 2.1.3**, we expected that we can detect the injected attacks in our data set using the L2-norm residuals from DKF. But the evidence from experiments showed that the L2-norm residual-based method did not detect any of the attacks.

The following major findings from the normalized residual-based method supported answering our research questions **RQ 2.1.4** (Can the normalized residual-based method proposed in [14] using DKF detect our injected attacks in our data set?).

- The normalized residual-based method detected only one of the attack (attack type CO) among the attacks that have been introduced in Tab. 5.1 of Chapter 5. The normalized residual-based method did not work for all data points but detected at least 1 data point in many manipulated test data sets.
- The normalized residual-based method detected changes quickly but also adapted to the changes quickly so that residuals became small quickly and remained within threshold. So after some anomalies occurred, it did not detect subsequent anomalies for long.
- The residuals were calculated from measurements, so they would adapt if we increase the offsets higher and higher slowly, but if a quick high increase is made in the offsets, then it would result in high residuals so that the attack would be detected.

We expected for our reasoning **RQ 2.1.4** that we can detect the injected attacks in our data set using the normalized residuals from DKF. But the evidence from experiments showed that the normalized residual-based method detected only some of the attacks.

CHAPTER 8

Stealthy Attacks

Notice of adoption from previous publications in Chapter 8
Parts of the contents of this chapter have been published in the following papers:

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
The stealthy attacks described in this chapter is based on the work done in [132] and [133]. A part of state estimation under attack scenario described in this chapter is based on the work done in [132].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, first we present necessary conditions for a stealthy attack on state estimation and the effect of the false data injection attacks on the state estimation process. In a second step, we present the experimental setup of the stealthy attacks in Sec. 8.2, then show results in Sec. 8.3

For state estimation (SE) with linear weighted least squares (LWLS) or Kalman filters (KFs), which are often used for SE in smart grids, a residual-based approach to bad data detection (BDD) has the advantage that residuals can be readily calculated as a by-product of the SE process. Most of the BDD algorithms make an assumption, when there is a bad measurement then the difference between the observed measurement and their corresponding estimated values becomes significant [40].

The SE process uses sensor (e.g., PMU) data, and bad data detection algorithms use the residuals from SE to detect bad data due to the measurement system's failures. In a setting where attackers can gain access to modify the sensor data, they can exploit the fact that SE is used to process the data and modify the sensor data so that the attacks remain stealthy in the SE process. Further, an attacker can evade bad data detection if the attacker uses knowledge about the smart grid system (e.g., anomaly detection system, the method used for state estimation) to craft the attack. A false data injection (FDI) attack is a stealthy attack if it does not trigger BDD alarms. The alarms are triggered if there is a deviation from the expected physical state.

Let m be the number of meters and $\boldsymbol{z}$ be the vector of m measurements that are sent to the state estimator. An attacker is able to change measurements by physically accessing the meter or accessing the communication channel. We recall that the manipulated measurement is represented by Eq. (8.1) [100].

$$\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a} \tag{8.1}$$

where $\boldsymbol{z}$ is the actual measurement, $\boldsymbol{a}$ is attack vector and $\boldsymbol{z}_a$ is the manipulated measurement vector.

Attackers aim to fool the energy management system (EMS) and a human operator with misinformation, such that a particular measurement is $\boldsymbol{z}_{k,a} = \boldsymbol{z}_k + \boldsymbol{a}_k$ and not $\boldsymbol{z}_k$ for time step k where $\boldsymbol{a}_k$ is an attack vector. Authors in [100] use the L2-norm of the residuals from LWLS SE for BDD. The L2-norm of the residuals in normal operation is $\|\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}\|$ and during an attack is $\|\boldsymbol{z}_a - \boldsymbol{H}\hat{\boldsymbol{x}}_{bad}\|$. An attacker that wants to remain undetected, needs to ensure that the L2-norm of the residuals from LWLS SE using the manipulated measurement is below the threshold as expressed in Eq. (8.2) (see Sec. 7.1.2).

$$\|\boldsymbol{z}_a - \boldsymbol{H}\hat{\boldsymbol{x}}_{bad}\| \leq t \tag{8.2}$$

where $\hat{\boldsymbol{x}}_{bad}$ is the estimated state considering manipulated measurement and t is the pre-defined threshold.

In [100] authors show that an attack of type $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ is not detected by residual-based BDD using the L2-norm if LWLS are used for state estimation. Thus the stealthy attacks are expressed as Eq. (8.3) in [100].

$$\|\boldsymbol{z}_a - \boldsymbol{H}\hat{\boldsymbol{x}}_{bad}\| \leq \|\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}\| + \|(\boldsymbol{a} - \boldsymbol{H}\boldsymbol{c})\| \tag{8.3}$$

where $\hat{\boldsymbol{x}}$ is true state, $\boldsymbol{c}$ is a vector of offsets and $\boldsymbol{a}$ is a vector of attack.

Our research question **RQ 2.2** about stealthy attacks reads:

- **RQ 2.2**: Can stealthy attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100] be detected by residual-based methods?

Further, we divide the research question **RQ 2.2** into the following sub-research questions:

- **RQ 2.2.1**: Can attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against state estimation from [100] remain stealthy if we analyze residuals from state estimation using LWLS?
  Rationale: [100] shows that attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remain stealthy for a residual-based detection methods for LWLS SE. If LWLS is used for SE and only one metric (voltage) is measured, residuals are zero for LWLS SE. Therefore, we assume that in a simple scenario with only voltage measurements, voltage manipulation and LWLS SE, attacks cannot be detected. If multiple metrics (e.g. voltage and current) are measured we assume that (as shown in [100]) an attacker can remain stealthy as long as the attacker can manipulate both metrics to bring them in the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$. That means that also for high values of c the attack cannot be detected. If the attacker can only manipulate one metric, we assume that the attack can be detected, because the second metric can be used to check the plausibility of the values.

- **RQ 2.2.2**: Can attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against state estimation from [100] be detected if we analyze residuals from state estimation using DKF?
  Rationale: In [100] stealthy attacks are shown for LWLS SE. As Kalman filters are widely used for SE in different domains, we also use DKF for SE and here we want to check if attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ also remain stealthy if DKF is used as SE method. Since DKF SE takes past values into account, we expect a different behaviour and rather assume that with DKF the attacks do not remain stealthy.

Table 8.1 shows the intention of using the stealthy attacks, data used for the experiment and findings. Details on parameter settings for the experiment are presented in Sec. 8.2.

Table 8.1: Overview of stealthy attacks.

Methods	**Data***	**Goals**	**Sections**
Stealthy attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$	Test data (01.04.2016)	- to answer **RQ 2.2.1** - to answer **RQ 2.2.2**	8.1 8.2 8.3

* Test data, one hour at 02:00-03:00 UTC is used.

8.1 Theoretical background

8.1.1 Stealthy attack on voltage measurements

We aim to check the conditions of stealthy attack of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ from [100] using only voltage measurements for state estimation. We recall the expression of the measurement $\boldsymbol{z}$ and true state $\boldsymbol{x}$ as

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \tag{8.4}$$

where $\boldsymbol{v}$ is the measurement noise, $\boldsymbol{H}$ equals to an identity matrix, measurement $\boldsymbol{z}$ is represented by Eq. (8.5) and true state $\boldsymbol{x}$ is represented by Eq. (8.6).

$$\boldsymbol{z} = \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} \tag{8.5}$$

$$\boldsymbol{x} = \begin{bmatrix} V_{re}^{t} \\ V_{im}^{t} \end{bmatrix} \tag{8.6}$$

Under the condition defined by Liu et al. in [100], an attack is undetected if $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$. The condition is expressed in Eq. (8.7), and the manipulated measurement under the condition is represented in Eq. (8.8).

$$\boldsymbol{H}\boldsymbol{c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{8.7}$$

$$\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a} = \begin{bmatrix} V_{re} \\ V_{im} \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{8.8}$$

The resulting manipulated measurements in rectangular coordinates are represented by Eq. (8.9) and Eq. (8.10).

$$z_1 = V_{re} + c_1 \tag{8.9}$$

$$z_2 = V_{im} + c_2 \tag{8.10}$$

8.1.2 Stealthy attack on voltage and current measurements

We aim to check the conditions of the stealthy attack of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ from [100] using both voltage and current measurements for state estimation. In this case, the measurement $\boldsymbol{z}$ is represented by Eq. (8.11). We recall that the true state $\boldsymbol{x}$ is represented by Eq. (8.6) (see Sec. 8.1.1) and the matrix $\boldsymbol{H}$ is represented by Eq. (8.12).

$$\boldsymbol{z} = \begin{bmatrix} V_{re} \\ V_{im} \\ I_{re} \\ I_{im} \end{bmatrix} \tag{8.11}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ G & -B \\ B & G \end{bmatrix} \tag{8.12}$$

The condition defined by Liu et al. in [100] to be a stealthy attack on voltage and current measurements is expressed in Eq. (8.13).

$$\boldsymbol{Hc} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ G & -B \\ B & G \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ Gc_1 - Bc_2 \\ Bc_1 + Gc_2 \end{bmatrix} \tag{8.13}$$

So if we add a constant to V_{re} and V_{im} and then add values to the current measurements that depend on c_1, c_2, G and B then the attack remains undetected by residual-based BDD methods for LWLS.

8.2 Experimental Setup

Based on the stealthy attacks definitions presented in Sec. 8.1, we manipulate measurements (see below) and analyze the residuals from SE using LWLS and DKF.

8.2.1 Manipulate only voltage measurements

We assume a case where an attacker manipulates only the voltage measurements. The attack vectors in the measurement of the current are zero. Thus the attack vector $\boldsymbol{a}$ is represented by Eq. (8.14).

$$\boldsymbol{a} = \begin{bmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{bmatrix} \tag{8.14}$$

Manipulated measurement can be rewritten in Eq. (8.15).

$$\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a} = \begin{bmatrix} V_{re} \\ V_{im} \\ I_{re} \\ I_{im} \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{bmatrix} \tag{8.15}$$

Thus the resulting manipulated measurements are

$$z_1 = V_{re} + c_1 \tag{8.16}$$

$$z_2 = V_{im} + c_2 \tag{8.17}$$

$$z_3 = I_{re} + 0 \tag{8.18}$$

$$z_4 = I_{im} + 0 \tag{8.19}$$

In order to investigate whether the attack remains stealthy with different detection methods, we add a constant offset attack (CO) that stars at $2,000^{th}$ data point and ends at $18,000^{th}$ data point. Figure (8.1) shows the manipulated real voltage and imaginary voltage.

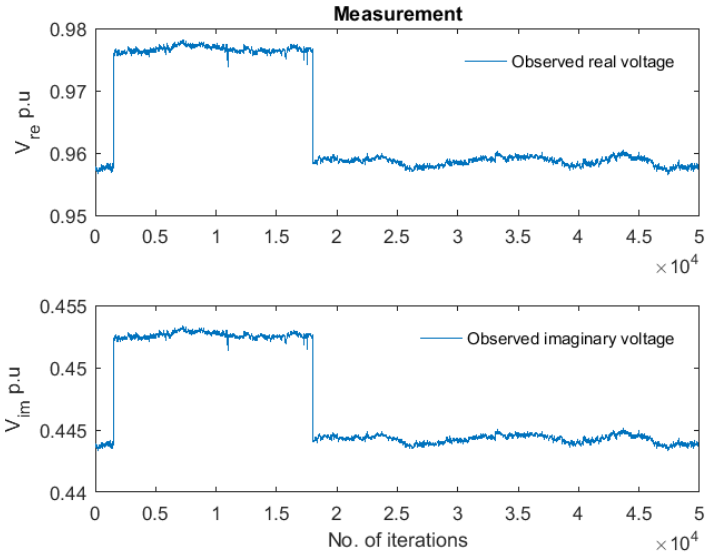


Figure 8.1: Visualization of measured real voltage and imaginary voltage during an attack.

8.2.2 Manipulate both voltage and current measurements

We assume a case where an attacker manipulates both the voltage and current measurements. Thus the attack vector $\boldsymbol{a}$ is represented by Eq. (8.20).

$$\boldsymbol{a} = \begin{bmatrix} c_1 \\ c_2 \\ Gc_1 - Bc_2 \\ Bc_1 - Gc_2 \end{bmatrix} \tag{8.20}$$

Manipulated measurement can be rewritten in Eq. (8.21).

$$\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a} = \begin{bmatrix} V_{re} \\ V_{im} \\ I_{re} \\ I_{im} \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ Gc_1 - Bc_2 \\ Bc_1 - Gc_2 \end{bmatrix} \tag{8.21}$$

Thus the resulting manipulated measurements are

$$z_1 = V_{re} + c_1 \tag{8.22}$$

$$z_2 = V_{im} + c_2 \tag{8.23}$$

$$z_3 = I_{re} + Gc_1 - Bc_2 \tag{8.24}$$

$$z_4 = I_{im} + Bc_1 - Gc_2 \tag{8.25}$$

In [100] the authors show that attacks of this type remain stealthy but only for the LWLS method. We first look at detection from LWLS but then also investigate whether those attacks get visible when we use DKF for the state estimation. We show both method (LWLS and DKF) for two different cases a) only with voltage measurements and b) with voltage and current measurements. For the stealthy attack on LWLS state estimation [100], we add one experiment with a very large offset (experiment 8.2), in order to check if the attack remains stealthy, even if the measurements deviate a lot from the original values.

Table 8.2 shows an overview of stealthy attacks, measured vectors, SE methods, manipulation of the vectors, parameter settings and injected attacks. Here we aim to analyze the residuals while starting and ending the attacks. We recall that the starting and ending data points of the stealthy attacks are different than those in the attacks generated using the attack model. We do this because here we aim to analyze residuals in both starting and ending data points of the attacks; in the attacks using the attack model there are no attack ending data points for the given time interval. Further, the manipulation attack is the CO type attack with different offset magnitude than the magnitude in the attack model as we do not compare the results of the stealthy attack to CO attack.

Table 8.2: Overview of experiment with stealthy attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$, attacks parameters setting and injected attack; SE = State estimation; Exp. = experiment; MV = measured vector; CO = constant offset. It is all for the test data* (01.04. 2016)

SE Method	Exp.	MV	Manipulation	Param. setting	Injected attacks
LWLS	8.1	Voltage	Voltage	start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)
LWLS	8.2	Voltage	Voltage	attack start = 2,000, end = 18,000	CO (offset = 10 p.u.)
LWLS	8.3	Voltage, current	Voltage	attack start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)
LWLS	8.4	Voltage, current	Voltage, current	attack start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)
DKF	8.5	Voltage	Voltage	attack start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)
DKF	8.6	Voltage, current	Voltage	attack start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)
DKF	8.7	Voltage, current	Voltage, current	attack start = 2,000, end = 18,000	CO (offset = 0.0195 p.u.)

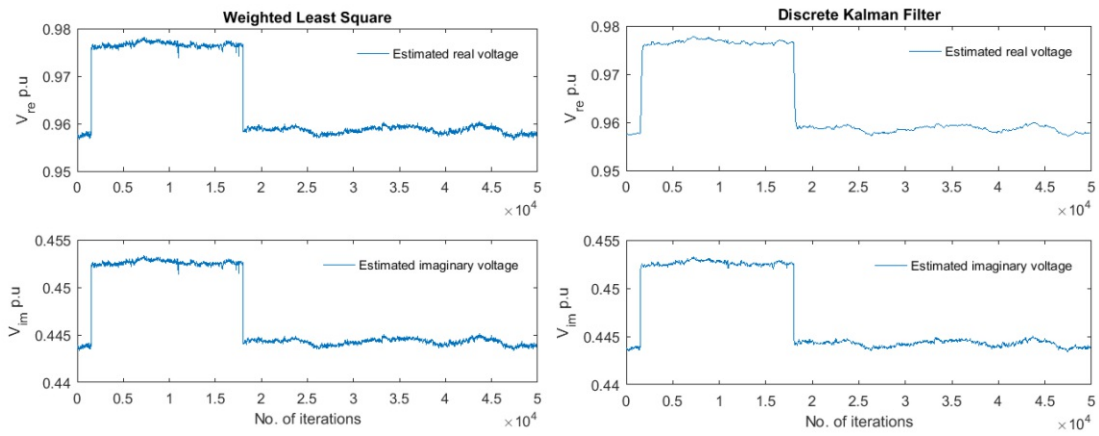
* Test data, one hour at 02:00-03:00 UTC is used.

8.3 Results

In this section, we present attacks, which meet the definition of stealthy attacks on measurements that are of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ and therefore should remain stealthy as shown in Sec. 8.1. The experiment setup is presented in Sec. 8.2. We analyze residuals of SE using LWLS and DKF with fixed phase angle.

8.3.1 State estimation based on voltage measurements

Here we first present SE results under an attack scenario using only voltage measurement. In this section, we are showing experiments 8.1, 8.2 and 8.5 of Tab. 8.2. An attacker adds a constant offset during the attack. In experiments 8.1 and 8.5, we assume the attacker adds an offset 0.0195 at $2,000^{th}$ data point so that the manipulated polar voltage value reaches 1.075 p.u. at the data point, and then add the same offset 0.0195 p.u. to all data values until the attack is over at data point 18,000. Since we fixed the phase angle, the constant offset in polar coordinates also will add a constant offset to real and imaginary parts.



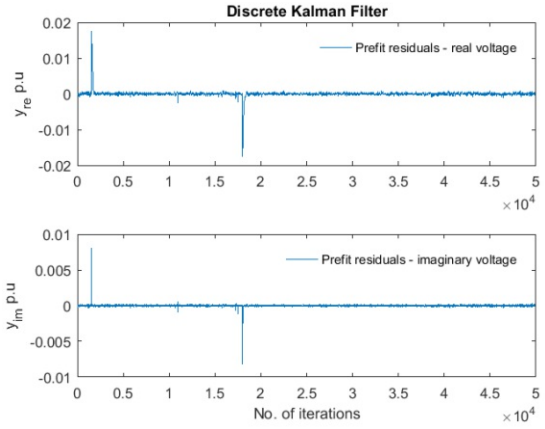
(a) Estimation using LWLS

(b) Estimation using DKF

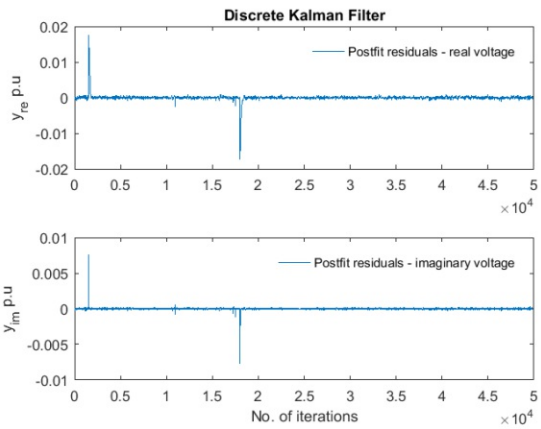
Figure 8.2: Estimated real voltage and imaginary voltage (Exp. 8.1 and 8.5 of Tab. 8.2).

With the estimation of LWLS method based on voltage data only, the estimation is equal to the actual measured value. Therefore the estimated signal from LWLS is exactly the same as the original observed signal and all residuals are zero. Also for LWLS we do not have a separate prediction step and therefore do not distinguish between pre-fit and post-fit residuals. Figure 8.2 visualizes estimated real voltage and imaginary voltage using LWLS and DKF during the attack mentioned above. Estimated real voltage and imaginary voltages from LWLS are shown in sub-figure 8.2a. Similarly, sub-figure 8.2b shows estimated real voltage and imaginary voltage using DKF. Similar to the estimation in normal operation, the main difference that can be seen is that SE using the DKF

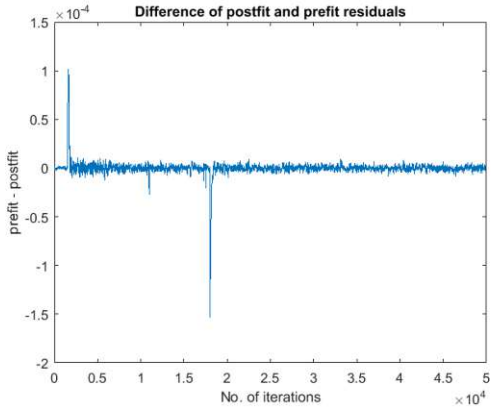
smooths out the voltage signals. Since with the fixed phase angle, we have a higher variation in the real voltage (see Sec. 6.2 of Chapter 6) the smoothing effect for the real voltage is larger than for the imaginary voltage.



(a) Pre-fit residuals of DKF



(b) Postfit residuals of DKF



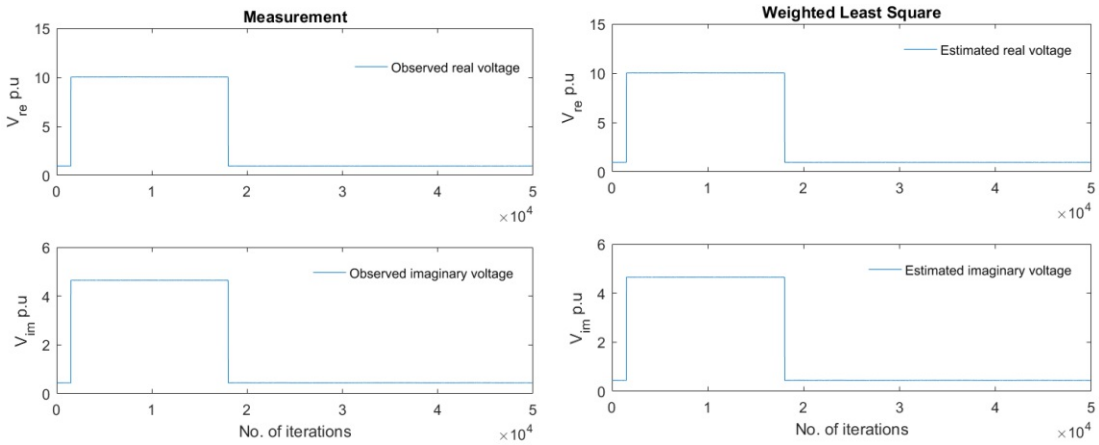
(c) Difference of pre-fit and post-fit residuals for real voltage using DKF

Figure 8.3: Residuals of real voltage and imaginary voltage under attack (Exp. 8.5 of Tab. 8.2).

Using LWLS with only voltage measurement, residuals are zero. Thus the attack can not be identified analysing results from the estimation process. Using DKF, estimated values follow the manipulated voltage signal pattern causing a delay between prediction and estimation. The delay causes high jumps in pre-fit and post-fit residuals shown in Fig. 8.3. The pre-fit residuals and post-fit residuals are shown in sub-figures 8.3a and 8.3b respectively. The difference of the pre-fit residuals and the post-fit residuals are shown in sub-figure 8.3c. Using DKF, when an attack starts, predicted value on the data point depends on the estimated value of the previous non-manipulated data point, whereas the estimated value at the data point considers both the predicted value and the observed

value. Therefore while starting an attack, pre-fit residuals have a higher magnitude than post-fit residual. When an attacker stops manipulating voltage values, prediction in the next data point still is affected by the attack, and as estimation considers prediction it is also effected by the previous manipulated data. Thus for some next data points the estimated value is closer to the predicted value, this results in a higher magnitude of post-fit residuals. From this we conclude, while also for this simple case the attack remains undetected for LWLS (as predicted by Liu et al. in [100]), it can be clearly identified if DKFs are used for the state estimation.

In experiment 8.2, we assume an attacker adds a high constant offset value of 10 p.u. to voltage measurements during the attack. We did this, to check if the attack still remains stealthy (as claimed in [100]). Figure 8.4 shows observed and estimated real voltage and imaginary voltage. After starting an attack at $2,000^{th}$ data point, the manipulated voltage is very high and continue till the attack ends at $18,000^{th}$ data point. We can see this in sub-figure 8.4a, the original signal is small compared to the attack. Similarly, from the sub-figure 8.4b, we can see estimated voltage using LWLS does not filter noise and estimated voltage is very high during the attack.



(a) Observed voltage

(b) Estimated voltage using LWLS

Figure 8.4: Observed and estimated real voltage and imaginary voltage (Exp. 8.2 of Tab. 8.2, the original signal is small compared to the manipulated signal).

Thus we conclude in SE using LWLS considering a simple setup with only voltage measurements, attacks cannot be detected by the residual-based method if LWLS is used for the SE but is detected if DKF is used for state estimation.

8.3.2 State estimation based on voltage and current measurements

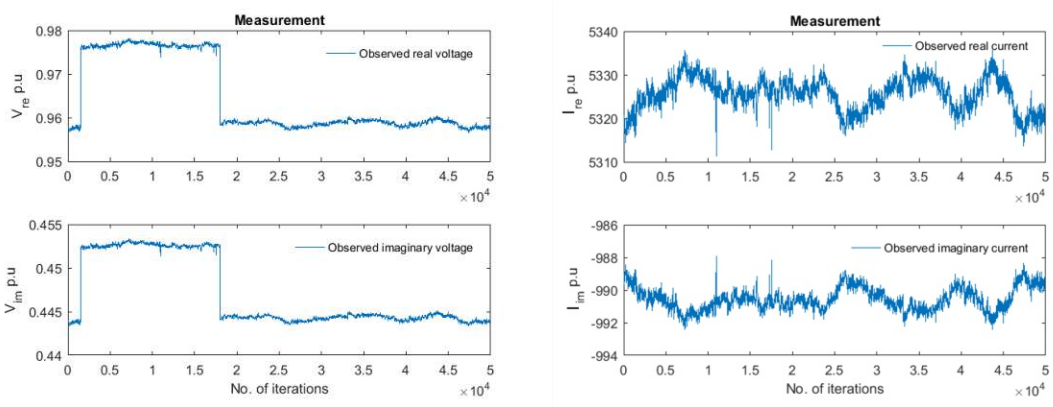
Here we present results for residual-based detection for a SE based on LWLS and DKF SE where for the SE uses both, voltage and current measurements, are used. We then

distinguish two different attacks a) one attack where only voltage measurements can be manipulated by the attacker and b) one attack where the attacker can manipulate both voltage and current measurements. So here the SE is always based on voltage and current measurements.

8.3.2.1 Manipulating only voltage measurements

Here, we consider an assumption presented in Sec. 8.2, the SE is based on voltage and current measurements but the attacker manipulates only voltage measurement. Thus the manipulated real and imaginary voltage measurements, and the non-manipulated real and imaginary current measurements are considered for state estimation.

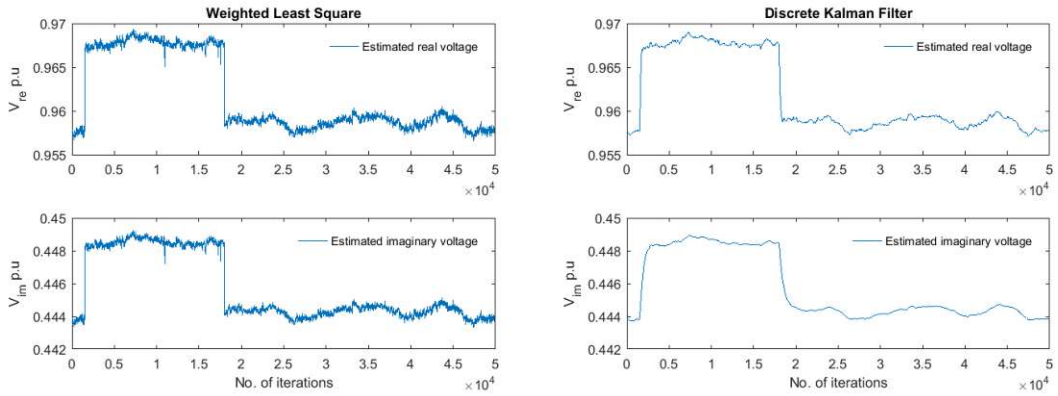
Observed voltage and current measurements are shown in Fig. (8.5). Sub-figure 8.5a depicts observed real voltage and imaginary voltage. The real and imaginary voltages are calculated from the manipulated polar voltage with fixed phase angle (fixed with first observed phase angle). Similarly, sub-figure 8.5b depicts observed real current and imaginary current. The real and imaginary currents are calculated from the actual (non-manipulated) measurement.



(a) Real and imaginary voltage (b) Real and imaginary current

Figure 8.5: Observed voltage and current measurements under attack scenario.

In experiment 8.1, estimated states using DKF and LWLS during the attack scenario are shown in Fig. (8.6). From the figure, one clearly can see the estimated states from both LWLS and DKF follow the attack. Sub-figure 8.6a depicts estimated real voltage and imaginary voltage using LWLS. Another sub-figure 8.6b depicts estimated real voltage and imaginary voltage using DKF, it shows the estimation process smooths out the real voltage and imaginary voltage signals. From here, one can see estimated real voltage and imaginary voltage in the sub-figure 8.6a (estimated with LWLS) are not as much smoothed as estimated by DKF (shown in sub-figure 8.6b).



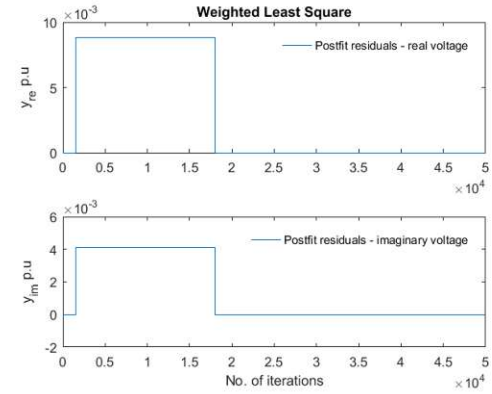
(a) SE using LWLS (b) SE using DKF

Figure 8.6: Estimated states using LWLS and DKF under an attack (Exp. 8.3 and 8.6 of Tab. 8.2).

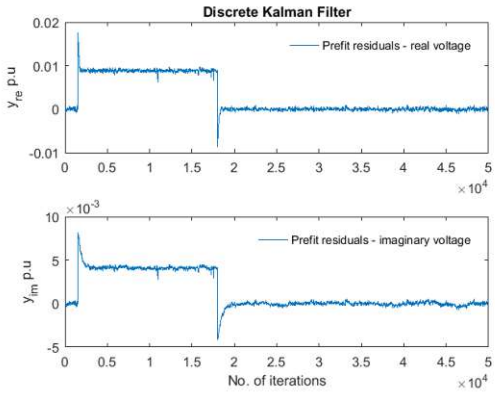
Residuals in DKF and LWLS are visualized in Fig. (8.7). Sub-figure 8.7a shows the residuals using LWLS. One can clearly see that the attack is visible in the residuals of LWLS. Sub-figure 8.7b shows pre-fit residuals of real and imaginary voltages using DKF. The attack clearly leads to some peak in the residuals. In contrast to the LWLS residuals the DKF residuals are much smaller and decrease immediately after the first peak. The reason for this is that in the DKF previous values are taken into account and this way the SE is "trained" to accept the new high value as normal. Sub-figure 8.7c shows post-fit residuals of real and imaginary voltage using DKF. As shown in sub-figure 8.7d, one can see pre-fit residuals are higher in attack starting data point and post-fit residuals are greater in attack ending data point (shown for real voltage). When an attack starts, predicted value on the data point depends on the estimated value of the previous non-manipulated data point, whereas estimated value at the data point considers both the predicted value and observed value. Therefore while starting an attack pre-fit residuals have higher magnitude than post-fit residuals. When an attacker stops manipulating voltage values, prediction in the next data point still is affected by the attack, and as estimation considers prediction it is also affected by the previous manipulated data.

From the sub-figures of residuals, one can see that residuals using DKF are higher than the residuals using LWLS because at a time step k, SE using LWLS considers only the measurements at the time step, whereas SE using DKF considers previous value and the measurements at the time step.

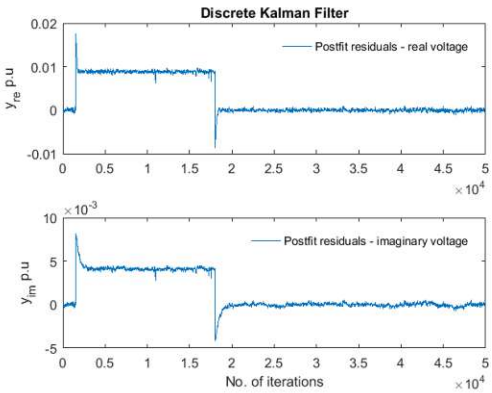
We conclude in SE using LWLS considering voltage and current measurements, if constant offsets are added only to voltage measurements then attack can be seen in the residuals of LWLS and DKF.



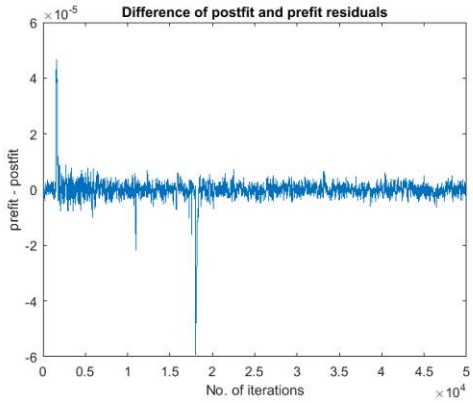
(a) Residuals of LWLS



(b) Pre-fit residuals of DKF



(c) Post-fit residuals of DKF

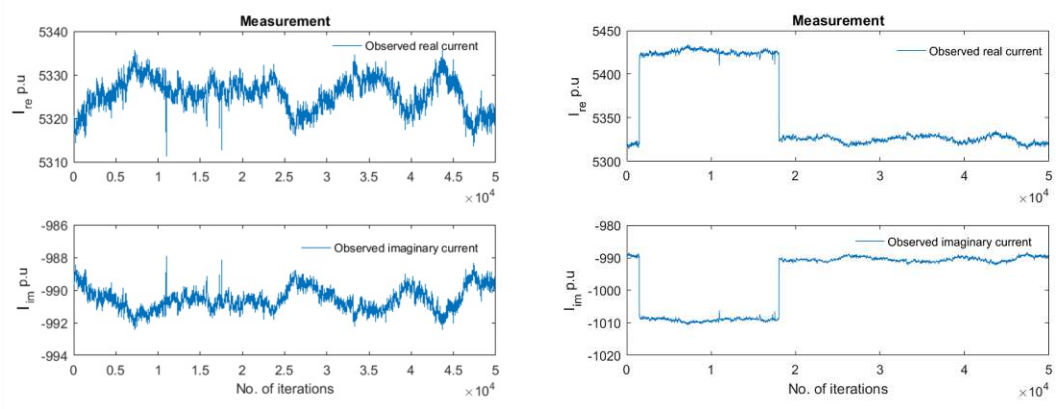


(d) Diff. of pre-fit and post-fit residuals of DKF (shown for real voltage).

Figure 8.7: Residuals of real and imaginary voltage under attack (SE is based on voltage and current; but only voltage is manipulated) (Exp. 8.3 and 8.6 of Tab. 8.2).

8.3.2.2 Manipulating both voltage and current measurements

Here we again assume SE is based on both voltage and current measurement both in a similar manner as presented in Sec. 8.1.2, but also the attacker can manipulate both voltage and current measurements. The real and imaginary voltages are calculated from the manipulated polar voltage with fixed phase angle (fixed with first observed phase angle). The real and imaginary currents are also calculated from the manipulated voltage using the measurement model shown by Eq. 4.64 in Sec. 4.3.2. Also using the measurement model, the calculation of real and imaginary voltages are shown in Eq. 4.65 and Eq. 4.66 respectively. Thus the manipulated real and imaginary voltage measurements, and the manipulated real and imaginary current measurements are considered for state estimation.



(a) Actual current measurement (b) Manipulated current measurement

Figure 8.8: Actual current measurement and current measurement during an attack.

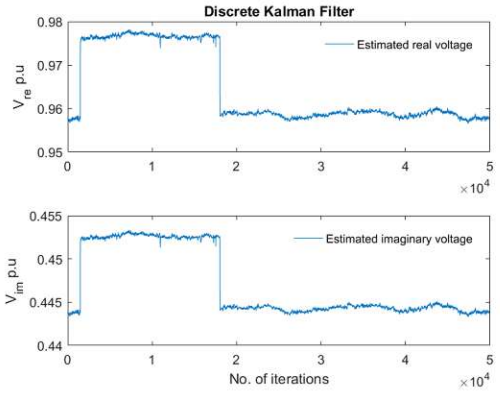
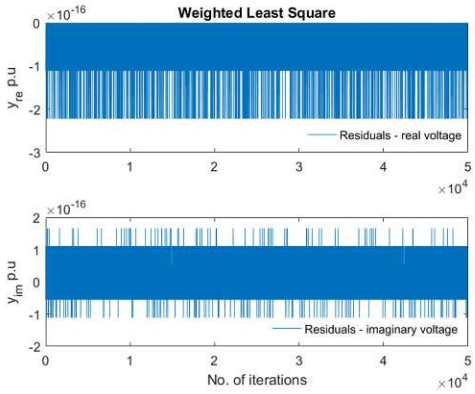


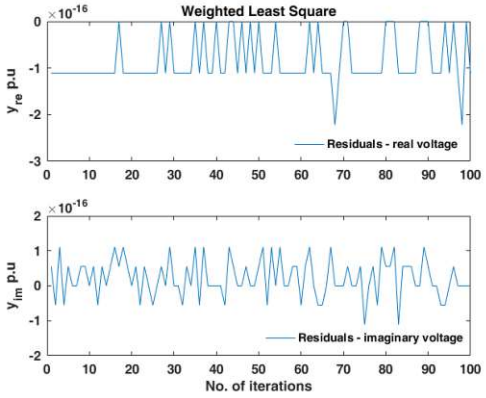
Figure 8.9: Estimated states using LWLS under an attack (Exp. 8.4 of Tab. 8.2).

The manipulated voltage measurement is shown by Fig. 8.1 The actual and manipulated current measurements are shown in Fig. 8.8. Comparing the sub-figures 8.8a and 8.8b, we can see real current increases and imaginary current decreases during the attack. From the Fig. 8.1, one can see the real voltage is higher than the imaginary voltage. Since the real part G of admittance matrix H has positive value and the imaginary part B has negative value, one clearly can see from the Eq. 4.65 that the real current increases and from Eq. 4.66 that the imaginary current decreases during the attack.

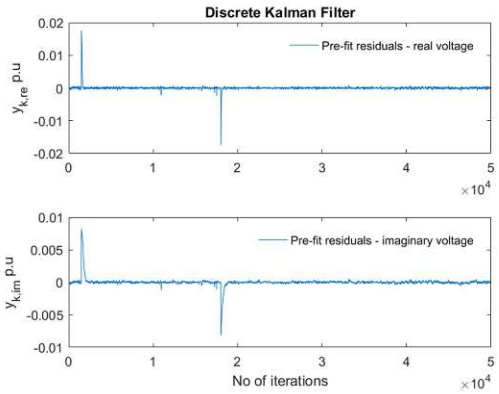
LWLS uses the manipulated voltage and current measurements (shown in Fig. 8.1 and Fig. 8.8b) and estimates voltage states. The estimated voltage states are shown in Fig. 8.9. Another sub-figure 8.10a shows residuals for both real and imaginary voltage which is just a tiny signal. The magnitude of the residuals is similar to the magnitude of the residuals in normal operation shown by Fig. 4.18 in Sec. 4.3.2.3 and the residuals are close to zero. From sub-figure 8.10b, we can see the estimated signal are close to



(a) Residuals using LWLS



(b) Residuals using LWLS (zoomed in)



(c) Pre-fit residuals using DKF

Figure 8.10: Residuals of real and imaginary voltages under an attack (Exp. 8.4 and 8.7 of Tab. 8.2).

observation, and there is tiny difference due to the admittance matrix (as the observed values are multiplied by the admittance matrix). Thus, we conclude the attack can not be detected looking at the residuals if the attacker can manipulate both voltage and current measurements. This is in-line with the work of Liu et al. in [100], that says that attacks remain stealthy for the LWLS residuals if the the attacker can fit them to the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$. In case that voltage and current measurements are used for the SE the attacker needs to be able to manipulate both, since otherwise (as shown in this section above) the attack is visible.

If DKF is used for SE instead of LWLS, the attack can be detected in the residuals as shown in sub-figure 8.10c. So, if a DKF is used for the SE, the attack is visible in the residuals even if the attacker can manipulate both, voltage and current.

We conclude that attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as defined by Liu et al. in [100] remain stealthy for a residual-based detection if LWLS SE based on voltage and current

measurements is used and both, voltage and current measurements, can be manipulated.

Nevertheless, if DKF SE is used based on voltage and current measurements, the attacks can be detected by residual-based detection.

8.3.3 Results Findings

From our experiments, we conclude the following:

- F 2.2.1: Constant offset attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remain stealthy for residual-based detection methods if we use LWLS as SE method based only on voltage measurement and only manipulate the voltage. In this case, the LWLS residuals are always zero (as shown in Exp. 8.1 and Exp. 8.2).
- F 2.2.2: Constant offset attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ can be detected by residual-based detection methods if we use LWLS as SE method based on voltage and current measurement and only the voltage can be manipulated (as shown in Exp. 8.3).
- F 2.2.3: Constant offset attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remain stealthy for residual-based detection methods if we use LWLS as SE method based on voltage and current measurement and both voltage and current can be manipulated (as shown in Exp. 8.4).
- F 2.2.4: Constant offset attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ can be detected for residual-based detection methods if we use DKF as SE method. This was shown for the three different cases: a)SE based on voltage with manipulation of voltage (as shown in Exp. 8.5), b) SE based on voltage and current with manipulation of voltage (as shown in Exp. 8.6) and c) SE based on voltage and current with manipulation of voltage and current (as shown in Exp. 8.7). In all three cases the attack was detected.
- F 2.2.5: With residual-based detection based on LWLS SE the residuals remain high (in the cases where the detection works) whereas with DKF we get one peak when the signal changes and then the signal adjusts to the manipulated values and considers the manipulated signal as the new normal.

The possibility of detecting the manipulation attacks on voltage and current measurements depends on the method and measured vectors used for state estimation. A summary of the results is shown in Tab. 8.3.

Table 8.3: Possibility of detecting stealthy attacks defined by Liu et al. in [100] using residual-based methods; Exp. = experiment; Det. = detection possible.

SE method	Exp.	Measured vector	Manipulation of	Detectable
LWLS	8.1, 8.2	Voltage	Voltage	No
LWLS	8.3	Voltage, current	Voltage	Yes
LWLS	8.4	Voltage, current	Voltage, current	No
DKF	8.5	Voltage	Voltage	Yes
DKF	8.6	Voltage, current	Voltage	Yes
DKF	8.7	Voltage, current	Voltage, current	Yes

8.4 Summary

In this chapter, we presented the stealthiness of the FDI attacks for residual-based detection methods based on LWLS and DKF. Two cases were shown; SE using i) only voltage measurements and ii) both voltage and current measurements. Then we provided an experimental setup of the data manipulation for the stealthy attacks.

We showed the relation of residuals and the SE methods (LWLS and DKF). The following finding help us to answer **RQ 2.2.1** (Can attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against state estimation from [100] remain stealthy if we analyze residuals from state estimation using LWLS?):

- Under the conditions defined by Liu et al. in [100] for LWLS SE, attacks remained stealthy for the case that all measured values used for the SE were manipulated.

For our reasoning **RQ 2.2.1**, we already expected that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against SE from [100] remain stealthy if SE is based on voltage measurement and voltage can be manipulated. The evidence from experiments confirmed that if the attacker could manipulate the voltage measurements then the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remained stealthy in the SE using LWLS and based on voltage. Further, as we already expected that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remain stealthy if SE is based on voltage and voltage can be manipulated adding high offsets. The evidence from experiments confirmed that the attacks remained stealthy if the attacker could manipulate the voltage measurements considered for LWLS SE based on voltage even if the c was quite large. Similarly, for our reasoning, we expected that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ remain stealthy if SE is based on voltage and current; and both voltage and current measurements can be manipulated. The evidence from experiments confirmed that the attacks remained stealthy only if the attacker could manipulate both voltage and current measurements considered for LWLS SE based on voltage and current.

The following finding help us to answer **RQ 2.2.2** (Can attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against state estimation from [100] be detected if we analyze residuals from state estimation using DKF?):

- The attacks defined in [100] did not remain stealthy if DKF was used for SE because the attacks were detected by analysing residuals from SE using the DKF.

For our reasoning **RQ 2.2.2**, we expected that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ do not remain stealthy if we use DKF for SE based on voltage and voltage could be manipulated. The evidence from experiments confirmed that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ are detected by analysing the residuals from SE using DKF and based on only voltage. Similarly, as we expected for our reasoning, the evidence from experiments confirmed that the attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ are detected in if SE is based on both voltage and current and either only voltage or both voltage and current could be manipulated. In both cases, we identified a different by analysing the residuals from SE using the DKF.

CHAPTER 9

Lightweight Statistical Methods

Notice of adoption from previous publications in Chapter 9
Parts of the contents of this chapter have been published in the following papers:

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
Median absolute deviation, Kullback-Leibler divergence, cumulative sum and weighted voting methods in this chapter are based on the work done in [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

In this chapter, we aim to detect the injected attacks introduced in Chapter 5. In a first step, we propose an anomaly detection model. In a second step, we present multiple statistical anomaly detection methods: median absolute deviation (MAD), Kullback-Leibler divergence (KLD), and cumulative sum (CUSUM). In a third step, we present the experimental setup of lightweight statistical methods - MAD, KLD, and CUSUM. In the experimental setup, usage of training data for setting thresholds for these methods is shown. Further, the effect of anomaly detection parameters on anomaly detection performance is presented. In a fourth step, we present results from the experiments and show receiver operating characteristic (ROC) curves of the methods. Our analysis shows, in contrast to L2-norm and normalized residual-based methods (presented in Chapter 7), lightweight statistical methods - MAD, KLD and CUSUM detect at least one anomaly (a malicious or benign anomaly that is also a malicious anomaly) during all attacks introduced in the

Tab. 5.1 of Chapter 5. Then we focus on enhancing the anomaly detection performance through a combination of methods and explore a way to combine methods in order to increase detection performance. First, we investigate using a weighted voting scheme where we assign weights on the methods based on their detection performance; second, we present the experimental setup and last the results from the approach are analyzed to better understand the anomaly detection performance on attack types. The goal is to analyze to which extent the combination approach can increase the overall detection performance.

An attacker can modify voltage measurements and, with some knowledge, can hide the attack in the normal operation of the state estimation and circumvent detection of the attack. Table 9.1 shows the possibility of detecting false data injection attacks generated using an attack model introduced in Chapter 5.

From the Tab. 9.1, we can see some attacks that are not detected by using residuals of LWLS (see exp. 8.1, 8.2 and 8.4) are detected using residuals of DKF (see exp. 8.5 and 8.7). Nevertheless a DKF is not always implemented because this is some effort to implement in a power system. The results analysis in Chapters 7 and 8 shows if a quick high increase is made in the offsets, then it would result in high residuals so that residual-based methods detect the attack if DKF is used for state estimation, but they have poor detection performance due to the following features (weakness that are particular to DKF):

- Residuals are calculated from measurements, so they would adapt if we increase the offsets slowly.
- The residual-based method detected changes quickly but also adapted to the changes quickly so that residuals became small quickly and remained within the threshold. So after some anomalies occurred, it did not detect subsequent anomalies for long.

In order to prevent such stealthy attacks, we propose using multiple statistical anomaly detection methods in an overall effort to achieve effective detection. To this end, we use lightweight statistical methods which have been applied to similar problems (e.g., in [107, 29, 179, 180, 142]).

Our research questions about using lightweight statistical methods read:

- **RQ 2.3**: Is it possible to detect the injected attacks with the lightweight statistical methods?

Further, we divide the research question RQ 2.3 into the following sub-research questions:

Table 9.1: Detection of FDI attacks (BDD methods detect only some of the FDI attacks as shown in chapters 7 and 8, here we summarize the detection of FDI attacks using the residuals-based BDD methods). Attack parameters as described in Sec. 5.3, methods and thresholds as described in sections 7.2 and 8.2; MV = measured vectors; SEM = SE methods; Det. = detection; DP = detection possible in our experiments; MO = manipulation of; Y = yes; N = no.

Exp.	MV	SEM	Det. methods	MO	Attacks	DP
7.1	Voltage	DKF	Plain RB	Voltage	SD	Y
					RSCV, ICOS IROCV	N[1]
7.2	Voltage	DKF	L2-norm	Voltage	CO, RO, ICO, IRO, IROMN, ICOHS	N[2]
7.3	Voltage	DKF	Normalized RB	Voltage	CO	Y
					RO, ICO, IRO, IROMN, ICOHS	N[1]
8.1 8.2	Voltage	LWLS	Plain residuals	Voltage	CO	N[3]
8.3	Voltage, current	LWLS	Plain residuals	Voltage	CO	Y
8.4	Voltage, current	LWLS	Plain residuals	Voltage, current	CO	N[3]
8.5	Voltage	DKF	Plain residuals	Voltage	CO	Y
8.6	Voltage, current	DKF	Plain residuals	Voltage	CO	Y
8.7	Voltage, current	DKF	Plain residuals	Voltage, current	CO	Y

[1] Slowly changing offsets are adopted in the residuals.
[2] With our method the threshold based on the training data was too high.
[3] Attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100].

- **RQ 2.3.1**: Is it possible to detect at least one of the injected malicious anomalies during our injected attacks with the lightweight statistical methods?
  Rationale: As it can be seen in Chapter 8, the attacks that are generated by our model are clearly observable in the signal; we might therefore assume that simple lightweight statistical methods will be sufficient to detect them. It would be beneficial to rapidly detect the attacks and with few computational resources, both to limit the potential consequence of the attack and to accommodate the rate that measurements are generated by a PMU (50Hz). Furthermore, as critical decisions will be based on the outcomes of the detection methods (e.g. regarding network switching decisions), it is beneficial that a human operator is able to understand the results from the detection methods and have them readily explainable (in contrast to, for example, some machine learning and deep learning methods where

explainability remains an open issue [157]). We propose that detecting at least one (or a small number of) anomalous data point(s) is sufficient to raise an alarm and invoke further actions, such as an in-depth analysis of the situation. However, this is only the case if the methods do not generate many false positives – if this were the case, single alarms would likely be lost or dismissed (as a weak indicator) by an operator and not acted on. Finally, the rationale for using several methods (as opposed to one that attempts to detect all the attack forms that our model can generate), is that different configurations of the model clearly generate distinct signals that, presumably, a carefully selected set of detection methods that use different statistical properties can detect.

- **RQ 2.3.2**: How long do the methods take until the first malicious anomaly during the attack is detected?
  Rationale: The earlier we detect an attack, the earlier we can invoke countermeasures (e.g., invoke additional analysis steps, substitute values for state estimation, etc.) The methods that detect the first malicious anomaly fast are beneficial because we can avoid that the attacks can cause damage on the state estimator and subsequently the power system. An aggressive attack can have significant changes in the statistical properties of data, but slow changes during an attack can take time to have significant changes in the statistical properties of the data. Therefore we investigate how long it takes until attacks are detected.

- **RQ 2.3.3**: How many of the malicious anomalies are detected?
  Rationale: The detection of at least one of the injected malicious anomalies during our injected attacks trigger an alarm so that operators conduct an in-depth analysis of the situation. In situations with many false positives, it can help to get the number of positives and only put an alarm if too many positives are detected. The number of detected malicious anomalies can help us to rate the detection performance of methods. In addition, the detection of more than one anomaly can help us to maintain the correctness of state estimation e.g., replacing the detected anomalies before sending the data to state estimation.

- **RQ 2.3.4**: To which extent does detection performance improve if we combine lightweight statistical methods?
  Rationale: The attacks that are generated by our model have different characteristics. Different detection methods use different statistical properties for detecting the attacks. Therefore, it is likely that some methods are well-suited to detect one form of an attack but fail for other attack forms. Thus, the different methods can detect different types of attacks, and anomaly detection performance can be improved by combining the results of the methods. A combination of lightweight statistical methods can produce trustworthy results.

To combine methods, we use a combination technique from literature, a weighted voting scheme which is originally applied to machine learning algorithms in [101]. We describe

the following lightweight statistical methods 1) median absolute deviation (MAD), 2) Kullback-Leibler divergence (KLD), 3) cumulative sum (CUSUM), and 4) combination of methods using weighted voting. Further, we make theoretical considerations whether the methods (MAD, KLD and CUSUM) are able to detect different attacks introduced in Tab. 5.1 of the Chapter 5.

Table 9.2: Overview of lightweight statistical methods.

Methods	**Data***	**Goal of experiment**	**Section**
MAD	Training data (22.03.2016-31.03.2016) Test data (01.04.2016-14.04.2016)	- to answer **RQ 2.3.1** - to answer **RQ 2.3.2** - to answer **RQ 2.3.3**	9.1.2 9.2.1 9.3
KLD	Training data (22.03.2016-31.03.2016) Test data (01.04.2016-14.04.2016)	- to answer **RQ 2.3.1** - to answer **RQ 2.3.2** - to answer **RQ 2.3.3**	9.1.3 9.2.2 9.3
CUSUM	Training data (22.03.2016-31.03.2016) Test data (01.04.2016-14.04.2016)	- to answer **RQ 2.3.1** - to answer **RQ 2.3.2** - to answer **RQ 2.3.3**	9.1.4 9.2.3 9.3
Weighted voting	Training data (22.03.2016-31.03.2016) Test data (01.04.2016-14.04.2016)	- to answer **RQ 2.3.4**	9.4

* For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

Table 9.2 shows the intention of using the lightweight statistical methods and data used for the experiment. Details on parameter settings for the experiment are presented in Sec. 9.2.

9.1 Theoretical Background

9.1.1 Anomaly Detection Model

Here we introduce our model for detecting anomalies. Different false data injection (FDI) attacks on PMU measurements can be detected by analysing different data features. We investigate the characteristics of important and distinct methods for detecting FDI attacks to a wide area monitoring system (WAMS). We are interested in detecting different types of attacks. A single method may not detect all types of attacks or may not perform best for all attack types. A combination of different methods may help to improve anomaly detection performance and enables to detect different attack types.

Our anomaly detection model has three main components, namely bad data detection as shown in Chapters 7 and 8, lightweight statistical methods, and a combination of lightweight statistical methods. Figure 9.1 shows the anomaly detection model. An overview of the model is presented in the following paragraphs.

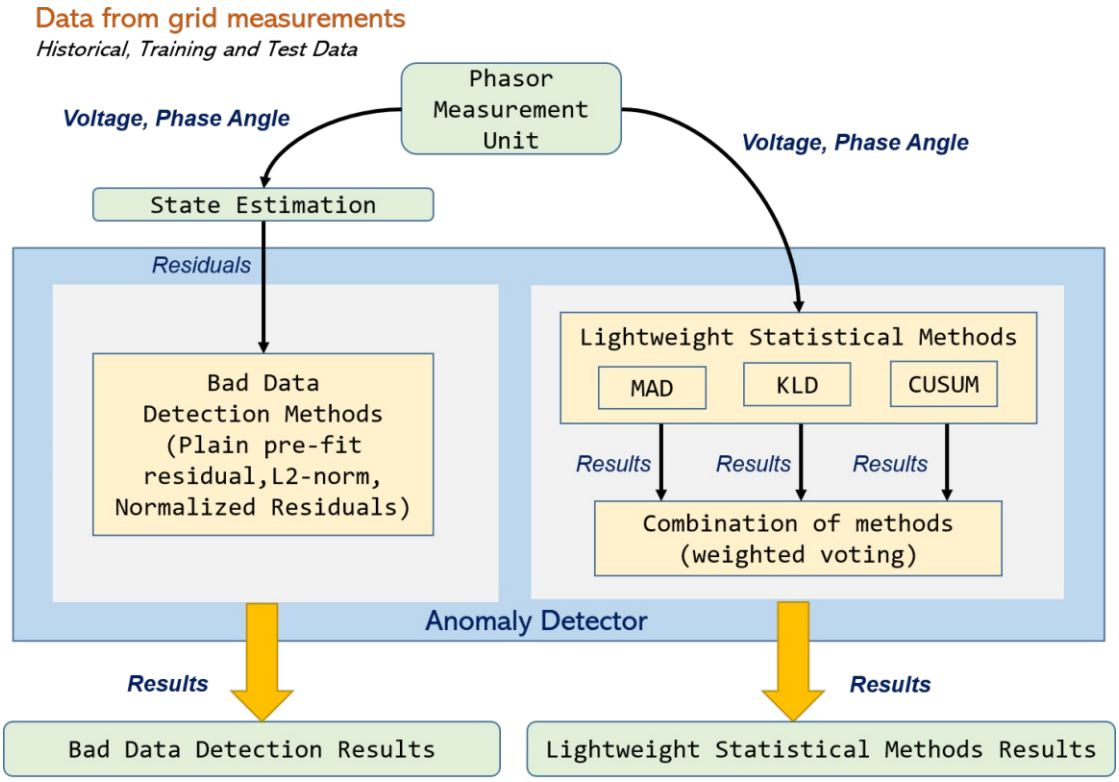


Figure 9.1: Anomaly detection model.

Errors in data sources or FDI attacks could influence state estimation. Therefore, bad data detection (BDD) methods are needed to be implemented in grid operators. Since BDD methods detect data that deviates from normal operation, they could also detect manipulated data [139]. We applied different residual-based BDD algorithms to the FDI attacks described in Sec. 5.3, our investigation found that we cannot detect anomalies in some of the FDI attacks using the residual-based methods. Sophisticated attacks can bypass the classical BDD detection methods [46]. Therefore to complement the detection, we argue that one has to use additional anomaly detection (AD) methods.

To more effectively detect the FDI attacks, we propose the use of three statistical anomaly detection methods [107, 29, 179, 180, 142]: i) a measure of dispersion – the MAD; (ii) a histogram (distribution) based method – the KLD; and (iii) a change point detection method – the CUSUM methods. We have chosen to apply simple statistical methods, in contrast to, e.g., machine learning-based approaches because a) we believe that they are

sufficient to detect our proposed attacks b) to promote the scrutability (understanding) of the results they produce – a desirable property for use in a critical infrastructure setting, such as power distribution networks and c) to keep computational overhead small. Further, considering the computational overhead in the critical infrastructure, we make a selection of lightweight algorithms as they can be deployed and executed with limited resources and memory.

The anomaly detector is connected to two components, namely bad data detection and lightweight statistical methods of the model. Here we check with lightweight statistical methods though attacks are not detected with residual-based bad data detection methods. PMU measurements are fed to the state estimation and the lightweight statistical methods of the anomaly detector. Residuals from the state estimation are fed to the bad data detection component of the anomaly detector. After processing the residuals, results are produced from the bad data detection methods. In the lightweight statistical methods detection component, after processing the measurements the detection results of the lightweight statistical methods are used for combining different methods. Then final anomaly detection results are produced from the combination method.

The lightweight statistical methods are expected to be variously suitable for detecting different attack types. Overall detection performance could be improved by combining their output. To this end, we propose the use of a combination technique in literature namely, a weighted voted scheme to increase precision of the results by reducing false alarms. The voting system is usually used as ensemble methods in machine learning algorithms [101], here we apply it for combining different statistical methods. Weights on the methods are assigned based on their anomaly detection performance metrics (e.g., TPR, TNR).

9.1.2 Median Absolute Deviation

Mean and standard deviation are sensitive to outliers and therefore are not efficient estimators [147, 95]. A robust alternative to the mean is median, the median is less biased than the mean by outliers. Median absolute deviation (MAD) is a robust estimator of dispersion, as it estimates the median of the absolute deviations from the median [146, 147]. MAD can be used in both normal and non-normal distributions [147].

The MAD is defined as Eq. (9.1) [146, 102, 147].

$$MAD = b \cdot median(|x_i - M|) \tag{9.1}$$

where b is a scale factor, M is the median of the data points, $x_i - M$ is the difference of each data point to the median. Thus the MAD is then defined as median of those differences.

In order to use the MAD as an estimator for dispersion, the value is multiplied by a constant scale factor b, which depends on the distribution and is set to $b = 1.4826$ for normal distributed data [146].

For setting a threshold for decisions based on the MAD, one can establish a MAD interval as follows:

$$median - l \cdot MAD < x_i < median + l \cdot MAD \tag{9.2}$$

Here l is the decision level. So similar to saying values should stay within 2 or 3 standard deviations from the mean, we here can say the values should stay within 2 or $3 \cdot MAD$ from the median.

We later use the MAD method in different processing steps. We first use it for labeling the training data. All data points outside the MAD limits are labelled as anomalous.

We then use the MAD method for anomaly detection with a threshold that we set based on training data. We calculate MAD from the training data and set the level of decision such that all normal data in the training data remain within the interval. The upper limit and lower limit of the interval defined by the MAD from the training data are set as a threshold for the classification of the test data.

We also use the MAD method to establish thresholds for the KLD values in a way that we calculate the MAD from the KLD sequence and set the level of decision from the training set (such that anomalies are outside the interval) and then use the resulting MAD interval (of the KLD sequence) to classify the test data.

Theoretical considerations about the detection performance: MAD is suitable to detect deviations from the median (less influenced by the outliers). For setting the MAD boundaries in the test data, we take the median and the MAD of the whole training data set as a reference to set the interval. Based on our data analysis, detectability depends on a) if the test data have a different median from the training data and b) if the test data has a higher deviation from the median of the training data (see Tab. A.4 and Tab. A.5 in Appendix A.11). Attacks are expected to be detected if the data value deviates significantly from the median of the training data.

Table 9.3 shows the expected detecting possibility and detection speed of the types of attacks using MAD. Measurement z_k at a time step k is detected as anomalous if the precondition is true. It is expected to detect CO attacks immediately, i.e, when an offset is large, then the first manipulated data value already remains outside of the interval. As ICO and ICOHS have increasing offsets these attacks are expected to be detect after some time. The MAD method can detect RO, IRO, IROMN attacks if they exceed the boundaries (upper limit or lower limit). Detection can happen due to three reasons a) the deviations from the median in test data differs from the one in training data b) median changes due to an attack c) deviations from median increases due to an attack.

Table 9.3: Detectability of attacks using MAD.

Attacks	Parameters	Precondition	Detection delay
CO	constant (c)	$\text{median}_{\text{train}} - \text{l} * \text{MAD}_{\text{train}} > z_k + c > \text{median}_{\text{train}} + \text{l} * \text{MAD}_{\text{train}}$	Immediately
RO	random (r)	$\text{median}_{\text{train}} - \text{l} * \text{MAD}_{\text{train}} > z_k + r > \text{median}_{\text{train}} + \text{l} * \text{MAD}_{\text{train}}$	Immediately
ICO, ICOHS	slope (s)	$\text{median}_{\text{train}} - \text{l} * \text{MAD}_{\text{train}} > z_k + s \cdot k > \text{median}_{\text{train}} + \text{l} * \text{MAD}_{\text{train}}$	Delayed
IRO, IROMN	random (r), slope (s)	$\text{median}_{\text{train}} - \text{l} * \text{MAD}_{\text{train}} > z_k + r_k + s \cdot k > \text{median}_{\text{train}} + \text{l} * \text{MAD}_{\text{train}}$	Immediately

9.1.3 Kullback-Leibler Divergence

The Kullback-Leibler divergence (KLD) measures the difference of two probability distributions over the same variable [91, 90]. The KLD of a distribution $Q(x)$ from a reference distribution $P(x)$ is a measure of information loss, if we use $Q(x)$ to approximate $P(x)$, i.e., it measures how close are the two distributions [91, 90].

The KLD between two discrete probability distributions $P(x)$ and $Q(x)$ over a discrete domain is represented by Eq. (9.3) [54, 31]

$$D_{KL}(P \mid\mid Q) = \sum_{x \in X} P(x) log \left(\frac{P(x)}{Q(x)} \right) \tag{9.3}$$

The KLD is ≥ 0 and only gets zero if $P(x)$ and $Q(x)$ are equal. The KLD can be used to check a distribution of observations against a reference distribution. In our case, we derive a reference distribution from historic data and then use the KLD to compare our test data with the reference distribution to see if there are distributions that are atypical (anomalous) for the given data.

Figure 9.2 depicts a diagram for calculating KLD. It shows how the KLD sequence is calculated from the reference histogram and the observation.

Figure 9.3 depicts a diagram for detecting anomalies using KLD. Both data points-based and window-based approaches are visualized in the figure. A window is marked as anomalous when it contains at least one anomalous data point. In this case, all data points in the anomalous window are marked as anomalous data points.

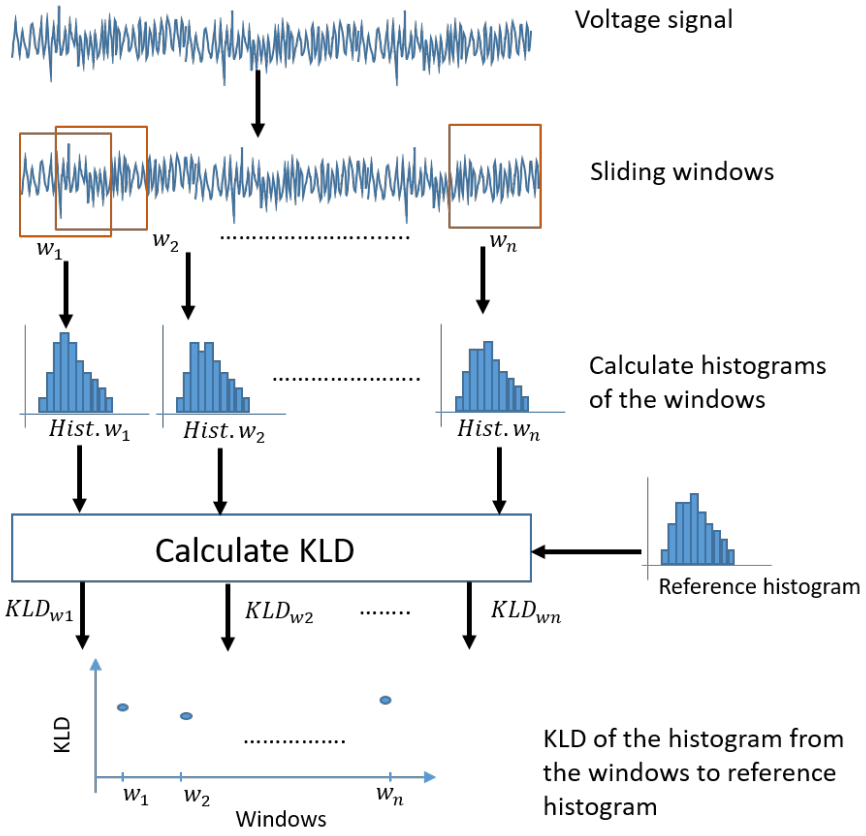


Figure 9.2: A diagram for calculation of KLD.

In the data point-based approach, we look at the fraction of correctly classified anomalous data points and the number of all data points. The accuracy of the data points-based approach is calculated as shown in Eq. (9.4).

$$ACC_{DP} = \frac{TP_{DP} + TN_{DP}}{all\ DP} \tag{9.4}$$

where TP_{DP} represents the correctly classified anomalous data points, TN_{DP} represents the correctly classified normal data points, and the number of all data points $all\ DP = TP_{DP} + FP_{DP} + TN_{DP} + FN_{DP}$.

In the window-based approach, we look at the fraction of correctly classified anomalous windows and the number of all windows. The accuracy of the window-based approach is calculated as shown in Eq. (9.5).

$$ACC_{Win} = \frac{TP_{Win} + TN_{Win}}{all\ Win} \tag{9.5}$$

where TP_{Win} represents the correctly classified anomalous windows, TN_{Win} represents

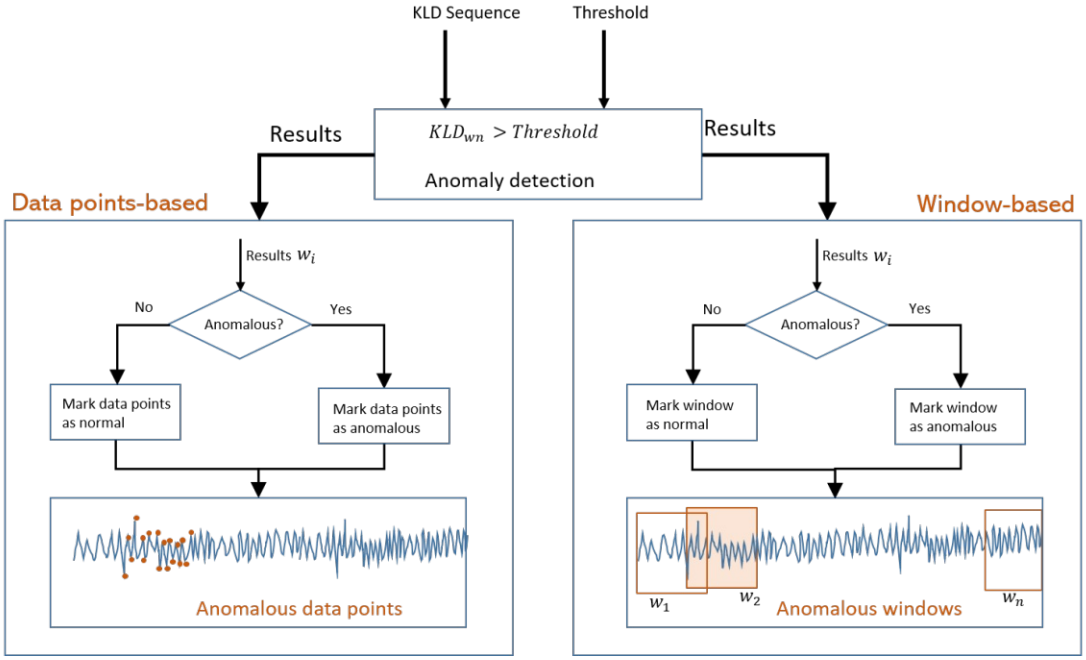


Figure 9.3: Anomaly detection using KLD (data points-based and window-based).

the correctly classified normal windows, and the number of all windows $all\ Win = TP_{Win} + FP_{Win} + TN_{Win} + FN_{Win}$

Theoretical considerations about the detection performance: KLD detects changes in the distributions. So it highly depends on the choice of the reference histogram (e.g., from historical, from training or from previous window), on the window size and on the shifting speed (see Sec. 9.2.2). In order to make a selection of a representative historic data, we consider the historic data from three weeks (see Sec. 6.4) and derive a minute reference histogram as a representative distribution. The one minute histogram represents the distribution of the data (see Sec. 9.2). So we make a comparison of the reference histogram to the histogram of 1 minute (the distribution of values from 1 minute). The reference histogram has a broad distribution in compare to histograms per hour (1.0463 to 1.0728 for an hour - see Sec. 9.2.2). As we set the threshold using the KLD values from the training data, we tolerate a high deviation because the training data already deviates from the historic data. Thus only when values in the test data deviate much more from the histogram from historic data than the values from the training data then it is detected as anomalous.

Table 9.4 shows the expected detecting possibility and detection speed of the types of attacks using KLD. It is expected that the KLD method detects CO attacks immediately only if c is large enough; ICO, IRO, IROMN and ICOHS attacks are expected to be detected after some time because distribution of the data points shifts as the magnitude of the offsets increase with time. Based on the data analysis, we saw that the mean of

Table 9.4: Detectability of attacks using KLD

Attacks	Parameters	Precondition	Detection delay
CO	constant (c)	$D_{KL}(Histo_{test}, Histo_{ref})$ $> t$ with $t = \max(D_{KL}(Histo_{train}, Histo_{ref}))$	Immediately
RO	random (r)	$D_{KL}(Histo_{test}, Histo_{ref}) > t$	Not Detected
ICO, ICOHS	slope (s)	$D_{KL}(Histo_{test}, Histo_{ref}) > t$	Delayed
IRO, IROMN	random (r), slope (s)	$D_{KL}(Histo_{test}, Histo_{ref}) > t$	Delayed

test data is different from training data. However, to be detected as an anomaly, the difference should be significant to change the histogram. So for the attack that makes small changes in mean may not be detected. For instance, for the RO attack (if the mean of the test data is similar to the mean of the training data), it may not significantly change the histogram and may not be detected. An anomaly is assumed to be detected if the 1 minute histogram of test data deviates more than the histogram of training data to the reference histogram.

9.1.4 Cumulative Sum

The cumulative sum (CUSUM) is a sequential analysis method to detect change points in time series. We use a two-sided CUSUM algorithm to detect changes (a decrease or an increase) in the means. Table 9.5 shows an overview of the notation used in CUSUM. Using the definition from [21], we define the sufficient statistic s_i for detecting a change of the mean from μ_0 to μ_1 in a Gaussian distribution with constant variance σ^2, as follows (Eq. 9.6, derived from the log-likelihood ratio as shown in [21]).

$$s_i = \frac{\mu_1 - \mu_0}{\sigma^2}\left(x_i - \frac{\mu_0 - \mu_1}{2}\right) \tag{9.6}$$

This can be rewritten as

$$s_i = \frac{\mu_1 - \mu_0}{\sigma^2}\left(x_i - \frac{\mu_0 - \mu_1}{2}\right) = \frac{b}{\sigma}\left(x_i - \mu_0 - \frac{\nu}{2}\right) \tag{9.7}$$

$$s_i = \frac{b}{\sigma}\left(x_i - \mu_0 - \frac{\nu}{2}\right) \tag{9.8}$$

with the change magnitude ν and the signal-to-noise ratio $b = \frac{\mu_1 - \mu_0}{\sigma}$.

For detecting a change point at time t_n from observations x_i to x_k we build the sum S_n from the s_i as shown in Eq. (9.9):

$$S_n = \sum_{i=1}^{i=k} s_i \tag{9.9}$$

Table 9.5: Notations used in Cumulative Sum.

Notation	Description
μ_0	Mean before a change
μ_1	Mean after a change
σ^2	Variance of distribution
s_i	Sufficient statistics
b	Signal-to-noise ratio
ν	Allowed change magnitude
x_i	Observation at time step i
x_k	Observation at time step k
S_n	Sum of s_i from $i = 1$ to $i = k$
μ_1^+	Deviation from μ_0 in positive direction
μ_1^-	Deviation from μ_0 in negative direction
g_n^+	Cumulative change in positive direction
g_n^-	Cumulative change in negative direction
g_{min}^+	Minimum value of g_n^+
g_{min}^-	Minimum value of g_n^-
h	Threshold
α	Allowed false alarm
N	Size of a block

Since we want to detect deviations from μ_0 in both directions the change is either $\mu_1^+ = \mu_0 + \nu$ or $\mu_1^- = \mu_0 - \nu$ and we use the two-sided CUSUM algorithm to detect if in either of the two cases the value exceeds the threshold h. As shown in [21], we incorporate $\frac{b}{\sigma}$ in the threshold and can therefore express the s_i in a more simple way as

$$s_i = x_i - \mu_0 - \frac{v}{2} \tag{9.10}$$

We then define g_n^+ and g_n^- as

$$g_n^+ = (g_{n-1}^+ + x_n - \mu_0 - \frac{\nu}{2}) \tag{9.11}$$

$$g_n^- = (g_{n-1}^- - x_n + \mu_0 - \frac{\nu}{2}) \tag{9.12}$$

The threshold is defined by Eq. (9.13) as in [117].

$$h = -\frac{\sigma}{\nu/2} ln\, \alpha \tag{9.13}$$

where ν is the maximum variation allowed in the mean of the signal, α is the false alarm probability and σ is the standard deviation of the signal.

In normal condition both g_n^+ and g_n^- continue with negative slopes indicating that the signal is inside the reference variation ν and has normal values. Thus the minimum value of g_n^+ (g_{min}^+) and minimum value of g_n^- (g_{min}^-) are updated in each data point.

Once a positive jump is detected then g_n^+ increases until it reaches a maximum value g_{max}^+ as the following data points do not show big differences anymore, after reaching the maximum value g_n^+ starts decreasing. The maximum value is the point at which the difference in means stops being significant. Figure 9.4 shows an illustrative plot where we can see the change in g_n^+ due to a significant change in mean of the signal.

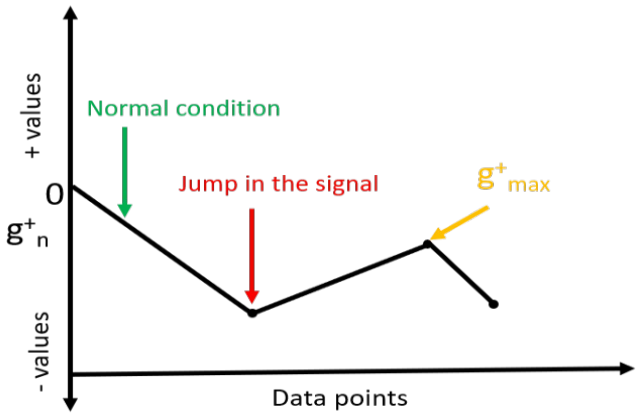


Figure 9.4: A sample CUSUM plot representing a change in g_n^+ due to a significant change in mean of the signal.

Similarly, once a negative jump is detected then g_n^- increases until it reaches a maximum value and again starts decreasing after reaching the maximum value g_{max}^-. The alarm is set to the point where either of them exceeds the threshold h.

$$\begin{gathered} g_n^+ - g_{min}^+ > h \\ OR \\ g_n^- - g_{min}^- > h \end{gathered} \tag{9.14}$$

It could be necessary to process blocks of data for a fast computation (e.g., online processing), thus we compute blocks of data with length N after the block is over g_n^+ is reset to:

$$g_0^+ = g_{max}^+ - g_{min}^+ \tag{9.15}$$

Similarly, we reset g_n^- to

$$g_0^- = g_{max}^- - g_{min}^- \tag{9.16}$$

Figure 9.5 depicts a sample plot for detecting a change using g_n^+. The plot shows the behaviour of the g_n^+ for normal operation and after starting a change. We can see the

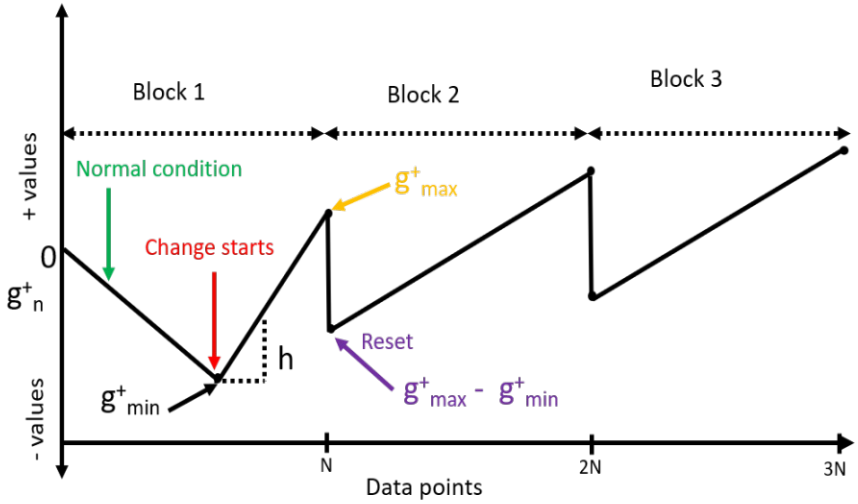


Figure 9.5: A sample CUSUM plot for detecting the change using g_n^+.

normal behaviour on the first block before starting a change, in latter blocks (blocks 2 and 3) g_n^+ always increases due to the change and reset at the end of each block.

Theoretical considerations about the detection performance: CUSUM detects variations in the mean. It tracks if the mean changes over time. It usually depends on the assumption that the data is normally distributed. One can see from the time series of the selected data in Sec. 6.4 that in many cases our PMU data is partially normally distributed and there are already abrupt changes of the mean in the normal data. To tolerate this, we set a large enough threshold. As we define the threshold using the overall mean value of all the training data and maximum variation in the mean of training data the detectability depends on if the mean of the test data deviates significantly from the mean of the training data. Data analysis shows that the mean of test data is different from the mean of training data (see Tab. A.4 and Tab. A.5 in Appendix A.11). The attacks that cause significant changes in mean are expected to be detected.

Table 9.6 shows the expected detecting possibility and detection speed of the types of attacks using CUSUM. In a CO attack the mean changes abruptly and should be detected immediately as changes in mean cross the allowed variation ν, and for ICO, IRO, IROMN and ICOHS attacks, mean has significant change if the magnitude of the offsets are large then they are expected to be detected. For the RO attack, we have the mean of the attack signal = 0, and the randomization component added negative and positive offsets such that the mean does not change. Thus RO is not expected to be detected, but due to changes in the actual data might cause the detection during the attack type RO.

Table 9.6: Detectability of attacks using CUSUM.

Attacks	Parameters	Precondition	Detection delay
CO	constant (c)	g_n^+ of $z_k + c - g_{min}^+ > h$ or g_n^- of $z_k + c - g_{min}^- > h$	Immediately
RO	random (r)	g_n^+ of $z_k + r - g_{min}^+ > h$ or g_n^- of $z_k + r - g_{min}^- > h$	Delayed
ICO, ICOHS	slope (s)	g_n^+ of $z_k + s \cdot k - g_{min}^+ > h$ or g_n^- of $z_k + s \cdot k - g_{min}^- > h$	Delayed
IRO, IROMN	random (r), slope (s)	g_n^+ of $z_k + r_k + s \cdot k - g_{min}^+ > h$ or g_n^- of $z_k + r_k + s \cdot k - g_{min}^- > h$	Delayed

9.2 Experimental Setup

We aim to detect the manipulation attacks introduced in Tab. 5.1 of Chapter 5. Some of these attacks are not detected by RB BDD (shown in Sec. 7.3.3). Here we aim at detecting the attacks with the methods MAD, KLD and CUSUM. We use the same data sets and process for defining normal and malicious data points as described in Sec. 6.5. Figure 6.14 shows the processing steps that are performed on the data. Depending on the method, we proceed with different steps.

Additionally, we present influencing factors of the anomaly detection methods. We briefly discuss on methods' parameters, dependency of the one parameters to other parameters of the methods, along with methods' goals and anomaly detection performances. Table 9.7 illustrates methods goals, parameters and the key parameters on which anomaly detection performance depends.

In contrast to existing work of the statistical methods using simulated-based data (e.g., in [165],[56],[58],[174],[57]), we use real measurement from the EPFL campus network (see Chapter 6). We recall that since we have noisy real data and we may have high thresholds as they are defined based on the real noisy data. The high thresholds may influence the detection performance of the statistical methods.

9.2.1 MAD

For the MAD method, we use the interval (with decision level 3.5) that was used for labelling the training and test data as the threshold. With this we can ensure that all BAs in the test data will be detected as anomalies. Using the MAD threshold, we classify the data points of the manipulated test data as normal or anomalous.

Table 9.7: Influencing factors of detection methods; DPs = data points.

Methods	**Detection goal**	**Parameters**	**Parameters dependency**	**Parameters for testing**
MAD	If data is outside of MAD interval	- Reference median - Time window for calculating reference median - Decision level for threshold interval - MAD scale factor - Threshold (interval)	- Median from training data - MAD scale factor (b) to 1.14826 - Threshold covers all DPs from training data	- Threshold
KLD	If distribution differs from reference distribution	- Reference histogram - Time window of histograms - Sliding time of histograms - Decision level for divergence - MAD scale factor - Threshold for divergence	- Time window set to 1 min - Sliding tine set to 1 sec - Median from training data divergence - MAD scale factor from training data divergence - Threshold covers all divergences from training data	- Reference histogram - Window size - Sliding size - Threshold for divergence
CUSUM	If mean changes over time	- Threshold for $g_n{}^+$, $g_n{}^-$ - Maximum allowed variation ν - Mean value μ_0 - Standard deviation σ - False alarm probability α	- Variation in mean from training data - Probability of false alarm - Standard deviation from training data	- Threshold for $g_n{}^+$, $g_n{}^-$

Parameters for anomaly detection: MAD checks whether a data point is outside the MAD interval, an interval represents lower boundary and upper boundary of the voltage magnitude (see Tab. 9.8).

MAD has a reference median calculated from the training dataset. Thus the reference median depends on the time window from which reference median is calculated. The interval for voltage depends on scale factor b which depends on the distribution of the reference data. In our case, we fixed the value of b, and b is set to $b = 1.14826$ [146] which is used for normal distribution. Decision level l is set such that it covers the MAD interval. Thus the two thresholds, upper and lower boundaries of the intervals of l depends on the reference data. In our case l is set to $l = 2$.

MAD's performance depends on how well the thresholds are set. We checked the distribution of the training voltage and found it has partial normal distribution and therefore we select $b = 1.4826$ [146] which is used for the normal distribution.

9.2.2 KLD

For the KLD method, we first need to calculate a reference histogram. For this, we use all the data points from the historical data. It is possible to compare and calculate an average histogram from histograms with the same range (lower and upper boundaries). Consequently, we set the histogram range to the minimum and maximum value of the historical data; the number of bins to 60 and determine the width of the bins (edges). For all histograms, we hold the same range, bins and edges. First, we calculate 60 histograms per day (1 hour) as we have a 1-minute window and only consider 1 hour per day; second, we calculate an average histogram (1-minute window) per day out of the 60 histograms, resulting in 31 histograms (1-minute window) from 31 days. Then, from all days, we calculate the average reference (1-minute) histogram. Figure 9.6 depicts the generated reference histogram. It uses 3,000 data points. The highest value is 1.0712 and the lowest is at 1.

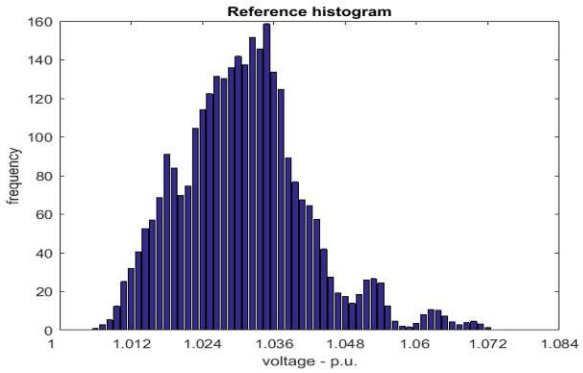


Figure 9.6: Reference histogram (source Paudel et al. [133])

We then calculate the KLD for the training data. Based on our analysis on different window sizes and shifting time, windows of size 1 minute that shift in steps of 1 second visualizes changes in the distribution. Thus, we use a sliding time window of size 1 minute

that progresses in steps of 1 second and then calculate for each window a histogram and compare it with the reference histogram by generating a KLD value. We use the sequence of KLD values from the training data to set a KLD threshold. For this we apply the MAD method to the KLD sequence and check how we would need to set the decision level so that all normal data points lie within the interval.

We then calculate the KLD sequence for the manipulated test data (using the reference histogram from the historical data) and then use the KLD threshold to classify the data points as normal or anomalous.

Parameters for anomaly detection: KLD checks whether the distribution of measurements differs from the reference distribution. Proper adjustment of time window of histogram and sliding time of histogram can increase anomaly detection performance. Thus, the threshold of KLD depends on the MAD scale factor b which depends on the distribution of training data KLD sequence. The MAD scale factor b is set to the reciprocal of the inverse of quantile function [146] (computed for 0.75 probability of the KLD sequence distribution). Decision level l is set for covering the normal behavior of the training data.

KLD's anomaly detection performance depends on how the reference histogram is calculated, and how representative is the histogram for normal traffic.

Since normal traffic changes over time, a better option can be having a sliding window and comparing the sliding window with more recent value. For instance, use the KLD differences to the previous time window. But in this case, the method adapts to changes. Similar to the residuals in state estimation, it therefore could be used by attackers to trick the system. As we aim detecting changes, we do not use the KLD differences of subsequent time windows.

9.2.3 CUSUM

For the CUSUM method, we use maximum variation ν, standard deviation σ, mean value μ_0 from the training data and a target false alarm rate of α between 0.1 and 0.9 to calculate a threshold. We then calculate the CUSUM for the manipulated test data and then use the CUSUM threshold to classify the data points as normal or anomalous.

CUSUM resets once the block of length N is computed. If N is large then false negative would be small but at the same time false positives may increase. We tried different values of N and looked at the trade of positives and negatives. In order to avoid large false positives, we have chosen small value of N = 3,000 (data of 1 minute). Then it continues calculating g_n^+ , g_n^- from the next data point.

Parameters for anomaly detection: CUSUM checks if the mean changes over time. The threshold for ${g_n}^+$ and ${g_n}^-$ is set using the maximum allowed variation ν, mean μ_0 and standard deviation σ from the training data. In addition, the probability of false alarm α is considered for calculating the threshold.

CUSUM's anomaly detection performance depends on the selection of values for calculating the threshold, for instance the value of allowed variation (value of ν), accepted false alarms (value of α). Block length N also has an influence on the detection performance. For instance, if N is too large then history data are taken into account so that the method gets confused and triggers false alarms.

Table 9.8: Overview of methods, parameter setting, thresholds and injected attacks; Exp. = experiment; DL = decision level; b = scale factor; t = threshold; LL = lower limit; UL = upper limit; WS = window size; ST = window sliding time; CO = constant offset; RO = random offset; ICO = incremental constant offset; IRO = incremental random offset; IROMN = incremental random offset with more noise; ICOHS = incremental constant offset with high slope.

Exp.	**Methods**	**Data***	**Param. setting**	**Threshold**	**Injected attacks**	**Sec.**
9.1	MAD	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	For labeling (DL = 3.5) For AD (DL = 2) b = 1.4826	For AD UL = 1.07 LL = 1.051	CO, RO ICO, IRO IROMN, ICOHS	9.3.1
9.2	KLD	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	DL = 3.2, ST = 1 sec, WS = 1 min	t = 8.95	CO, RO ICO, IRO IROMN, ICOHS	9.3.1
9.3	CUSUM	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	$\nu = 0.0076$, $\sigma = 0.0046$, $\alpha = 0.005$ N = 3,000	t = 6.41	CO, RO ICO, IRO IROMN, ICOHS	9.3.1

* For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

Table 9.8 shows an overview of lightweight statistical methods (MAD, KLD and CUSUM), parameter settings, thresholds and injected attacks. We recall that the thresholds of the methods are defined for polar voltage and no transformation into rectangular coordinates is necessary.

9.3 Results

Here we present the results of the three lightweight statistical methods from experiments 9.1, 9.2 and 9.3. We use anomaly detection results for data points-based and windows-based approaches. In the data points-based approach, a method checks for each data point if it is an anomaly. In the window-based approach, a method checks for a window (a whole set of subsequent data points) if the characteristics of the window differ from the normal behavior. For MAD and CUSUM, we only use a data points-based approach. For KLD we use both, a data points-based (KLD-DPB) and a window-based (KLD-WB) approach as they reflect how both approaches would be used in practice. We compare how close the histogram of the data points of a sliding window is to a reference histogram. If the difference exceeds the threshold, then in both approaches (KLD-DPB and KLD-WB) all data points of the sliding window are marked as an anomaly. For the performance of the KLD-DPB approach, we compare anomalous data points to total data points. And for the performance of KLD-WB, we compare anomalous windows to total windows as explained in Sec. 9.1.3.

9.3.1 Detection of Anomalies per Attack

9.3.1.1 Constant Offset Attack

The CO attack is well detected by all methods, if we count the detection of at least one malicious data point during the attack. Table 9.9 shows that the CO attack type is detected in all test data sets by all methods. Nevertheless, if we look at the detection delay we can see big differences. MAD picks up the attack always immediately when the first anomalous data point occurs (see Tab. 9.10).

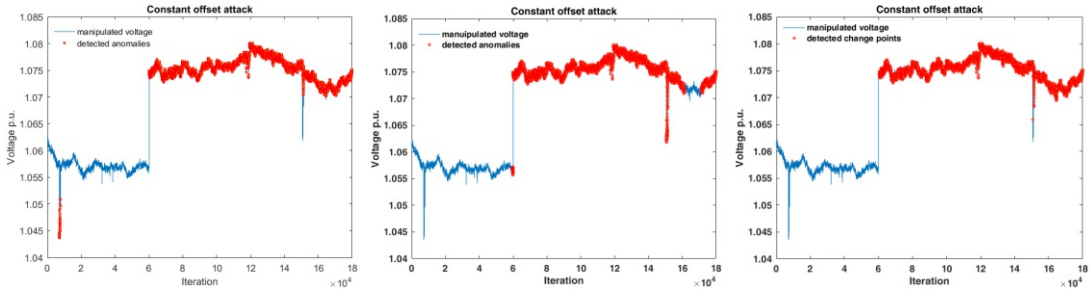
The KLD-DPB detects the attack earlier then the window-based approach. KLD-WB needs longer than MAD to detect that a change has occurred (between 1499 and 3049 data points). This can be explained, because the window progresses over the attack data points and only after some time the histogram of the data points in the window differs sufficiently from the reference histogram so that it is detected as an anomaly.

CUSUM needs longer than MAD and shorter than KLD to detect the attacks (570 to 653 data points). This is because it needs some time until the constant change influences the mean value.

In the following we look further into details to explain the different effects.

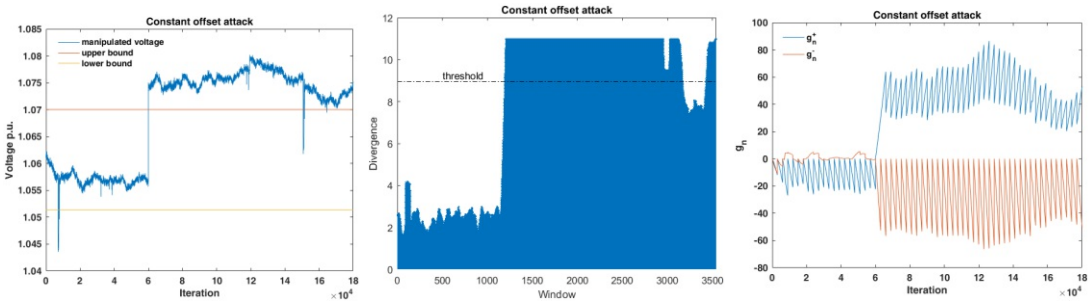
Figure 9.7 shows the manipulated voltage for attack CO for the first (out of the 14) test data set (April 1, 2016). The anomalies that were detected by the three different methods are shown as red data points. Figure 9.8a shows upper and lower bound of simple MAD where one can see the attack crosses the upper bound immediately. It can be seen in the sequence of KLD values shown in Fig. 9.8b, where each data point represents the KLD

which compares the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the reference histogram. From Fig. 9.8c one can see the calculated CUSUM sequence for the CO attack.



(a) MAD (b) KLD (data points-based) (c) CUSUM

Figure 9.7: Visualization of detected anomalies in constant offset attack (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).



(a) MAD interval (b) KLD sequence (c) CUSUM sequence

Figure 9.8: MAD interval, KLD and CUSUM sequences for constant offset attack.

One can see how the simple MAD method picks up the anomaly generated by the attack immediately (at the $60,001^{st}$ measurement) and detects most of the following anomalous data points (Fig. 9.7a). This is not surprising because the attack just suddenly increases the voltage and the MAD just checks for a threshold. The MAD also correctly detects the benign anomaly between data points 7,093 and 7,815.

For the KLD-DPB method one can see that the attack is detected but a bit earlier (Fig. 9.7b) because we mark all data points of a window as anomalies if the window contains one anomalous data. The CUSUM method detects a change point at data point 570, it takes a bit more time (until the data point 570) to recognize the change. Also one can clearly see how the benign anomaly in the beginning is not detected as a change point by CUSUM, because it just consist of a few outliers that do not significantly change the mean. This is according to the observation described in Sec. 9.3.4.4

9.3.1.2 Random Offset Attack

The malicious data points due to RO attack are quite difficult to detect. From Tab. 9.9, we can see the number of average detected anomalous data points is the smallest than the other attack types. With the MAD method the overall accuracy for all test data is only 36.10% and the recall is 4.65%. That means we miss more than 95% of the anomalous data points. KLD-DPB performance is similar to the MAD, the overall accuracy is 36.48% and the recall is 7.94% but the KLD-WB performs worse. CUSUM also detects a few anomalous data point of the attacks type RO (recall= 34.32%). KLD-WB detects less then 2 % of the anomalies (recall= 1.6%) in all 14 data sets.

The bad detection performance from KLD and CUSUM can be explained, because the random offset is performed by adding values from a random normal distribution with mean $\mu = 0$ (see table 5.1). For KLD it seems that most of the added random values stay within the reference histogram, because the reference histogram was selected from 3 weeks and therefore is already quite broad. Also with a mean $\mu = 0$ the attack does not influences the mean over time and therefore will not exceed the CUSUM threshold.

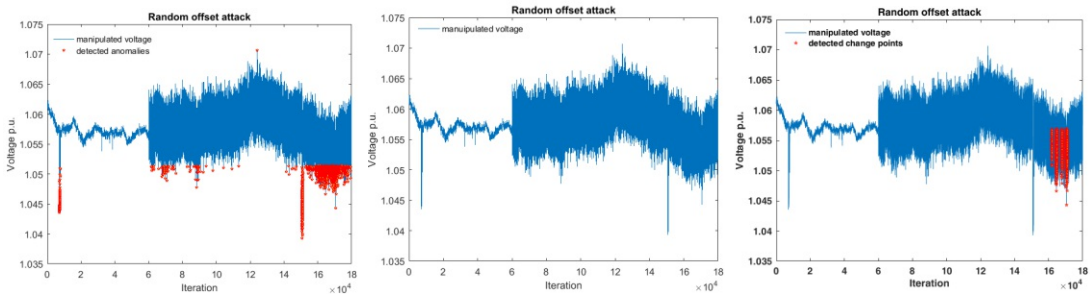
Figure 9.9 shows the manipulated data for attack RO for the first day of the test data set. It can be seen from Fig. 9.10a that the MAD method only detects those random data points which by chance exceeded the threshold. KLD (both KLD-DPB and KLD-WB) detects not a single anomaly in the first data set, because even with the random added data points the histograms in the sliding windows do not differ much from the reference histogram. This can be seen in the sequence of KLD-WB values shown in Fig. 9.10b, where each data point represents the KLD value calculated from the comparison of the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the reference histogram. On average, KLD-DPB detects only 3,018 data points on all 14 test data sets (see Tab. 9.9). From Fig. 9.10c one can see that also the calculated CUSUM values stay within the threshold 6.41 in most of the test data sets (see Tab. 9.9). In the first test data set (April 01, 02:00-03:00), CUSUM detects change points between data points 161,904 and 171,000 as g_n^- crosses the threshold. One can see from sub-figure 9.10c that g_n^- has significant changes between the data points 161,904 and 171,000.

9.3.1.3 Incremental Constant Offset Attack

ICO attack is detected (at least one malicious anomaly) in all test data sets by all methods (see Tab. 9.9). Figure 9.11 visualizes detected anomalies on the first test data set.

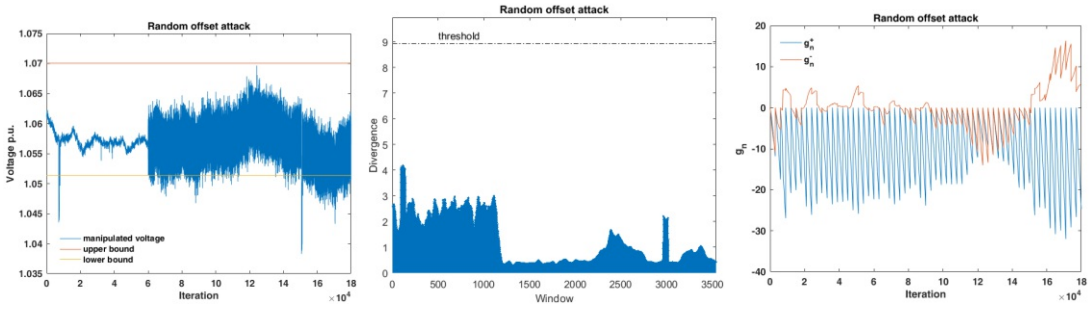
Due to the incremental increase, the attack is only detected after it exceeds a threshold. Therefore the overall performance for MAD and KLD is slightly worse than for the constant offset and the detection takes much longer (see Table 9.10).

CUSUM also detects the attack (see first red point in Fig. 9.11c) but also quite late (the fastest detection was 12249 data points after the attack start). From Fig. 9.12a one



(a) MAD (b) KLD (c) CUSUM

Figure 9.9: Visualization of detected anomalies in random offset attack (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).



(a) MAD interval (b) KLD sequence (c) CUSUM sequence

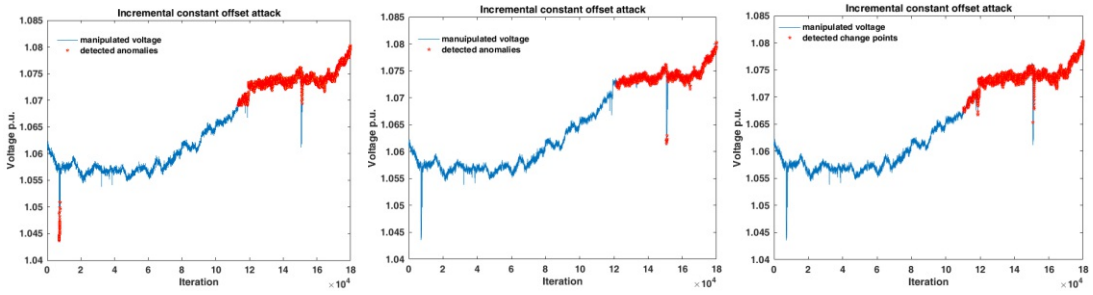
Figure 9.10: MAD interval, KLD and CUSUM sequences for random offset attack (source Paudel et al. [133]).

can see the manipulated voltage crosses the upper boundary of MAD only after adding the high offsets. Figure 9.12b shows the sequence of KLD values, where each data point represents the KLD which compares the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the reference histogram. From Fig. 9.12c one shows the calculated CUSUM sequence for the ICO attack.

9.3.1.4 Incremental Random Offset Attack

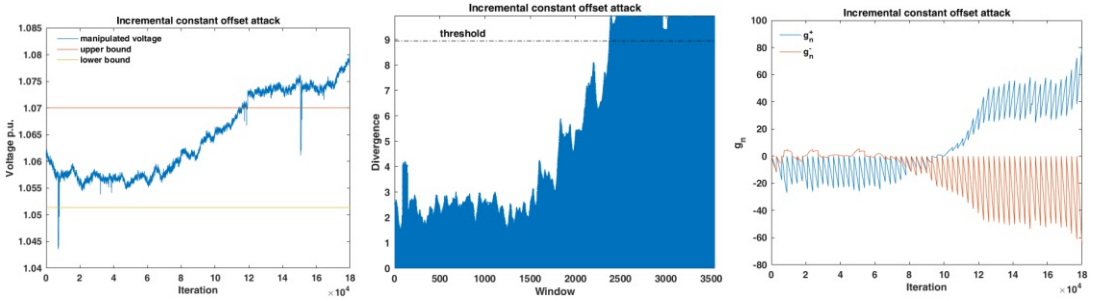
IRO attack is also detected (at least one malicious anomaly) in all test data sets by all methods (see Tab. 9.9) but on average it takes longer than for the pure incremental offset (ICO attack) until anomalies are detected (see Fig. 9.13). This can be explained by the random component that in some cases prevents that thresholds are exceeded.

Figure 9.14a shows the MAD interval on IRO attack. Figure 9.14b shows the sequence of KLD values, where each data point represents the KLD which compares the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the



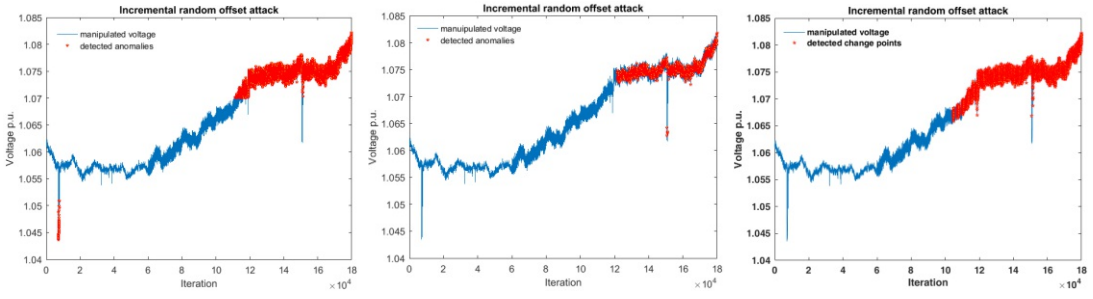
(a) MAD (b) KLD (c) CUSUM

Figure 9.11: Visualization of detected anomalies in incremental constant offset attack (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).



(a) MAD interval (b) KLD sequence (c) CUSUM sequence

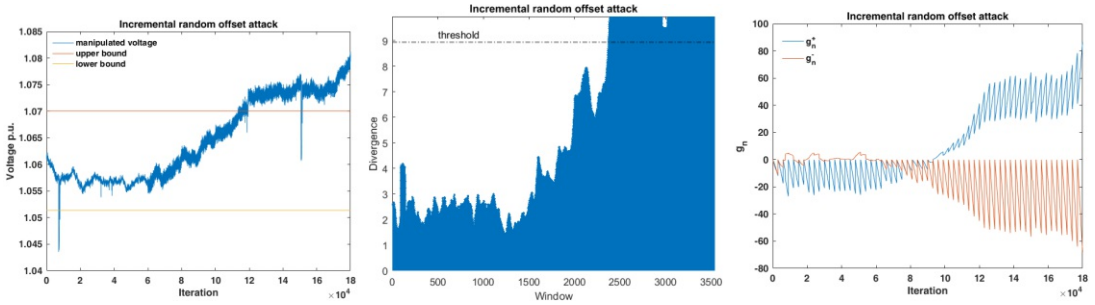
Figure 9.12: MAD interval, KLD and CUSUM sequences for incremental constant offset attack.



(a) MAD (b) KLD (c) CUSUM

Figure 9.13: Visualization of detected anomalies in incremental random offset attack (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).

reference histogram. From Fig. 9.14c one shows the calculated CUSUM sequence for the IRO attack.

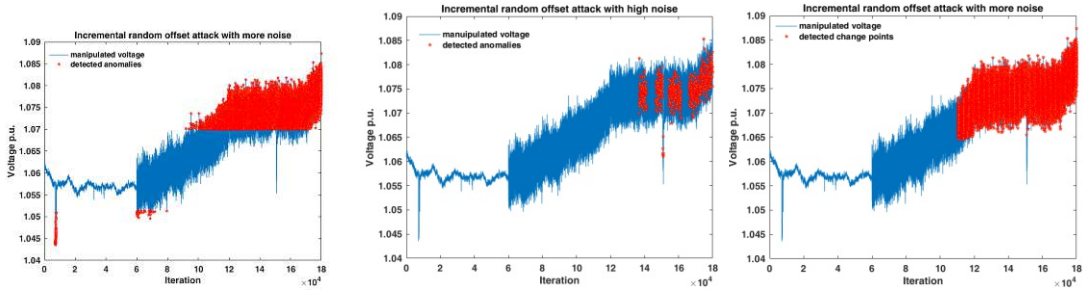


(a) MAD interval (b) KLD sequence (c) CUSUM sequence

Figure 9.14: MAD interval, KLD and CUSUM sequences for incremental random offset attack.

9.3.1.5 Incremental Random Offset Attack with More Noise

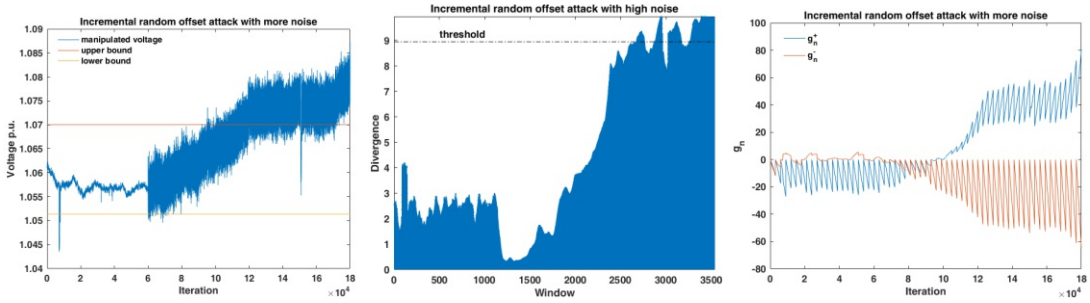
Table 9.9 shows that the IROMN attack is also detected on all test data sets by all methods but on average it takes longer than for the IRO attack until anomalies are detected (see Fig. 9.15). This can be explained by the random component that in some cases prevents that thresholds are exceeded.



(a) MAD (b) KLD (c) CUSUM

Figure 9.15: Visualization of detected anomalies in incremental random offset attack with more noise (shown on April 01, 02:00-03:00).

From Fig. 9.16a one can see even with a high random component some data points remain within the boundary. Figure 9.16b shows the sequence of KLD values, where each data point represents the KLD which compares the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the reference histogram. From Fig. 9.16c one shows the calculated CUSUM sequence for the IRO attack.



(a) MAD interval (b) KLD sequence (c) CUSUM sequence

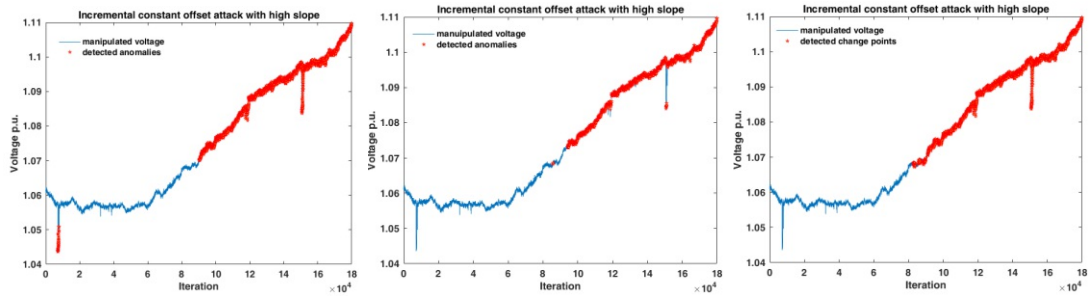
Figure 9.16: MAD interval, KLD and CUSUM sequences for incremental random offset attack with more noise.

9.3.1.6 Incremental Constant Offset Attack with High Slope

The ICOHS attack is detected well by all methods (see Tab. 9.9) Figure 9.17 shows detected anomalies in first test data sets.

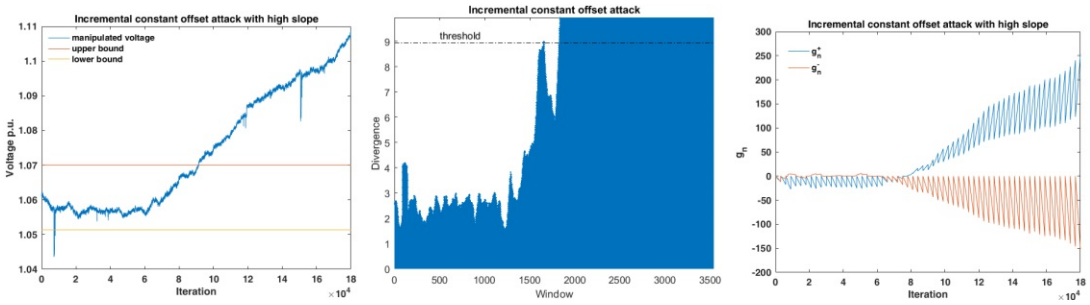
Due to the incremental increase with high slope, the attack is only detected after it exceeds a threshold. Therefore the overall performance for MAD and KLD is slightly worse than for the constant offset and the detection takes much longer (see Tab. 9.10). But it is better than for the incremental constant offset.

From Fig. 9.18a one can see due to increment in slope, the attack is detected earlier than ICO and a jump (between 14 and 16) remain outside of the MAD interval. CUSUM also detects the attack (see Fig. 9.17c). The fastest detection was 1661 data points after the attack start. Figure 9.18b shows the sequence of KLD values, where each data point represents the KLD which compares the histogram calculated from a sliding window (size 1 min, sliding step 1 sec) of the data with the reference histogram. From Fig. 9.18c one shows the calculated CUSUM sequence for the ICOHS attack.



(a) MAD (b) KLD (c) CUSUM

Figure 9.17: Visualization of detected anomalies in incremental constant offset attack with high slope (shown on April 01, 02:00-03:00) (source Paudel et al. [133]).



(a) MAD interval (b) KLD sequence (c) CUSUM sequence

Figure 9.18: MAD interval, KLD and CUSUM sequences for incremental constant offset attack with high slope.

9.3.2 Attack Detection

An attack is assumed to be detected if at least one of the malicious data points is detected as an anomaly. We show the detected data points that are calculated from all data points from all 14 manipulated test data sets (which always include 14 attacks of one type). Here we use original labels (i.e., BA or MA for an anomaly and normal for a normal data point) for the calculation of the performance metrics. Labels provided by the detection methods as explained in Sec. 6.5 are compared to the original labels.

Table 9.9 shows the different detected attack types for all methods. The table depicts the number of test datasets in which the attacks are detected, and the average detected anomalous (malicious and/or benign) data points for the 14 test data sets. Malicious and benign data points on the 14 test data sets are shown in Tab. 6.3 of Chapter 6. In the Tab. 9.9, we consider detection of only those anomalies during the attacks (the benign anomalies after staring and before ending the attacks are counted as malicious). It can happen that BAs are above threshold but here we show what can happen only during attacks.

For the attacks at least one data point was detected except for the random offsets attack. Detected data points using MAD, KLD and CUSUM for each data sets are shown in Tab. A.3 of Appendix A.10. We detect at least one anomalous data point in all attacks.

Detection delay (measure in number of data points after the attack started) results are shown in Tab. 9.10. In the window-based approach, we measure the number of windows after the attack started as a detection delay. KLD window-based detection delay results are shown in Tab. 9.11. Additionally, Tab. 9.10 shows how fast the attacks are detected. It therefore only summarizes the detection delay for the malicious anomalies and does not consider the detection delay for the benign anomalies that occured before the attack. The detection delay shows the number of data points between attack start and the first detection of an anomaly. The table shows the minimum, maximum and average detection delay observed in the 14 test data sets.

Table 9.9: Detected data points using MAD, KLD and CUSUM methods (rounded average values are shown for the detected data points); an attack is detected if one data point is detected.

Method	**Detected Attacks (average detected data points)**					
	CO	RO	ICO	IRO	IROMN	ICOHS
MAD	14 (114,088)	14 (5,586)	14 (71,771)	14 (71,871)	14 (71,977)	14 (97,871)
KLD	14 (119,289)	1 (3,018)	14 (82,614)	14 (81,525)	14 (68,468)	14 (106,896)
CUSUM	14 (119,379)	6 (41,196)	14 (92,773)	14 (93,437)	14 (92,625)	14 (105,760)

Table 9.10: Minimum, maximum and average anomaly detection delay of different methods (source Paudel et al. [133]).

Methods	**Attack**	**Detection Delay**		
		min	**max**	**average**
MAD	CO	1	1	**1.00**
	RO	4	35,382	**4,696.10**
	ICO	2,055	70,669	34,593.50
	IRO	1,847	70,669	32,126.00
	IROMN	1	21,824	**2,763.07**
	ICOHS	2,044	39,230	18,626.00
KLD	CO	1	1	**1.00**
	RO	70,350	70,350	70,350
	ICO	350	58,650	**7,792.86**
	IRO	19,850	58,500	38,475.00
	IROMN	28,900	73,750	49,821.43
	ICOHS	9,250	22,700	**12,714.29**
CUSUM	CO	570	653	621.29
	RO	1,810	10,1903	23,875.57
	ICO	1,574	56,952	27,226.64
	IRO	1,740	50,814	**26,563.07**
	IROMN	1,580	56,935	27,374.57
	ICOHS	1,661	23,994	14,239.64

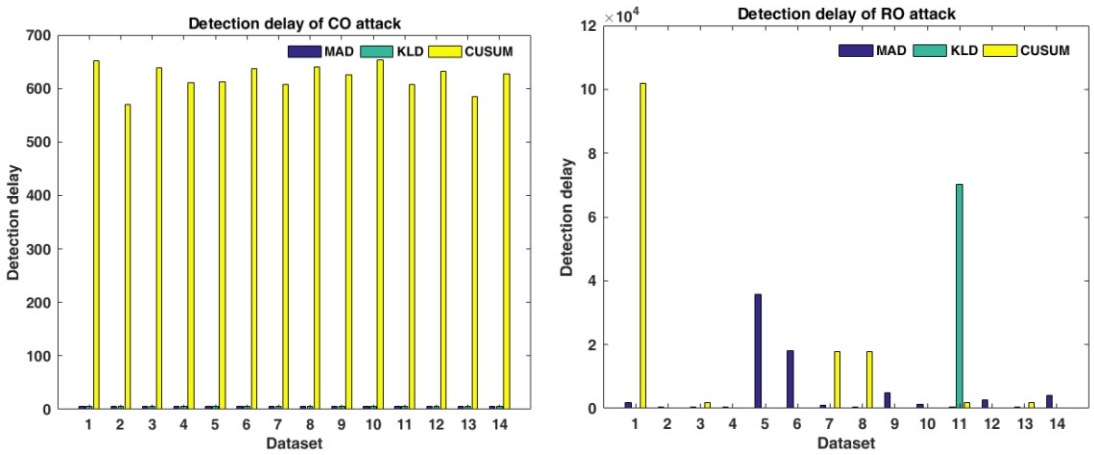
The attack detection delays shown in the Tab. 9.10 are illustrated, visualizing the attack detection for all test datasets. Figure 9.19 shows detection delay of CO and RO attacks. Sub-figure 9.19a shows the detection delay of CO attack using MAD, KLD and CUSUM. Detection delay of the CO attack using MAD and KLD is 1 in all test datasets. Therefore, bars of the detection delay using MAD and KLD are very short. From the sub-figure, we can see CUSUM has long bars as it is slower than MAD and KLD methods while

Table 9.11: Minimum, maximum and average anomaly detection delay of KLD window-based (source Paudel et al. [133]).

Methods	Attack	Detection Delay		
		min	max	average
KLD	CO	1,499	3,049	2,470.43
	RO	73,349	73,349	73,349
	ICO	66	2,091	1,185.214
	IRO	22,849	104,649	60,620.43
	IROMN	637	2,095	1,423.214
	ICOHS	12,249	46,399	28,370.43

detecting the CO attack; and able to detect only at 570 earliest and delays up to 650.

Similarly, sub-figure 9.19b shows the detection delay for RO attack. MAD detects RO type attack in 8 datasets early (between 4 and 707), a bit later in 4 dataset (between 1,455 to 4,271) and a delayed detection in 2 datasets (at 17,612 and 35,382). From the sub-figure, one can see MAD detects RO attack on second, third and fourth datasets very fast (as the bars are very short). KLD detects RO attack only on 1 dataset (eleventh dataset) with detection delay 70,350. CUSUM detects the attack only in 6 datasets; among them detection in 3 datasets are earlier (1810, 1935 and 1956) than in 2 datasets (at 17,727 and 17,923) and a delayed detection in 1 dataset (at 101,903). Thus, we can see that methods KLD and CUSUM do not have bars in many datasets; for instance, in fifth and sixth datasets.

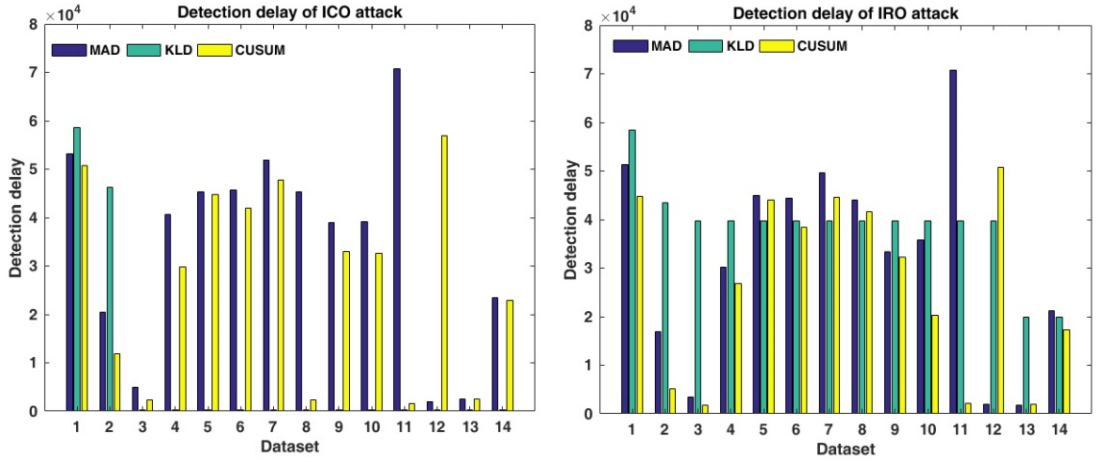


(a) CO attack (b) RO attack

Figure 9.19: Anomalies detection delay in constant offset and random offset attacks.

Figure 9.20 shows the detection delay of ICO and IRO attacks. From sub-figure 9.20a one can see MAD detects anomaly in only 3 datasets early, and delayed detection in 11 datasets. KLD detects anomalies in 12 datasets quickly and delayed in remaining 2

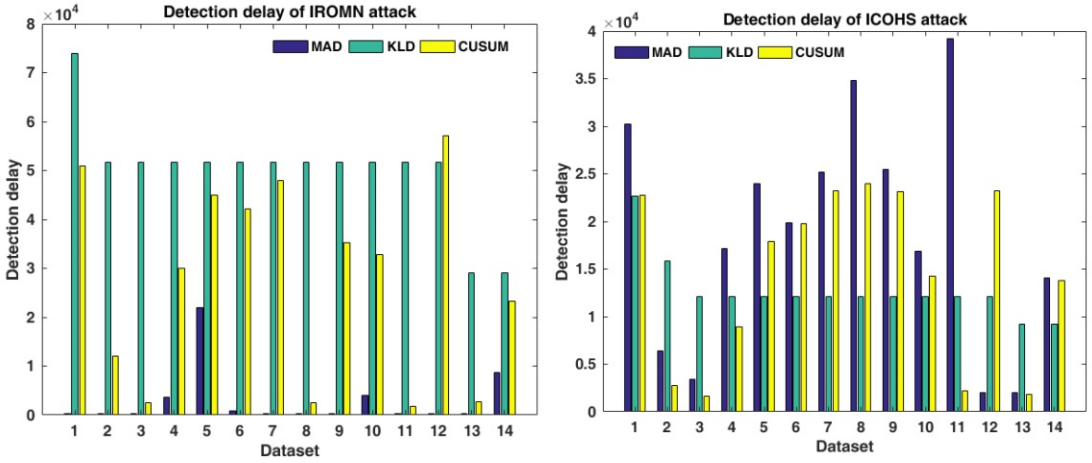
datasets. CUSUM also detects anomalies in 4 datasets early and delayed detection in 10 datasets. Figure 9.20b shows the detection delay of IRO attack on 14 test datasets. MAD detects the ICO attack in 3 datasets earlier and delayed detection in 11 datasets. Detection using KLD is delayed in all datasets. Detection using CUSUM is earlier in 4 datasets and delayed in 10 datasets.



(a) ICO attack (b) IRO attack

Figure 9.20: Anomalies detection delay in ICO and IRO attacks.

Figure 9.21 shows the detection delay in IROMN and ICOHS attack. Sub-figure 9.21a shows the detection delay of IROMN attack. MAD detects IROMN in 10 datasets very quick. KLD has delayed detection delay and has same detection delay in 51,450 in 11 datasets. CUSUM detects the attack in 4 datasets earlier and has a delayed detection in 10 datasets. Similarly, Fig. 9.21b shows the detection delay of ICOHS attack. MAD detects IROHS in 2 datasets and has delayed detection delay in 12 datasets. KLD has delayed detection in all datasets. CUSUM has detection delay in 4 datasets earlier than in 10 datasets.



(a) IROMN attack (b) ICOHS attack

Figure 9.21: Anomalies detection delay in IROMN and ICOHS attacks.

9.3.3 Detection of manipulated data points

We show the overall performance metrics that are calculated from all data points from all 14 manipulated test data sets (which always include 14 attacks of one type). Here we use original labels (i.e., BA or MA for an anomaly and normal for a normal data point) for the calculation of the performance metrics. Labels provided by the detection methods as explained in Sec. 6.5 are compared to the original labels.

Table 9.12 shows the detection performance (data points-based) for all methods and the different attack types. The table depicts minimum, maximum and average values of the performance metrics from 14 test data sets; the best results per attack are shown in bold letters. Accuracy and recall of MAD, KLD and CUSUM for 14 test datasets will be illustrated in Sec. 9.4.3. The KLD window-based detection performance is shown in Tab. 9.13.

All methods detect the CO attack well. The RO attack is better detected by CUSUM than by MAD and KLD. Similarly, the ICOHS and IROMN are better detected than the ICO and IRO attacks by all methods.

MAD has a lower false positive rate than KLD and CUSUM in all attack types. Similarly, MAD also has higher precision than KLD and CUSUM in attack types CO, ICO, IRO, IROMN and ICOHS.

For the RO attack, CUSUM has higher precision than MAD and KLD. We observed a high variation in the recall. For instance, highest recall values are seen for the data of day 2 and the lowest recall values are seen for the data of day 4. MAD detects only a few malicious data points in the manipulated signal (details are shown in Fig. A.14 in Appendix A.7) which causes the minimum value 0.01% in recall of MAD. Other methods

Table 9.12: Anomaly detection performance of different methods. The values shown are the minimum, maximum and average anomaly detection performance metrics from the 14 test data sets. Best results per attack are shown in bold letters (source Paudel et al. [133]).

Methods	Attack	Accuracy average (min/max)	Recall average (min/max)	FPR average (min/max)	Precision average (min/max)
MAD	CO	96.38% (78.61/100)%	95.03% (68.29/100)%	**0.91%** (0/12.02)%	**99.52%** (92.60/100)%
	RO	36.10% (33.23/57.66)%	4.65% (0.01/42.50)%	**0.91%** (0/12.02)%	91.09% (87.61/100)%
	ICO	72.89% (39.43/94.38)%	59.81% (15.16/91.94)%	**0.91%** (0/12.02)%	**99.20%** (71.61/100)%
	IRO	72.93% (39.40/94.39)%	59.86% (15.11/91.95)%	**0.91%** (0/12.02)%	**99.25%** (71.55/100)%
	IROMN	70.42% (40.40/90.09)%	56.11% (16.60/85.50)%	**0.91%** (0/12.02)%	**99.20%** (73.43/100)%
	ICOHS	87.37% (73.22/97.40)%	81.52% (65.84/96.10)%	**0.91%** (0/12.02)%	**99.44%** (91.64/100)%
KLD	CO	96.70% (93.90/99.64)%	99.36% (91.27/100)%	8.63% (0.76/9.95)%	95.84% (95.28/99.59)%
	RO	36.48% (30.75/49.39)%	7.94% (0/27.79)%	6.36% (0/7.44)%	71.41% (0/88.23)%
	ICO	77.08% (67.10/82.39)%	68.82% (50.88/77.29)%	6.36% (0/7.44)%	95.59% (94.76/100)%
	IRO	76.47% (67.18/86.50)%	67.91% (51.01/83.46)%	6.36% (0/7.44)%	95.53% (94.75/100)%
	IROMN	69.22% (56.46/75.94)%	57.04% (35.00/67.63)%	6.36% (0.00/7.44)%	94.72% (93.90/100)%
	ICOHS	**90.57%** (84.04/92.39)%	**89.04%** (76.18/92.29)%	6.36% (0/7.44)%	96.55% (96.04/100)%
CUSUM	CO	**97.44%** (87.19/99.66)%	**99.43%** (98.98/99.51)%	6.56% (0/37.42)%	96.81% (84.17/100)%
	RO	**54.00%** (33.23/96.67)%	**34.32%** (0/98.49)%	6.56% (0/37.42)%	**91.28%** (84.02/100)%
	ICO	**82.66%** (67.86/96.38)%	**77.28%** (52.54/98.69)%	6.56% (0/37.42)%	95.93% (84.06/100)%
	IRO	**83.02%** (69.58/97.08)%	**77.83%** (57.66/98.55)%	6.56% (0/37.42)%	95.96% (84.01/100)%
	IROMN	**82.57%** (67.85/96.37)%	**77.16%** (52.55/98.68)%	6.56% (0/37.42)%	95.92% (84.06/100)%
	ICOHS	89.87% (80.20/98.43)%	88.09% (80.01/98.62)%	6.56% (0/37.42)%	96.41% (83.99/100)%

Table 9.13: Anomaly detection performance of KLD window-based approach. The values shown are the minimum, maximum and average anomaly detection performance metrics from of the 14 test data sets (source Paudel et al. [133]).

Methods	Attack	Accuracy	Recall	FPR	Precision
KLD (window-based)	CO	87.80% (60.82/98.59)%	82.22% (42.21/97.92)%	0.19% (0/2.63)%	99.89% (97.64/100)%
	RO	32.75% (29.77/47.51)%	1.60% (0/22.58)%	0.19% (0/2.63)%	94.76% (0/100)%
	ICO	62.42% (40.93/77.46)%	45.04% (12.88/66.75)%	0.19% (0/2.63)%	99.80% (98.03/100)%
	IRO	64.93% (40.90/85.48)%	48.73% (12.83/78.58)%	0.19% (0/2.63)%	99.82% (98.10/100)%
	IROMN	57.12% (40.82/68.67)%	37.29% (12.71/54.58)%	0.19% (0/2.63)%	99.76% (97.76/100)%
	ICOHS	83.11% (73.81/93.11)%	75.35% (61.38/89.83)%	0.19% (0/2.63)%	99.88% (98.59/100)%

KLD and CUSUM do not even detect any anomalous data points in the manipulated signal (details are shown in Fig. A.29 in Appendix A.8 and in Fig. A.43 in Appendix A.9) which causes the minimum value 0% in recall of KLD and CUSUM.

CUSUM has the higher accuracy and recall than KLD and MAD in CO, RO, ICO , IRO and IROMN attack types. For the ICOHS attack, KLD-DPB has higher accuracy and recall than MAD and CUSUM.

The number of data points marked as anomaly in both KLD-DPS and KLD-WB approaches are equal. But the detection performance of KLD-WB approach differs as we count the number of windows instead of the number of data points.Total number of windows in KLD-WB approach is 3540, it has influence in the overall performance of KLD-WB approach as it results in fewer false positive rate and higher precision than KLD-DPB approach.

9.3.3.1 ROC Curves

In order to see how the detection rates change for different parameter settings, we here look at the ROC curves for the different methods. ROC curves can be used to select a threshold that would fit best to the desired outcome with respect to FPR and TPR. With the ROC curves, we also can see how a specific algorithm performs for the different attack types.

Here we discuss the detection performance of the lightweight statistical methods (MAD, KLD and CUSUM) using their ROC curves. Detection of the attack types CO, RO, ICO, IRO, IROMN and ICOHS using the lightweight statistical methods at different

thresholds are discussed in the following paragraphs. Anomaly detection results shown in the previous sections are the experimental results using the threshold, which is shown as a plus sign (+) in each method's ROC curve.

MAD Main finding: MAD has better detection performance on CO then on ICO, IRO, IROMN and IROHS attack types; and a very bad detection performance on RO attack type.

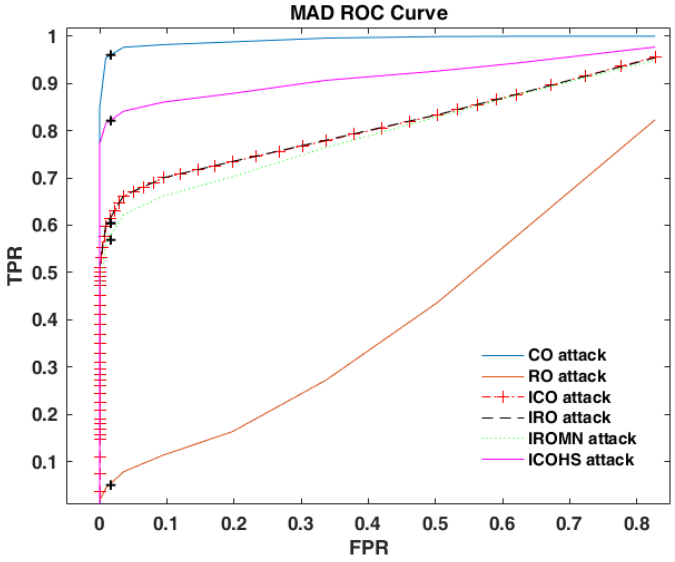


Figure 9.22: ROC curve of MAD.

Figure 9.22 shows the ROC curve for MAD. From the figure, one can see attack type CO is well detected by MAD. Detection performance on RO type attack is very bad. The slope of the ICOHS attack influences MAD's detection performances. From the ROC curves of ICO and ICOHS attacks, we can see the detection performance has significant difference due to the higher slope. Curves of ICO, and IRO attacks almost overlap but curve of IROMN attack differs slightly i.e., the influence of the random offsets depends on the magnitudes of randomization, IROMN has more noise than IRO attack. TPRs and FPRs of the attacks of same slope are very similar.

The threshold at decision level 2 which covers the minimum and maximum values in the training data has less FPR and high TPR than other thresholds. It has FPR 0.0091 and high TPR using the threshold (see Tab. 9.12). For the smaller thresholds than at decision level 2, increment in FPR is higher than increment in TPR. As expected with smaller thresholds, MAD detects more anomalies but also has higher probability of triggering false alarms, i.e high TPRs and high FPRs.

KLD (data points-based) Main finding: KLD has better detection performance on CO attack than on other attack types.

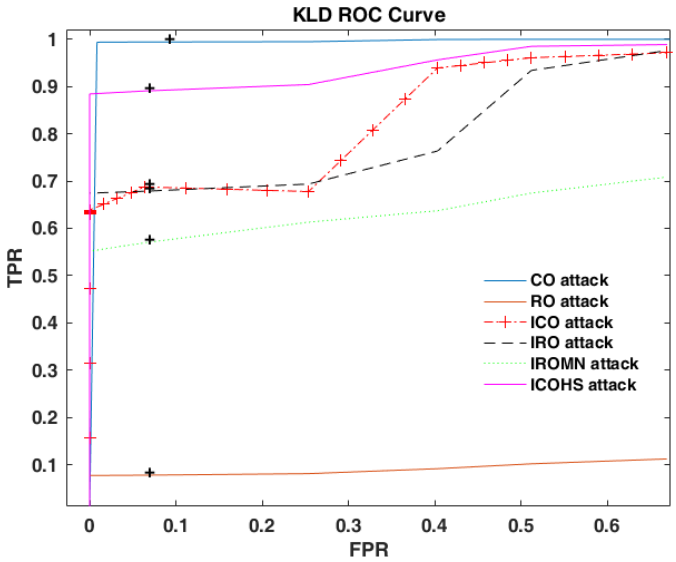


Figure 9.23: ROC curve of KLD.

Figure 9.23 shows KLD's ROC curve. From the figure, one can see KLD detects CO well, does not detect attack type RO and also does not have good detection performance on ICO, IRO, IROMN attacks. In a similar manner to MAD, anomaly detection performance of KLD is also influenced by the slope. We can see the influence from the curves of ICO and ICOHS attacks. Similarly, from the curves of ICO, IRO and IROMN attacks we can see randomization influences its detection performance. More noise reduces the detection performance (see curve of IRO and IROMN attacks).

With the threshold 8.95 at decision level 3.2, it covers the minimum and maximum divergence of the training data to the reference data. As shown in Tab. 9.12, KLD also has less FPR (8.63% for CO attack and 6.36% for RO, ICO, IRO, IROMN, ICOHS attacks) and high TPR (99.43%) using the threshold. With smaller thresholds, it only increases FPR until it reaches very small threshold. For instance, at decision level 2.5 KLD has TPR = 99.95% and FPR = 40.52%.

CUSUM Main finding: CUSUM detects attack types CO, ICO, IRO, IROMN and ICOHS. In addition, it has better detection performance than MAD and KLD on RO attack type.

Figure 9.24 shows CUSUM's ROC curve. From the figure, we can see CUSUM detects attack type CO very well and also detects RO attack type. Similar to methods MAD and KLD, slope influences CUSUM's anomaly detection performance. Curves of ICO and

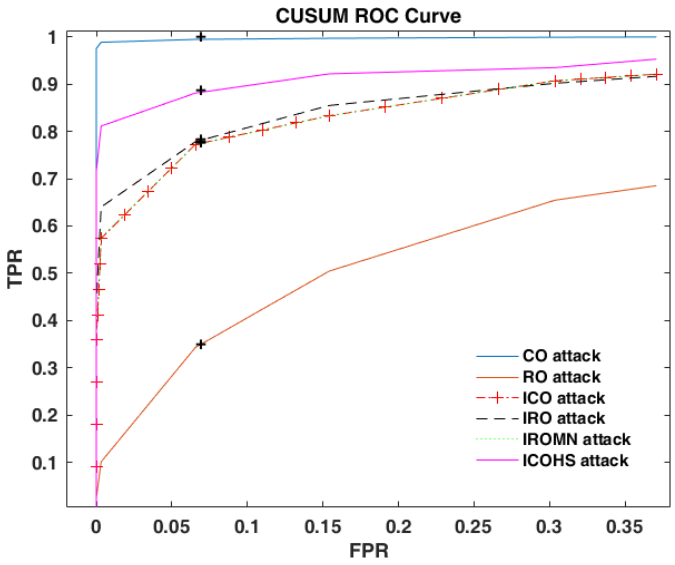


Figure 9.24: ROC curve of CUSUM.

ICOHS attacks differ significantly. From the curves of ICO, IRO and IROMN attacks, one can see noise influences the detection. IRO attack is detected better than ICO attack. Similarly, due to more noise IROMN attack is detected better than IRO attack.

Anomaly detection results of CUSUM using threshold 6.41 has high TPR and less FPR than using other thresholds. The results using the threshold are shown in Table 9.12. With the smaller thresholds, increment in FPR is higher than in TPR.

9.3.4 General Observations

After going into the details of the individual attacks, here we make some general observations:

9.3.4.1 MAD does not detect all anomalous data points

After some time MAD detects at least one malicious data point within all attacks as anomalous. Therefore, although MAD does not detect all maliciously modified values it would detect at least one malicious data point as part of the attack.

9.3.4.2 KLD window-based detection leads to many false positives

The KLD window-based approach determines whether a window of test data is anomalous. As a window is counted as anomalous when it contains at least one anomalous data

point, some of the data from the anomalous window may not be a part of the attack. To calculate the detection performance in the window-based approach, all of the data within an anomalously labelled window is marked as such. In other words if a window has 1 anomalous data point then the whole window is marked as anomalous; in this case 1 data point is a true positive and 2,999 data points are false positives. Consequently, more data points are marked as anomalous, resulting in a higher recall and false positive rates; this was confirmed by calculating the results by discriminating between normal and anomalous data within a window.

9.3.4.3 CUSUM generates a high number of false negatives

Our implementation of CUSUM processes blocks of data and resets g_n^+ and g_n^- after each block. In each block, a change point is detected only when the increase in g_n^+ or g_n^- reaches the defined threshold. An alarm is triggered (or the change is detected) only when the increase in g_n^+ or g_n^- reaches the threshold (see Fig. 9.5 in Sec. 9.1.4). Thus, the instances of the detected change point are missed until the change is detected as we continue marking anomalies after detecting the change point. Repetition of this in each block increases false negatives.

9.3.4.4 CUSUM misses isolated outliers

Despite the correct detection of nearly all malicious data points (all except for attack type ICOHS) CUSUM misses some benign anomalies. Many BAs are correctly classified by CUSUM and RB, but some BAs are missed. This is mainly due to isolated BAs with a very short duration. They are too small in duration to introduce a significant change in the mean. CUSUM would also miss any malicious increase or decrease of the voltage if it is only over a short time period because in a short time period, the changes in mean may not have a significant deviation. So only using CUSUM alone as detection method would leave many possibilities to an attacker to circumvent detection.

9.3.4.5 Methods differ significantly regarding detection delay

As can be seen in Tab. 9.10 the number of modified data points that pass undetected until a method detects the attack differs significantly for different methods and different attacks. For instance, the CO attack type is detected immediately by MAD while KLD, and CUSUM's detection is delayed.

9.3.4.6 Methods differ significantly regarding the number of detected data points

The number of detected data points in the 14 test data sets differs significantly for the different methods and the different attacks. From Tab. 9.9, we can see an attack type is

detected on the same number of test data sets and the number of detected data points are different (e.g., in ICO, IRO, IROMN).

9.3.4.7 Methods differ significantly regarding detection of RO attack type

We can see from Tab. 9.9, detection of RO attack type differs significantly for different methods. MAD detects RO attack type on 14 test data sets, whereas KLD detects only on 2 test data sets, and CUSUM detects only on 6 test data sets.

9.3.5 Results Findings

Besides the general observations already discussed in Sec. 9.3.4 we can conclude the following findings from the results:

- F 2.3.1: MAD works for the distribution of our noisy real data as it detects all attacks (at least one anomalous data point) on all test data sets. Though MAD detects all attacks, it does not work well for all data points; this might be due to our assumption of a normal distribution although the distribution of our real data is partially normally distributed.
- F 2.3.2: MAD performs well and fast for sudden increases, but fails for random offset that remains below the threshold. The detection is delayed for incremental offset attacks because at the beginning the manipulated values are very small and remain within the threshold boundaries.
- F 2.3.3: KLD (both data points-based and window-based) detects all attacks (at least one data point) on all test data sets except the RO attack. RO attack type is performed by adding values from a random normal distribution, so for KLD it seems that most of the added random values stay within the reference histogram (as the reference histogram has a broad distribution for an hour).
- F 2.3.4: KLD's (both data points-based and window-based) detection is delayed for incremental offset attacks because distribution after the manipulation stays within the limits of the reference histogram at the beginning of the attacks. But once KLD detects anomalies then it consistently detects subsequent anomalies.
- F 2.3.5: CUSUM shows a better performance than MAD and KLD for correctly identifying anomalous data points. It detects all attacks (at least one data point) on all test data sets except RO attack. In RO attack, random negative and positive values are added, which do not change the mean much (as it added $\mu = 0$ from the distribution), so it stays within the allowed variation in the mean of most of the test data.

- F 2.3.6: CUSUM's detection is delayed for incremental offset attacks because at the beginning changes in mean due to the attacks stay within the allowed variation in the mean.
- F 2.3.7: Detection delay varies a lot among methods and attack types, MAD picks up changes earlier then KLD or CUSUM. Further, results analysis shows adding a random component in the signal delays detection. For instance, incremental random offset attack detection is delayed then incremental constant offset by all methods. Thus, adding some random noise can be of advantage for attackers to hide malicious activities and prevent early detection.
- F 2.3.8: Though all of the methods have quite bad detection performance on RO attack, some malicious behavior not detected by residual-based bad data detection methods are detected by the statistical methods MAD, KLD and CUSUM (e.g., detection of incremental offsets attacks). This finding could be a valuable input to researcher using statistical methods.

As a consequence we conclude that one should not rely on a single method but instead a combination of several methods in order to prevent that an attacker can circumvent detection.

9.4 Combination of Methods

Anomaly detection performance can be improved by combining output of different methods. Different methods focus on different properties for anomaly detection and thus can detect different types of attacks as shown in Sec. 9.3. A combined application of methods can be used for better detecting anomalous behaviour in a system. In other words, one can select a type of combination method based on what is to be achieved, either generate more false alarms together with hitting more anomalous data points or hitting anomalous data points with reducing false alarms. In other words, a combination method can be selected depending on the goal, for instance achieving higher precision, detection of at least one data point per attack, higher hit rate etc. To this end, we propose using a combination technique namely, weighted voting. Output of the lightweight statistical anomaly detection methods (MAD, KLD and CUSUM) are combined using the weighted voting. In contrast to existing work (that use weighted voting for combining machine learning techniques) in smart grid domain (e.g., in [108],[101]), we use weighted voting to combine lightweight statistical anomaly detection methods.

9.4.1 Theoretical Background

Weighted voting is a technique for combination of methods. In weighted voting [45] weights are assigned for each of the methods. The weights are assigned based on their

anomaly detection performance on the test dataset. Figure 9.25 visualize steps of weighted voting method.

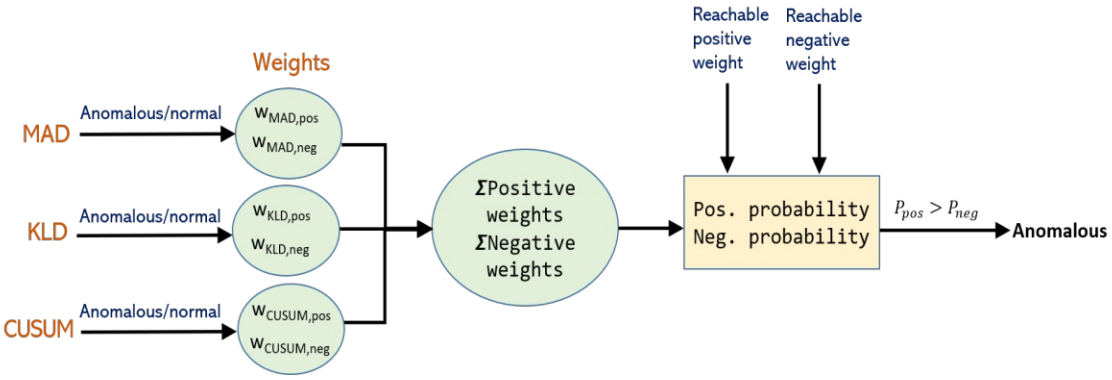


Figure 9.25: Weighted Voting Approach.

Here, we review the weighted voting method in [101]. A function is used for calculating weights for the methods. The function integrates true negative rate (TNR) and recall (true positive rate) for assigning weights for each methods. The function is expressed in Eq.(9.17) as in [101].

$$w(x) = \frac{1}{(1-x)\cdot a + b} \tag{9.17}$$

where x is a TNR or TPR (recall), a and b are control variables of the weight assignment function. Recall is used if a data point is detected as an anomalous otherwise TNR is used.

One can see from the Eq.(9.17), if value of x (TNR or recall) is closer to 1 then there would be better prediction i.e., higher values of the TNR and the recall lead to higher weights to the methods.

A smaller x results in higher $(1-x)$ value. This leads to lower weight value. Here the value of a comes in place and controls steepness of the slope. A lower performance method together with small a can result on higher weight. Thus the value of a is selected in a way that a method with better performance has higher weight than the methods with worse performance.

The variable b is used to control magnitude of the weights. Smaller values of b results in higher weights and vice versa. Thus the value of b is adjusted for controlling the maximum weight so that it avoids infinite weight value.

Using the function (represented by Eq. (9.17)), positive weight is calculated as

$$w_{pos} = \frac{1}{(1-Recall)\cdot a + b} \tag{9.18}$$

Similarly, negative weight is calculated as

$$w_{neg} = \frac{1}{(1 - TNR) \cdot a + b} \tag{9.19}$$

The reachable positive weight is the sum of positive weight of all methods. It is represented as

$$w_{all,pos} = w_{MAD,pos} + w_{KLD,pos} + w_{CUSUM,pos} \tag{9.20}$$

Similarly, reachable negative weight is the sum of the negative weight of all methods. It is represented as

$$w_{all,neg} = w_{MAD,neg} + w_{KLD,neg} + w_{CUSUM,neg} \tag{9.21}$$

The probability of being anomalous is calculated using the reachable positive weight ($w_{all,pos}$) as

$$P_{pos} = \frac{\sum w_{pos}}{w_{all,pos}} \tag{9.22}$$

where $\sum w_{pos}$ is the sum of positive weights of the methods that detect an attack type.

Similarly, probability of being negative is calculated using the sum of negative weights of the methods ($w_{all,neg}$) as

$$P_{neg} = \frac{\sum w_{neg}}{w_{all,neg}} \tag{9.23}$$

where $\sum w_{neg}$ is the sum of negative weights of the methods that do not detect an attack type.

Finally, we compare P_{pos} (probability of being anomalous) and P_{neg} (probability of being normal) to detect anomalies. If the probability of being anomalous is greater than the probability of being normal i.e. $P_{pos} > P_{neg}$ then a data point is detected as an anomaly.

Here we explain an example of anomaly detection using the weighted voting method. There may be some attack scenarios where only one method is good at detecting the type of attack. For instance, Tab. 9.14 shows sample recall and TNR for three methods. From the Tab. 9.14 , one can see that method 3 has higher recall than method 1 and method 2, and method 1 has higher TNR than method 2 and method 3.

Table 9.14: An example recall and true positive rate.

Method	**Recall**	**TNR**
Method 1	10%	95%
Method 2	20%	89%
Method 3	80%	90%

Table 9.15: Positive and negative weights for the recall and TNR shown in Tab. 9.14; Pos. = positive, Neg. = negative; see Sec. 9.4.2 for values of a and b.

Method	**Pos. weight**	**Neg. weight**
Method 1	1.10	16.66
Method 2	1.23	8.33
Method 3	4.76	9.10
Sum	7.09	34.09

For the recall and TNR shown in the Tab. 9.14, positive and negative weights derived using the formulae in equations 9.18 and 9.19 are shown in Tab. 9.15. The weights vary between 0.99 and 100.

Let us suppose only the method 3 detects a data point as an anomaly then positive weight = 4.76 and negative weight = 16.66 + 8.33 = 24.99. The probability of being an anomaly equals to $4.76/7.09 = 0.67$ and probability of being a normal data point equals to $24.99/34.09 = 0.73$. Since $0.73 > 0.67$ the data point is not detected as an anomaly.

9.4.2 Experimental Setup

For setting up an experiment, first we need to select a threshold for the methods MAD, KLD and CUSUM. Then we use the anomaly detection results using the thresholds.

Threshold selection: Thresholds could be selected in different ways. **Optimal threshold** can be selected from the analysis of ROC curves in Sec. 9.3.3.1. The methods have different detection performance using different thresholds for each attack types. For instance, a threshold can be an optimal threshold for detecting an attack type but not for detecting another attack type. From CUSUM's ROC curve shown in Fig. 9.24, we can see threshold 6.41 is an optimal threshold for ICO, IRO, IROMN and ICOHS; and a smaller threshold (3.62) is an optimal threshold for CO attack. **Min/max values of training data** could be used to define the thresholds. For the thresholds, we use the one in Sec. 7.2 in chapter 7 and Sec. 9.2 in this chapter. MAD of voltage values at decision level 2 covers the minimum and maximum voltage values of training data. Thus, we use lower bound (1.051) and upper bound (1.07) at the decision level as threshold. Similarly, MAD of distances at decision level 3.2 covers the minimum and maximum distance values of training data to the reference data. As we are looking at the divergence, we set the maximum distance (8.95) as a threshold. A threshold for CUSUM is defined using the standard deviation (σ) and maximum variation (ν) of training data combined with a small probability (0.005) of false alarm (α). The derived threshold (6.41) is set as a threshold for CUSUM. Here we have chosen a threshold for each methods that are shown as plus sign (+) in ROC curves (experimental results using the thresholds are shown in Sec. 9.3). Table 9.16 shows an overview of the thresholds for the statistical methods.

Table 9.16: An overview of thresholds for statistical methods, MAD, KLD and CUSUM.

MAD	KLD	CUSUM
DL = 2 UB = 1.070 LB = 1.051	DL = 3.2 t = 8.95	$\nu = 0.0076$ $\sigma = 0.0046$ $\alpha = 0.005$ t = 6.41

Weighted voting uses the recall and TNR of the anomaly detection performance. Table 9.17 shows recall and TNR of the anomaly detection methods for each attack types. In attack type CO, CUSUM has highest recall (0.9943) and MAD has highest TNR (0.9909). In attack type RO, recall of all methods are not that good but among the methods CUSUM has highest recall (0.3432). Similarly, CUSUM has highest recall in ICO, IRO and IROMN attacks. MAD has the highest TNR (0.9909) in all types of attacks. KLD has highest recall for attack types ICOHS.

Table 9.17: Recall and true negative rates of MAD, KLD and CUSUM.

Attack	Metrics	MAD	KLD	CUSUM
CO	Recall	95.03%	99.37%	99.43%
	TNR	99.09%	91.38%	93.44%
RO	Recall	4.65%	7.94%	34.32%
	TNR	99.09%	93.64%	93.44%
ICO	Recall	59.81%	68.82%	77.28%
	TNR	99.04%	93.64%	93.44%
IRO	Recall	59.86%	67.91%	88.09%
	TNR	99.09%	93.64%	93.44%
IROMN	Recall	56.11%	57.04%	77.16%
	TNR	99.04%	93.64%	93.44%
ICOHS	Recall	81.52%	89.04%	88.09%
	TNR	99.09%	93.64%	93.44%

Using the rates in Tab. 9.17, we calculate positive and negative weights per method for all attack types. Aiming to get higher weights for good methods, similar to work in [101], we tried different values of a and b and set values of a and b to 1 and 0.01 respectively (as they provide the expected results) in the function shown in Eq. (9.17). In other words, our intention of selecting the values of a and b is to assign higher weights (positive or negative) for the methods with good performance and increase the likelihood of a correct prediction. Thus the value of a controls the proportion of the weights and value of b avoids infinite weight value.

Table 9.18 shows derived positive and negative weights of the methods for each attack types. Since weights of the methods are based on the detection rates, weights vary for each attack types. MAD has high impact to detect malicious data point as it has

Table 9.18: Positive and negative weights of MAD, KLD, and CUSUM.

Attack	Weight	MAD	KLD	CUSUM	Sum
CO	Positive	16.75	59.88	63.69	140.32
	Negative	52.36	10.40	13.23	75.98
RO	Positive	1.04	1.07	1.50	3.61
	Negative	52.36	13.59	13.23	79.17
ICO	Positive	2.43	3.11	4.22	9.75
	Negative	51.02	13.59	13.23	77.83
IRO	Positive	2.43	3.02	4.32	9.77
	Negative	52.36	13.59	13.23	79.17
IROMN	Positive	2.23	2.27	4.19	8.70
	Negative	52.08	13.59	13.23	78.90
ICOHS	Positive	5.13	8.36	7.75	21.24
	Negative	52.36	13.59	13.23	79.17

high negative weights in all attack types. Last column in the table 9.18 shows the sum of all positive or negative weights for each of the attacks. It represents the maximum possible negative and positive weights for each attack types while combining the methods. The reachable positive weight (shown in Eq. (9.20)) could be achieved only if all of the methods detect a data point as an anomaly at same time. Similarly, the reachable negative weight (shown in Eq. (9.21)) could be achieved only if all of the methods miss an anomaly at same time.

The positive weights and negative weights based on methods' anomaly detection results help to predict whether a data point is anomalous or normal. The prediction is done by comparing the probability of being anomalous or normal.

Table 9.19: Overview of weighted voting, parameter setting and injected attacks; Exp. = experiment; CO = constant offset; RO = random offset; ICO = incremental constant offset; IRO = incremental random offset; IROMN = incremental random offset with more noise; ICOHS = incremental constant offset with high slope; Sec = section.

Exp.	Method	Data[‡]	Param. setting	Injected attacks	Sec.
9.4	Weighted voting	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016 - 14.04.2016)	Weights (see Tab. 9.18) of each methods for each attack types	CO, RO ICO, IRO IROMN, ICOHS	9.4.3

[*] For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

[1]For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

Table 9.19 shows an overview of weighted voting, parameter settings and injected attacks. We recall that the TPR (recall) and TNR of the methods are used to assign weights for the methods.

9.4.3 Results

Here we describe our experimental results in detail from experiment 9.4. We show overall detection performance metrics calculated from all 14 manipulated test data sets.

Table 9.20 shows results of the weighted voting method. The values shown in the table are the average, minimum and maximum anomaly detection performance metrics from the 14 test data sets. Table 9.21 shows how fast the attacks are detected after combining the results using the combination method.

Similarly, Tab. 9.22 shows the average anomaly detection performance of the individual methods and the combination method (weighted voting) from the 14 test data sets. From the table, one can directly compare the anomaly detection of the individual methods to the combined results. Further, looking at figures in sections 9.3.1 and 9.4.3.2, we can see that only MAD detects benign anomalies before starting attacks therefore weighted voting misses the benign anomalies in most of the cases. Further, in some cases (e.g., in CO, IROMN attacks) the individual methods detect anomalies in different locations but weighted voting consistently detects anomalies once it detects an attack which produces more precise results than the individual methods as we can see in the Tab. 9.22.

Table 9.20: Minimum, maximum and average anomaly detection performance metrics of the weighted voting method from the 14 test data sets.

Methods	**Attack**	**Accuracy average (min/max)**	**Recall average (min/max)**	**FPR average (min/max)**	**Precision average (min/max)**
Weighted Voting	CO	99.48% (96.66/100)%	99.95% (99.46/100)%	1.46% (0/10.02)%	99.28% (95.23/100)%
	RO	37.27% (33.23/60.41)%	6.45% (0/44.74)%	0.99% (0/8.27)%	92.89% (52.33/100)%
	ICO	78.72% (68.60/94.19)%	68.60% (52.89/92.49)%	0.99% (0/8.27)%	99.29% (94.34/100)%
	IRO	79.01% (69.80/93.82)%	69.03% (54.92/91.95)%	0.99% (0/8.27)%	99.29% (94.18/100)%
	IROMN	75.19% (67.87/90.16)%	63.30% (52.03/86.46)%	0.99% (0/8.27)%	99.23% (93.38/100)%
	ICOHS	90.46% (84.51/97.21)%	86.19% (76.88/96.42)%	0.99% (0/8.27)%	99.43% (95.61/100)%

Table 9.21: Minimum, maximum and average anomaly detection delay of the weighted voting method.

Methods	Attack	Detection Delay		
		min	max	average
Weighted Voting	CO	1	1	**1.00**
	RO	1,825	101,905	35,647.38
	ICO	1,574	53,170	24,943.57
	IRO	2,048	51,333	32,387.07
	IROMN	1,580	51,459	26,990.50
	IROHS	2,044	23,994	15,568.50

Table 9.22: Average anomaly detection performance of the individual and the weighted voting methods from the 14 test data sets. Best results per attack are shown in bold letters.

Methods	Attack	Accuracy	Recall	FPR	Precision
MAD	CO	96.38%	95.03%	**0.91%**	**99.52%**
	RO	36.10%	4.65%	**0.91%**	91.09%
	ICO	72.89%	59.81%	**0.91%**	99.20%
	IRO	72.93%	59.86%	**0.91%**	99.25%
	IROMN	70.42%	56.11%	**0.91%**	99.20%
	ICOHS	87.37%	81.52%	**0.91%**	**99.44%**
KLD	CO	96.70%	99.36%	8.63%	95.84%
	RO	36.48%	7.94%	6.36%	71.41%
	ICO	77.08%	68.82%	6.36%	95.59%
	IRO	76.47%	67.91%	6.36%	95.53%
	IROMN	69.22%	57.04%	6.36%	94.72%
	ICOHS	**90.57%**	**89.04%**	6.36%	96.55%
CUSUM	CO	97.44%	99.43%	6.56%	96.81%
	RO	54.00%	**34.32%**	6.56%	91.28%
	ICO	82.66%	**77.28%**	6.56%	95.93%
	IRO	**83.02%**	**77.83%**	6.56%	95.96%
	IROMN	**82.57%**	**77.16%**	6.56%	95.92%
	ICOHS	89.87%	88.09%	6.56%	96.41%
Weighted Voting	CO	**99.48%**	**99.95%**	1.46%	99.28%
	RO	37.27%	6.45%	0.99%	**92.89%**
	ICO	78.72%	68.60%	0.99%	**99.29%**
	IRO	79.01%	69.03%	0.99%	**99.29%**
	IROMN	75.19%	63.30%	0.99%	**99.23%**
	ICOHS	90.46%	86.19%	0.99%	99.43%

Table 9.23: Detected data points using weighted voting scheme (rounded average values are shown for the detected data points).

Method	Attacks (average detected data points)					
	CO	RO	ICO	IRO	IROMN	ICOHS
Weighted Voting	14 (119,995)	8 (7,742)	14 (82,348)	14 (82,869)	14 (75,992)	14 (103,478)

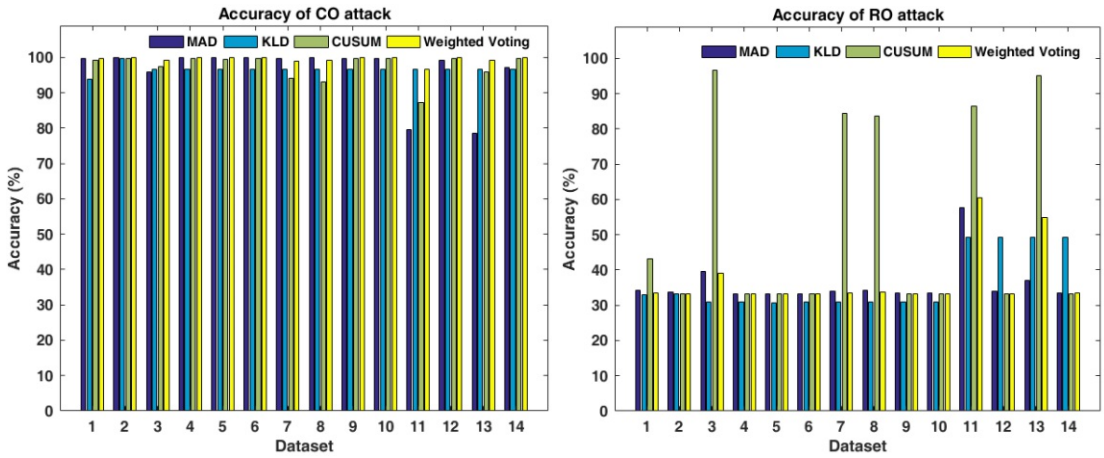
Table 9.23 shows the different detected attack types using weighted voting scheme. The table depicts the number of test datasets the attacks are detected, and the average detected data points on 14 test data sets. Malicious and benign data points on the 14 test data sets are shown in Tab. 6.3 of Chapter 6. Detected data points using weighted voting on each data sets are shown in Tab. A.2 of Appendix A.10.

Table 9.22 shows that weighted voting has higher precision in most of the attacks (RO, ICO, IRO and IROMN). In the following paragraphs, we illustrate the accuracy and recall of methods MAD, KLD, CUSUM and weighted voting shown in the Tab. 9.22 for all test datasets. The minimum and maximum values of the accuracy and recall for weighted voting is shown in Tab. 9.20 and for MAD, KLD and CUSUM are shown in Tab. 9.12 in Sec. 9.3.3.

Accuracy: Accuracy of methods MAD, KLD, CUSUM and weighted voting are illustrated from Fig. 9.26 to 9.28. Figure 9.26 shows accuracy of CO and RO attacks. Sub-figure 9.26a shows the accuracy of CO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.26a, we can see MAD has minimum accuracy which is 78.61% for dataset 11 (as shown in Tab. 9.12), and the accuracy varies up to 100%. Further from the sub-figure 9.26a, one can see that minimum accuracy of KLD is above 80% (for dataset 11) but does not reach 100%. In all of the datasets, weighted voting has higher accuracy than KLD and CUSUM. In some datasets (e.g., datasets 1, 4) weighted voting and MAD have equal accuracy (100%) but in most of the datasets weighted voting has higher accuracy.

Similarly, sub-figure 9.26b shows the accuracy for RO attack. As we know that detection performance of RO attack is not good, from the sub-figure 9.26b one can see MAD has accuracy between 33.23% and 40% in 13 datasets and maximum accuracy is 57.66% in a dataset. KLD has accuracy between 30.75% and 35% in 10 datasets and 49.39% in 4 datasets. CUSUM has more accuracy than MAD, KLD and weighted voting in 6 datasets and varies up to 96.67%. Weighted voting has higher accuracy than MAD and KLD in 2 datasets. In 7 datasets weighted voting and CUSUM have equal accuracy.

In datasets 3, 7, 8, 11 and 13 though CUSUM has higher accuracy weighted voting does not have high accuracy as CUSUM. This can be explained using recall, TNR, weights and probability of being an anomaly. We calculate the probability of being an anomaly using the weights shown in the Tab. 9.18 in Sec. 9.4.2. The weights are calculated using the recall and TNR shown in Tab. 9.17. From the Tab. 9.17, one can see that for RO attack



(a) CO attack (b) RO attack

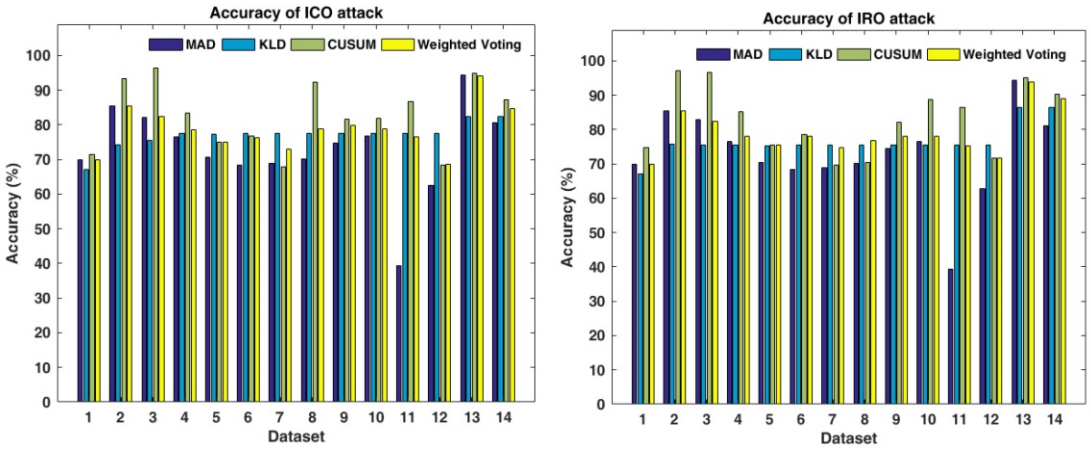
Figure 9.26: Accuracy in constant offset and random offset attacks.

i) recall of CUSUM is higher than recalls of MAD and KLD, and ii) TNR of MAD is higher than KLD and CUSUM, and KLD and CUSUM have almost equal TNR. During RO attack if only CUSUM detects a data point as an anomaly then positive weight is calculated using CUSUM's recall whereas negative weight is calculated using TNRs of MAD and KLD which results in a higher negative weight. Therefore, weighted voting does not detect the data point as an anomaly. For instance, if MAD and KLD detect a data point as normal; and CUSUM detects the data point as an anomaly. Then total positive weight = 1.50, total negative weight = 52.36+13.59 = 65.94, positive probability = 1.50/3.61 = 0.41 and negative probability = 65.94/79.17 = 0.83. In this case (RO attack), as negative probability is greater than positive probability (0.83 > 0.41) weighted voting detects the data point as normal.

Detection due to the changes in the original signals after starting an attack results in CUSUM's higher accuracy but if there is no attack then such changes in the signal cause a lot of false alarms.

Figure 9.27 shows accuracy in ICO and IRO attacks. Sub-figure 9.27a shows the accuracy of ICO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.27a, one can see in most of the datasets accuracy of the methods are between 70% and 80%. CUSUM has higher accuracy than other methods in 9 datasets and weighted voting also has higher accuracy than MAD and KLD in 6 datasets.

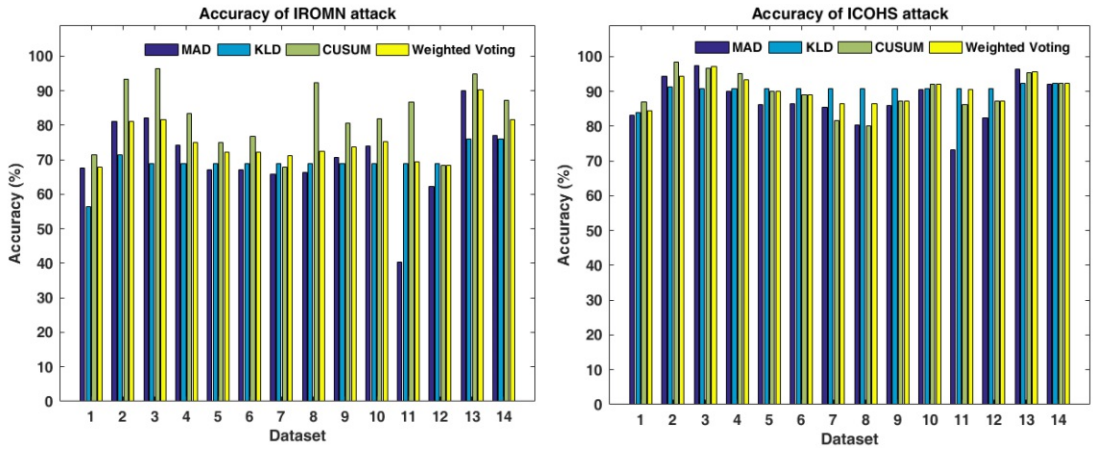
Sub-figure 9.27b shows the accuracy of IRO attack using MAD, KLD, CUSUM and weighted voting. Similar to ICO attack, for many datasets accuracy of the methods are between 70% and 80%. From the sub-figure, one can see MAD is better than KLD in 6 datasets and KLD is better than MAD in 8 datasets. Here also CUSUM has more accuracy than other methods in 10 datasets and weighted voting has more accuracy in 8 datasets.



(a) ICO attack (b) IRO attack

Figure 9.27: Accuracy in ICO and IRO attacks.

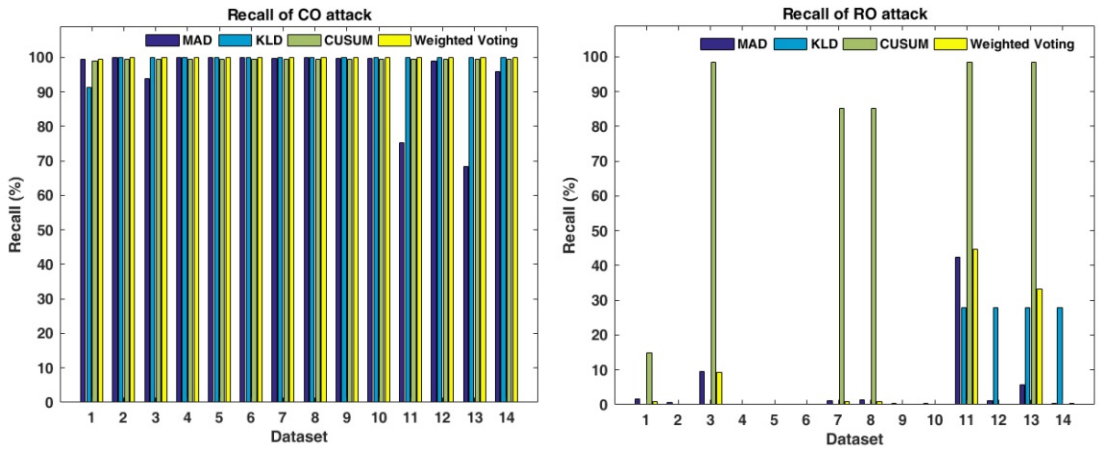
Figure 9.28 shows the frequency of accuracy in IROMN and ICOHS attacks. Sub-figure 9.28a shows the accuracy of IROMN attack using MAD, KLD, CUSUM and weighted voting. Similar to ICO and IRO the accuracy of the methods on many datasets are around 80%. CUSUM has more accuracy than other methods in 12 datasets and has highest accuracy 96.39%. Another sub-figure 9.28b shows the accuracy of ICOHS attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.28b, one can see that the methods have accuracy between 80% and 90%. It means accuracy of ICOHS is higher then IROMN in all datasets, one can also see by comparing sub-figure sub-figure 9.28a and 9.28b).



(a) IROMN attack (b) ICOHS attack

Figure 9.28: Accuracy in IROMN and ICOHS attacks.

Recall: Recall of methods MAD, KLD, CUSUM and weighted voting are illustrated from Fig. 9.29 to 9.31. Figure 9.29 shows the recall in CO and RO attacks. Sub-figure 9.29a shows the recall of CO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.29a, one can see MAD has recall 68.29% in dataset 13 and 75.30% in dataset 11, except these all methods have recall almost 100% in rest of the datasets. Sub-figure 9.29b shows the recall of RO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.29b, one can see that RO attack is not detected in most of the datasets and if detected then CUSUM has higher recall than other methods. CUSUM has 98% recall in 3 datasets (datasets 3, 11, 13). This is due to abnormal behaviour (significant changes) in the original signals (see figures 6.10, 6.11, 6.12 in Sec. 6.4). CUSUM detects such changes in the original signals, further if the changes affect the distribution, then MAD and KLD also detect the changes. Therefore, the changes after starting an attack results in higher recall in 5 datasets.

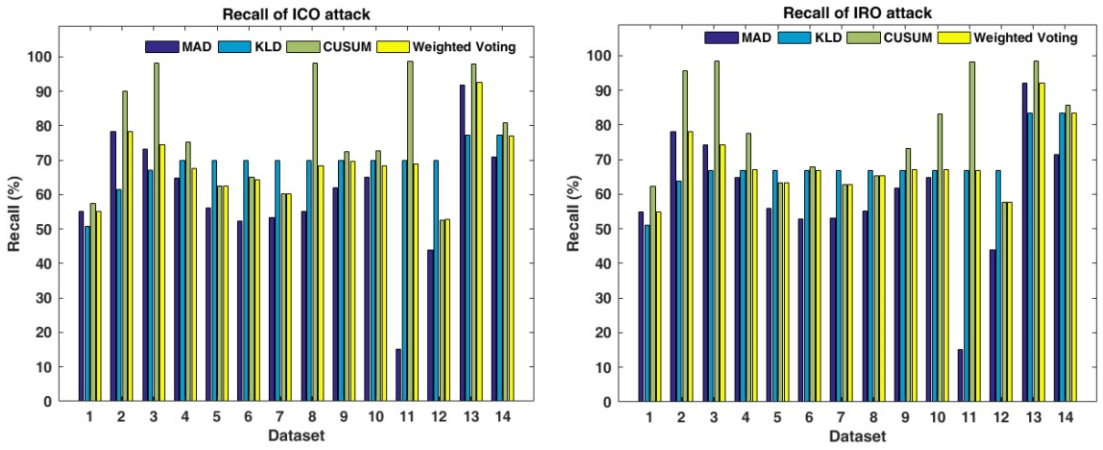


(a) CO attack (b) RO attack

Figure 9.29: Recall in constant offset and random offset attacks.

Figure 9.30 shows the frequency of recall in ICO and IRO attacks. Sub-figure 9.30a shows the recall of ICO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.30a, one can see that majority of recall is around 65%. KLD has higher recall than other methods in 4 datasets. CUSUM has higher recall than other methods in 10 datasets. Sub-figure 9.30b shows the recall of IRO attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure 9.30b, one can see that recall of the methods in IRO are similar to ICO attack.

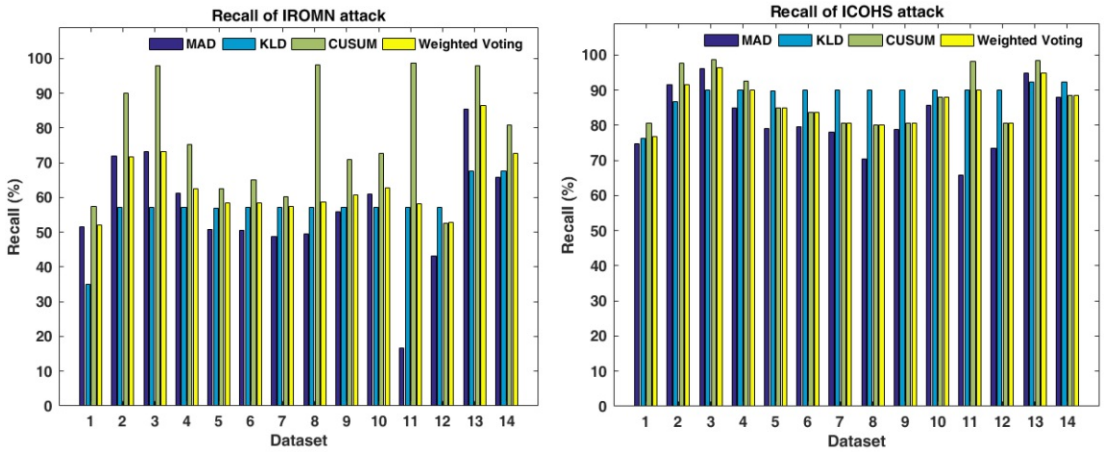
Figure 9.31 shows the recall in ICO and IRO attacks. Sub-figure 9.31a shows the recall of IROMN attack using MAD, KLD, CUSUM and weighted voting. From the sub-figure, one can see that CUSUM has higher recall than other methods in most of the datasets and weighted voting has higher recall than MAD and KLD in 9 datasets. Sub-figure 9.31b shows the recall of ICOHS attack using MAD, KLD, CUSUM and weighted voting. The sub-figure shows that recall of ICOHS attack is higher than IROMN attack and



(a) ICO attack (b) IRO attack

Figure 9.30: Recall in ICO and IRO attacks.

majority is around 85%. KLD has higher recall than other methods in 8 datasets and CUSUM has higher recall than other methods in 6 datasets.



(a) IROMN attack (b) ICOHS attack

Figure 9.31: Recall in IROMN and ICOHS attacks.

9.4.3.1 General observations

Here we make some general observations on anomaly detection performance of the combination methods.

Weighted voting decreases false alarm Weighted voting triggers an alarm if and only if an anomaly is detected with higher probability of being an anomalous than not being an anomalous. The probability is calculated based on the anomaly detection results from multiple methods. More than one method can detect anomalies in the same location due to extreme outliers or due to long term abnormal behavior in the signal. Thus weighted voting generates more trustworthy results than the KLD and CUSUM (see Tab. 9.22) as it triggers fewer false alarms (only 0.99%). Anomaly detection performance of weighted voting shows that it ends up with more precise detection results than the individual methods.

Weighted voting detects an attack with a single method The weights of each method for detecting anomalous data points in attack types are different. In some cases, even with one method, the positive probability for detecting a data point as anomalous is higher than the negative probability using negative weights of more methods. In such cases, even with a single method, weighted voting surprisingly may detect the data point as anomalous. For instance, in attack type IRO, weighted voting detects anomalous data point between data point 7,093 and 7,815, even if the data points are detected as anomalous by only one method MAD.

Weighted voting detects at least some data points in all attack types The attack type RO is detected by MAD and KLD with low recall but CUSUM detects RO with higher recall. The weighted voting method detects the attack type RO with less recall but in more datasets than the individual methods (see tables 9.9 and 9.23). Overall results show the combination method detects all types of attacks.

Weighted voting is slower in detection In general, weighted voting has detection delay larger than one of the methods for detecting the attacks (see tables 9.21 and 9.10) but produces trustworthy results, as it triggers less false alarms (see Tab. 9.22). By comparing the figures in Sec. 9.3.1 to 9.4.3.2, one can see the detection of weighted voting is delayed than the detection of one of the statistical methods.

9.4.3.2 Detailed Detection Results per Attack

Constant offset attack Weighted voting method detects attack type CO immediately. Figure 9.32 visualizes detected anomalies using weighted voting method in constant offset attack.

In the first test data, it does not generate any false alarms and also misses BAs between 7,093 and 7,815 data points. It has higher precision 99.28% than KLD and CUSUM, and it also detects CO attack on first malicious data point. Further, it has higher recall than the individual methods (see Tab. 9.22).

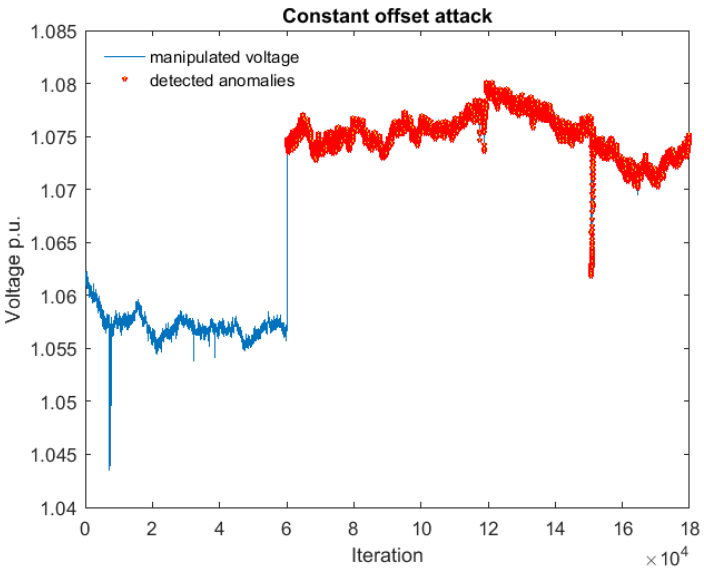


Figure 9.32: Visualization of detected anomalies in constant offset attack (shown on April 01, 02:00-03:00).

Random offset attack RO type attack is a challenging attack, but the weighted voting method detects at least some data point as anomalous in the attack type RO. Detection with weighted voting is delayed because weighted voting detected it only when CUSUM detects the RO attack although MAD detects the attack earlier (see sub-section 9.3.1.2). Figure 9.33 shows detected anomalies in random offset attack.

The Variation in recall, FPR and precision is high in RO attack. These can be explained. As only MAD detects the attack, weighted voting does not detect any points as anomalous on the data set of day 3 so that they have minimum value 0.00% in recall and in FPR (see Tables 9.12 and 9.20).

From the Fig. 9.33, one can see weighted voting does not detect BAs data points 7093 and 7815 in this attack type (on first test data) and detects malicious data points with delay. MAD method and CUSUM cause the detection of malicious data points. Detection delay of the voting method vary from 1,825 to 101,905. MAD starts detecting some data points during the attack at 4^{th} data point after that attack started and CUSUM starts detecting the attack at data point 1,810. And these two methods detect the attack at $1,825^{th}$ data point after the attack started, it leads weighted voting detects earliest at this data point.

Incremental constant offset attack Weighted voting method detects attack type ICO with delay. Figure 9.34 shows detected anomalies in incremental constant offset attack.

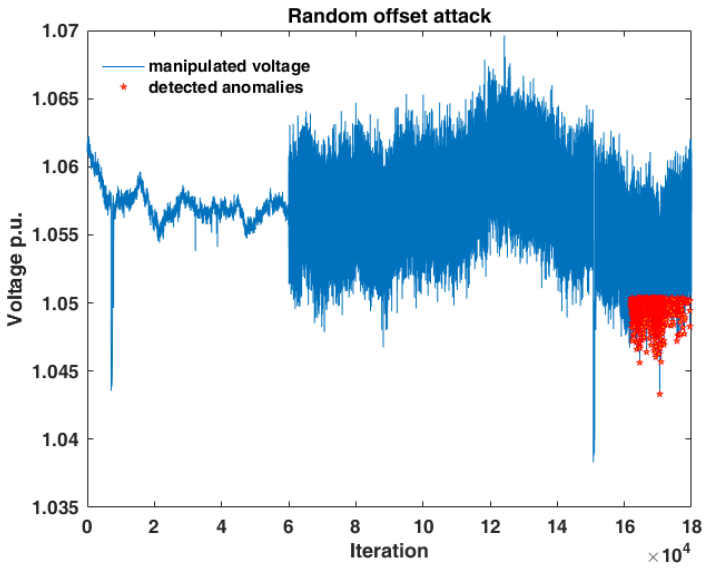


Figure 9.33: Visualization of detected anomalies in random offset attack (shown on April 01, 02:00-03:00).

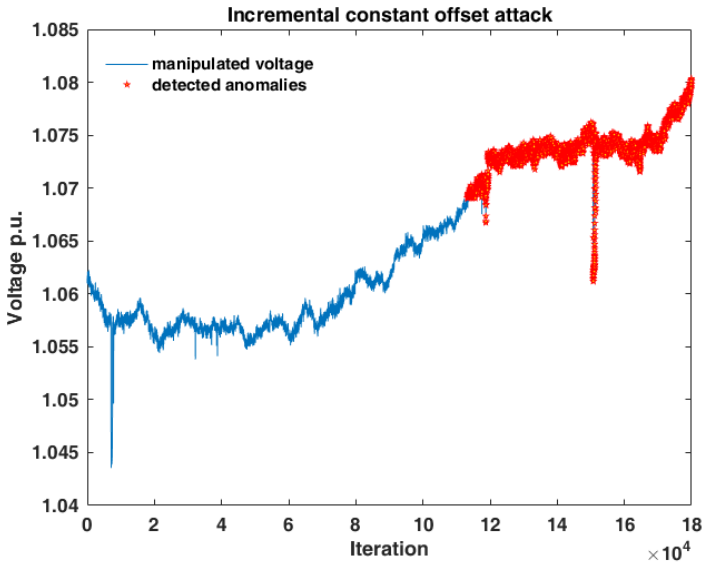


Figure 9.34: Visualization of detected anomalies in incremental constant offset attack (shown on April 01, 02:00-03:00).

The detection by weighted voting delays because only the positive weights of KLD is not enough, and waits until CUSUM detects the attack. Weighted voting has 99.29% precision which is higher than single methods. The detection delay of anomalies varies

from 1,574 to 53,170. KLD starts detecting some data points during the attack at 350^{th} data point while CUSUM starts detecting the attack only at data point 1,574. Therefore, weighted voting detects the attack at data point 1,574.

Incremental random offset attack Attack type IRO is also detected by the voting method. Figure 9.35 shows detected anomalies using the voting method in incremental random offset attack.

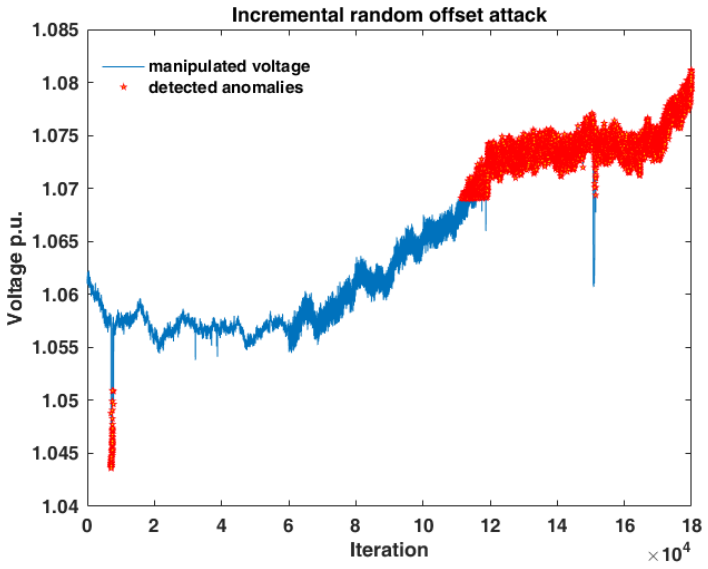


Figure 9.35: Visualization of detected anomalies in incremental random offset attack (shown on April 01, 02:00-03:00).

Regarding BAs between 7,093 and 7,815 data points in IRO attack type, only MAD detects the BAs. From the table 9.18, we can see for the attack type IRO positive weight is 2.4307 out of possible weight 9.7687 and negative weight is 26.8145 out of possible weight 79.1705. From the weights, probability of being an anomalous (25%) is higher than the probability of not being an anomalous (22%), and thus the weighted voting detects the BAs. Regarding the detection speed, weighted voting detects the attack earliest at data point 2,048. It is due the detection by CUSUM as MAD already starts detecting the attack at data point 1,847.

Incremental random offset attack with more noise Figure 9.36 shows detected anomalies in incremental random offset attack. The weighted voting method detects attack type IROMN. Detection delay of weighted voting varies from 1,580 to 51,459 whereas simple combination detects 1^{st} malicious data point and varies to 21,824 depending on the signal. Methods MAD and CUSUM detect IROMN attack at data point 1,580,

therefore the weighted voting detects the IROMN attack earliest at the data point. At the beginning of detection, it detects the data points only on the upper side of the signal. The detection in upper side is due to the detection of MAD (see Fig. 9.15 in Sec. 9.3.1). After some time when CUSUM starts detecting the attack, weighted voting detects the attack on both side (upper and lower side) of the signal (from data point 138,000).

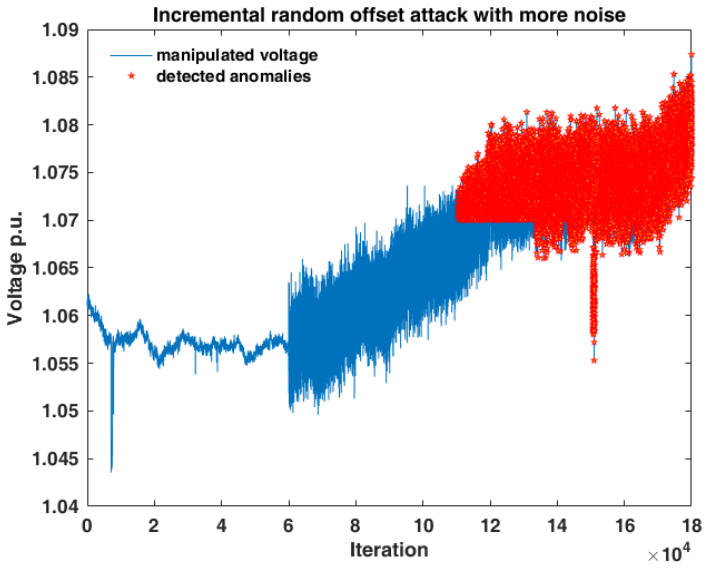


Figure 9.36: Visualization of detected anomalies in incremental random offset attack with more noise (shown on April 01, 02:00-03:00).

Incremental constant offset attack with high slope The weighted voting method detects attack type incremental constant offset attack with high slope. Figure 9.37 shows detected anomalies in incremental constant offset attack. Weighted voting has more precise results (99.43%) than KLD (96.55%), CUSUM (96.41%) methods (93.48%).

One can see that even with higher slope the detection is delayed up to the data point 13,994. The earliest detection of this type of attack by weighted voting is at data point 2,044 which is caused due to the detection of MAD and CUSUM.

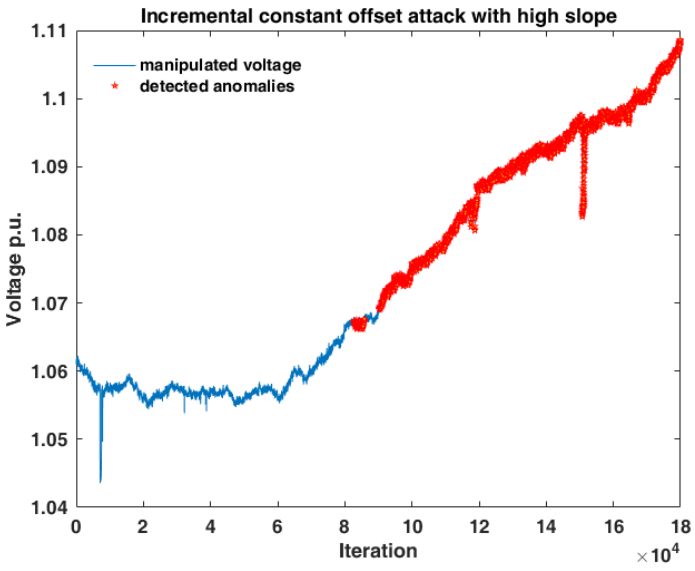


Figure 9.37: Visualization of detected anomalies in incremental constant offset attack with high slope (shown on April 01, 02:00-03:00).

9.4.3.3 Results Findings

- F 2.3.9: The combination method weighted voting works for the combination of statistical methods. It detects at least one of the anomalous data points in all attack types, so alarms are triggered for all attacks such that an operator in a power system is informed and an appropriate protection action can take in place.
- F 2.3.10: The weighted voting method detects at least some anomalous data points in the attack type RO, and also detects anomalous data points on more test data sets than the KLD and CUSUM methods. This evidence confirms that a combination method improves anomaly detection performance.
- F 2.3.11: Weighted voting generates less false alarms than the MAD and KLD methods. So it provides better precision than individual methods for most of the attack types. Thus, this evidence confirms that the combined results are more trustworthy than the results of individual methods, so that an operator in a control center could trust the results.

Table 9.24 shows an overview of attack detection using methods, MAD, KLD, CUSUM and weighted voting. The sign ● represents an attack is detected immediately, sign ◑ represents delayed detection, and the number in brackets represents the number of datasets for which an attack was detected. Here an attack is said to be detected immediately if an attack is detected within the first 100 malicious data points, otherwise

it is said to be delayed detection. From the Tab. 9.24, one can see that although weighted voting has delayed detection, it detects all attack types in more datasets than the methods KLD and CUSUM. From the analysis, we conclude that one should use combination method in order to detect at least some data points on all types of attacks.

Table 9.24: An overview of attack detection using methods MAD, KLD, CUSUM, and weighted voting. Sign ● represents an attack is detected immediately, and sign ◑ represents delayed detection; an attack is meant to be detected immediately if an attack is detected with in 100 data points.

Attacks	Weighted Voting	MAD	KLD	CUSUM
CO	●(14)	●(14)	●(14)	◑(14)
RO	◑(8)	●(6), ◑(8)	◑(1)	◑(6)
ICO	◑(14)	●(3) ◑(11)	◑(14)	◑(14)
IRO	◑(14)	◑(14)	◑(14)	◑(14)
IROMN	◑(14)	●(4), ◑(7)	◑(14)	◑(14)
ICOHS	◑(14)	◑(14)	◑(14)	◑(14)

9.5 Summary

In this chapter, we presented our model for detecting anomalies. The model was designed to implement different anomaly detecting methods.

We proposed using lightweight statistical methods for anomaly detection. For this we have chosen three different methods 1) a simple threshold based on the MAD, 2) a histogram (distribution) based approach using the KLD with a sliding window and 3) the CUSUM as a representative of a change point detection method.

Methods for defining thresholds for anomaly detection were presented for the lightweight statistical methods and some details on calculating parameters of the methods. Further, influence of method-parameters on anomaly detection performance was discussed for each of the methods.

Then we presented the results of the lightweight statistical methods; MAD, KLD and CUSUM. The results showed that many attacks not detected by the residual-based methods (see Chapter 7) are detected by the lightweight statistical methods but as we fixed the phase angle by the first phase angle, the data differs in the residual-based and the lightweight statistical methods. And therefore cannot directly be compared. Further, performance metrics and detection speed were presented for each attacks types. Visualization of the detected anomalies on attack types provided details on anomaly detection.

Additionally, we provided the ROC curves of the statistical methods (MAD, KLD and CUSUM). To this end, the analysis of the attack detection with different thresholds for

each method was presented.

The following major results findings supported answering the research questions **RQ 2.3.1** (Is it possible to detect at least one of the injected malicious anomalies during our injected attacks with the lightweight statistical methods?), **RQ 2.3.2** (How long do the methods take until the first malicious anomaly during the attack is detected?) and **RQ 2.3.3** (How many of the malicious anomalies are detected?):

- The lightweight statistical methods detected some malicious behaviour that were not detected by the residual-based methods in Chapter 7 but as we fixed the phase angle the data differs in the residual-based and the lightweight statistical methods. At least one anomalous data point was detected in each attack type so that alarms were triggered for all attacks.
- Different methods detected different attack types, and the detection delay varied a lot among methods and the attack types.
- Almost all malicious anomalies during an attack were detected if the anomaly detection was fast as we assumed an attack is detected if the first anomalous data point is detected.

As we expected for our reasoning **RQ 2.3.1**, the evidence from the experiment showed that we could detect at least one malicious data point during all attacks with the lightweight statistical methods. Additionally, the evidence from the experiment showed that depending on the attack types and characteristics of the methods, the anomaly detection could be achieved immediately or be delayed. For instance, MAD detects RO attack type (in case of large offset) immediately, using CUSUM it is expected a delayed detection and using KLD it is not expected to be detected.

As we expected for our reasoning **RQ 2.3.2** the experimental evidence showed that i) attacks that caused significance change in the statistical properties (e.g., CO attack) are detected earlier and ii) attacks that caused slow changes in the statistical properties (e.g., ICO, IRO attacks) were detected with delay. Further, as we already expected for reasoning **RQ 2.3.3**, nearly all malicious anomalies were detected for the early detected attacks.

We argued that multiple methods should be used together via a combination method to prevent that attackers can circumvent detection and propose combining results from the different anomaly detection methods.

We proposed using weighted voting scheme for methods combination. Experimental setup for assigning weights to the methods were presented based on their detection performance. Then we combined results from different anomaly detection methods using the weighted voting method. An analysis of the combined results showed the combination method enhanced anomaly detection performance by detecting at least some anomalous data points on all types of false data injection attack introduced in Sec. 5.3. Combined results

using weighted voting method were more precise than the individual methods (MAD, KLD and CUSUM) methods in most of the attack types. The following major results finding supported answering the research question **RQ 2.3.4** (To which extent does detection performance improve if we combine lightweight statistical methods?):

- The combination method weighted voting enhanced anomaly detection performance as it detected all attacks (at least some anomalous data points); and the results had higher precision than the individual methods as it triggered less false alarms. Thus, the combined results were more trustworthy than the results of the individual methods.

As we expected for our reasoning **RQ 2.3.4**, the evidence from the experiment showed that i) different methods had their own strengths and weaknesses, when detecting different types of attacks, and ii) the combination of results of the lightweight statistical methods improved the anomaly detection performance of the injected false data injection attacks on the test data sets.

CHAPTER 10

Mitigating the Effects of Attacks on State Estimation

Notice of adoption from previous publications in Chapter 10
The contents of this chapter have not been published so far.

In this chapter, we give an overview of how inconsistent states can be recovered using previous consistent (normal) states and propose an approach for maintaining correctness of state estimation. First we present our approach, second, we show how state estimation is affected by attacks, third how the effects of attacks on state estimation can be mitigated with the proposed approach. And then we show experimental results and discuss how the effects of attacks on the state estimation is mitigated using the approach.

Consistency of a system state can be analyzed based on the data collected and the functionalities of a system. For instance, a state that meets normal behavior of a system can be called a consistent or a normal state. Due to faults or attacks, a system can be caused to be inconsistent with other states. Here we aim at recovering from inconsistent states.

Figure 10.1 depicts an overview of a process recovering inconsistent state using information from a consistent state. We make an assumption that a state at time step k is estimated based on the actual (original) measurements. The left part of the figure shows an inconsistent state caused due to faults or attacks, and right part of the figure shows a model for prediction uses information (e.g., data) from the consistent state for instance, compare the estimated state at time step k to previous consistent state at time step $k-1$.

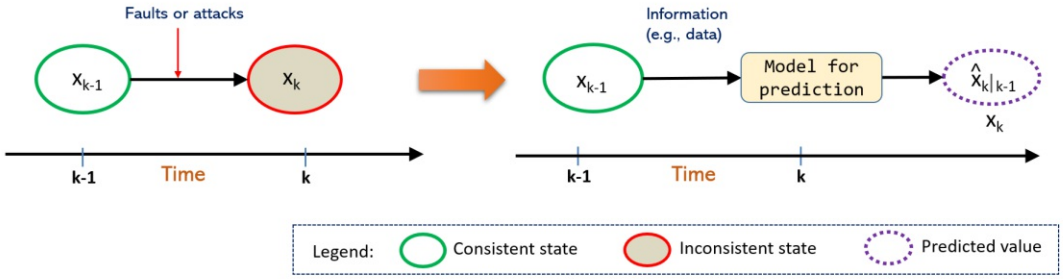


Figure 10.1: An overview of an inconsistent state recovery process.

Kong et al. [85, 84] present an approach for predicting a consistent state at the current time by using previous consistent states. In this work, historical information (global consistent states) has been used for making a prediction of a roll forward state. The proposed approach in [85, 84] uses a system model for current state prediction. As shown in Fig. 10.1, the system model uses the latest stored correct (globally consistent) state for the prediction of the current state. The authors mention that instead of using the prediction by their model, prediction by Kalman filter can be used for such state recovery, and leave it as a future work. Here we want to bridge the gap and show how prediction by Kalman filter can be used for the state recovery. Table 10.1 shows an overview of difference of our approach to the approach in [85, 84]. In a similar manner in [85, 84], we assume other existing mechanism discover whether a particular sensor is faulty or compromised. In our case, we can use the detection methods shown in Chapters 7 and 9. After discovering the fault or attack, we execute our model for state recovery.

In this chapter, we aim to mitigate the effects of attacks on state estimation (SE) by correcting the detected anomalous data before sending them to a state estimator. As we correct the anomalous data, the estimated states based on the corrected data are different from the original (actual) states. Though the states are different from the original states, they represent previously known normal behaviour from the system; if the state of the actual system does not change significantly, the corrected values should closely represent the actual system state. In this way, we aim to provide a form of SE integrity. In this sense, in the following sections of this chapter, we use the term preservation of SE integrity for mitigating the effects of attacks on SE by correcting the detected anomalous data. Our research question for preserving SE integrity reads:

- **RQ 3.1**: To what extent can the effects of FDI attacks on SE in electric power systems (EPSs) be mitigated by replacing detected anomalies with values derived from past data?
  Rationale: In [85, 84], it is shown that an inconsistent state can be rolled forward to current state if past consistent states are used for predicting the current state. The

Table 10.1: An overview of our contribution.

Appraoch in [85, 84]	**Our approach**
- assume some mechanism exists for attacks/faults detection	- detect attacks using the methods presented in Chapters 7 and 9
- focus on cyber-physical-state	- focus only on cyber-state
- use a Linear-Time Invariant (LTI) model based on historical information and control inputs for a system state recovery	- use Kalman filter (KF) based on measurements and without control inputs to mitigate the effects of attacks on state estimation
- use LTI model for prediction of current state	- use KF model for prediction and estimation of current state
- the model use historical information (e.g., previous global consistent state based on measurements) and control inputs from physical states between previous global consistent state and inconsistent state to predict states of failed (compromised) elements (e.g., sensor)	- the KF model use only measurements for prediction and estimation of current state - replace detected anomalous data with the predicted value of Kalman filter which is called corrected value - also use the predicted and the corrected value (no control input) to estimate current system state
- the predicted state is considered as current consistent system state	- the estimated state is considered as current consistent system state
- detection delay and difference of the recovered state to the reference are used to evaluate the approach.	- evaluate how the approach reduces unnecessary voltage reporting to a control center with different detection methods and the corrected value
- future work in [85]: prediction can be done by Kalman filter	- covers the future work in [85]

correctness of SE is important as control actions in a power system will be taken based on the decisions of the SE; the importance of SE correctness or SE integrity preservation motivates us to investigate methods for the SE integrity preservation. We assume SE integrity can be preserved by replacing the detected anomalous measurements with the predicted value of Kalman filter before sending them to SE. As can be seen in Chapters 8 and 10, the false data injection attacks introduced in Chapter 5 are detected faster or slower depending on the method. In this way, we can do both: detect anomalies in the measurement data and preserve state estimation by using past data.

Here, we use Kalman filter for SE. When a state is detected as anomalous by anomaly detection methods then the state is considered as inconsistent or abnormal state of a system. In order not to influence SE with the anomalous states, the predicted state by

the Kalman filter is then used for replacing the anomalous state.

Table 10.2 shows the intention of using the SE integrity preservation model and data used for the experiment. Details on parameter settings for the experiment are presented in Sec. 10.2.

Table 10.2: Overview of state estimation integrity preservation, Sec. = section.

Method	**Data***	**Goal of experiment**	**Sec.**
Anomalous data replacement	Training data (22.03.2016-31.03.2016) Test data (01.04.2016-14.04.2016)	- to answer **RQ 3.1**	10.1 10.2 10.3

* For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

All of the notations used in this chapter are illustrated in Tab. 10.3.

Table 10.3: Notations used in state estimation integrity preservation.

Notation	**Description**
$\boldsymbol{H}$	Observation model
$\boldsymbol{z}_k$	Observed measurement
$\Delta\boldsymbol{z}$	An offset due to an attack
$\boldsymbol{z}_{k,a}$	Manipulated measurement
$\boldsymbol{z}_{k,c}$	Corrected (substituted) value
$\hat{\boldsymbol{x}}_{k\|k-1}$	Predicted state
$\hat{\boldsymbol{x}}_{k\|k}$	Estimated state
$\boldsymbol{L}_k$	Kalman gain
$\hat{\boldsymbol{x}}_{k\|k,a}$	Estimated state based on manipulated measurement
$\Delta\hat{\boldsymbol{x}}_{k\|k}$	Difference between estimated state based on manipulated and actual measurements
$\hat{\boldsymbol{x}}_{k\|k,c}$	Estimated state based on corrected (substituted) values
$\boldsymbol{S}_{err}$	Sum of difference of estimated voltage during attack and in normal operation
$\boldsymbol{S}_{diff}$	Sum of voltage of SE based on substituted value and actual measurement

10.1 Theoretical Background

10.1.1 Approach

In our scenario, PMU measurements are used for estimating states of a system. Estimated states based on manipulated measurements can lead to wrong control decisions and can cause impact to devices and also in human lives. We aim at avoiding such impacts by preserving SE integrity. To this end, we substitute detected anomalous data by the values predicted by the Kalman filter. Here we present our approach for SE integrity preservation.

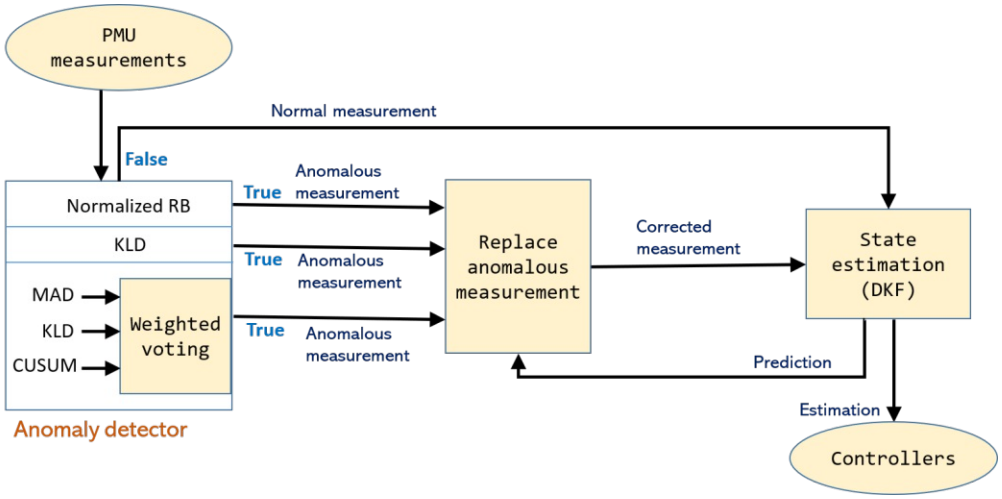


Figure 10.2: State estimation integrity preservation model.

Figure 10.2 shows our approach for replacing detected anomalous data and preserving SE integrity. If there is no anomaly detection scheme, then the PMU measurement (normal or anomalous) is directly used for SE which will estimate the wrong state if one or more observations are anomalous or manipulated. Thus, the PMU measurement needs to be sent to an anomaly detector where anomaly detection method is deployed, so that the detected anomalous data can be corrected before sending it to the state estimator. If a data point is accidentally detected as an attack (false positive) then it replaces the normal data point which will influence the state estimation in a way that it predicts the wrong state. We investigate on protecting SE integrity during abnormal behaviour of a system and propose a model which detects anomalies, corrects the detected anomalous data before feeding them to state estimator.

PMU measurement is fed to an anomaly detector where methods for detecting anomalies are deployed. For our experiment, we selected only some anomaly detection methods because we want to compare SE integrity preservation results from different types of methods. To this end, we use a method from bad data detection (BDD), we decided to

use KLD and a method that combines these approaches (in this case all three lightweight statistical methods are used). Methods are selected based on our experimental results analysis in Chapter 9 and 9. L2-norm RB method does not detect any type of attacks (generated by FDI attack model presented in Sec. 5.3). Therefore, this method is not considered here. Normalized RB method detects one of the attacks (type CO attack) and also detects extreme benign anomalies (see Sec. 7.3.3). Both slow-changing attacks and abrupt-changing attacks can circumvent the detection methods. But the slow changes are reflected in distribution. Kullback-Leibler divergence (KLD) detects these types of attacks and consistently detects subsequent anomalies. KLD has higher true positives than MAD and CUSUM (see Sec. 9.3)) Further a type of combination approach, weighted voting (WV) combines anomaly detection results from median absolute deviation (MAD), KLD and cumulative sum (CUSUM), and enhance anomaly detection performance generating low false alarm (see Sec. 9.4.3). Therefore, we selected KLD and the combined method for our experiments. We assume the selected methods are deployed in the anomaly detector and anomaly detection results of each methods are treated separately.

Data different than original data (rectangular coordinates with fixed phase) is used in state estimation, it results the prediction which is not close to the original data. The substitution of anomalous data with the prediction will result in the state that is not close to the original data. Though the resulting state is not close to the original state, the substitution avoids the effect of an attack by replacing the manipulated data with representative normal data in the state estimation process.

If a measurement is detected as being anomalous then the measurement is replaced by prediction otherwise the measurement is sent to the estimator. SE based on the corrected or normal measurement is sent to the operators. Operators take decisions for control actions based on the estimated states.

The input for the RB method is different than for the other methods. Therefore, we cannot directly compare the errors reported to CC while deploying RB method to the errors while deploying other methods.

Anomalous data replacement can preserve SE integrity and reduce reporting the wrong measurements to the CC. Methods to preserve the integrity of SE is presented in next section. But with false positive it influences the state estimation in a way that it predicts the wrong state. Therefore, we make an assumption that we do not have too high false positive.

10.1.2 Integrity of State Estimation

A state estimator estimates power system states based on observations. We use PMU measurements for SE using a DKF and estimate voltage states. If estimated voltage is normal no reactions are needed to be triggered, and if the estimated voltage is under voltage or over voltage then necessary protection actions need to be applied in the system.

When an attacker is able to manipulate voltage values, these manipulated voltage values are used for estimating states. Estimated states based on the manipulated values can be different than the real states of the system. Table 10.4 shows possibilities how an attack can fake the system's states. The possibilities are i) real state is normal and estimated (manipulated) state shows either over voltage or under voltage ii) real state is under voltage and estimated state shows either over voltage or normal iii) real state is over voltage and the estimated state is either normal or under voltage.

Table 10.4: Possibilities of fake states.

Real state	Normal	Under votlage	Over voltage
Estimated state	Over voltage	Normal	Normal
	Under voltage	Over voltage	Under voltage

Due to the data manipulation attack, operators may trigger protection actions in the system based on the fake estimated states. So, unnecessary actions are triggered in the system leading to critical condition. For example if the system is in under voltage and fake state shows it is in over voltage, then reactions can lead the system to invoke wrong power control actions. For SE with Kalman filters, estimated state at time step k is based on the predicted state ($\hat{\boldsymbol{x}}_{k|k-1}$) and the observation vector ($\boldsymbol{z}_k$). Here we recall SE using DKF in normal operation in Eq. (10.1) (as presented in Sec. 4.2.1).

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{10.1}$$

where $\boldsymbol{L}_k$ is Kalman gain at time step k.

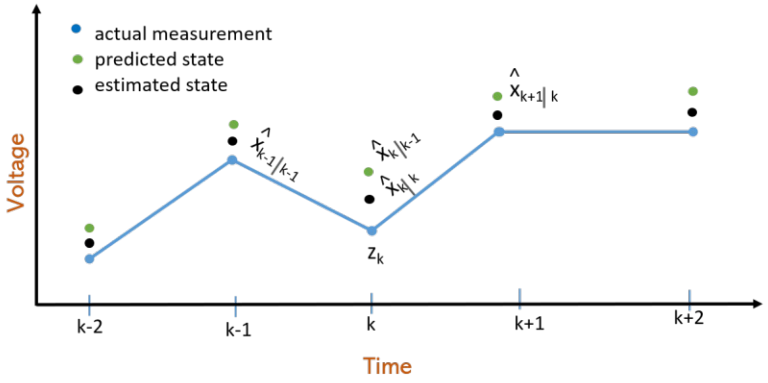


Figure 10.3: State estimation process based on actual signal (without manipulation).

Figure 10.3 depicts the SE process at a time step k based on the actual signal. At the time step k, $\boldsymbol{z}_k$ is the original (actual) measurement, first a prediction $\hat{\boldsymbol{x}}_{k|k-1}$ is calculated based on the previous estimated value $\hat{\boldsymbol{x}}_{k-1|k-1}$ and then a new estimate $\hat{\boldsymbol{x}}_{k|k}$ for time step k is calculated from the prediction $\hat{\boldsymbol{x}}_{k|k-1}$ at time step k and the measurement $\boldsymbol{z}_k$ at

time step k. In the next time step $k+1$, state is predicted from estimated value $\hat{\boldsymbol{x}}_{k-1|k-1}$ and estimation process continues considering observed measurement and the prediction.

The manipulated voltage value is represented by Eq. (10.2)

$$\boldsymbol{z}_{k,a} = \boldsymbol{z}_k + \Delta \boldsymbol{z} \tag{10.2}$$

where $\Delta \boldsymbol{z}$ is an offset due to an attack at time step k.

SE using a DKF under attack is represented by Eq. (10.3)

$$\hat{\boldsymbol{x}}_{k|k,a} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_{k,a} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{10.3}$$

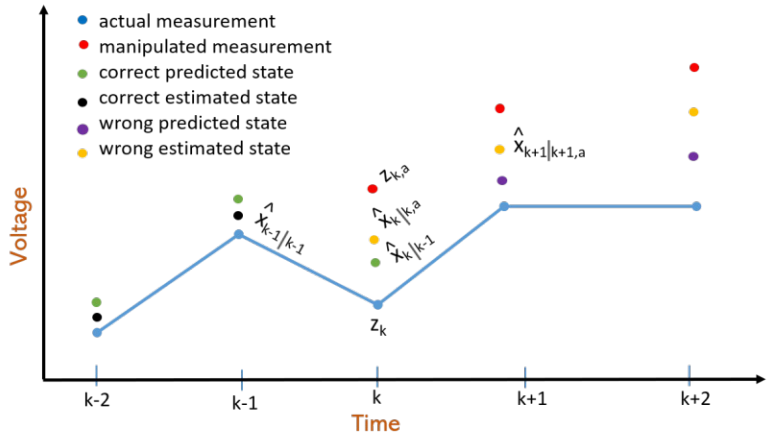


Figure 10.4: State estimation process based on manipulated signal.

Figure 10.4 depicts SE process at a time step k based on manipulated signal. Suppose an attack starts at time step k. At the time step k, $\boldsymbol{z}_{k,a}$ is manipulated measurement and the prediction of the state at time step k (denoted as $\hat{\boldsymbol{x}}_{k|k-1}$) from previous estimated value $\hat{\boldsymbol{x}}_{k-1|k-1}$. Current state ($\hat{\boldsymbol{x}}_{k|k,a}$) is estimated from the predicted state $\hat{\boldsymbol{x}}_{k|k-1}$ and the manipulated measurement $\boldsymbol{z}_{k,a}$. From this step we can see estimation process estimates wrong value considering manipulated measurement for estimation. In the next time step $k+1$, prediction is done from the wrong estimated value $\hat{\boldsymbol{x}}_{k|k,a}$ and estimation process continues considering the wrong prediction and observed measurement.

Integrity of SE affected due to an attack can be formulated as Eq. (10.4) using the equations (10.3) and (10.1).

$$\Delta\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k,a} - \hat{\boldsymbol{x}}_{k|k} \tag{10.4}$$

where $\hat{\boldsymbol{x}}_{k|k}$ is the estimated state based on the original (non manipulated) measurement at time step k and $\hat{\boldsymbol{x}}_{k|k,a}$ is estimated state based on the manipulated measurement at time step k.

10.1.3 Calculating Voltage Differences

Unnecessary voltage can be reported to an operator due to an attack because wrong states can be estimated based on the manipulated measurements. We aim to calculate total amount of the voltage error reported to a CC due to an attack. To this end, we analyse how much unnecessary voltage is reported to the CC with and without the anomaly detection scheme.

In our use case (as shown in Chapter 4), we have discrete time signal. Here, we want to calculate voltage in a discrete time signal to investigate how an attack affects the integrity and how we can preserve using anomaly detection schemes.

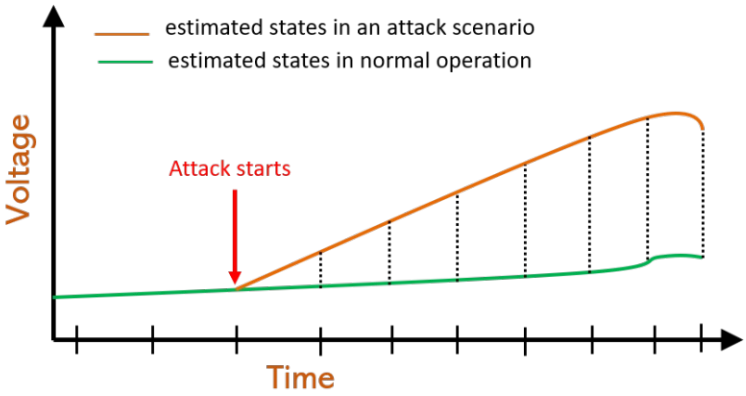


Figure 10.5: An example figure of estimated states in normal operation and in an attack scenario.

Figure 10.5 shows an example figure of estimated states in normal operation and in an attack scenario, the area between two the curves represents the error reported to the operator due to an attack.

Total difference of estimated voltages in normal operation and in an attack scenario over time (i.e., the sum of the errors) can be calculated using Eq. (10.5).

$$S_{err} = \sum_{i=1}^{N} \mid \hat{\boldsymbol{x}}_{i|i,a} - \hat{\boldsymbol{x}}_{i|i} \mid \tag{10.5}$$

where time interval between two consecutive data points is constant.

We take the sum of the errors due to an attack (S_{err}) as a metric to compare how much the reported values differ from the original measured values. Thus, we take S_{err} as a metric to see if our methods works well for preserving the SE integrity. Calculation of the unnecessary amount of energy or voltage is very important as it invokes protection actions such as load shedding, active or reactive power control in the power system. Thus we focus on preserving SE integrity by replacing detected anomalous state with predicted

state. Contribution of anomaly detection methods to preserve SE are discussed in the next section.

10.1.4 Preservation of Estimated State Integrity

The situation of a system is evaluated based on estimated states. Operators in a CC being aware of a system's current situation trigger necessary control actions. Therefore here we aim increasing trustworthiness of estimated states by detecting anomalous data and correcting them before sending it to the estimator.

We replace the detected anomalous data by the predicted value which we name as corrected value. Thus, anomalous data are corrected and sent to the state estimator. It helps increasing trustworthiness of estimated states. The "corrected" value differs from the "correct" value (which is the actual measurement). Corrected voltage value is represented by Eq. (10.6).

$$\boldsymbol{z}_{k,c} = \hat{\boldsymbol{x}}_{k|k-1} \tag{10.6}$$

where $\hat{\boldsymbol{x}}_{k|k-1}$ is the predicted value.

Estimated state using the corrected measurement is represented by Eq. (10.7)

$$\hat{\boldsymbol{x}}_{k|k,c} = \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{z}_{k,c} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{10.7}$$

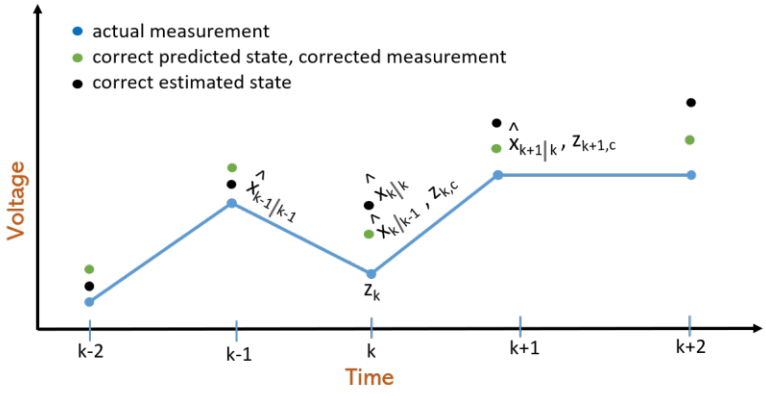


Figure 10.6: State estimation process considering anomaly detection and corrected measurement.

Figure 10.6 depicts SE process at a time step k considering anomaly detection and bad measurement replacement scheme. Current state is predicted (denoted as $\hat{\boldsymbol{x}}_{k|k-1}$) from previous estimated value $\hat{\boldsymbol{x}}_{k-1|k-1}$. Suppose an attack starts at time step k and is detected immediately at the time step k. Then, the manipulated measurement at time step k $\boldsymbol{z}_{k,a}$ is replaced by the predicted value $\hat{\boldsymbol{x}}_{k|k-1}$, and the corrected value is denoted as $\boldsymbol{z}_{k,c}$. A corrected estimated state at time step k (denoted as $\hat{\boldsymbol{x}}_{k|k,c}$) is estimated from

the predicted value $\hat{\boldsymbol{x}}_{k|k-1}$ and the corrected measurement $\boldsymbol{z}_{k,c}$. Thus in next time step $k+1$, state is predicted from $\hat{\boldsymbol{x}}_{k|k,c}$ and the estimation process continues considering corrected data and correct prediction.

Error on reporting the voltage to CC can vary depending on the detection delay of anomaly detection methods but also since the corrected value may differ from the original measurement. Thus, the sum of the errors after the correction (the difference between the corrected value and the real state) can be calculated using Eq. (10.8).

$$S_{diff} = \sum_{k=1}^{N} \mid \hat{\boldsymbol{x}}_{k|k,c} - \hat{\boldsymbol{x}}_{k|k} \mid \tag{10.8}$$

We take the difference as a metric and do experiments with different anomaly detection methods. Further, we compare for which method we get the smallest difference.

Residual-based algorithm checks bad measurements but the slow changes in measurement can circumvent the residual-based detection. Therefore, sometimes malicious measurements can hide in normal operation and keep having negative impact in the system.

10.2 Experimental Setup

We use PMU data from EPFL campus PMU network for our experiment (see Chapter 6). 50 frames/sec are reported from a PMU. i.e. a frame is sent every 20 milliseconds. Data preprocessing (see Sec. 6.5) and experimental setup for normalized residuals based method (see Sec. 7.2.3), KLD (see Sec.9.2.2) remain the same as presented in Chapter 9. Experimental setup of weighted voting remains the same as described in Sec. 9.4.2 of Chapter 9.

We use S_{diff} as a metric to compare how the data replacement works with the different AD methods. An attacker may bypass a detection system, but measurement changes can be detected using a different method (may be with detection delay). A combination method's decision has better detection performance than a single method and can thus help to minimize the issue of feeding anomalous data to the state estimator.

SE using Kalman filter is executed in normal operation and attack scenarios. If an anomaly is detected then the anomalous data is replaced by the predicted value. In addition, we run SE with and without the proposed scheme. We use discrete Kalman filter as shown in Sec. 4.2.1.

10.2.1 With attacks, detection and data substitution

In this setup, states are estimated with attacks, detection methods and data replacement. The proposed scheme has three methods for detecting anomalies. Detected bad data is

corrected based on the detection results of each method. Figure 10.7 shows experimental setup of the methods.

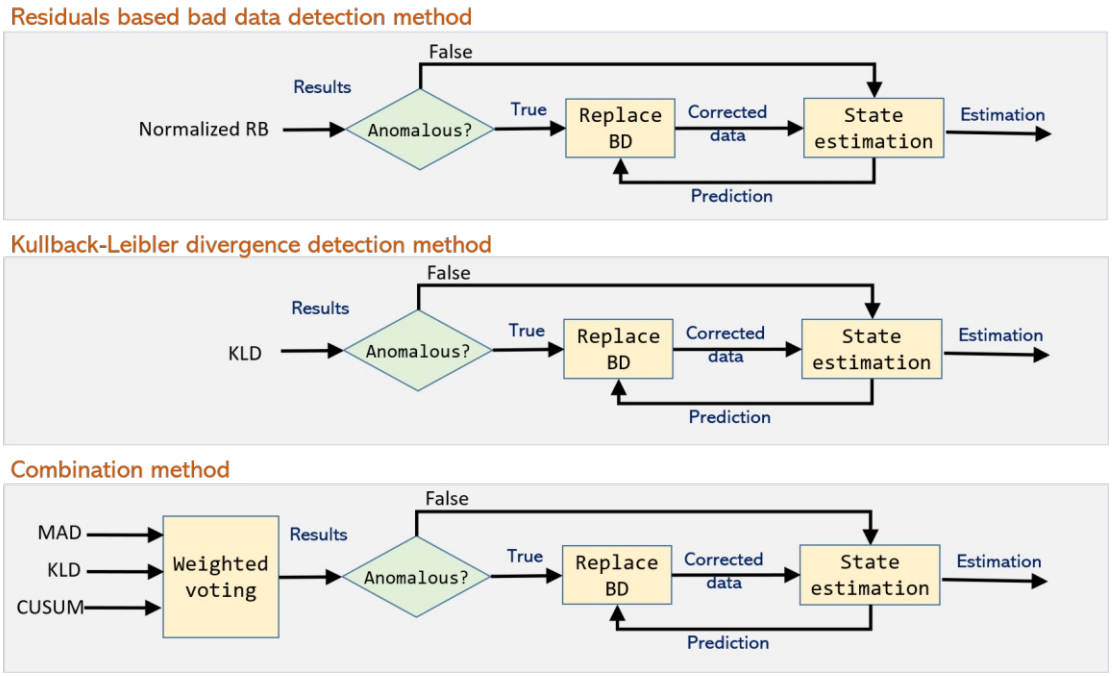


Figure 10.7: State estimation with attacks and the proposed integrity preservation scheme.

We test the integrity preservation with three different AD methods: normalized RB, KLD, combination method (weighted voting). We use KLD among other lightweight statistical methods (MAD and CUSUM) shown in Chapter 9 because it consistently detects the slow changes or abrupt changes attacks with higher true positives.

If the methods detect an observation at time step k as anomalous then the observation is replaced by the predicted value at the time step k. In other hand if a method has high false positives then it impacts the state estimation as we correct a measurement that has not been manipulated.

Table 10.5 shows an overview of SE integrity preservation, parameter settings and injected attacks. Predicted and estimated states under this setup are used for results analysis. Results are presented in the next section.

Table 10.5: Overview of state estimation integrity preservation, parameter setting and injected attacks; Exp. = experiment; CO = constant offset; RO = random offset; ICO = incremental constant offset; IRO = incremental random offset; IROMN = incremental random offset with more noise; ICOHS = incremental constant offset with high slope; Sec = section.

Exp.	Methods	Data*	Param. setting	Injected attacks	Sec.
10.1	Anomalous data replacement	Training data (22.03.2016 - 31.03.2016) Test data (01.04.2016)	- Results from normalized residual-based method with threshold 10.7, - KLD with threshold 8.95 - weighted voting	CO, RO ICO, IRO IROMN, ICOHS	10.3

* For all the given days of training and test data, one hour at 02:00-03:00 UTC is used.

10.3 Results

10.3.1 State estimation in normal operation

In this section, we present estimated voltage states based on actual voltage. We do not deploy any anomaly detection methods and execute the SE process. Extreme values due to jumps in the signal have less influence in SE as Kalman filter puts less trust in the measurements and follows the prediction model more closely.

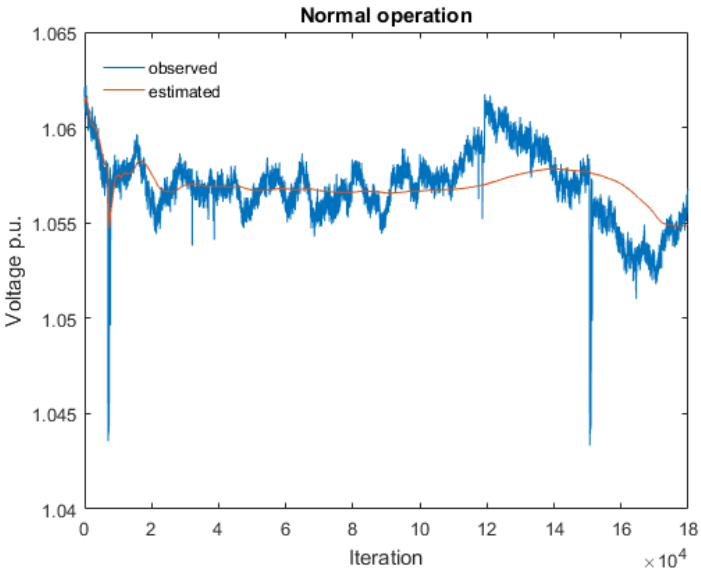


Figure 10.8: State estimation in normal operation (y axis represents polar voltage).

Figure 10.8 shows the observed voltage signal without attack and the estimated voltage. Blue signal shows the observed signal and orange line shows the estimated signal. Since the estimation process smooths out noise, the estimated signal does not fluctuates like the original signal.

The estimated signal aligns with the nature of the signal for instance between data points 7,093 and 7,815 voltage signal drops and slowly following the nature of the voltage signal the estimated signal also drops between the data points 7,093 and 7,815. Also after the drop at data point 120,000 the estimated signal slowly follows the nature of the actual voltage signal. The original values will be reported to the control center, so they see the real drops (and also could raise an alarm).

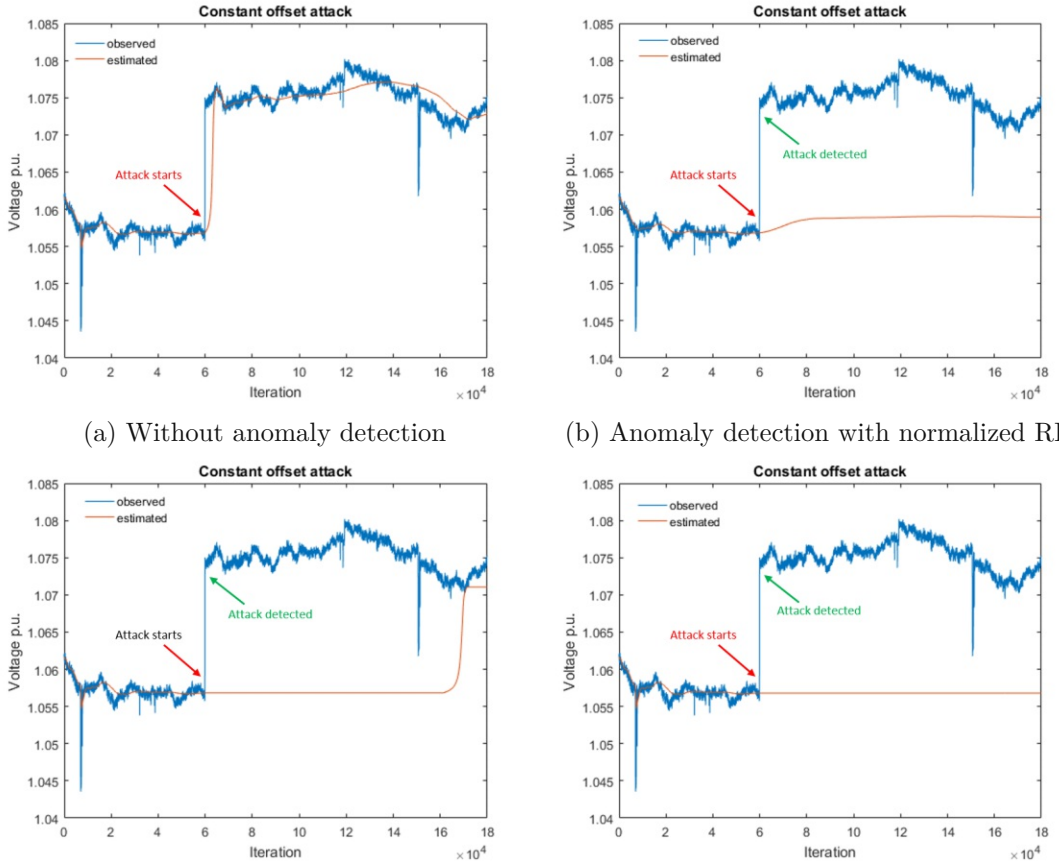
10.3.2 Anomalous data replacement and state estimation

In this section, we present SE under attack scenarios. Here we compare SE results before and after deploying anomaly detection methods. Anomaly detection is performed on the reported measurement values. For the comparison we execute SE process with the proposed scheme. We execute different anomaly detection methods and replace detected anomalous data before sending data to state estimator.

10.3.2.1 Constant offset attack

We execute normalized RB, KLD and weighted voting methods, and replace detected anomalous data before executing SE during constant offset attack. Further, for comparison of SE results we execute the SE process during the constant offset attack without deploying any anomaly detection method.

Figure 10.9 shows estimated voltage states during the constant offset attack with and without protection scheme for the test data set on April 1, 2016. Sub-figure 10.9a shows estimated states without deploying any anomaly detection method. Therefore, from the sub figure we can see the estimated signal follows the manipulated signal. Sub-figure 10.9b shows estimated signal after deploying normalized RB BDD method. The normalized RB BDD method is executed before estimating states and if any data point is detected as anomaly then the data point is replaced by predicted value. As presented in Chapter 7, normalized RB BDD method detects benign anomaly between data points 7,093 and 7,815 (due to a jump between the data points), and replaces the anomalous data with the prediction. When an attack starts at $60,001^{st}$ data point, we can see estimation is already affected due to the attack and the estimated values increase. When the normalized RB BDD method detects a data point as an anomaly, the anomalous data is replaced with the predicted value, and in the next iteration pre-fit residuals remain high and detects another data point also as an anomaly. Due to the replacement scheme, all detected malicious data points are replaced by the predicted values. On the other hand due to the substitution also the peaks will be missing. But since the CC also gets the



(a) Without anomaly detection

(b) Anomaly detection with normalized RB

(c) Anomaly detection with KLD

(d) Anomaly detection with weighted voting

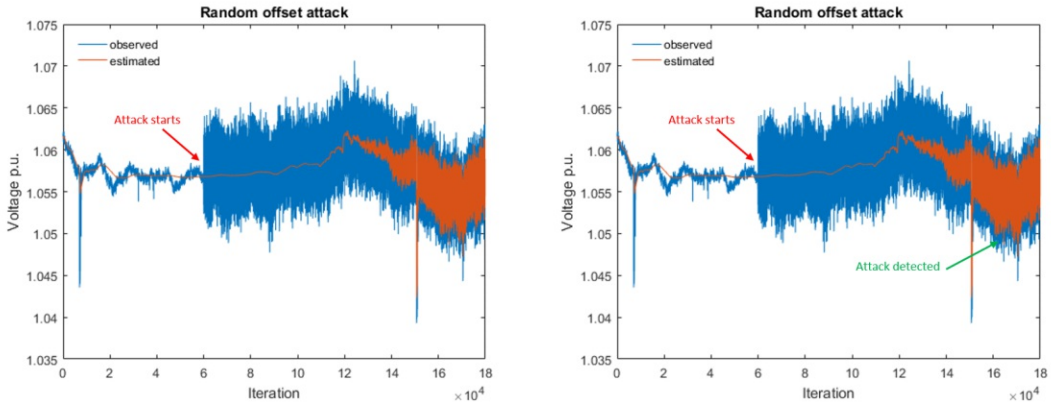
Figure 10.9: Observed and estimated signal in standard operation (a) and with different anomaly detection methods (b,c,d) and data replacement for constant offset attack.

(manipulated) measurement values, they will see the value but cannot asses if it is due to the manipulation or due to another cause.

Similarly, another sub-figure 10.9c shows results from SE where KLD method is executed before sending data to state estimator. From the sub-figure, we can see KLD is quick in detecting the anomaly but near data point 160,000 the divergence of the manipulated signal to the reference decreases and stays below the defined threshold. The changes in distribution caused due to the attack stay within the threshold because the original signal decreases near data point 160,000 and the manipulated values stay within the reference histogram distribution as it has quite broad distribution per hour (see Chapter 9). As anomalies are not detected, states are estimated based on the manipulated measurements. From the sub-figure 10.9c, one can see that the estimated state is based on the manipulated voltage which causes that the estimated signal raises (increases) until it detects an anomaly again (see increasing red line near data point 160,000). Sub-figure

10.9d shows weighted voting detects anomalies till the end, and is better than other methods.

10.3.2.2 Random offset attack



(a) Without anomaly detection (b) Anomaly detection with weighted voting

Figure 10.10: Observed and estimated signal in standard operation (a) and with weighted voting method (b) and data replacement for random offset attack.

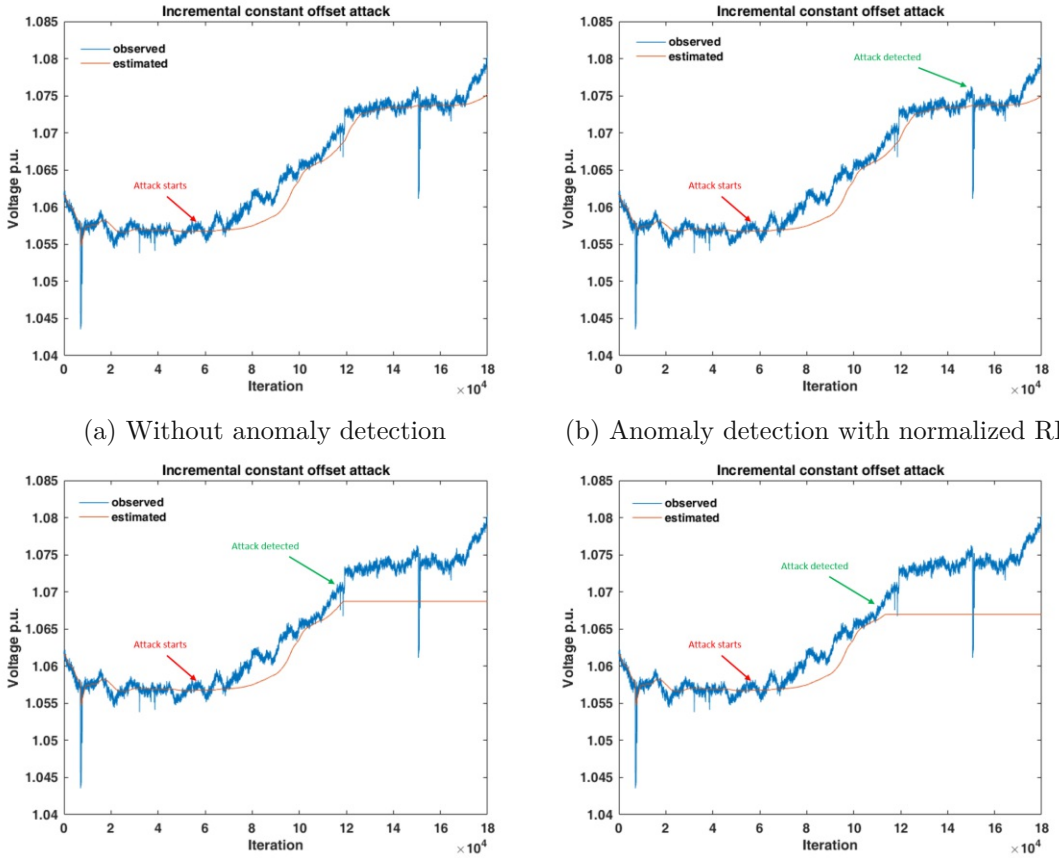
As presented in Chapter 7, normalized RB and KLD do not detect any malicious data points during the random offset attack. But weighted voting detects some malicious data points during this type of attack. Therefore we here only did experiments with weighted voting. Figure 10.10 shows the estimated voltage states during the random offset attack.

The estimation process is able to filter out noise up to point 12,000. After that the estimated signal starts becoming noisy because since anomaly is not detected, the SE process considers the noisy measurements for estimating the states. If anomaly is detected detected then also after the initial detection only some points are considered anomalous e.g. “even after the first point is detected as anomaly several subsequent manipulated data points are not detected as anomalies”.

Although the variation in the estimated signal is not normal, the random offset attack is able to bypass the detection methods. So even with weighted voting the attack is only detected very late. As only few data points are detected as anomalies by weighted voting (see Fig. 9.33 in Chapter 9); only data points that are detected as anomalous are substituted; therefore one cannot see the substitution in Fig. 10.10b.

10.3.2.3 Incremental constant offset attack

The SE process is executed during the incremental constant offset attack without deploying any of the anomaly detection methods.



(a) Without anomaly detection

(b) Anomaly detection with normalized RB

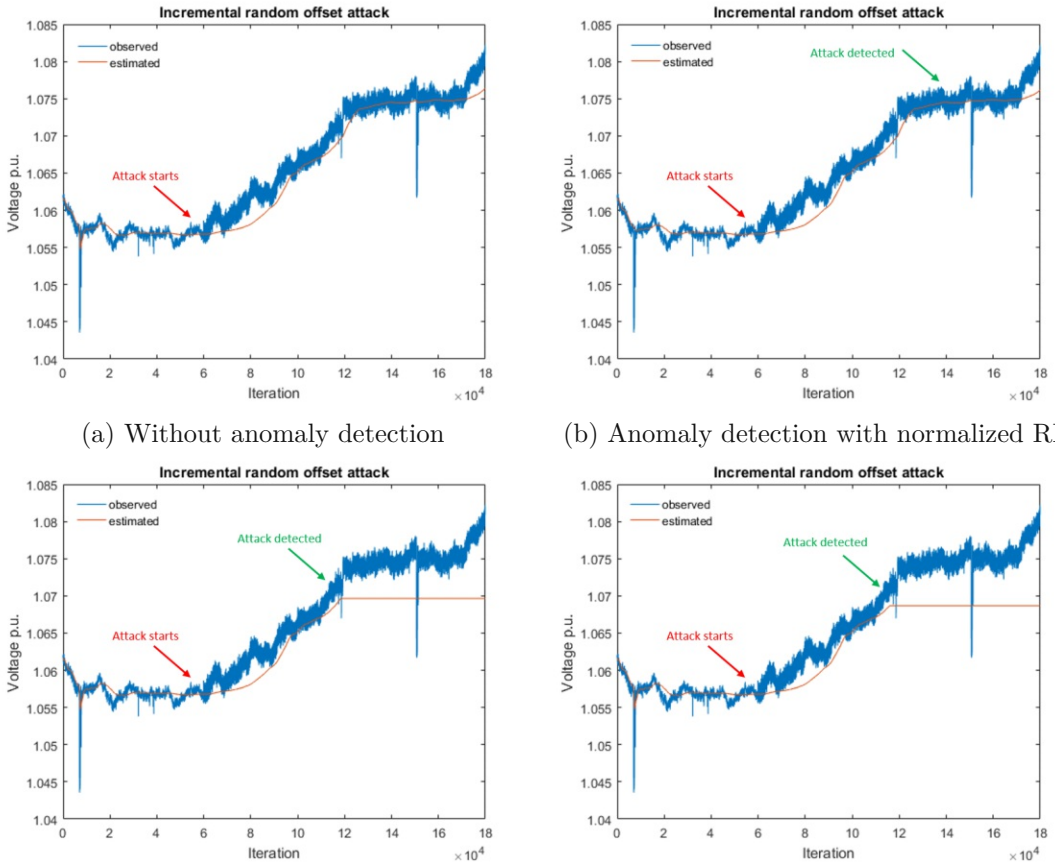
(c) Anomaly detection with KLD

(d) Anomaly detection with weighted voting

Figure 10.11: Observed and estimated signal in standard operation (a) and with different anomaly detection methods (b,c,d) and data replacement for incremental constant offset attack.

Figure 10.11 shows estimated voltage states during the incremental constant offset attack with and without protection scheme. Sub-figure 10.11a shows estimated states without deploying any anomaly detection method during incremental constant offset attack. Sub-figure 10.11b shows estimated signal after deploying normalized RB BDD method. Normalized RB BDD method is executed before sending data to the estimator. As presented in Chapter 7 normalized RB BDD method detects this type of attack very late. The detected anomalies are the benign anomalies caused due to the high jump in the signal, we can see from the sub-fig. 10.11b that SE could not be protected using this method because of the late detection and not all manipulated data points after the first point are detected. Similarly, sub-figures 10.11c and 10.11d show estimated signal after deploying KLD and weighted voting methods respectively. From the sub-figures we can see contribution of KLD and weighted voting in protecting SE.

10.3.2.4 Incremental random offset attack



(a) Without anomaly detection

(b) Anomaly detection with normalized RB

(c) Anomaly detection with KLD

(d) Anomaly detection with weighted voting

Figure 10.12: Observed and estimated signal in standard operation (a) and with different anomaly detection methods (b,c,d) and data replacement for incremental random offset attack.

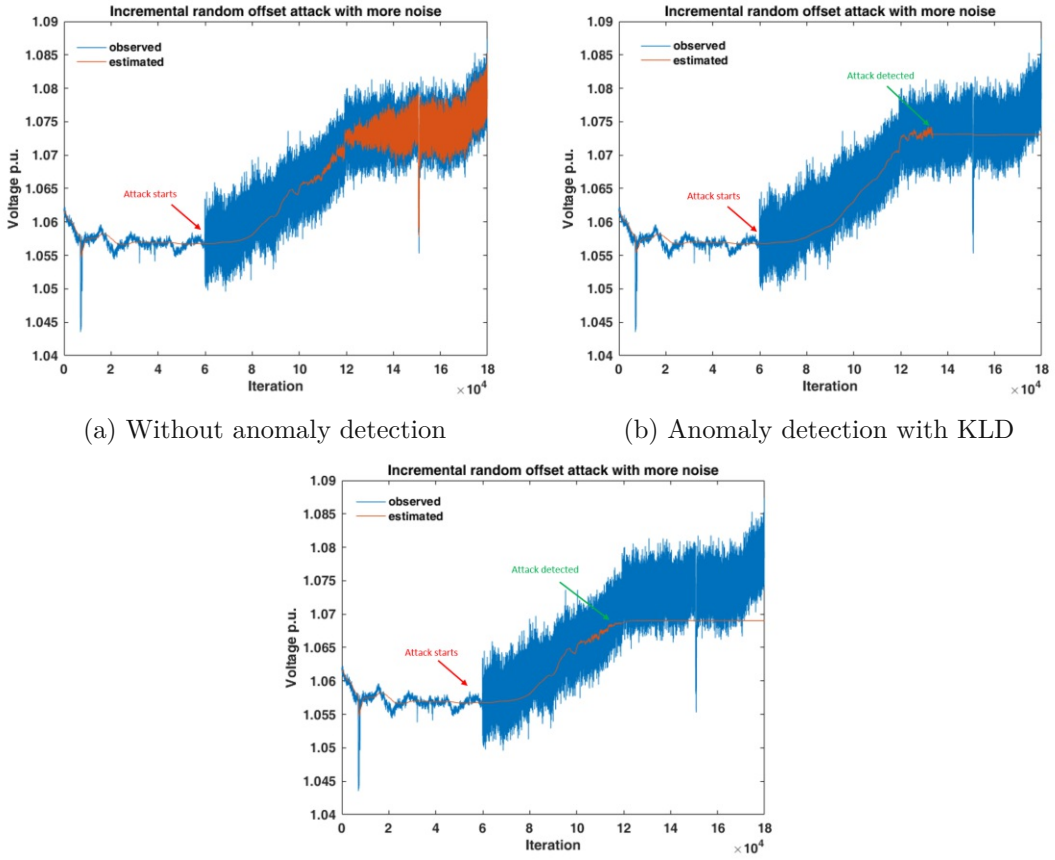
As presented in Chapters 7 and 9, normalized RB, KLD and weighted voting detect malicious data points during incremental random offset attack. Therefore, we here did experiments with the normalized RB, KLD and weighted voting methods where the detected anomalous data are substituted before sending the data to SE. In addition one without substitution as reference for comparison, the SE process is executed during incremental random offset attack without deploying any of the anomaly detection methods.

Figure 10.12 shows estimated voltage states during incremental random offset attack with and without protection scheme. Sub-figure 10.12a shows estimated states without deploying any anomaly detection method during incremental random offset attack. Sub-figure 10.12b shows estimated signal after deploying normalized RB BDD method. Normalized RB BDD method is executed before sending data to the estimator. As

presented in Chapter 7, normalized RB BDD method detects this type of attack very late (detection is only the benign anomalies caused due to the high jump), we can see from the sub-fig. 10.12b that SE could not be protected using this method.

Sub-figures 10.12c and 10.12d show estimated signal after deploying KLD and weighted voting methods respectively. From the sub-figures we can see anomaly detection is delayed. As a consequence, it delays the contribution of KLD and weighted voting in preservation of SE integrity.

10.3.2.5 Incremental random offset attack with more noise



(a) Without anomaly detection

(b) Anomaly detection with KLD

(c) Anomaly detection with weighted voting

Figure 10.13: Observed and estimated signal in standard operation (a) and with different anomaly detection methods (b,c) and data replacement for incremental random offset attack with more noise.

As presented in Chapters 7 and 9, normalized RB, KLD and weighted voting detect malicious data points during incremental random offset attack with more noise. Therefore,

we here did experiments with the normalized RB, KLD and weighted voting where the detected anomalous data are substituted before sending the data to SE. In addition one without substitution as reference for comparison, the SE process is executed during the incremental random offset attack with more noise without deploying any of the anomaly detection methods.

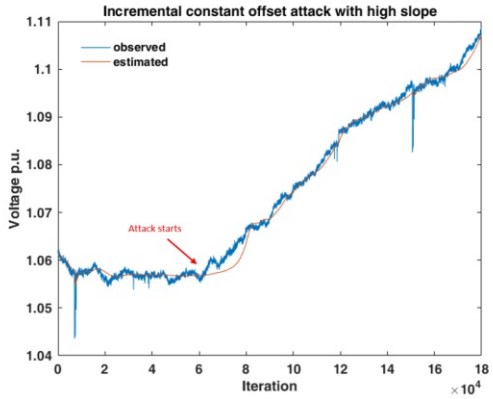
Figure 10.13 shows estimated voltage states during incremental random offset attack with and without protection scheme. Sub-figure 10.13a shows estimated states without deploying any anomaly detection method during incremental random offset attack. As presented in Chapter 7, normalized RB BDD method does not detect this type of attack, so the figure is the same as sub-figure 10.13a, and SE could not be protected using this method. Sub-figures 10.13b and 10.13c show estimated signal after deploying KLD and weighted voting methods respectively.

From the sub-figures we can see anomaly detection is delayed. This can be caused due to the random component, in some cases it prevents that thresholds are exceeded. If the detection is delayed, then the new estimate also is at a much higher level than the original (unmanipulated). As a consequence, it delays the contribution of KLD and weighted voting in preservation of SE integrity.

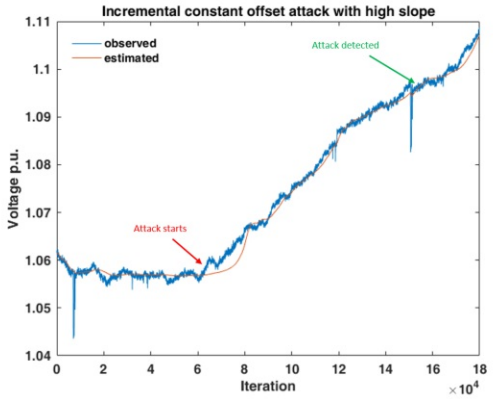
10.3.2.6 Incremental constant offset attack with high slope

Normalized RB, KLD and weighted voting detect malicious data points during incremental constant offset attack with high slope. Therefore, we here did experiments with the methods where the detected anomalous data are substituted before sending the data to SE. In addition one without substitution as reference for comparison, the SE process is executed during the incremental constant offset attack without deploying any of the anomaly detection methods.

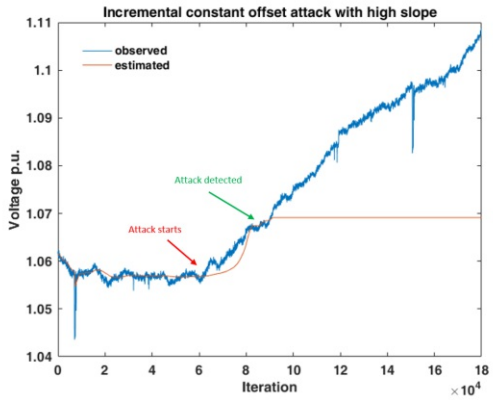
Figure 10.14 shows estimated voltage states during incremental constant offset attack with high slope with and without protection scheme. Sub-figure 10.14a shows estimated states without deploying any anomaly detection method during incremental constant offset attack with high slope. Sub-figure 10.14b shows estimated signal after deploying normalized RB BDD method. Normalized RB BDD method is executed before sending data to the estimator. As presented in Chapter 7 the normalized RB BDD method detects anomalies (counted as malicious anomalies) caused due to a high jump in this type of attack, we can see from the sub-fig. 10.14b that SE could not be protected using this method. Similarly, sub-figures 10.14c and 10.14d show estimated signal after deploying KLD and Weighted Voting methods respectively. From the sub-figures we can see contribution of KLD and weighted voting in protecting SE.



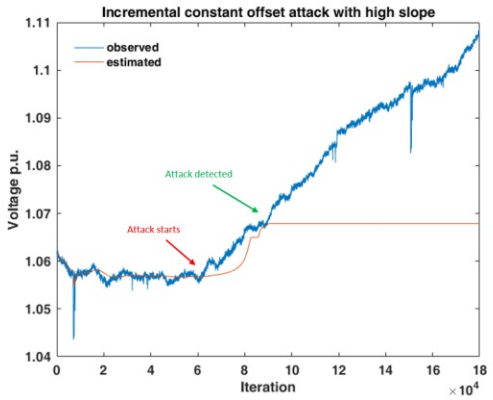
(a) Without anomaly detection



(b) Anomaly detection with normalized RB



(c) Anomaly detection with KLD



(d) Anomaly detection with weighted voting

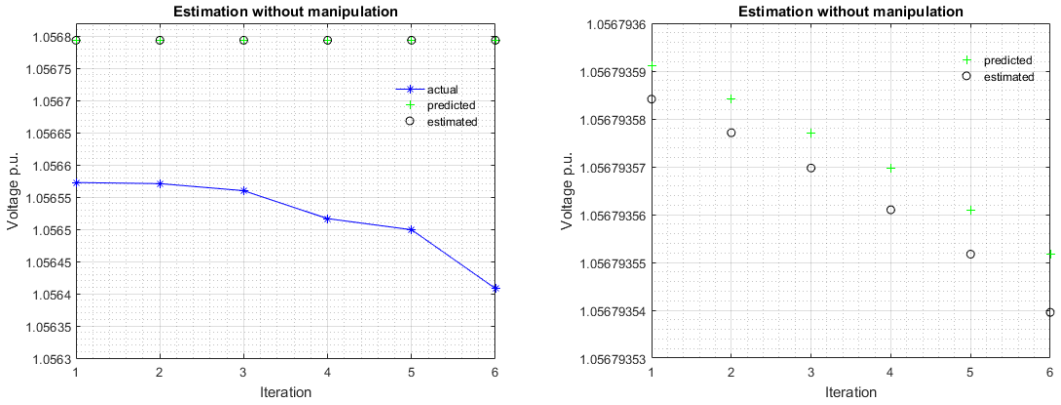
Figure 10.14: Observed and estimated signal in standard operation (a) and with different anomaly detection methods (b,c,d) and data replacement for incremental constant offset attack with high slope.

10.3.3 Effects on Voltage Estimates

Here we want to provide details or further insights about the effects of attacks in voltage estimates and the process of preserving the voltage estimation. We present estimated voltage values in different scenarios and provides details about how SE is preserved using anomaly detection methods and anomalous data replacement. Further examples of SE based on actual measurement, and manipulated measurement with and without the proposed scheme are shown.

We show how the estimation process works without manipulation, how it changes with the manipulation, and how anomaly detection and bad measurement replacement correct the SE.

Figure 10.15 shows an example of SE process based on actual measurements. Sub-figure

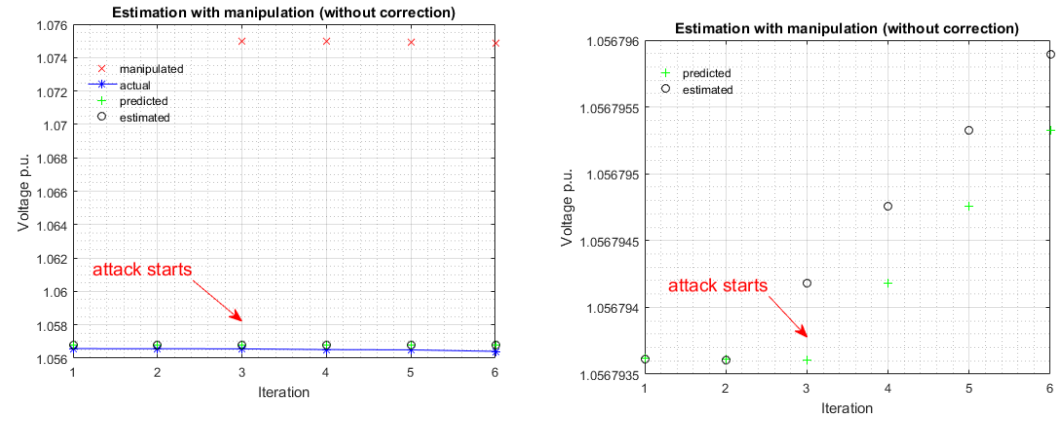


(a) Actual, prediction and estimation

(b) Zooming predicted and estimated values

Figure 10.15: State estimation without manipulation.

10.15a visualizes actual measurements by blue curve, predicted and estimated values by green and black points respectively. From the sub-figure, we can see prediction and estimation are very similar but differ from the actual values because prediction of current step depends on previous estimated state, and estimation of current step depends on predicted and observed (actual) values. The estimated values are close to the prediction as they both are related to the previous estimated values and observed values. Another sub-figure 10.15b zooms into predicted and estimated values to show that they slightly differ. From the sub-figures, we can see the estimated value lies between the prediction and observation because the state is estimated using the predicted and observed (original) value.



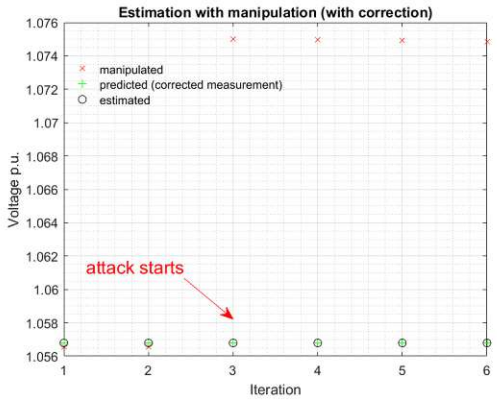
(a) Actual, manipulated measurement, prediction and estimation

(b) Zooming into predicted and estimated values

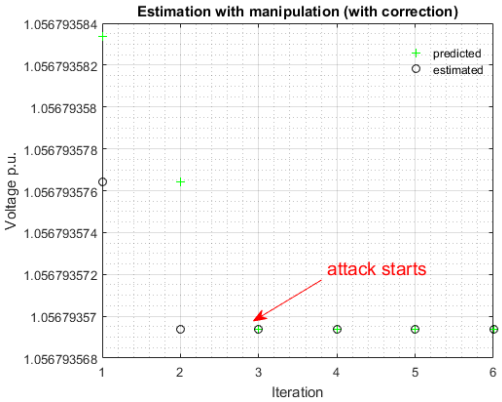
Figure 10.16: State estimation with manipulation and without correction.

Figure 10.16 shows an example of SE process based on manipulated measurements.

Sub-figure 10.16a visualizes actual, manipulated measurements together with prediction and estimation. When an attack starts at the third data point, manipulated measurement has the higher value than the actual value. From sub-figure 10.16b, we can see estimated value at the third data point increases and lies between manipulated value and predicted value instead of between original and predicted values. In the next steps, predicted and estimated values keeps on increasing.



(a) Actual, manipulated measurement, prediction and estimation



(b) Zooming predicted and estimated values

Figure 10.17: State estimation with manipulation and correction.

Figure 10.17 shows an example of SE process with correction of anomalous measurement. Sub-figure 10.17a shows manipulated measurements, predicted values, estimated values and substituted values (substituted by prediction). From the sub-figure, we can see predicted and corrected (substituted) values overlap, and the estimated value is very similar to the predicted value (it is zoomed-in in sub-figure 10.17b). In the first and second data points, predicted and estimated values have very small differences, but from the third data point onwards prediction and estimation overlap (i.e., they are equal). It is due to replacement of bad measurement.

Table 10.6 illustrates actual, measured, predicted and estimated values that are shown in Fig. 10.17. The fourth column shows whether the observation is detected as an anomaly. Depending on the anomaly detection, it is decided whether the corresponding measurement should be corrected (it is shown in column six). For the first two data points, measurements are not detected as anomaly and there is no need of correction. Once anomalies are detected from the third data point, the measurements are corrected by replacing them with the predicted value. From the table starting from the third data point, corrected values, predicted values, and estimated values are equal.

Table 10.6: Estimation with correction; Orig. = original; Report. = reported; Correct. = corrected; A = anomalous; NA = not applicable; Y = yes; N = no; $Diff_{est,org}$ = difference between estimated and original value.

Orig.	**Report.**	**A**	**Predict***	**Correct.**	**Estimate**	$Diff_{est,org}$
1.0565727	1.05657272	N	1.05679358	NA	1.05679357	220.8×10^2
1.0565709	1.05657095	N	1.05679357	NA	1.05679356	222.6×10^2
1.0565600	1.07500000	Y	1.05679356	1.05679356	1.05679356	233.5×10^2
1.0565166	1.07495661	Y	1.05679356	1.05679356	1.05679356	276.9×10^2
1.0564994	1.07493936	Y	1.05679356	1.05679356	1.05679356	294.1×10^2
1.0564080	1.07484802	Y	1.05679356	1.05679356	1.05679356	385.5×10^2

* Predicted value at time step k is always the estimated value from $k-1$.

Now we proceed to discuss the long term effect of voltage estimation without/with manipulation and without/with detection and correction process.

We calculate difference between estimated voltage in attack scenarios and estimated voltage in normal operation, denoted as S_{err}. S_{err} is the amount of unnecessary voltage due to voltage manipulation (error in reporting voltage due to an attack).

Table 10.7 shows an overview of unnecessary voltage amount reported to CC in different attack scenarios, and total estimated voltage after applying AD methods in the attack scenarios and its effect in estimated voltage integrity preservation. The voltage values are in p.u. From the table (S_{err} shown in second column), we can see that highest amount of unnecessary voltage is reported to CC during the ICOHS attack.

Table 10.7: Estimated voltage values (in p.u.) in attack scenarios, and state estimation preservation using different anomaly detection methods; S_{err} = sum of difference between estimated voltage in attack scenarios (without substitution) and normal operation (i.e., error in voltage reporting), S_{diff} = sum of difference of estimated values after applying AD method together with anomalous data replacement and estimated values in normal operation.

Attack	S_{err}	S_{diff} after applying method		
		Normalized RB BDD	KLD	Weighted Voting
CO	21.55×10^2	2.3698×10^2	1.0091×10^2	0.0169×10^2
RO	0.9783×10^2	1.009×10^2	97.78×10^2	0.0013×10^2
ICO	12.45×10^2	12.45×10^2	9.656×10^2	8.541×10^2
IRO	13.65×10^2	13.68×10^2	10.80×10^2	8.646×10^2
IROMN	13.64×10^2	13.64×10^2	12.87×10^2	10.41×10^2
ICOHS	30.23×10^2	30.25×10^2	12.42×10^2	10.98×10^2

Here, we proceed to the estimation results in attack scenarios while applying anomaly detection methods and bad data replacement. Total estimated voltage after applying anomaly detection methods together with bad data replacement in each of the attack

scenarios (shown in first column) is shown in third column (see S_{diff}). It shows the voltage reported to an operator after deploying the anomaly detection and mitigation scheme (i.e., voltage reported while deploying normalized RB BDD, KLD and weighted voting). The amount of voltage depends on anomaly detection delay.

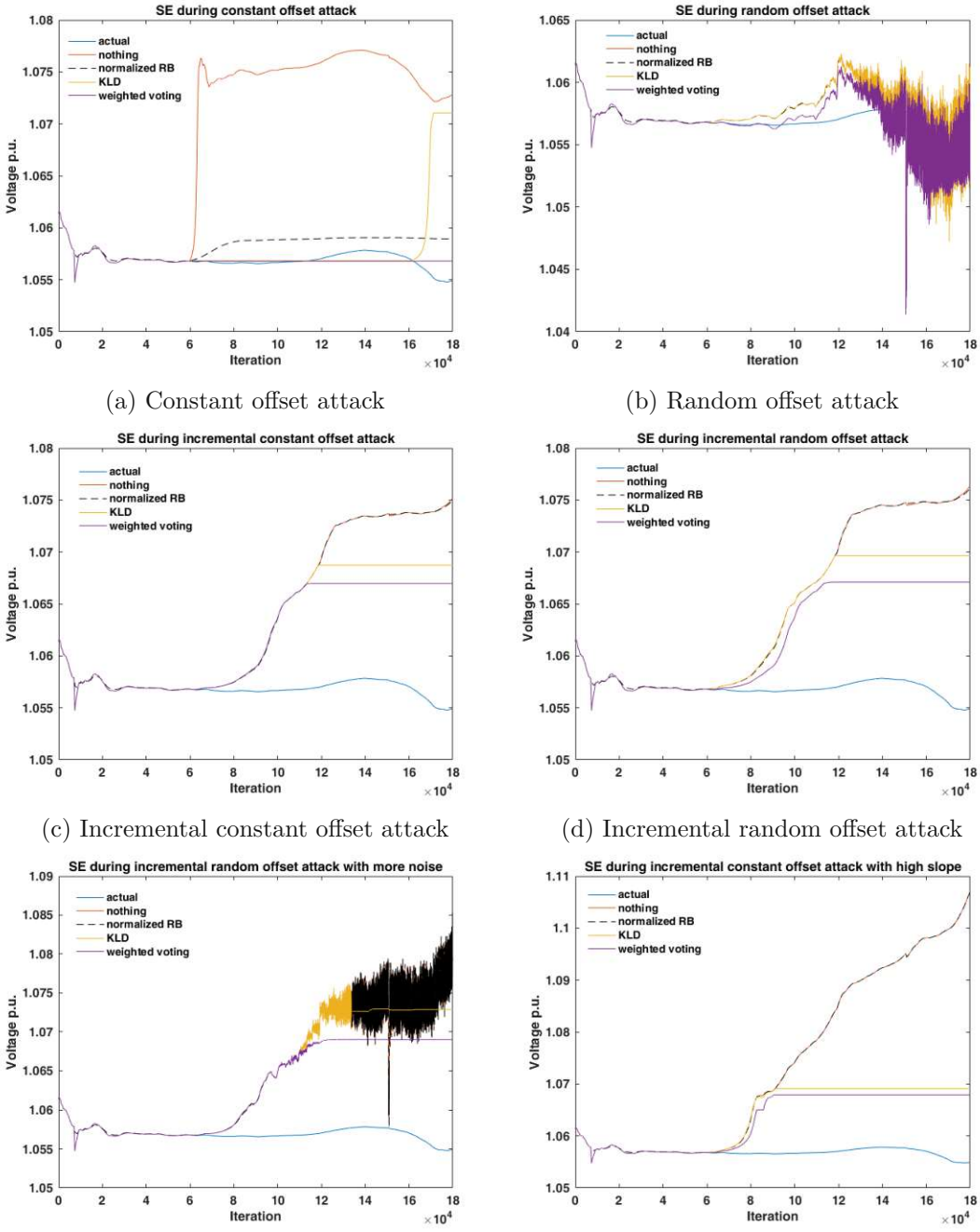
Total difference between estimated values after applying the detection methods together with bad data replacement and estimated values in normal operation (named as S_{diff}) is shown in third column of the table. We want to minimize the error due to an attack and show that the error is minimized if the substitution is done.

Figure 10.18 visualizes estimated voltage signals while deploying different methods in attack scenarios. Different curves in the sub-figures visualize estimated voltage signals in five conditions (given below). Thus estimated voltage values in five conditions, differences between them, illustrated in Table 10.7 are visualized in Fig. 10.18.

- Actual (blue curve): estimated based on actual (non-manipulated or normal) voltage.
- Nothing (orange curve): estimated voltage signal during attack without any detection method.
- Normalized RB (black curve): estimated voltage signal during attack while applying normalized RB BDD method and substitution.
- KLD (yellow curve): estimated voltage signal during attack while applying KLD method and substitution.
- Weighted voting (purple curve): estimated voltage signal during attack while applying Weighted voting method and substitution.

Mapping of values in Table 10.7 to the figures 10.18 is based on the following information.

- Sum of differences between orange and blue curve over time is calculated as S_err.
- Sum of estimated voltage represented by black/yellow/purple curve over time is calculated S_{ADR}.
- Sum of differences between black/yellow/purple curve and blue curve over time is calculated as S_{diff}.



(a) Constant offset attack

(b) Random offset attack

(c) Incremental constant offset attack

(d) Incremental random offset attack

(e) ICO attack with more noise

(f) ICO attack with high slope

Figure 10.18: State estimation integrity preservation using different methods in attack scenarios CO, RO, ICO, IRO, IROMN and ICOHS.

10.3.4 Results Findings

Our results show that the proposed SE integrity preservation scheme depends on anomaly detection delay and detection performance of the methods.

Here we conclude our results analysis.

- F 3.1.1: SE integrity preservation works well in CO type attack as it is detected immediately by all methods (normalized residual-based, KLD and weighted voting) and states can be corrected from the beginning.
- F 3.1.2: SE integrity preservation is challenging in RO type attack because detection of the RO attack type is also challenging. As expected, if an anomaly is not detected then SE integrity preservation does not work, for instance normalized RB methods and KLD do not detect RO type attack so in this attack type SE integrity cannot be preserved. for instance SE integrity preservation in IRO type attack is challenging than in ICO type attack as the detection of IRO is delayed than the detection of ICO.
- F 3.1.4: Combination method - weighted voting has better integrity preservation than the normalized residual-based and KLD methods because the combined results has better anomaly detection performance than the methods (normalized residual-based and KLD).

Since we do not know the original measurements the corrected state usually deviate from the original state. This leads to state deviations especially if attacks persist for a longer time span (as in our case). Also for a long sequence of anomalous values all corrected subsequent values equal since we lack fresh measurement information. Therefore for the preservation of the state estimation, it would be useful to also detect the end of an attack.

10.4 Summary

In this chapter, we presented our proposed approach for maintaining correctness of SE. The approach was designed to replace detected anomalous data and preserve SE integrity.

We showed how SE was affected by attacks. For this we presented possibilities how an attack could fake the system's states.

Voltage in discrete time signal is presented to calculate total amount of voltage over time. For this, amount of unnecessary voltage reported to the control center (i.e., error on voltage reporting to CC due to an attack) is analysed with and without the anomaly detection scheme.

We aimed at increasing trustworthiness of estimated states by detecting anomalous data and correcting them before sending to the estimator, and showed how estimated state integrity could be preserved.

Settings on estimating states with and without anomaly detection, and the proposed anomalous data replacement schemes were presented in the experimental setup. The results confirmed that the proposed SE integrity preservation scheme depended on anomaly detection delay and performance of the methods.

The following major results findings supported answering our research question **RQ 3.1** (To what extent can the effects of FDI attacks on SE in electric power systems (EPSs) be mitigated by replacing detected anomalies with values derived from past data?):

- SE integrity preservation works well if an anomaly was detected, for instance SE integrity preservation worked well for CO attack type and challenging for RO attack type.
- SE integrity was preserved by replacing detected anomalous measurements that have been detected by the residuals-based and lightweight statistical methods and the quality of integrity preservation depended on the detection delay.
- Combined anomaly detection results using weighted voting had better integrity preservation than the normalized residual-based and KLD methods as the weighted voting are trustworthy than the normalized residual-based and KLD methods.
- If we detect a false positive and correct the value that has not been manipulated then with a correction also the corrected value and true value have a difference. This difference can increase over time. Therefore, it would be also important to detect the end of an attack.

As we expected for reasoning **RQ 3.1**, the evidence from the experiments showed that the effects of attacks on SE was mitigated by replacing detected anomalous data by the predicted values from the Kalman filter.

CHAPTER 11

Summary and Conclusions

Notice of adoption from previous publications in Chapter 11
Parts of the contents of this chapter have been published in the following papers:

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. 4th International Symposium for ICS and SCADA Cyber Security Research, *2016*

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids, *CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM*

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018, *pages 16:1–16:10, 2018*

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. In preparation[a]

Explanation text, on what parts were adopted from previous publications:
Text is based on the work done in [129], [130], [132] and [133].

S. Paudel implemented the methods and conducted all the experiments. Theoretical considerations and experiment planning was done together with all co-authors, the text and figures in the papers were created together by all authors.

[a]The paper is in preparation.

11.1 Summary

A smart grid relies heavily on ICT in order to incorporate new functions into electricity grid monitoring and control. Increased deployment of information technology in smart grids opens new attack vectors and requires the deployment of methods to detect unusual activities. The fast changing landscape of threats in cyber-physical systems, the increasing complexity of control systems and the need to work in adversarial settings makes anomaly detection in smart grids very challenging.

This research focused on how power system security can be improved by detecting FDI attacks against WAMSs. We investigated FDI attacks at different attack entry points of a WAMS, and their impacts on the smart grid system. We also developed an attack model for generating different types of attacks. By providing an in-depth analysis of attack possibilities on WAMSs using attack trees and validating them through experiments and data analysis, we showed that a combination of different statical methods is necessary for effective detection of such attacks.

Figure 11.1 shows an overview of research questions and contributions in this work. Activities performed while addressing the research questions are summarized in the following paragraphs.

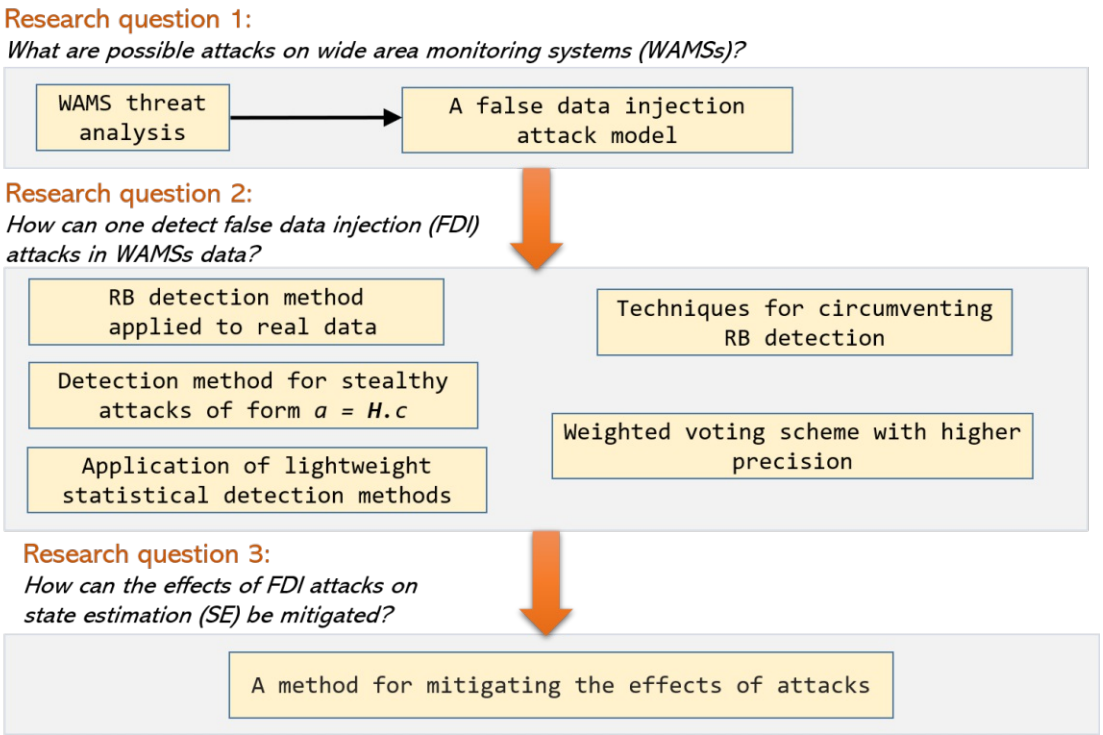


Figure 11.1: An overview of research questions and contributions.

11.1.1 Conclusions for Research Question 1

We investigated various possibilities of data integrity attacks at different points in a WAMS. For this we used a generic model of a WAMS, which is comprised of all possible components that could be present in the system. We considered different attack entry points and elaborated on the consequences for attackers and mitigation strategies. Six attack scenarios were considered based on a hierarchical WAMS structure with the following key components: PMUs, PDCs, super PDCs, PGWs, access and core routers. We analysed their impacts on the system. We identified that there are some gaps, e.g., no existing technique directly addresses PGWs misconfiguration. The different attack entry points, attack scenarios on the key components of a WAMS and their impacts on the system supported answering research question **RQ 1.1**.

We investigated attack models for cyber and physical attacks, which can be combined and can be a part of an APT in a WAMS using attack trees, and provided insights into sub goals that can be used to craft an attack for reaching a higher goal. We pointed out the different aspects (WAMS environment specific) of using physical and cyber means for attacks. Subsequently, we developed a generic attack tree for compromising a device in a WAMS. Based on the generic tree, we provided two specific attack trees in order to trigger wrong control decisions: i) maliciously provoking a blackout and ii) manipulating input data for grid control. The attack trees have been used as a method to model attacks in smart grid environments (e.g., for SCADA systems or smart meters). Since WAMSs provide many attack entry points, we believe that a detailed insight into WAMS attack possibilities is critical for smart grid protection. We considered different devices, their interfaces, hardware and software components in the WAMS, and investigated the possibilities that an attacker can use to launch severe attacks using these components. We described a generic attack model for compromising a device. Using the generic model as a building block, we showed how an attacker can launch specific attacks with two example models: i) provoking a power blackout; and ii) manipulating a phase angle. The attack models showed the different paths for launching the attacks from different entry points. The attack vectors, generic and specific attack trees in the context of a WAMS supported answering research question **RQ 1.1** (How can an attacker cause false data injection attacks in a wide area monitoring system?) and **RQ 1.2** (How can multiple different false data injection attack forms be expressed in one comprehensive attack model?).

Attack models helped us to understand i) the different ways of launching an attack in order to achieve the final goal in different ways ii) the vulnerabilities, security issues, and possible threats on the paths for cyber and physical attacks. The models also helped us to assess which branches are easier to achieve for attackers. Further, they provided strategic guidance for the deployment of suitable countermeasures. Therefore, attack models for WAMS environments provided useful insights to improve wide area monitoring security. The possibilities of launching an attack, vulnerabilities, security issues, possible threats and strategic guidance for suitable countermeasures supported answering **RQ**

1.2.

We developed an FDI attack model that generates types of attacks namely, CO, RO, ICO, IRO, IROMN and ICOHS. Additional attacks (SD, RSCV, IROCV and IROS) were generated by extending attack parameters and their values. The generation of multiple different FDI attack forms using one comprehensive attack model supported answering **RQ 1.2**.

11.1.2 Conclusions for Research Question 2

We showed how an attacker can modify voltage measurements and circumvent detection methods in systems that use Kalman filter based state estimation. Such data integrity attacks on PMU measurements stay under the safety limits and poison the measurements. We used linear state estimation (LSE) with weighted least squares (WLS) and discrete Kalman filter (DKF) using public PMU measurements from a real power grid. We reviewed an anomaly detection method that uses pre-fit residuals from Kalman filter based state estimation to detect anomalous measurement values in a power grid system. We implemented the method in MATLAB. We then showed different scenarios on how an attacker can modify voltage measurements and can hide the attack in the normal operation of the state estimation. The attacks remain stealthy because the attacker slowly makes the changes such that residuals stay below the threshold. The evidence from the experiments showed that one can detect the attacks using alternative methods. For instance, observing the changes in the histograms of the pre-fit residuals, and the evolution of the pre-fit residuals over time, one could yield insights about a potential attack. In order to prevent such stealthy attacks, we suggested to include further detection methods in an overall effort to achieve effective countermeasures. The evidence from the experiments of residual-based bad data detection methods supported answering research question **RQ 2.1** (To what extent can residual-based bad data detection methods detect different FDI attacks?).

We investigated stealthy attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ against state estimation from [100] and presented the stealthiness of the false data injection attacks on the state estimation using the weighted least squares. Under the conditions defined by Liu et al. in [100] for LWLS state estimation, attacks remained stealthy only if the attacker could manipulate all of the measurements considered for state estimation and in our experiments the attacks are detected for DKF state estimation. So with DKFs we are able to detect the stealthy attack. The evidence from the experiments of stealthy attacks of form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ under the conditions defined by Liu et al. in [100] supported answering research question **RQ 2.2** (Can stealthy attacks of the form $\boldsymbol{a} = \boldsymbol{H} \cdot \boldsymbol{c}$ as described in [100] be detected by residual-based methods?).

We presented an investigation into the characteristics of important and distinct methods for detecting false data injection against WAMSs. To this end, we have chosen bad data detection methods and lightweight statistical methods: a) three residual-based approaches

(plain pre-fit residuals, L2-norm and normalized residuals) using pre-fit residuals from Kalman filter based state estimation, as an example of an adaptive bad data detection method, b) a measure of dispersion using the median absolute deviation (MAD), c) a distribution-based approach using the Kullback-Leibler Divergence (KLD) and d) the cumulative sum (CUSUM) as a representative of a change point detection method. After data analysis, we made a selection of representative PMU measurements for our experiment. The selected data was partitioned into historic, training and test data sets. The training data sets were used to derive suitable thresholds for the methods. The measurements that appeared later in time were used as test data. To see how differently these methods perform, we injected six different attack types into the test data sets. Then we analysed the anomaly detection performance using the bad data detection methods and the lightweight statistical methods. In addition, the anomaly detection performance of the statistical methods when using different thresholds were discussed using the ROC curves. Further, we analysed the detection delay of the methods that showed how fast the different methods raise an alarm for the different attack types.

Our findings showed that different methods detect different attack types and there is no single superior method that performs best for all attack types. Traditional residual-based bad data detection methods may be tricked by attackers. For instance, L2-norm does not detect any of the attacks and three of the attacks are not detected by the normalized residual-based method. But if additional anomaly detection methods are applied, the detection performance can be significantly improved even with simple methods for some attacks. CUSUM has better recall than MAD and KLD on CO, RO, ICO, IRO and IROMN attacks as it is better in correctly identifying anomalous data points. KLD has better recall than MAD and CUSUM on ICOHS attack. Our analysis showed that the lightweight statistical methods detected at least one anomaly during all attacks (more data points are detected in many cases). Especially the combination of methods from different detection concepts (e.g., simple threshold, distribution-based, change point detection) proved to be powerful in order to detect a broad variety of attacks. A further important finding is that the detection delay varies a lot and highly depends on the attack type and the method. The analysis of the combined results showed the combination of methods enhanced anomaly detection performance and provided trustworthy results. This needs to be taken into account since this influences the time needed to invoke countermeasures and therefore can influence the damage caused by an attack. From our findings, we argued that grid operators need to combine a set of multiple different methods in order to detect different attack types. It helped in detecting sophisticated attacks, which might be able to bypass a detection method.

We further investigated on the combination of methods to enhance anomaly detection performance. A combination method weighted voting is applied to the anomaly detection results of the three lightweight statistical methods MAD, KLD and CUSUM, then the anomaly detection performance on the six types of attacks were analysed. Our findings showed i) we should use a combination of methods to detect at least some anomalous data point in all types of attacks and ii) we can select a combination method based on the

goal (requirements) of combination, for instance achieving higher precision, detection of at least one data point per attack, higher recall etc. The evidence from the experiments of the lightweight statistical methods (MAD, KLD, CUSUM) and a combination method (weighted voting) supported answering research question **RQ 2.3** (Is it possible to detect the injected attacks with lightweight statistical methods?).

11.1.3 Conclusion for Research Question 3

We investigated mitigating the effects of attacks on state estimation by replacing the anomalous measurement with the predicted values based on past data. An anomalous data replacement scheme was applied on the anomaly detection results from normalized residuals, KLD and weighted voting. The states were estimated based on the corrected measurement. We then performed an anlysis of mitigating the effects of attacks on voltage state estimation.

Our findings showed that the effects of attacks on voltage estimation can be mitigated by correcting measurements before sending them to the estimator. The weighted voting has better mitigation performance than the normalized residual-based and KLD methods, as the combined results have better anomaly detection performance than the normalized residual-based and KLD methods. Another finding is that the detection delay influences the mitigation of the effects of attacks on state estimation. Selection of an appropriate anomaly detection threshold influences anomaly detection delay and thus the mitigation of the effects of attacks.

The evidence from the experiments of state estimation with the corrected data (replacing detected anomalous data by predicted value of Kalman filter) supported answering research question **RQ 3.1** (To what extent can the effects of FDI attacks on state estimation in electric power systems (EPSs) be mitigated by replacing detected anomalies with values derived from past data?).

11.2 Research Outlook and Future Directions

The research questions in this thesis have been addressed by applying several different methods, however, several research topics and future directions have emerged during the course of this research.

11.2.1 Improvements in Anomaly Detection Performance

Anomaly detection performance always depends on the data we use on the methods, parameter settings and changes in the signal. The detection performance can be improved by using more data sets and by adjusting the influencing factors of the anomaly detection methods. For instance, regarding the KLD method, a careful selection of a reference

histogram, its time window, sliding time, threshold etc. can improve anomaly detection performance. Further, frequent changes in the signals (training and test) influences the anomaly detection performance in our scenarios. In particular, setting the thresholds in a suitable way is crucial for the detection performance.

Selection of appropriate values of the method's parameters can also improve the detection performance of combination methods. Further, an appropriate setting of parameters of combination methods also influence the overall detection performance. For instance, an appropriate selection of probability threshold for weighted voting improves anomaly detection performance.

A possible future direction of research could be parameter (influencing factors) optimization of methods to improve performance. We propose using a parameter optimizing method for selecting best parameters of the methods to improve anomaly detection performance (e.g., in [173, 134]). For instance, we could combine different parameter values of the methods and check the combination for better performance.

For more sophisticated attacks also the application of machine learning methods might be useful. But in our scenarios, machine learning methods are not really needed and we consider a desirable property for use in a critical infrastructure setting with few computational resources. There is a lot of space for setting thresholds for the statistical methods, machine learning is an option and would work if we use them for setting thresholds of the statistical anomaly detection methods. For instance, in a similar manner as in [125] we could use machine learning experiments to set thresholds for the anomaly detection methods.

Contextual information of a system can help identifying situations of a system. Combination of the contextual information and the decisions (results) from the different methods can improve overall anomaly detection performance (e.g., in [158, 128]).

11.2.2 Knowledge Based Anomaly Identification System

Highly dynamic measurements gathered from the distributed PMUs are used for state estimation and reflect the dynamic performance of a power system in real time. Data injection attacks in PMU measurements can lead to incorrect SE and invoke wrong control actions. Therefore, an operator needs to know the root cause of an anomaly to apply an appropriate remedy. Early anomaly detection, identifying its cause, and applying appropriate remedy can help avoiding such critical situation.

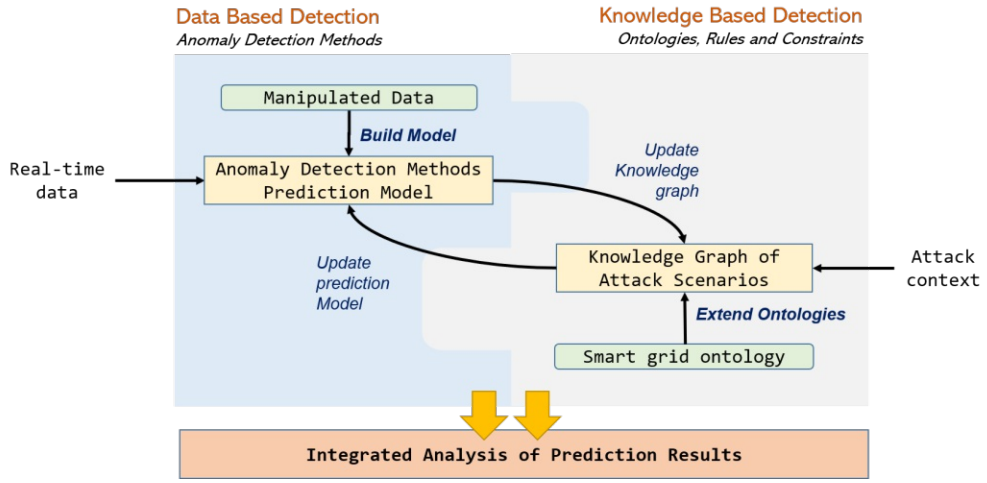


Figure 11.2: A potential next step of the research would be to integrate a knowledge-based approach.

A possible future direction of research could be to integrate contextual information and data analytics results for threat and attack detection. We propose using a knowledge-based anomaly identification system (see Fig. 11.2), which uses an ontology to identify anomalous behaviour. The ontology uses knowledge about the system and decisions from the anomaly detection algorithms for analysis. If an observation is detected as an anomaly then an appropriate remedy is applied before sending it to a control center for estimating states.

APPENDIX A

Appendix

A.1 List of Notations

$\boldsymbol{A}$	State-transition model
$\boldsymbol{B}$	Control input model
$\boldsymbol{H}$	Observation model
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{u}$	Control input
$\boldsymbol{v}$	Measurement noise
$\boldsymbol{v}_k$	Measurement noise (time-variant)
$\boldsymbol{w}$	Process noise
$\boldsymbol{w}_k$	Process noise (time-variant)
$\boldsymbol{Q}_k$	Process noise covariance matrix (time-variant)
$\boldsymbol{R}$	Measurement noise covariance matrix
$\boldsymbol{P}_{k\|k-1}$	Predicted process covariance matrix (time-variant)
$\boldsymbol{P}_{k\|k}$	Process covariance matrix (time-variant)
$\boldsymbol{z}_k$	Real measurement (time-variant)
$\boldsymbol{z}$	Real measurement
$\boldsymbol{z}_v$	Observed measurement
$\boldsymbol{z}_e$	Estimated measurement
$\boldsymbol{z}_{vk}$	Observed measurement (time-variant)
$\boldsymbol{y}_k$	Pre-fit residual (time-variant)
$\boldsymbol{y}_{k/k}$	Post-fit residual (time-variant)

$\boldsymbol{x}_k$	Real state (time-variant)
$\hat{\boldsymbol{x}}_{k\|k-1}$	Predicted state using DKF (time-variant)
$\hat{\boldsymbol{x}}_{k\|k}$	Estimated state using DKF (time-variant)
$\boldsymbol{L}_k$	Kalman gain (time-variant)
γ	Decision level (in DKF)
$\hat{\boldsymbol{x}}_{LWLS,k}$	Estimated state at time step k using WLS
$J(x)$	objective function of weighted least squares
R_{jj}	variance of the j^{th} measurement
G	gain matrix of WLS
$\Delta \boldsymbol{z}$	An offset due to an attack
$\boldsymbol{z}_{k,a}$	Manipulated measurement
$\boldsymbol{z}_{k,c}$	Corrected (substituted) value
$\hat{\boldsymbol{x}}_{k\|k,a}$	Estimated state based on manipulated measurement
$\Delta\hat{\boldsymbol{x}}_{k\|k}$	Difference of estimated state based on manipulated and actual measurements
$\hat{\boldsymbol{x}}_{k\|k,c}$	Estimated state based on corrected (substituted) measurement
$\boldsymbol{S}_{err}$	Sum of difference of estimated voltage during attack and in normal operation
$\boldsymbol{S}_{diff}$	Sum of voltage of state estimation based on substituted value and actual measurement

A.2 Command for KF concept validation

A discrete plant as expressed in [105] as a state space system is defined as

$$Plant = ss(A, [B \quad B], H, D, T, \text{`}inputname\text{`}, \{\text{`}u\text{`}, \text{`}w\text{`}, \text{`}v\text{`}\}, \text{`}outputname\text{`}, \{\text{`}z\text{`}, \text{`}zv\text{`}\}) \tag{A.1}$$

where A is state transition model, B is control input model, H is measurement model, D links real measurement to control input u, as measurement is based only on the measurement model we set D to 0, setting T to -1 marks this model as a discrete model, w is process noise, v is measurement noise and y is real measurement. u and w are inputs to the model and y is output of the model.

A.3 Measurement

A.3.1 Measurement Matrix

A.3.1.1 H for voltage-phasor measurements

Elements of H_V, h_1 to h_4 for multiple buses are represented as A.2

$$h_1{}^{l,re}_{h,re} = \begin{cases} 1 & if\ l = h \\ 0 & if\ l \neq h \end{cases} \tag{A.2}$$

$$h_2{}^{l,re}_{h,im} = 0 \tag{A.3}$$

$$h_3{}^{l,im}_{h,re} = 0 \tag{A.4}$$

$$h_4{}^{l,im}_{h,im} = \begin{cases} 1 & if\ l = h \\ 0 & if\ l \neq h \end{cases} \tag{A.5}$$

In the equations (A.2) - (A.5) superscripts I and h refer to the bus, re refers to the real part and im refers to the imaginary part of the voltage measurements, re refers to the real part and im refers to the imaginary part of the state variables. Here, element $H_1{}^{i,re}_{h,re}$ links real part of the measurement at bus i ($V_{i,re}$) to the imaginary part of the state at the bus ($V_{h,re}$).

A.3.1.2 H for current-injection-phasor measurements

The current injection phasor I_j at bus l is represented as Eq. (A.6)

$$I_j = \sum_{h=1}^{s} Y_{lh} V_h \tag{A.6}$$

Real part of the current injection phasor is represented by Eq. (A.7)

$$I_{j,re} = \sum_{h=1}^{s} [G_{lh} V_{h,re} - B_{lh} V_{h,im}] \tag{A.7}$$

Similarly imaginary part of the current injection phasor is represented by Eq. (A.8)

$$I_{j,im} = \sum_{h=1}^{s} [G_{lh} V_{h,im} + B_{lh} V_{h,re}] \tag{A.8}$$

$H_{I_{inj}}$ can be derived from the equations (A.7) and (A.8) as

$$H_{I_{inj}} = \begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix} \tag{A.9}$$

where

$$h_1{}^{l,re}_{h,re} = G_{lh} \tag{A.10}$$

$$h_2{}^{l,re}_{h,im} = -B_{lh} \tag{A.11}$$

$$h_3{}^{l,im}_{h,re} = B_{lh} \tag{A.12}$$

$$h_4{}^{l,im}_{h,im} = G_{lh} \tag{A.13}$$

A.4 Influence of Phase Angle Variation

In a real power system frequency and phase angle changes. We adopt the changes using a rotation matrix of the system. A rotation matrix defines rotation (clockwise or anticlockwise) of a power system [55, 17]. Rotation matrix for clockwise rotation is defined as R_k:

$$R_k = \begin{bmatrix} cos\theta_k & -sin\theta_k \\ sin\theta_k & cos\theta_k \end{bmatrix} \tag{A.14}$$

We adjust a state transition matrix A_k in a Discrete Kalmam Filter (DKF) model for the real networks applications. The state transition matrix considers rotation of the phase angle θ_k (represented by R_k). For a time step k, A_k is represented by Eq. (A.15).

$$A_k = \begin{bmatrix} cos\theta_k \cdot I & -sin\theta_k \cdot I \\ sin\theta_k \cdot I & cos\theta_k \cdot I \end{bmatrix} \tag{A.15}$$

where θ_k is a phase angle at step k and I is a $n \times n$ identity matrix where n is dimension of a state x_k.

A DKF process model after adjusting A_k for the real system as shown by Eq. (A.16).

$$x_k = A_k x_{k-1} + w_k \tag{A.16}$$

Thus for state estimation considering rotation matrix of the system, the rotation matrix needs to be adjusted in the state transition matrix A_k (see Eq. (A.15)). But the model we use from literature [139] assumes fixed phase angle and time invariant state transition matrix A as an identity matrix. In the literature the model is validated using simulation. Therefore in our use case we also assume constant phase angle and A as an identity matrix where rotation matrix is not used. Here, we recall the DKF process model as

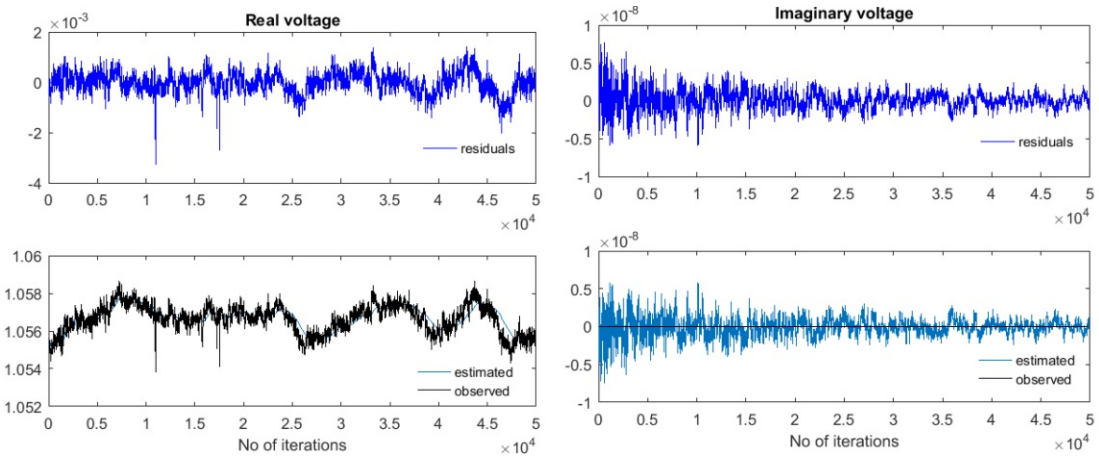
$$x_k = Ax_{k-1} + w_k \tag{A.17}$$

From Eq. (A.15), we can figure out that if the angle (θ) becomes 0 radian then A results to an identity matrix.

For a time step k, real voltage $V_{k,re}$ and imaginary voltage $V_{k,im}$ from polar voltage V_k and phase angle $\theta = 0$ is represented as

$$V_{k,re} = V \cdot cos(0) = V \tag{A.18}$$

$$V_{k,im} = V \cdot sin(0) = 0 \tag{A.19}$$



(a) Real voltage.

(b) Imaginary voltage.

Figure A.1: Estimated real and imaginary voltage (voltage and residuals are in p.u.).

We fix the phase angle value to zero and run the state estimation using DKF. Figure A.1 shows estimated real and imaginary voltages, and the corresponding pre-fit residuals (using the $V_{k,re}$ and the $V_{k,im}$). Sub-figure A.1a shows observed and estimated real voltage and pre-fit residuals. From the sub-figure we can see estimation process smooths out the signal. But from sub-figure A.1b we can see estimated signal is noisy then observed signal.

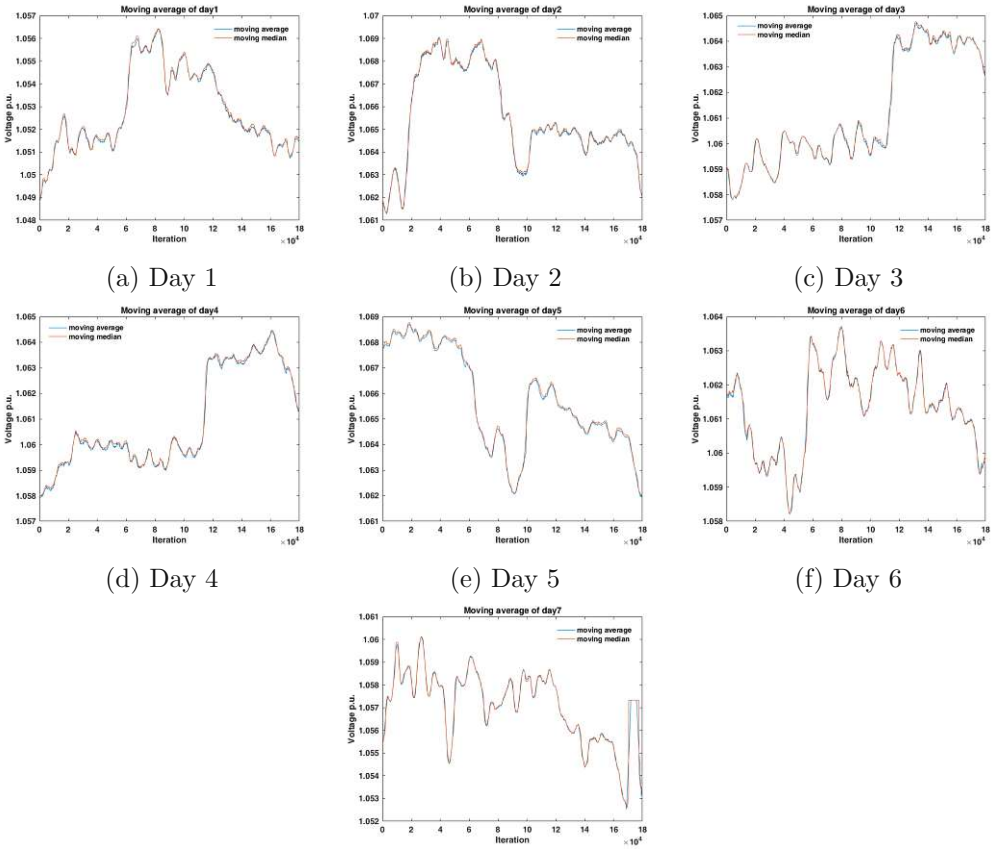
We conclude, state estimation using phase angle zero contrasts to the goal of filtering process. And as we use pre-fit residuals of real-and-imaginary voltage for detecting bad measurements this may not effectively detect bad measurements. Further, we make a use of a pre-fit residual-based approach that has been proposed by Pignati et al. [139]. They use the method for simulation-based data where frequency was fixed and phase angle did not vary. Therefore, here we fix phase angle by the first observed phase angle.

A.5 Moving Average, Median and Variance

Here we present moving average, median and variance of training and test data sets. An average, median and variance of 3,000 data points are considered in a window, and

the window slides in each data point. Thus, a new data point is considered and the oldest data point is removed while moving the window. First, we show moving average, median and variance of training data; second we show moving average, median, and variance of actual test data; then we show the moving average, median and variance of the manipulated test data. To this end, moving average, median and variance of all attacks (CO, RO, ICO, IRO, IROMN, and ICOHS) on the test data are shown.

Figure A.2 shows moving average and moving median of all training data sets. Sub-figures A.2a to A.2g in the Fig. A.2 show moving average and moving median from day 1 to day 7 of the training data.



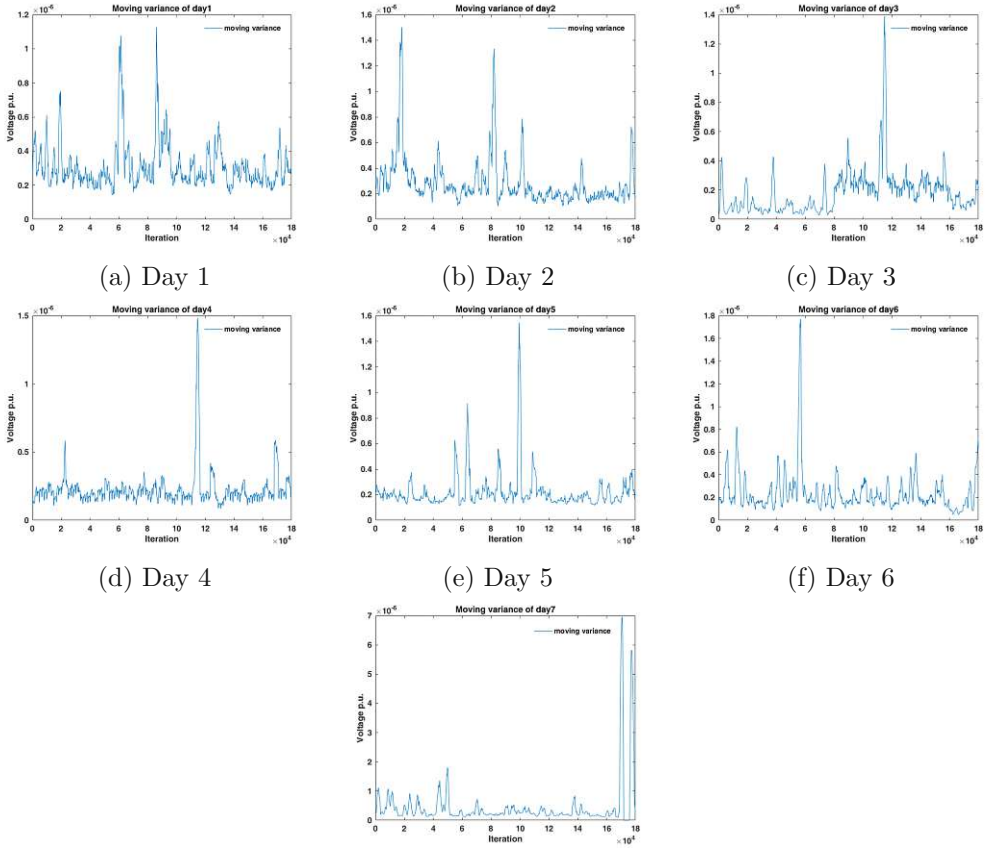
(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7

Figure A.2: Moving average and moving median of training data per day.

Figure A.3 shows moving variance of all training data sets.. Sub-figures A.3a to A.3g in the Fig. A.3 show moving variance from day 1 to day 7 of the training data.



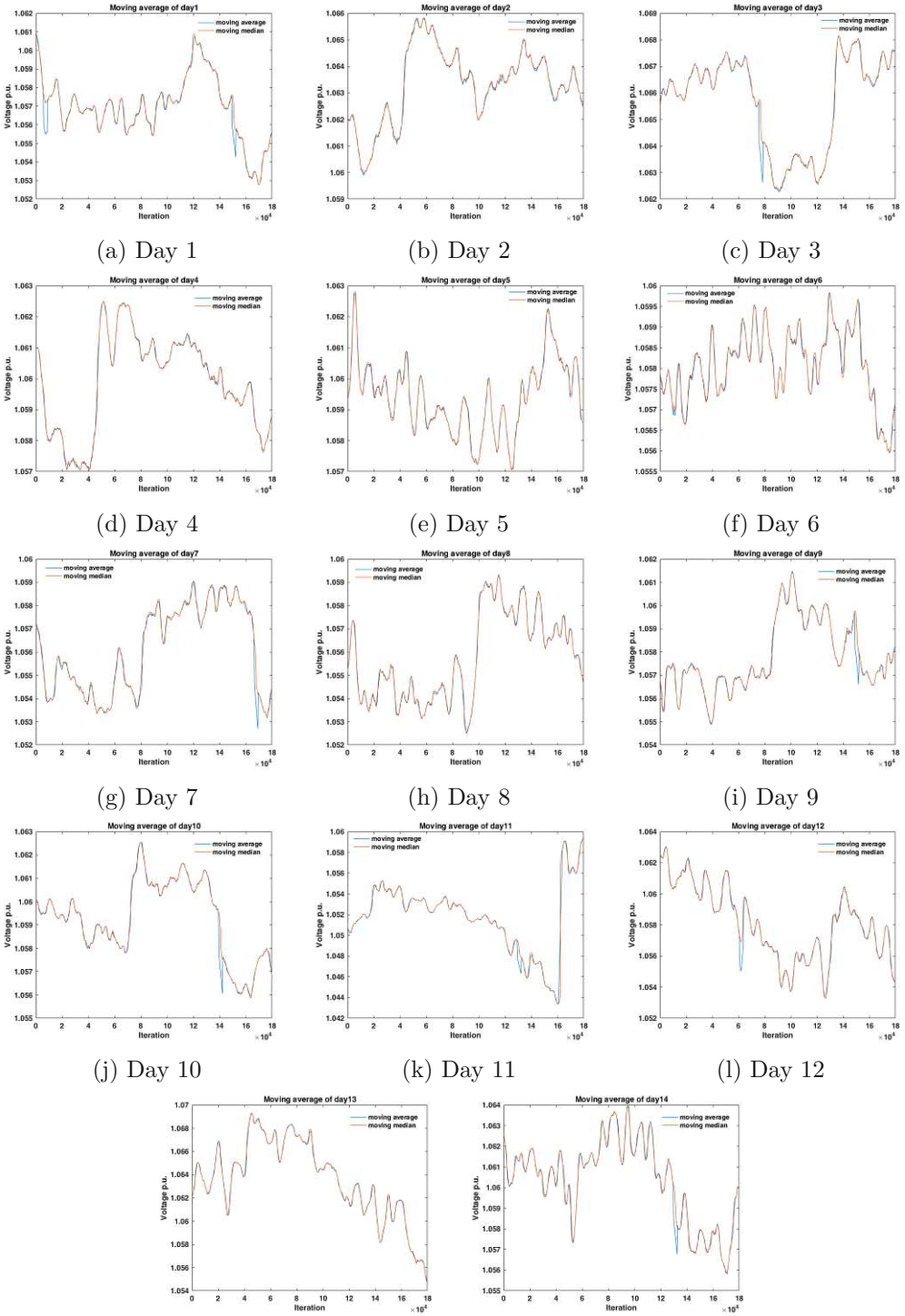
(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7

Figure A.3: Moving variance of training data per day.

Similarly, Fig. A.4 shows moving average and moving median of all actual test data sets. Sub-figures A.4a to A.4 show moving average and moving median from day 1 to day 14 of the actual test data. Figure A.5 shows moving variance of all actual test data sets. Sub-figures A.5a to A.5n in the Fig. A.5 show moving variance from day 1 to day 14 of the actual test data. Figure A.6 shows moving average of the manipulated test data (attacks on the test data). Sub-figures A.6a to A.6n in Fig. A.6 show moving average from day 1 to day 14 of the attacks on the test data from day 1 to day 14.



(a) Day 1 (b) Day 2 (c) Day 3

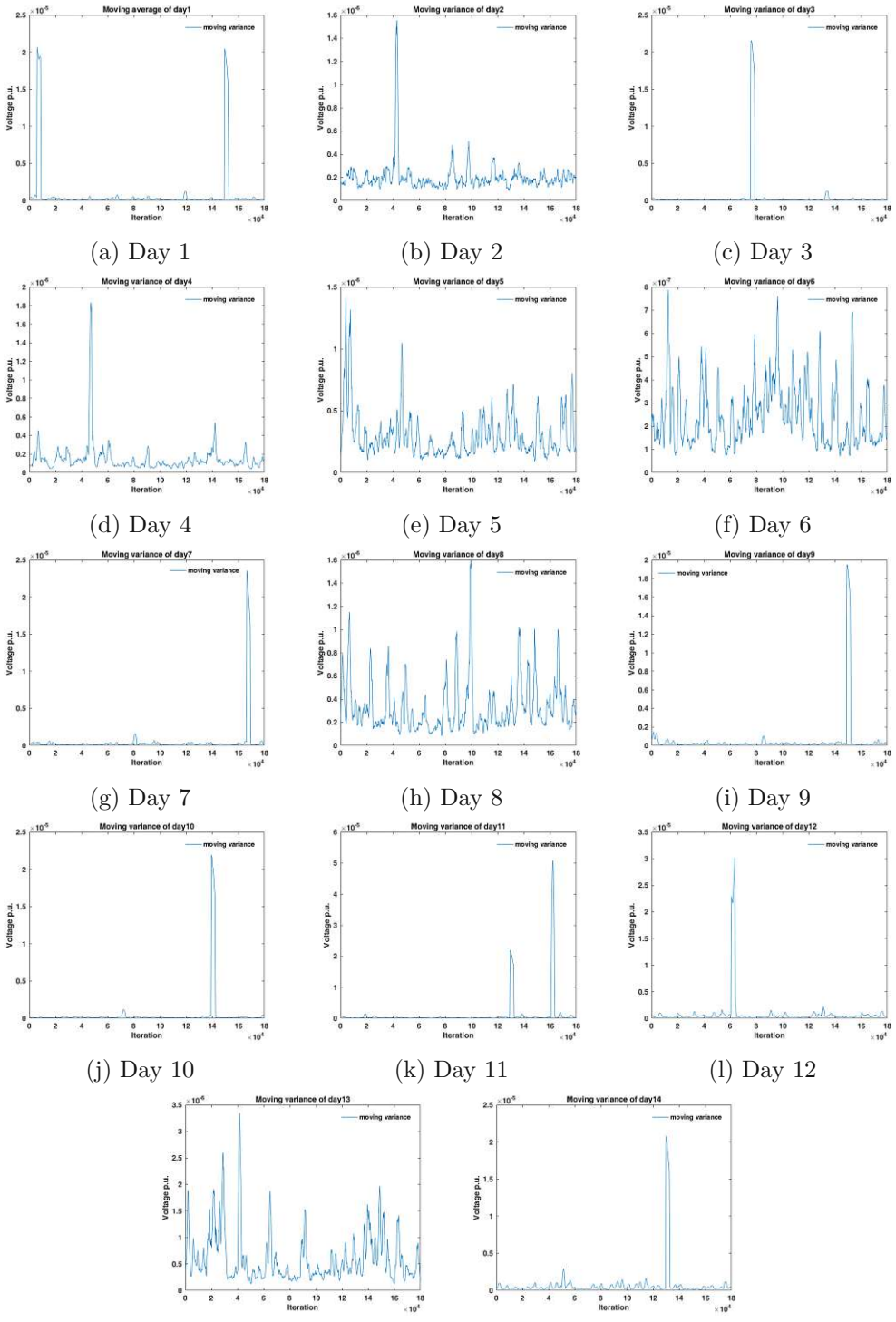
(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7 (h) Day 8 (i) Day 9

(j) Day 10 (k) Day 11 (l) Day 12

(m) Day 13 (n) Day 14

Figure A.4: Moving average and moving median of actual test data per day.



(a) Day 1 (b) Day 2 (c) Day 3

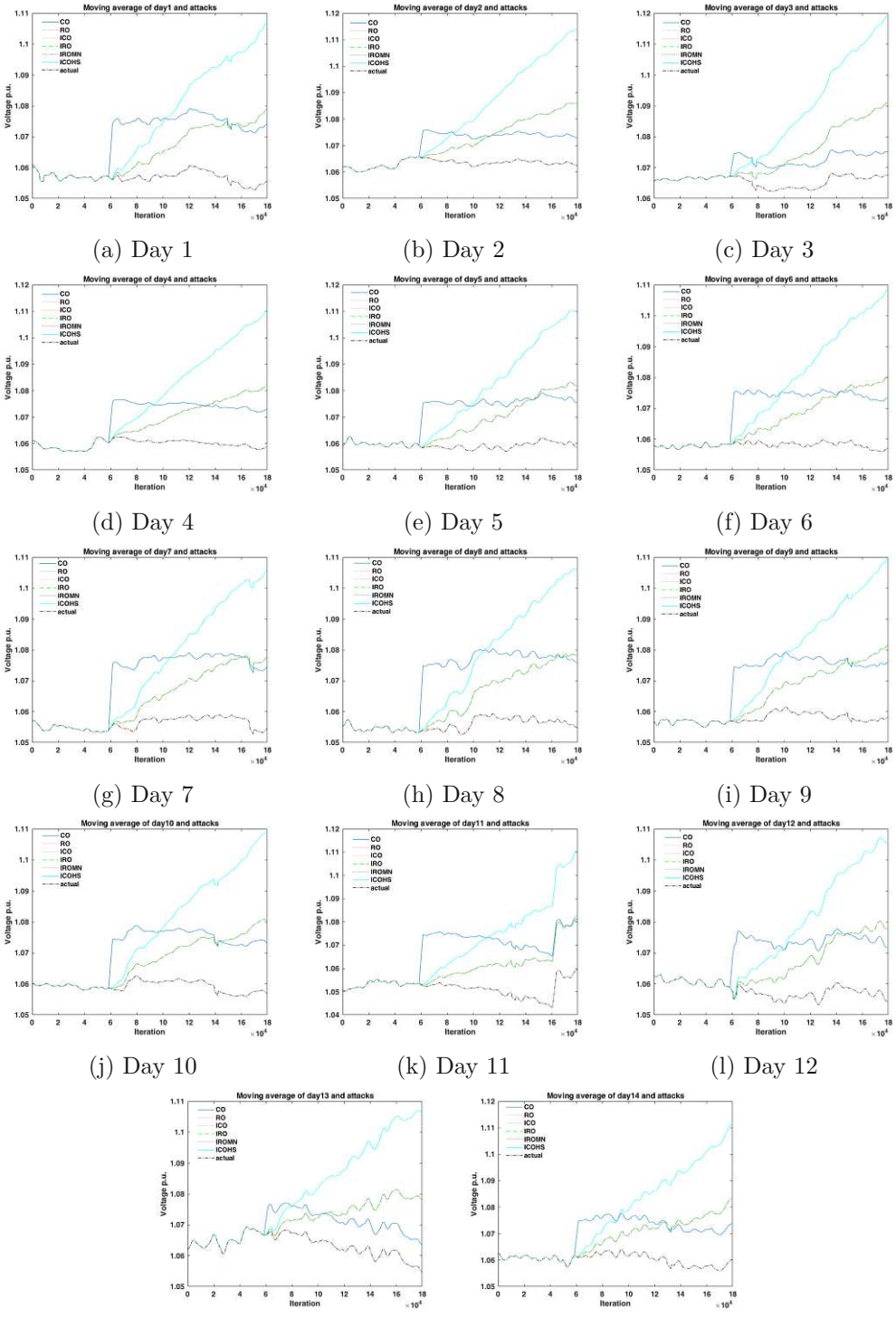
(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7 (h) Day 8 (i) Day 9

(j) Day 10 (k) Day 11 (l) Day 12

(m) Day 13 (n) Day 14

Figure A.5: Moving variance of actual test data per day.



(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7 (h) Day 8 (i) Day 9

(j) Day 10 (k) Day 11 (l) Day 12

(m) Day 13 (n) Day 14

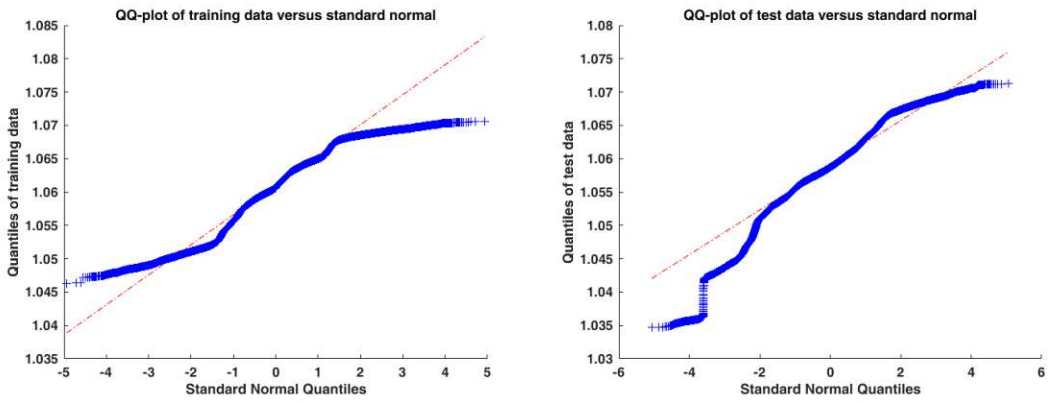
Figure A.6: Moving average of test data with attacks per day.

Our analysis of the moving average, median and variance of the training and test data concludes the following:

- Moving average of actual and RO attack almost overlap.
- Moving average of ICO, IRO and IROMN attacks almost overlap.
- Moving average of ICOHS attack has significant difference to ICO.
- Moving average has significant difference due to slope. Difference in slope changes magnitudes of offsets in one direction, it makes a difference in average value.
- Randomization of signal (adding random negative and random positive offsets) does not have much effect in average. But adding only positive or only negative random offsets can change average and may trigger an alarm.

A.6 Quantile-Quantile plots

Here we compare distribution of training and test data to the standard normal distribution. We visualize quantile-quantile of the given data versus quantile values from a theoretical normal distribution using qqplot in MATLAB. The red lines in the plots represent the theoretical normal distribution and the given data points are plotted in blue + markers. If the distribution of given data points is normal then the data plot appears linear, so that red line and blue data plot are close



(a) Training data (b) Test data

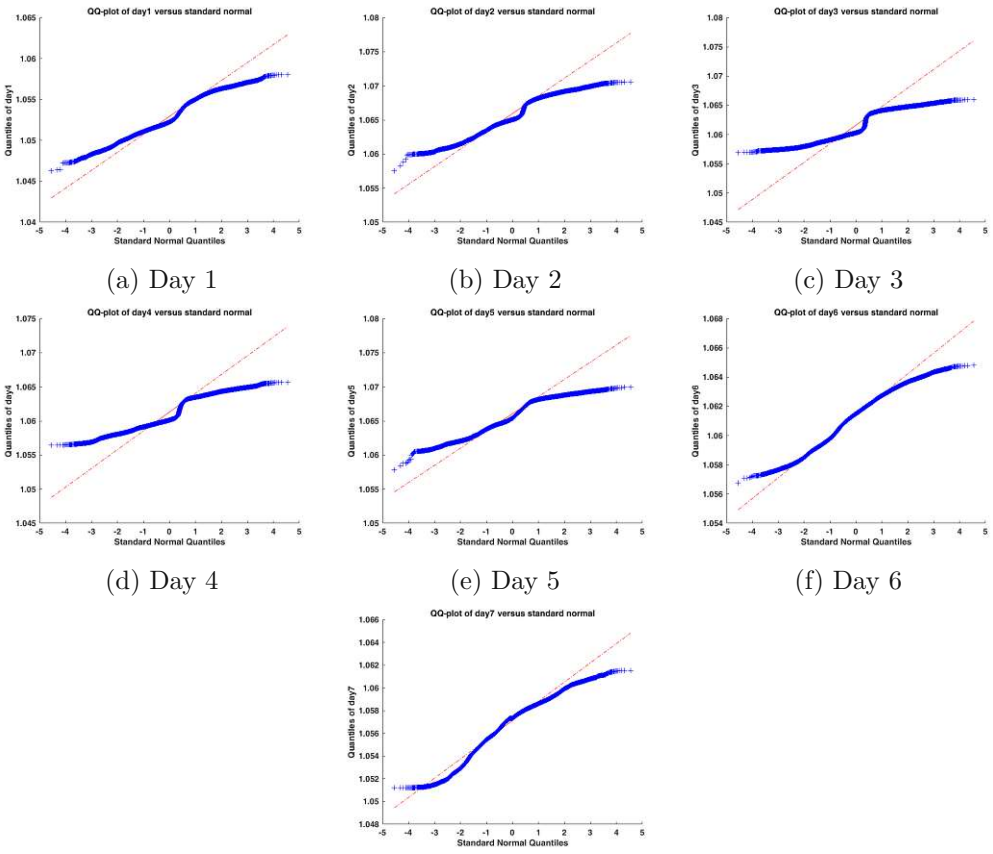
Figure A.7: Quantile-quantile plot of all 7 days training data and all 14 days test data.

Figure A.7 shows quantile-quantile plot of all 7 days training data and all 14 days test data. The quantile-quantile plot of training data (shown in sub-figure A.7a) has non-linear data plot which is an indication that the training data is slightly different than a normal

distribution. From the sub-figure A.7a, we can see the distribution of first and third quantiles of the data is partially normal distributed because blue plot deviates a bit from the theoretical normal distribution at the beginning and end of the plot (these are caused due to the multiple peaks in histogram – see Fig. 6.6a in Sec. 6.4). The second quantile is normally distributed and thus the blue plot produces approximately a straight line that follows the normal distribution (red line). Sub-figure 6.6b shows the quantile-quantile plot of test data.

From the sub-figure, we can see test data is a bit more normal than the training data, but it still differs from a normal distribution. The distribution of first quantile of the test data deviates from theoretical normal distribution (this is due to the tail on left side of histogram – see Fig. 6.6b in Sec. 6.4). The distribution of second and third quantile of the test data is normally distributed.

Figure A.8 shows moving quantile-quantile plot of all training data sets. Sub-figures A.8a to A.8g in the Fig. A.8 show quantile-quantile plot from day 1 to day 7 of the training data.

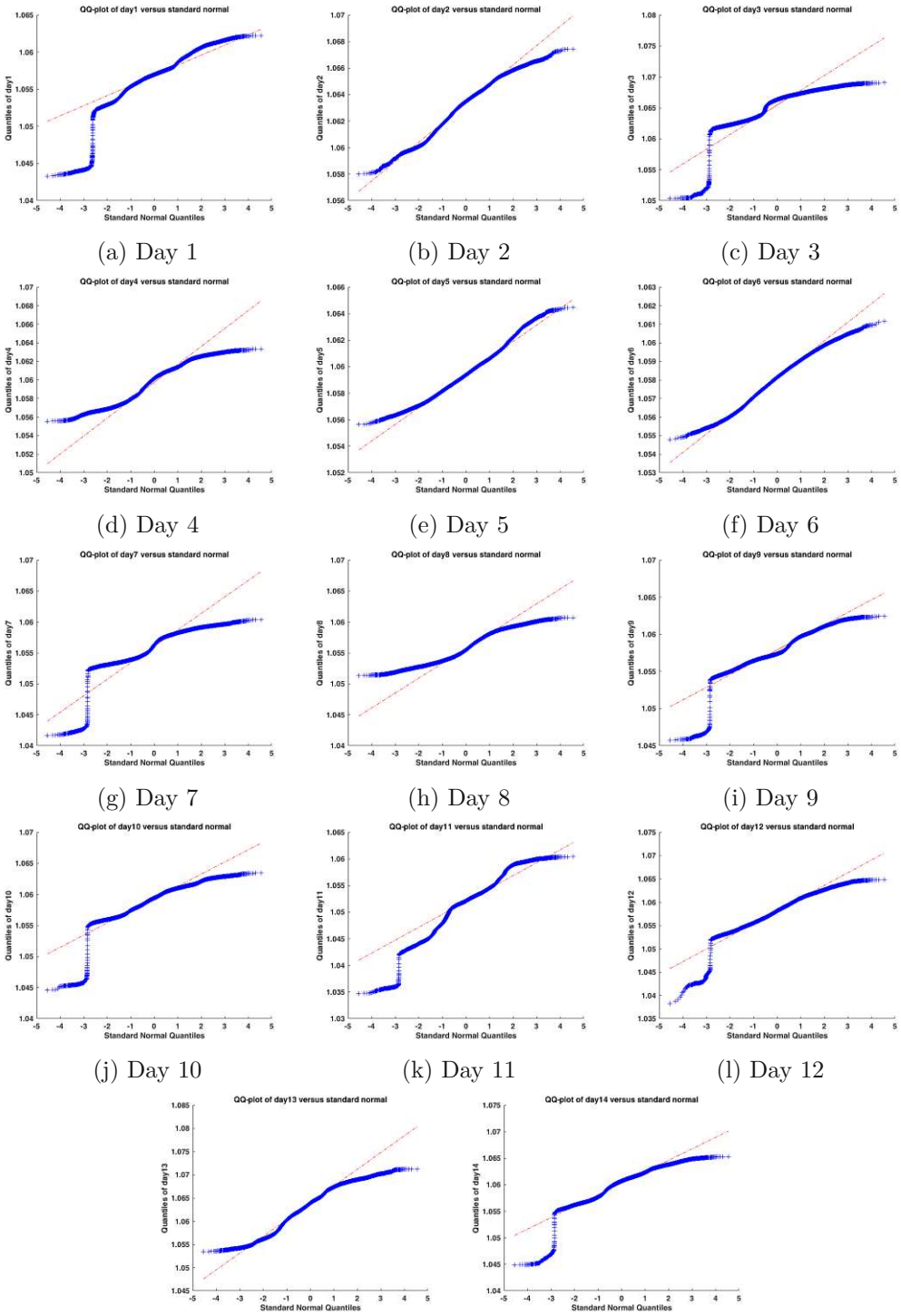


(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7

Figure A.8: Quantile-quantile plot of training data per day.



(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7 (h) Day 8 (i) Day 9

(j) Day 10 (k) Day 11 (l) Day 12

(m) Day 13 (n) Day 14

Figure A.9: Quantile-quantile plot of actual test data per day.

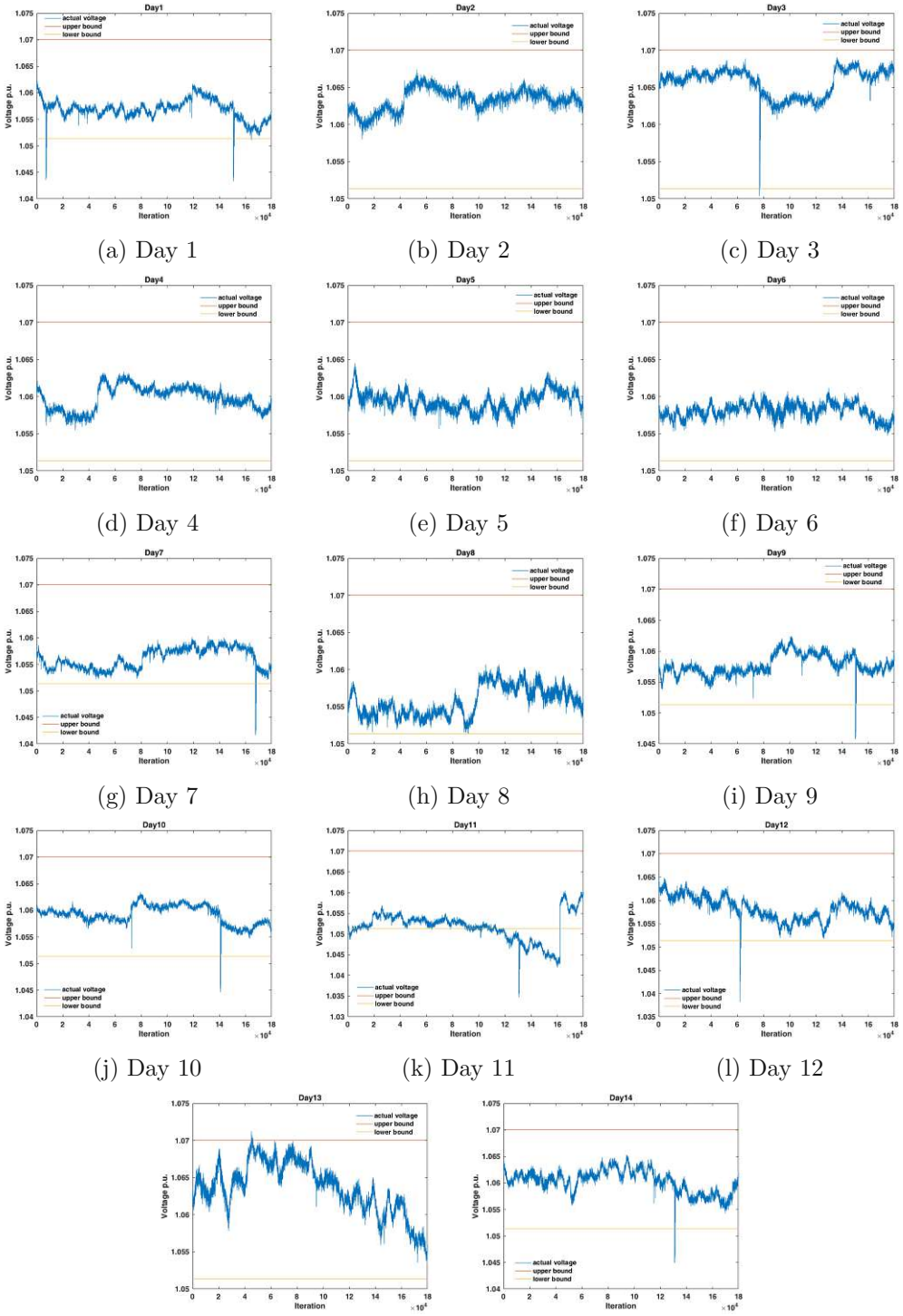
Similarly, Fig. A.9 shows quantile-quantile plots of all actual test data sets. Sub-figures A.9a to A.9 show quantile-quantile plot of day 1 to day 14 of the test data. From the figure, we can see some test data sets are normally distributed (e.g., qq-plots of day 2, day 4, day 5).

A.7 MAD Interval on Test Data

Here we visualize the MAD interval on test data. First, MAD interval are visualized on actual test data then the intervals are visualized on the manipulated test data. We show MAD intervals for all attacks (CO, RO, ICO, IRO, IROMN, and ICOHS) on the test data.

Figure A.10 shows MAD interval on all actual test data. Sub-figures A.10a to A.10n show the MAD intervals from day 1 to day 14 of the actual test data. Figures A.11 to A.24 show MAD interval on test data from day 1 to day 14 with attacks. Similarly, sub-figures in the figures show the MAD intervals on CO, RO, ICO, IRO, IROMN, and ICOHS attacks for the corresponding day.

From the figures one can see, generally the manpulated data points remain outside of the interval if the magnitude of added offset is high. But if the the original signal is close to the defined threshold (upper or lower bound) then small offsets can trigger alarms. An attack with high slope may cross the defined bound earlier and trigger an alarm earlier than an attack with lower slope.



(a) Day 1 (b) Day 2 (c) Day 3

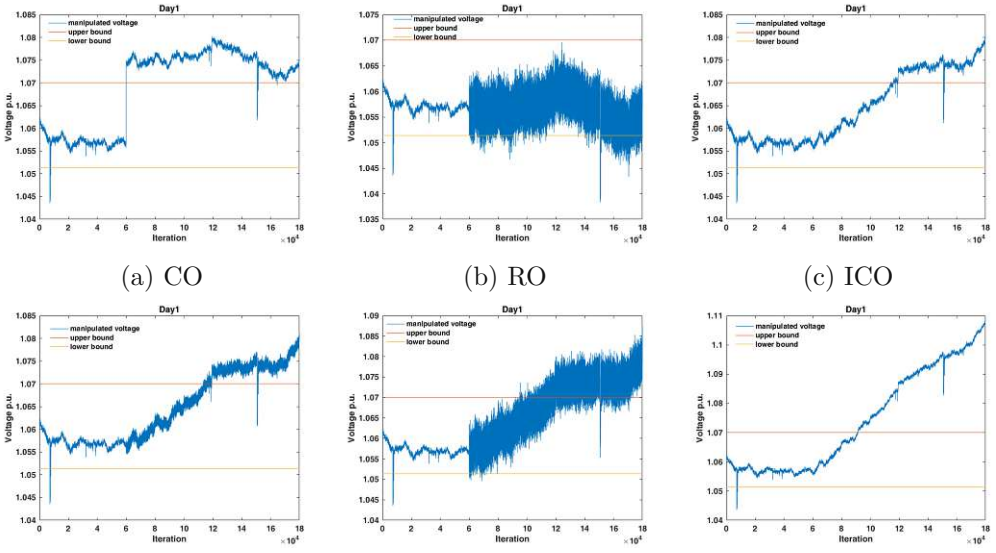
(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7 (h) Day 8 (i) Day 9

(j) Day 10 (k) Day 11 (l) Day 12

(m) Day 13 (n) Day 14

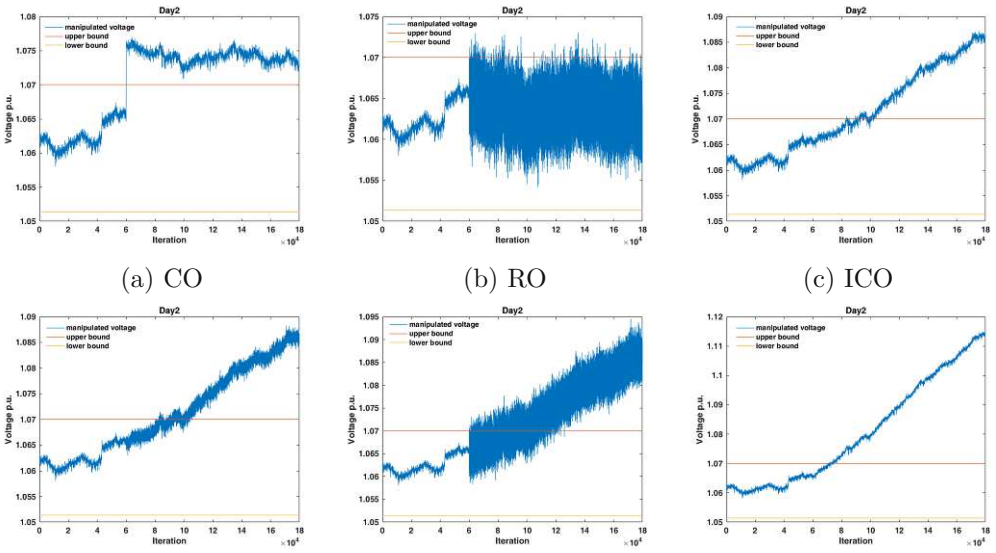
Figure A.10: MAD interval on test data.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

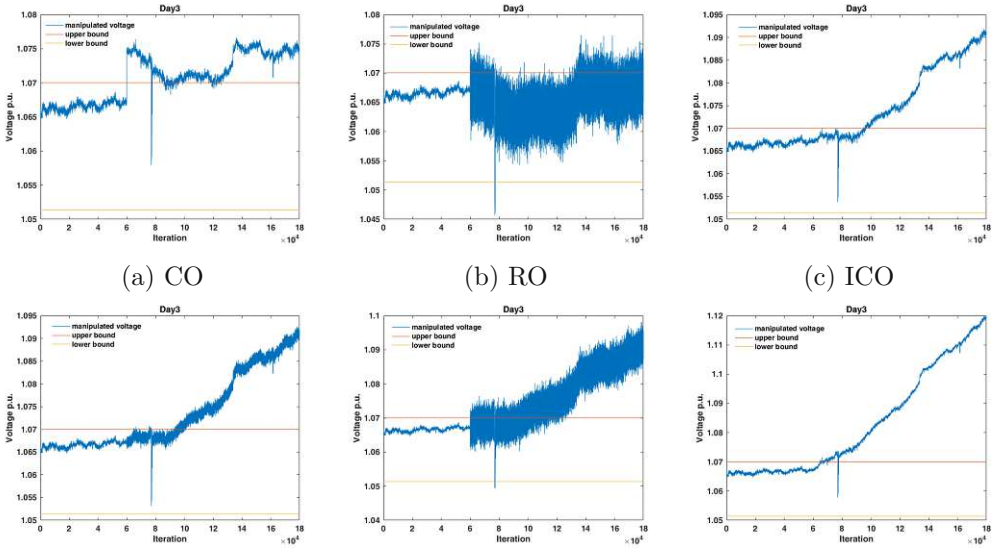
Figure A.11: MAD interval on test data - day 1 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

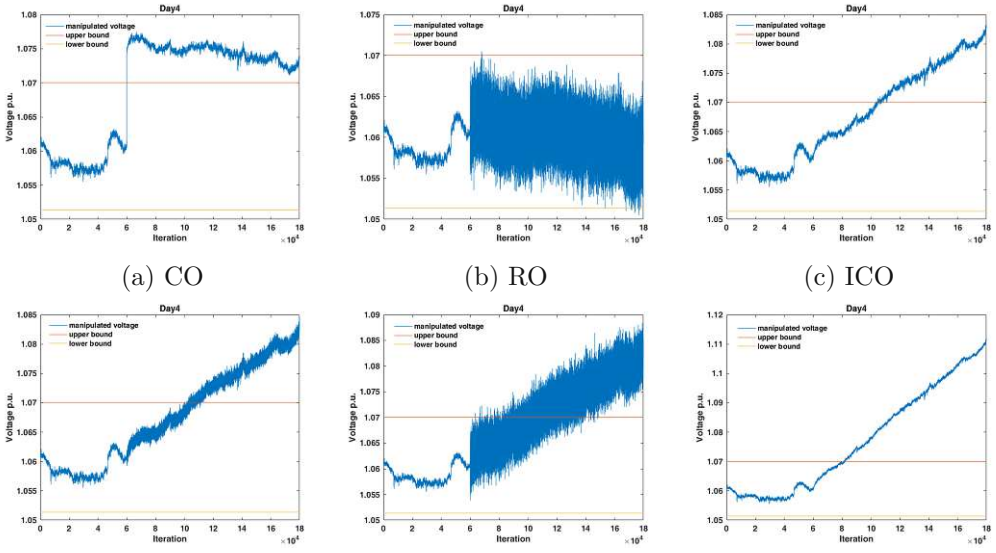
Figure A.12: MAD interval on test data - day 2 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

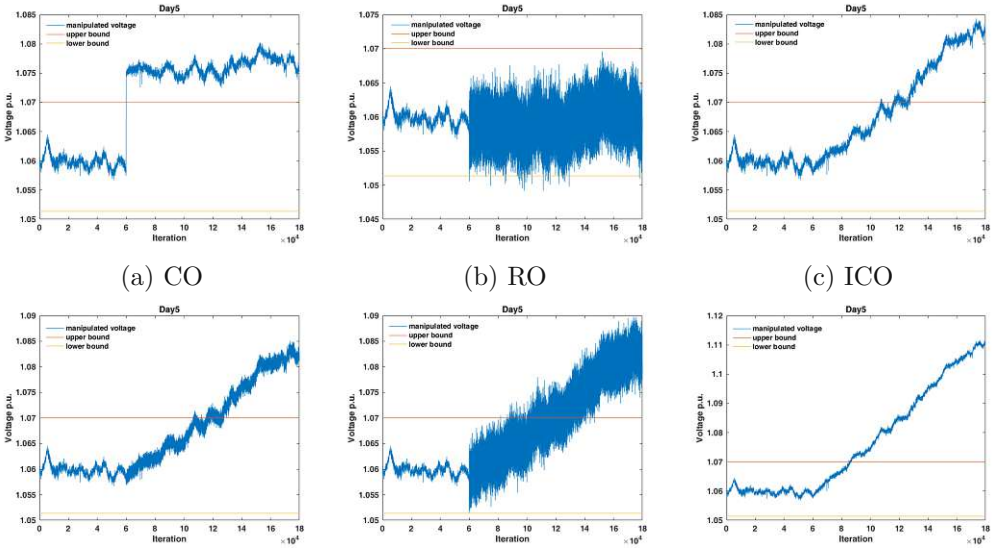
Figure A.13: MAD interval on test data - day 3 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

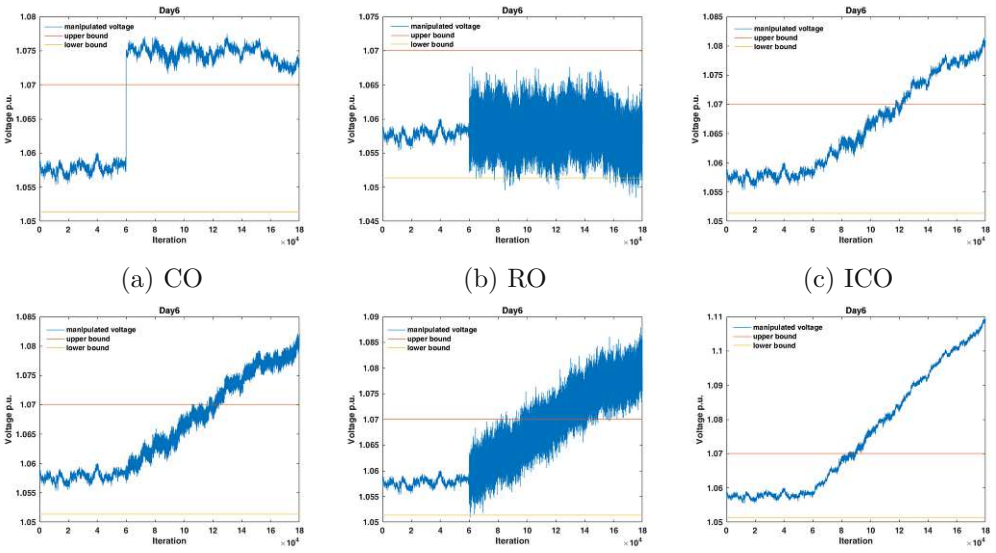
Figure A.14: MAD interval on test data - day 4 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

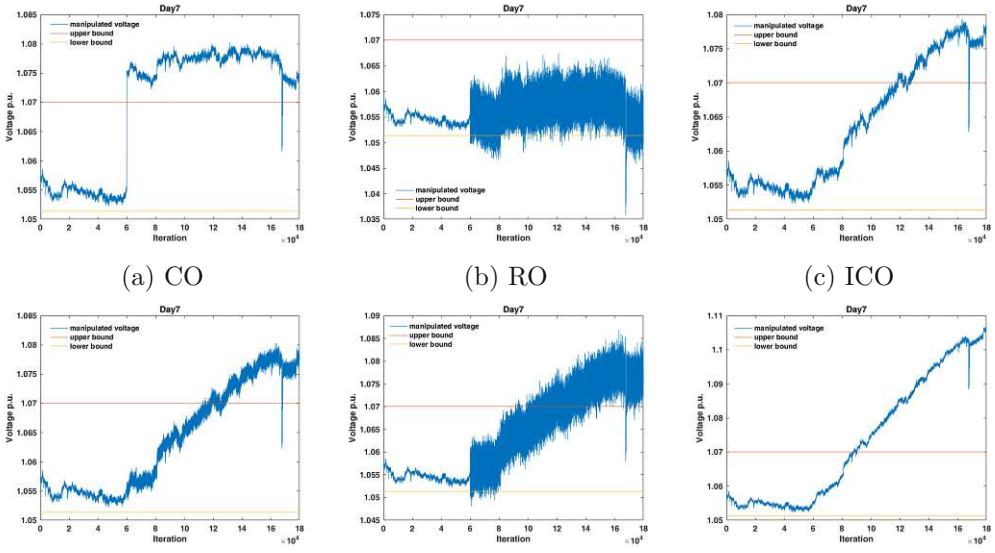
Figure A.15: MAD interval on test data - day 5 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

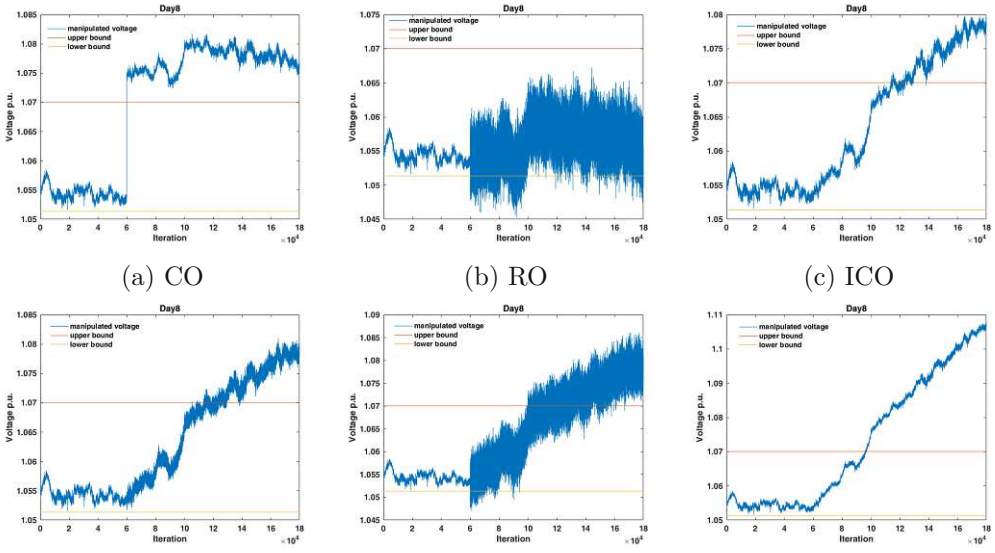
Figure A.16: MAD interval on test data - day 6 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

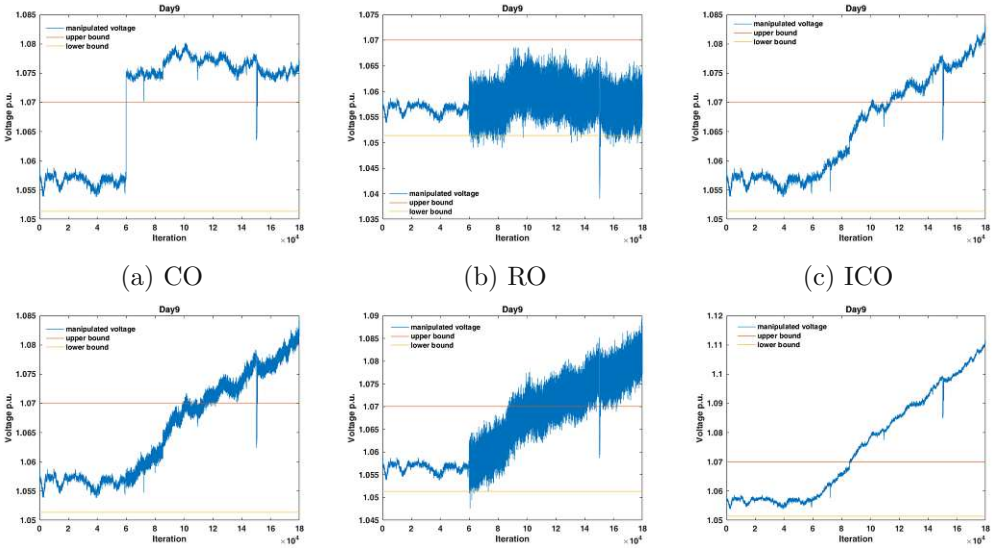
Figure A.17: MAD interval on test data - day 7 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

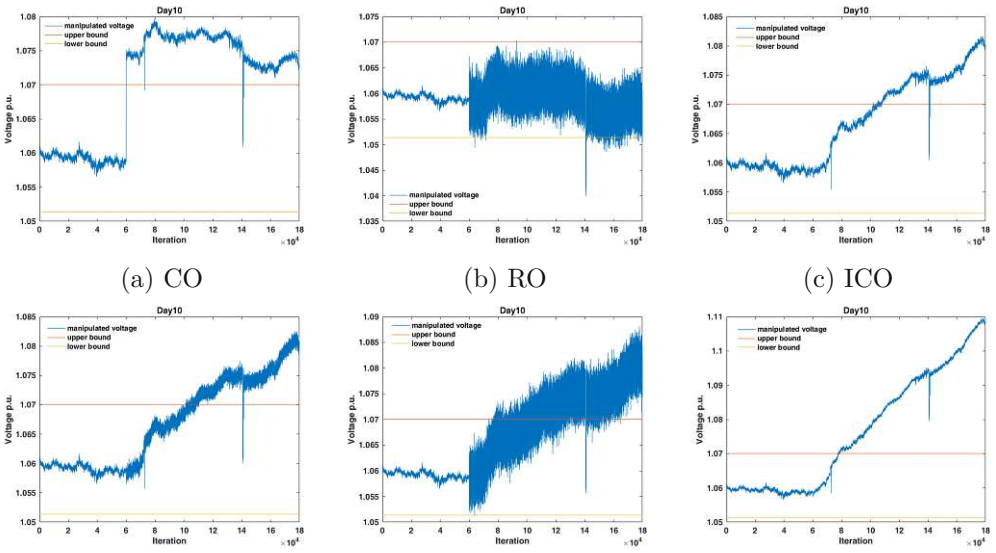
Figure A.18: MAD interval on test data - day 8 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

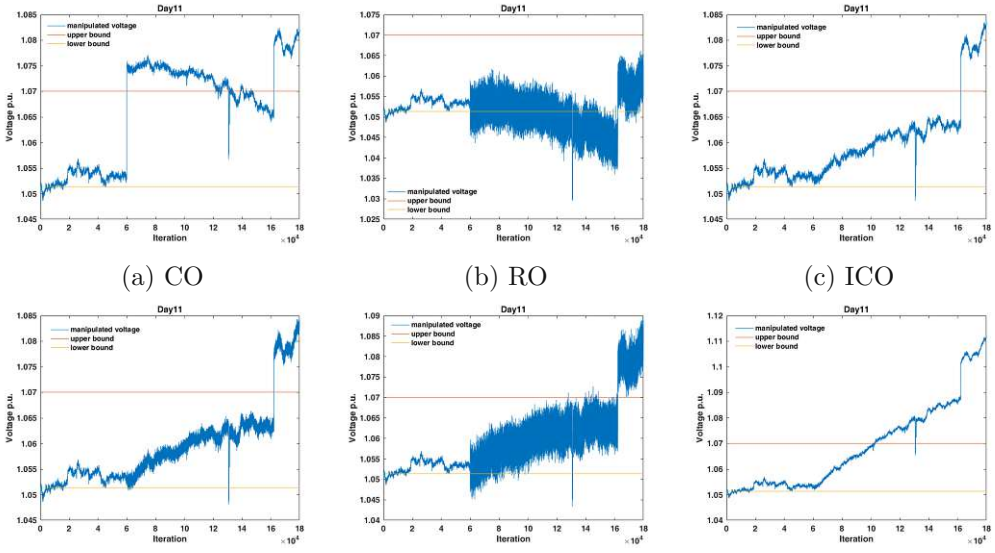
Figure A.19: MAD interval on test data -day 9 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

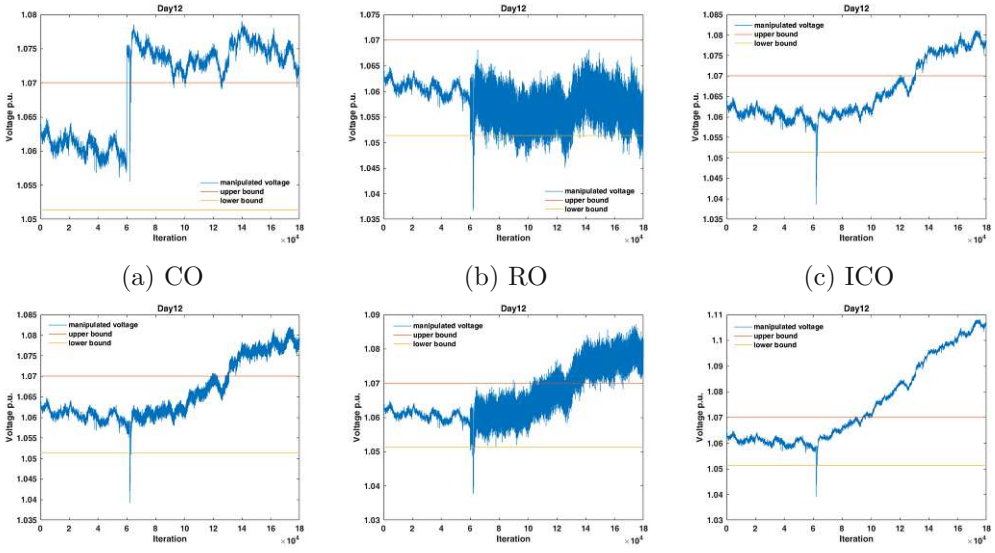
Figure A.20: MAD interval on test data - day 10 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

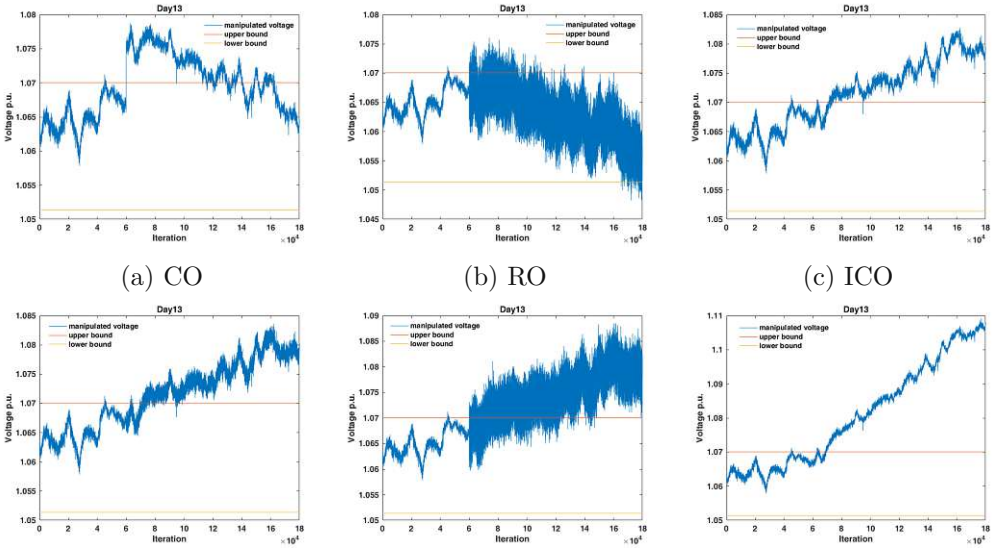
Figure A.21: MAD interval on test data - day 11 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

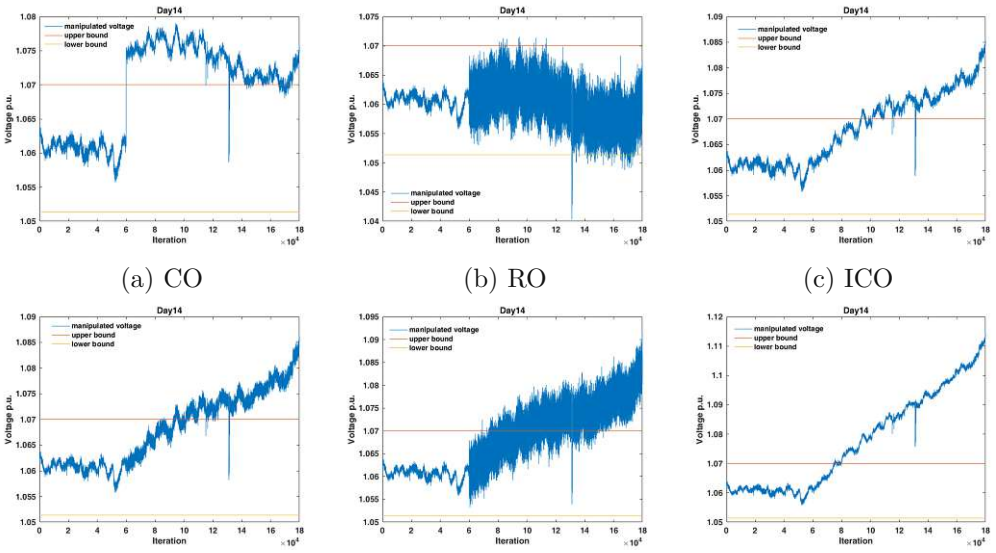
Figure A.22: MAD interval on test data - day 12 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.23: MAD interval on test data - day 13 with attacks.



(a) CO (b) RO (c) ICO

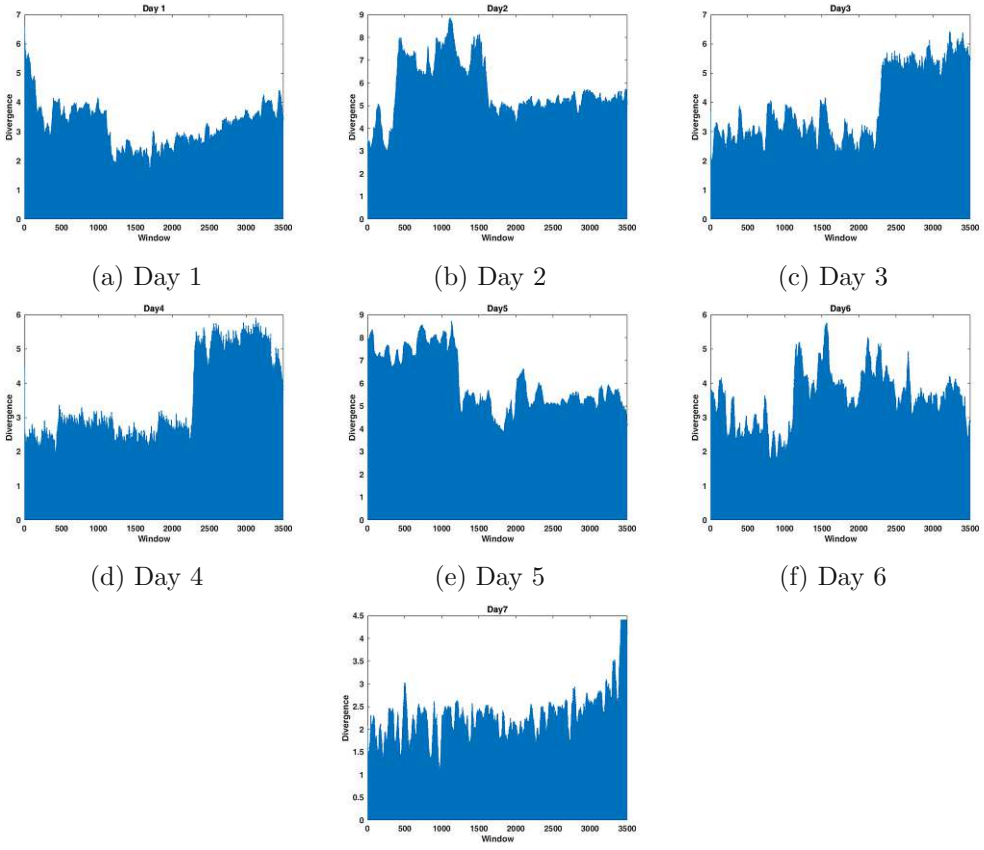
(d) IRO (e) IROMN (f) ICOHS

Figure A.24: MAD interval on test data - day 14 with attacks.

A.8 KLD Sequence

Here we visualize the KLD sequence of training and test data. For the test data, we visualize KLD sequence of the actual and manipulated data. Figure A.25 shows KLD sequence of training data per day. Sub-figures A.25a to A.25g show the KLD sequence from day 1 to day 7 of the training data. Similarly, figures A.26 to A.39 show KLD sequence of test data with attacks. Sub-figures in the figures show KLD sequences of CO, RO, ICO, IRO, IROMN, and ICOHS attacks for the corresponding day.

An analysis of the figures shows reference data has broad distribution and thus actual test data has high divergence to the reference data. In attack type RO, the manipulated test data is even closer to the reference data and thus the divergence to the reference data decreases. KLD divergence keeps on increasing and stops increasing at some point (i.e., at maximum divergence value). Therefore, we can see that the KLD stops increasing after the maximum divergence value in some figures e.g., in A.26a.

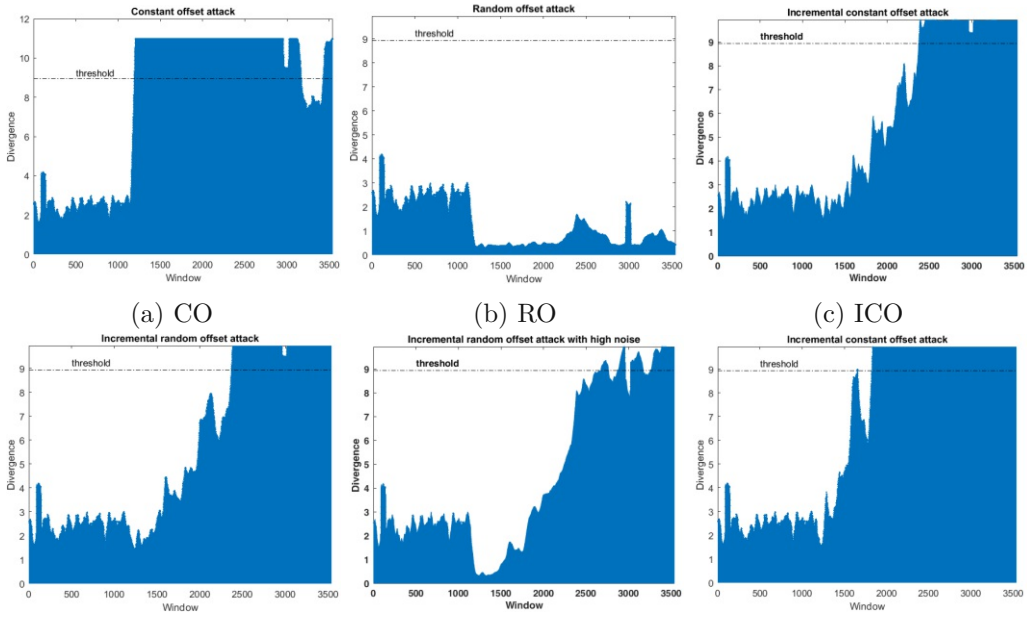


(a) Day 1 (b) Day 2 (c) Day 3

(d) Day 4 (e) Day 5 (f) Day 6

(g) Day 7

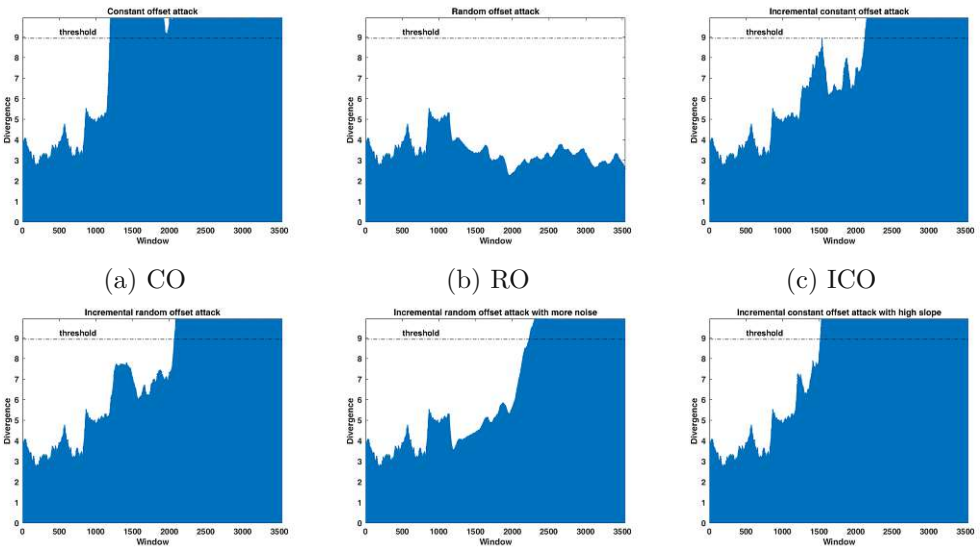
Figure A.25: KLD sequence of training data per day.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

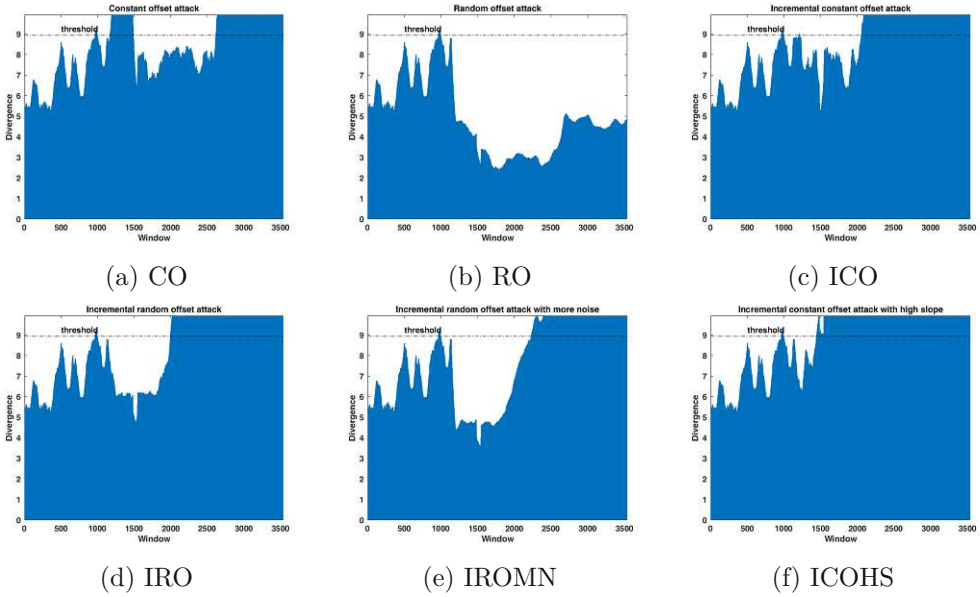
Figure A.26: KLD sequence of test data - day 1 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

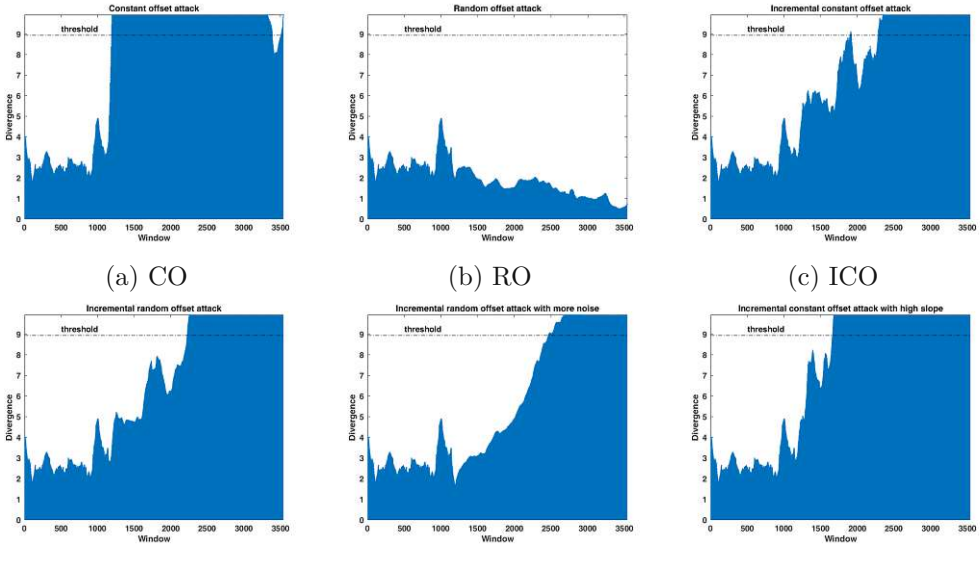
Figure A.27: KLD sequence of test data - day 2 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

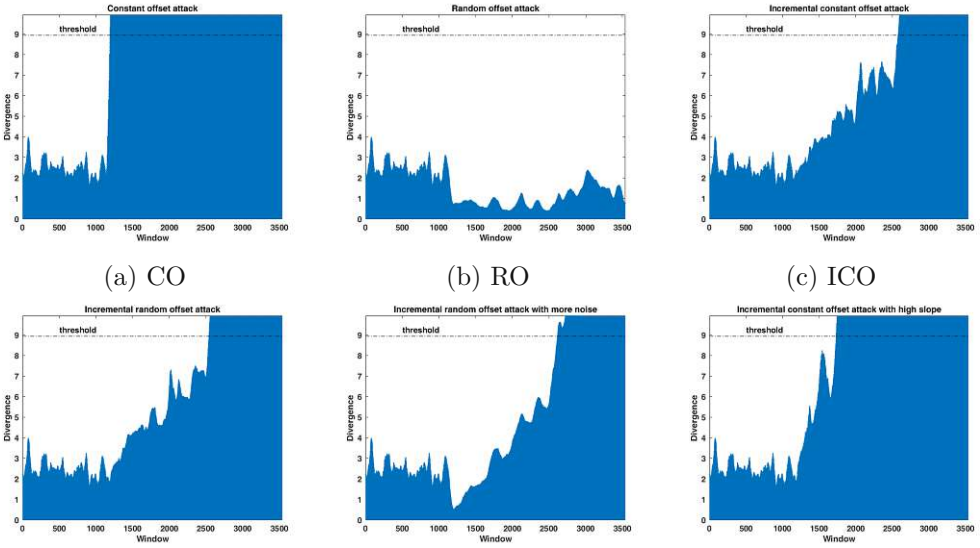
Figure A.28: KLD sequence of test data - day 3 with attacks. Detection of anomalies in RO signal is false positive.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

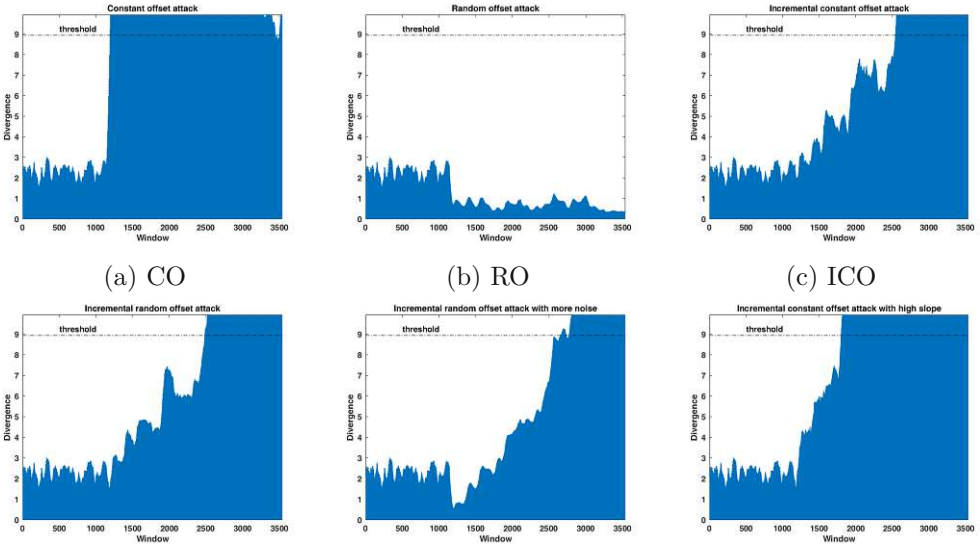
Figure A.29: KLD sequence of test data - day 4 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

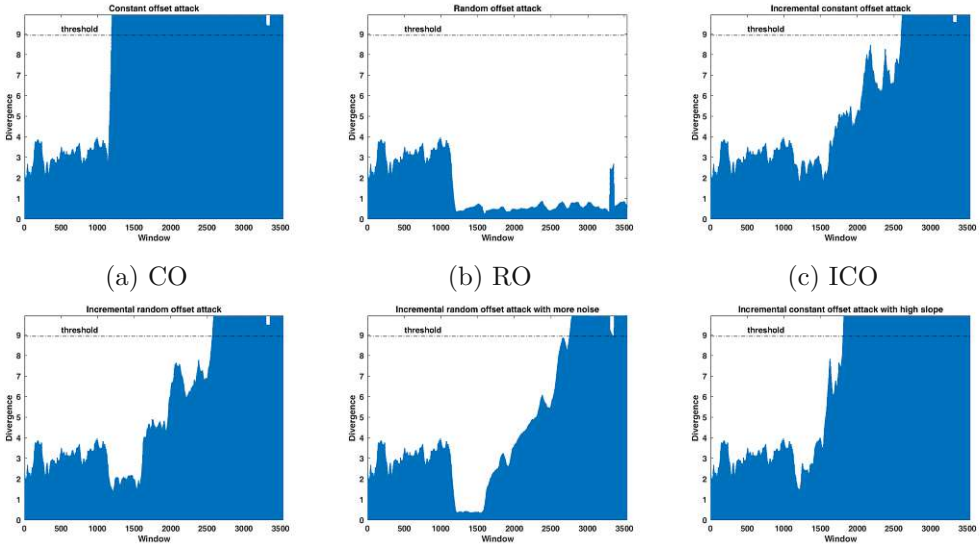
Figure A.30: KLD sequence of test data - day 5 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

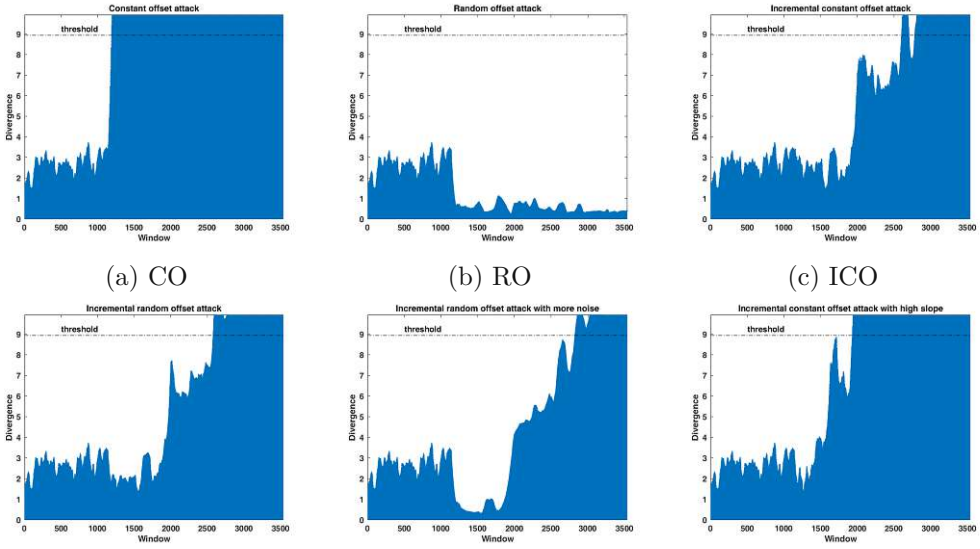
Figure A.31: KLD sequence of test data - day 6 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

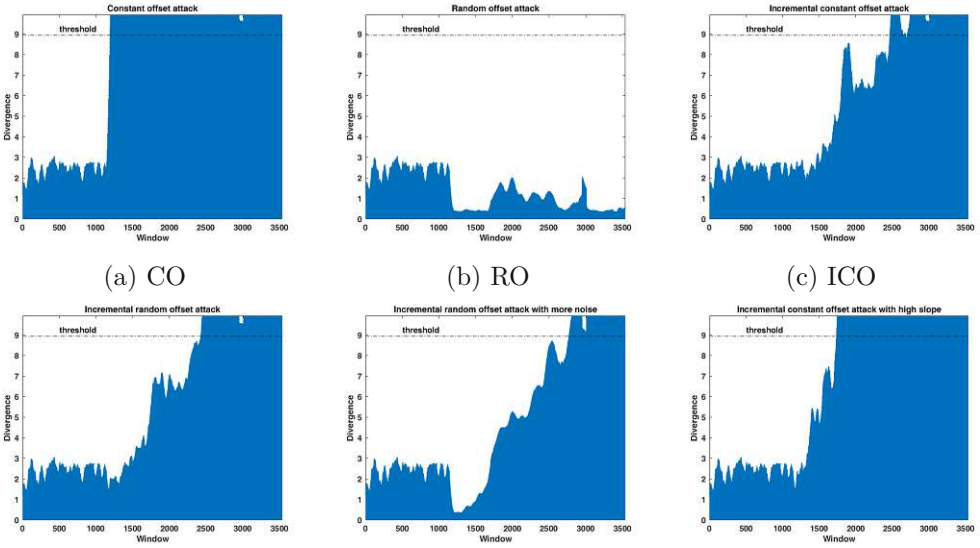
Figure A.32: KLD sequence of test data - day 7 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

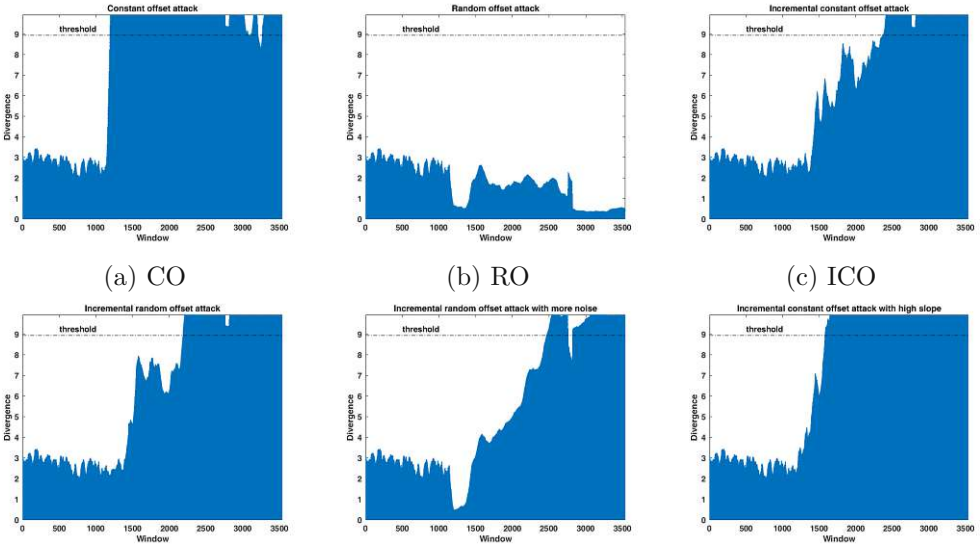
Figure A.33: KLD sequence of test data - day 8 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

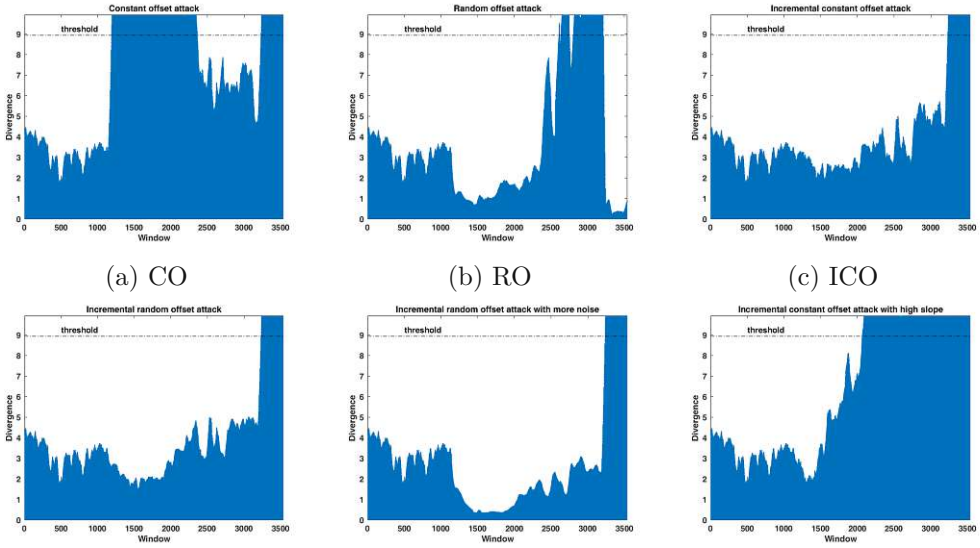
Figure A.34: KLD sequence of test data - day 9 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

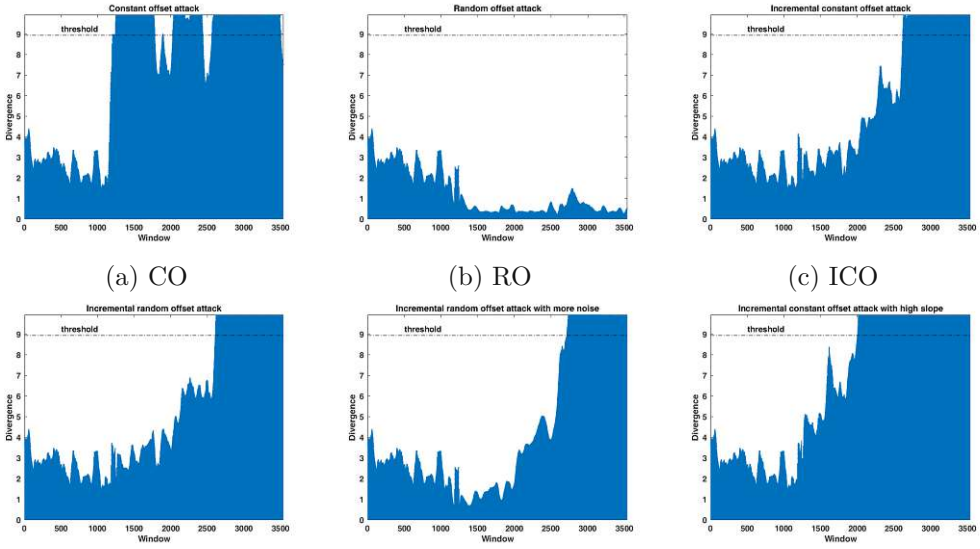
Figure A.35: KLD sequence of test data - day 10 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

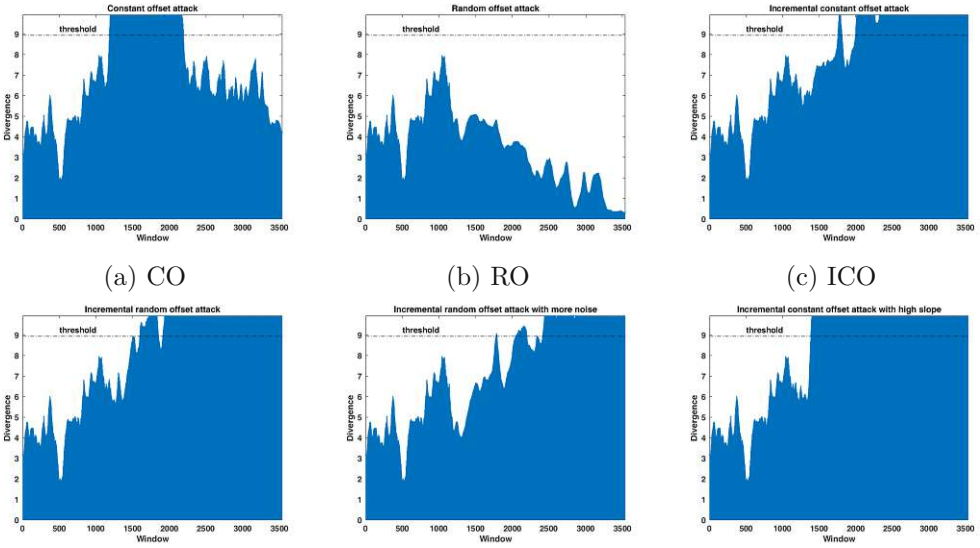
Figure A.36: KLD sequence of test data - day 11 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

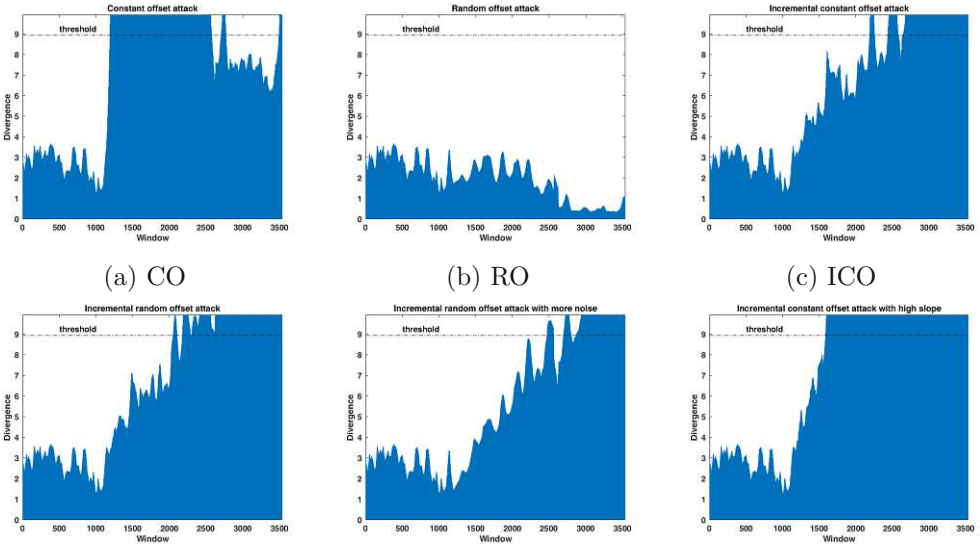
Figure A.37: KLD sequence of test data - day 12 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.38: KLD sequence of test data - day 13 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

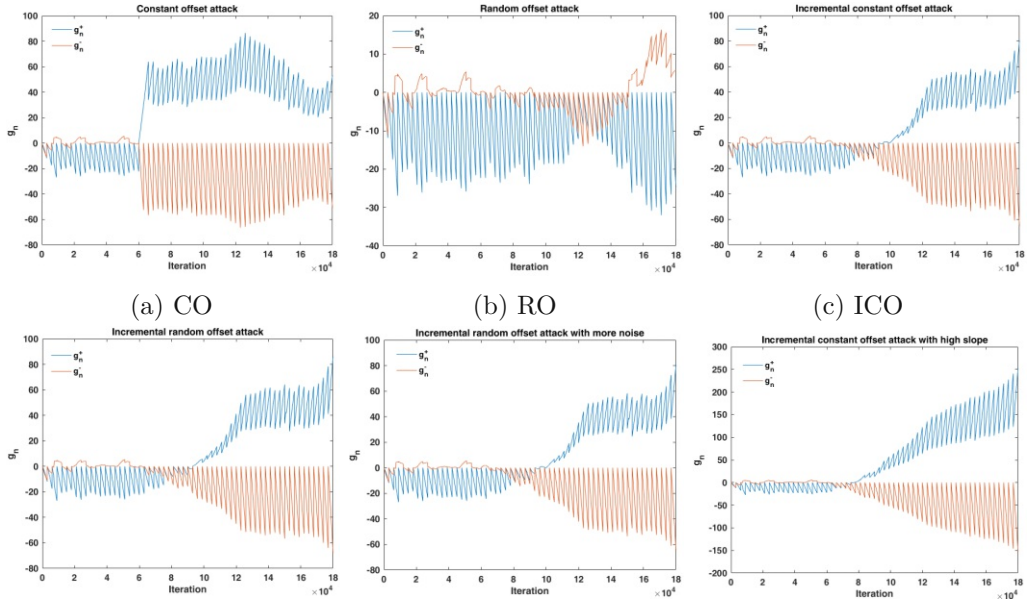
Figure A.39: KLD sequence of test data - day 14 with attacks.

A.9 CUSUM Sequence

Here we visualize the CUSUM sequence of manipulated test data. CUSUM sequence of all attacks (CO, RO, ICO, IRO, IROMN, and ICOHS) on different days of the test data are visualized in separate figures.

Figures A.40 to A.53 show CUSUM sequence of test data from day 1 to day 14 with attacks. Sub-figures in the figures show the CUSUM sequence of the corresponding day.

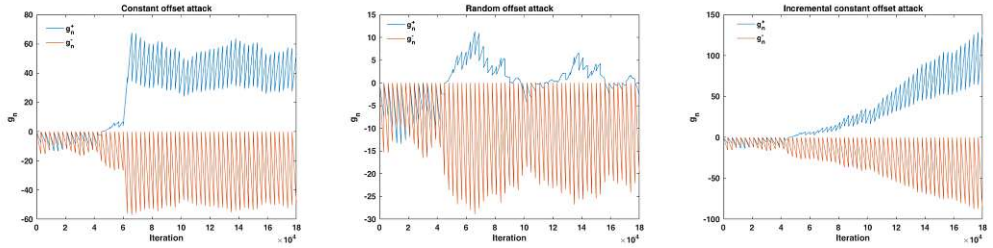
An analysis of the figures shows there are already abrupt changes in trainig data and test data. In addition, g_n^+ and g_n^- of actual signal and RO attack on test data are similar.



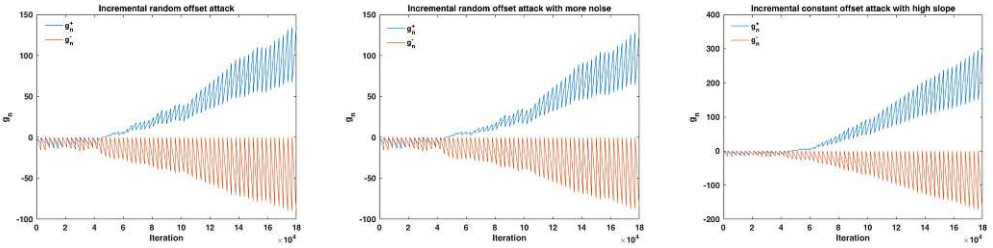
(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.40: CUSUM sequence of test data - day 1 with attacks.

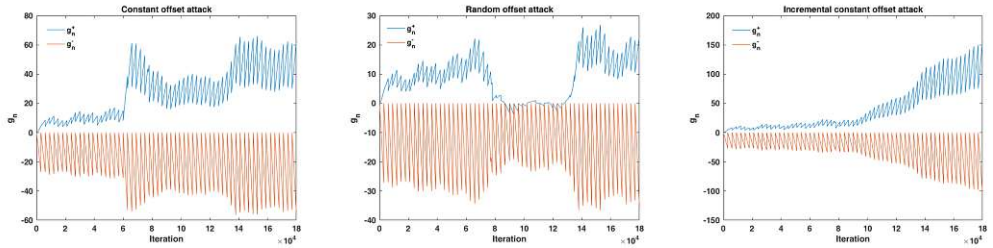


(a) CO (b) RO (c) ICO

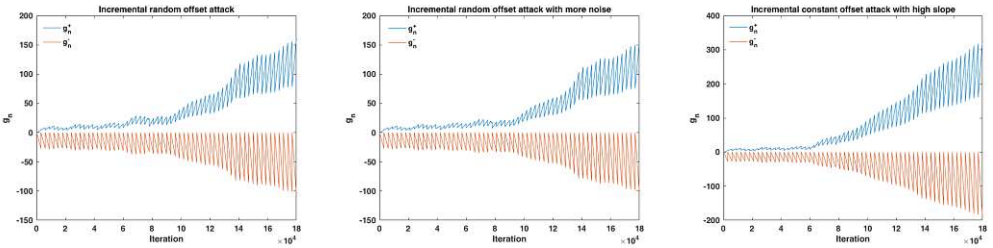


(d) IRO (e) IROMN (f) ICOHS

Figure A.41: CUSUM sequence of test data - day 2 with attacks.

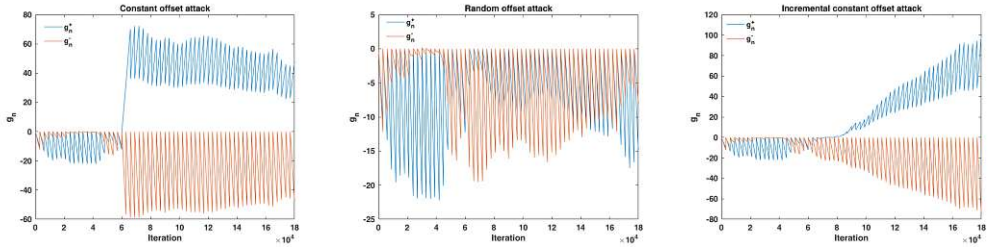


(a) CO (b) RO (c) ICO

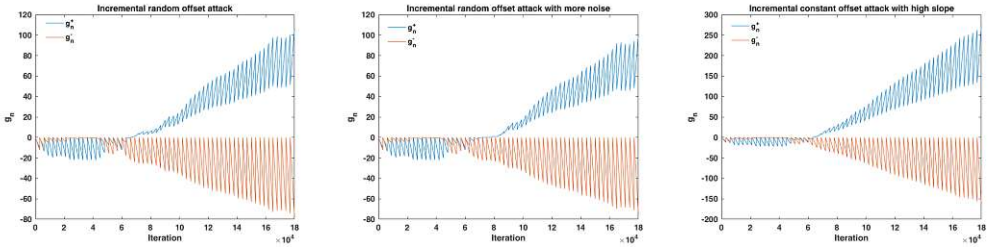


(d) IRO (e) IROMN (f) ICOHS

Figure A.42: CUSUM sequence of test data - day 3 with attacks.

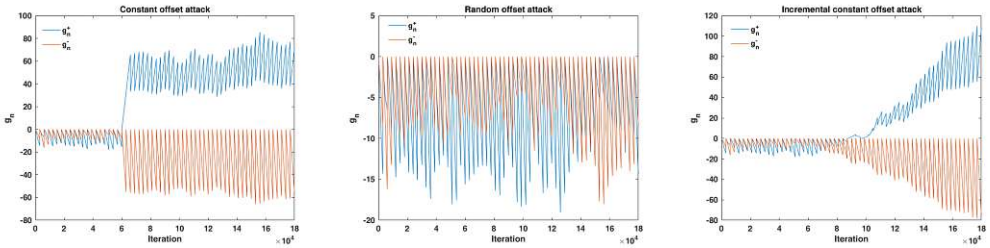


(a) CO (b) RO (c) ICO

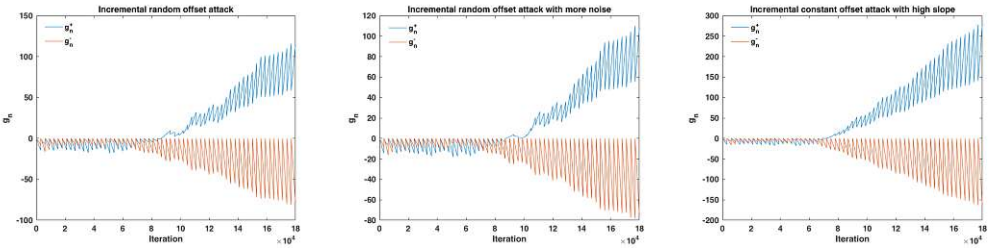


(d) IRO (e) IROMN (f) ICOHS

Figure A.43: CUSUM sequence of test data - day 4 with attacks.

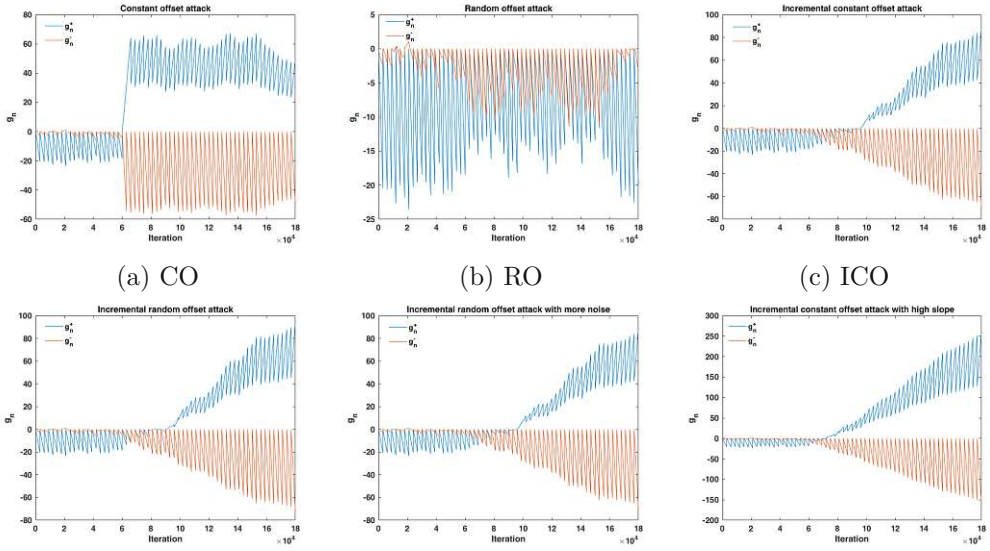


(a) CO (b) RO (c) ICO



(d) IRO (e) IROMN (f) ICOHS

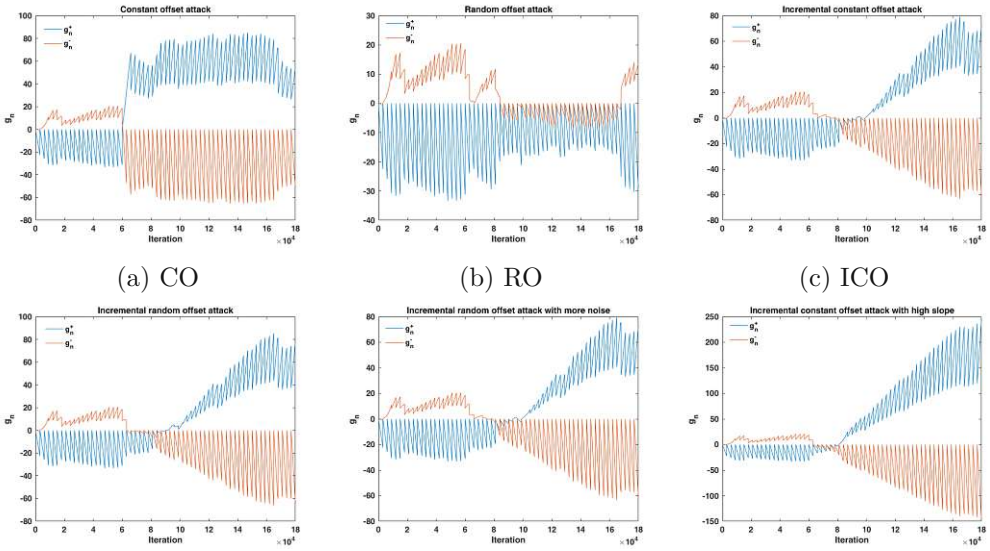
Figure A.44: CUSUM sequence of test data - day 5 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

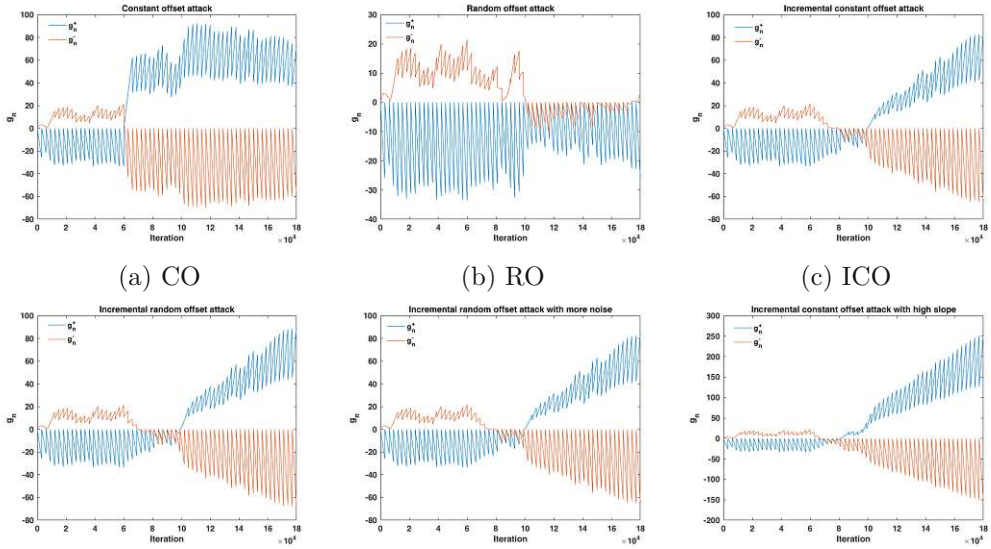
Figure A.45: CUSUM sequence of test data - day 6 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

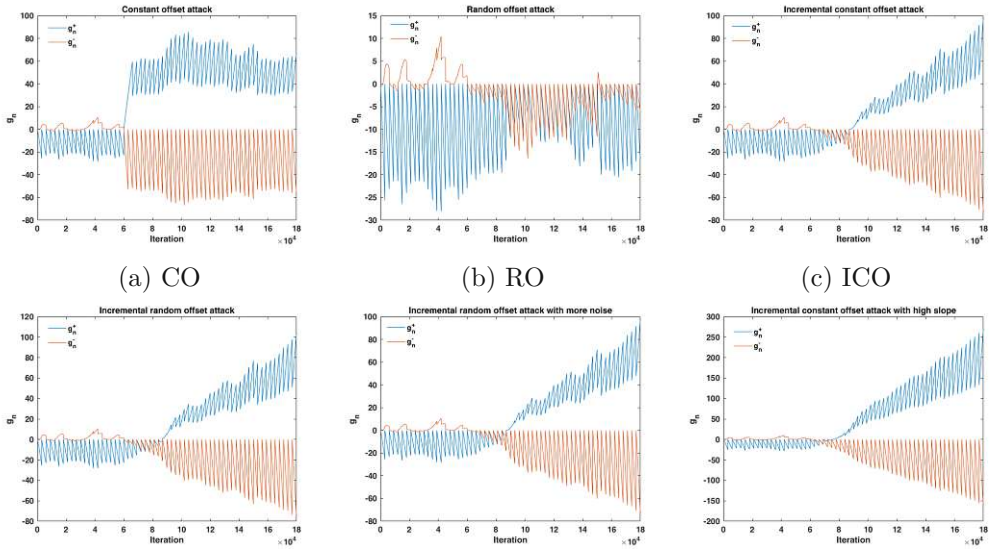
Figure A.46: CUSUM sequence of test data - day 7 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

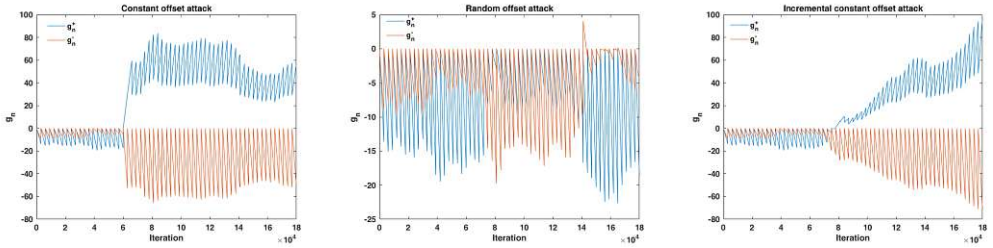
Figure A.47: CUSUM sequence of test data - day 8 with attacks.



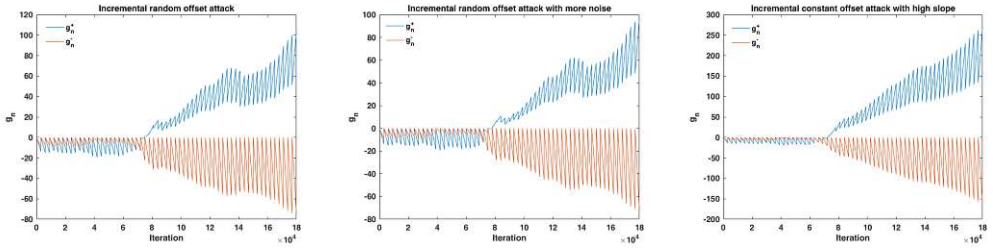
(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.48: CUSUM sequence of test data - day 9 with attacks.

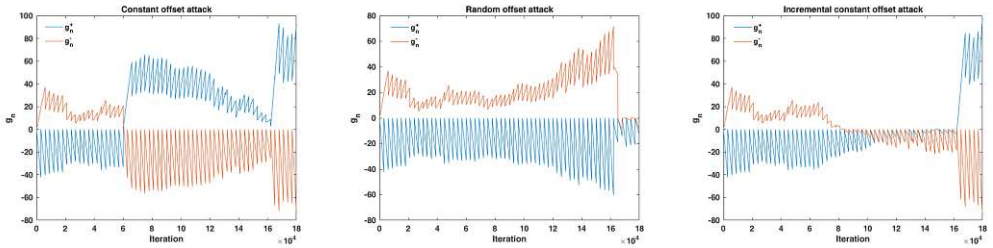


(a) CO (b) RO (c) ICO

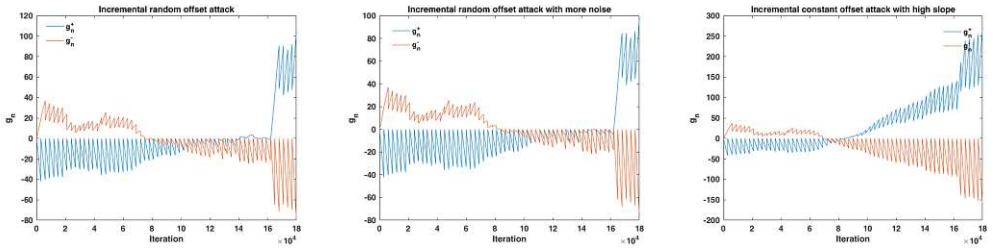


(d) IRO (e) IROMN (f) ICOHS

Figure A.49: CUSUM sequence of test data - day 10 with attacks.

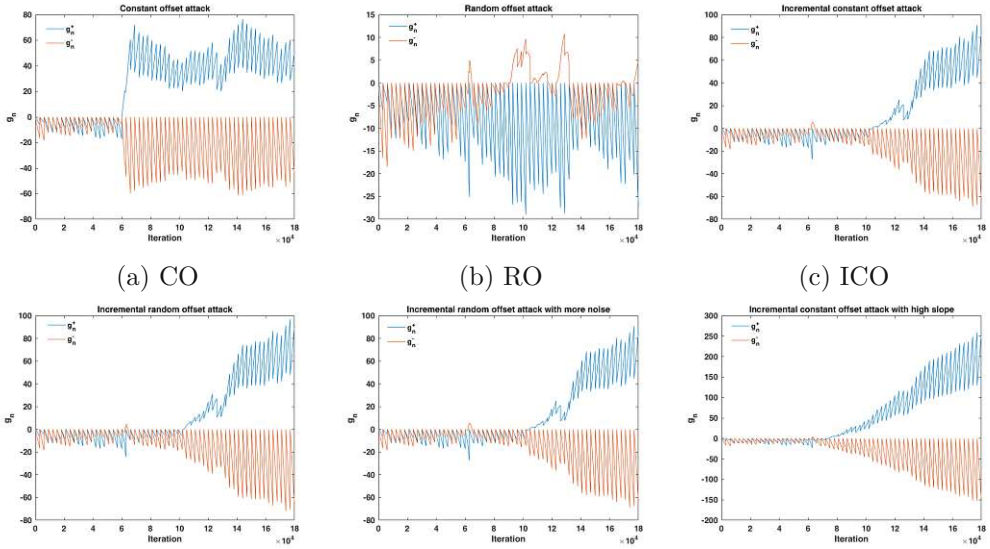


(a) CO (b) RO (c) ICO



(d) IRO (e) IROMN (f) ICOHS

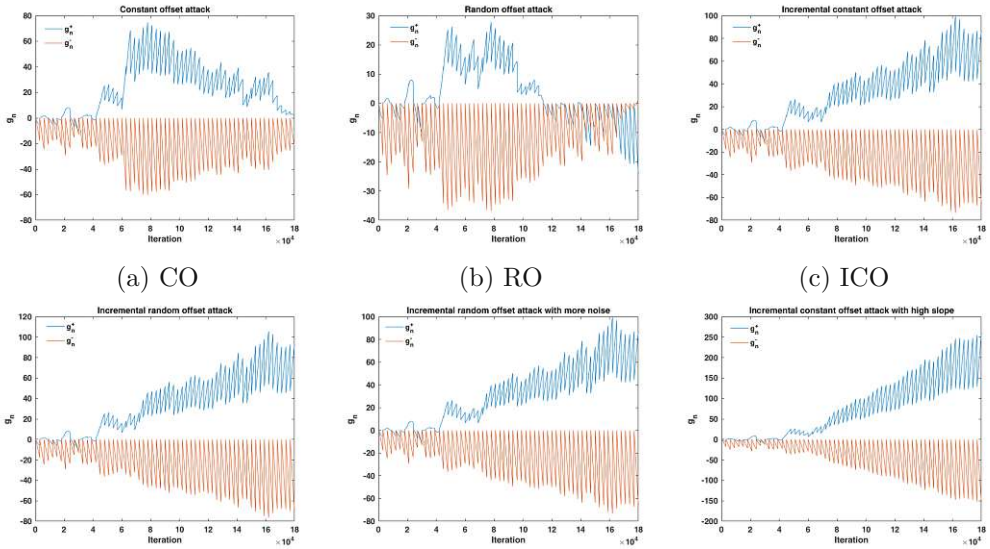
Figure A.50: CUSUM sequence of test data - day 11 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

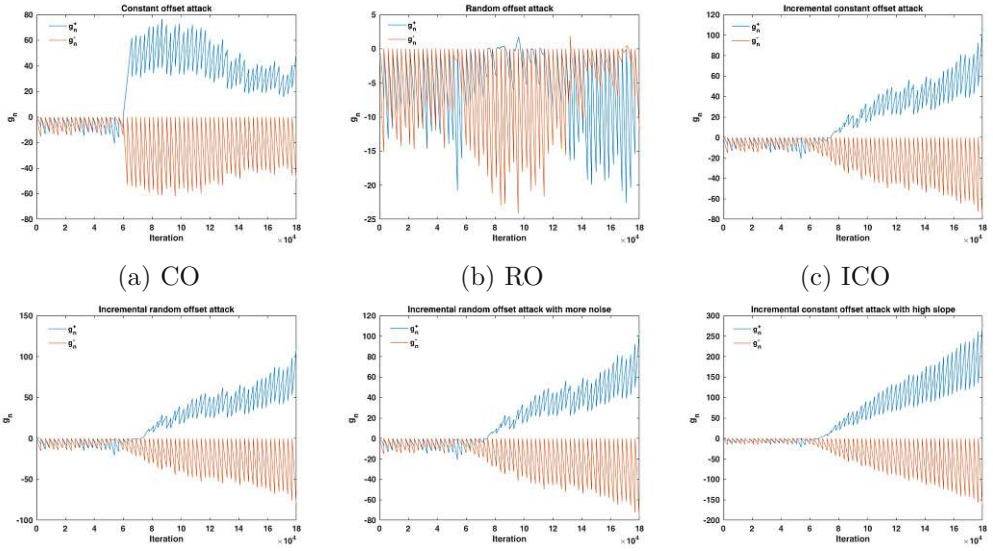
Figure A.51: CUSUM sequence of test data - day 12 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.52: CUSUM sequence of test data - day 13 with attacks.



(a) CO (b) RO (c) ICO

(d) IRO (e) IROMN (f) ICOHS

Figure A.53: CUSUM sequence of test data - day 14 with attacks.

A.10 Detected Data Points in Each Test Data Sets

Here we show data points detected as anomalous by methods, normalized residual-based bad data detection method, lightweight statistical methods (MAD, KLD, and CUSUM), and combination method (weighted voting).

Table A.1 shows data points detected as anomalous by the normalized residual-based method on all test data sets. Similarly, Tab. A.3 shows for the methods, MAD, KLD and CUSUM; and Tab. A.2 shows for weighted voting.

Table A.1: Detected data points using normalized residual-based method.

Method	**Data set**	**Attacks (detected data points in each test data)**					
		CO	RO	ICO	IRO	IROMN	ICOHS
Normalized	1	3597	286	487	520	286	289
	2	2002	0	0	0	0	0
	3	61	164	0	0	51	0
	4	1169	0	0	0	0	0
	5	2482	0	2909	2808	0	73211
	6	3219	0	0	0	0	0
	7	3149	0	402	397	0	290
	8	2827	0	0	0	1	0
	9	3636	0	0	19	0	51
	10	4197	4	380	382	20	314
	11	3447	52	3026	2789	10	2061
	12	2240	409	422	422	416	420
	13	0	0	0	0	0	0
	14	2316	0	281	272	0	176

Table A.2: Detected data points using weighted voting method.

Method	**Data set**	**Attacks (detected data points in each test data)**					
		CO	RO	ICO	IRO	IROMN	ICOHS
Weighted voting	1	119924	984	66473	66221	62738	92694
	2	120000	0	93906	93670	86082	109821
	3	120000	11152	89243	89001	87868	115699
	4	120000	0	81267	80375	75168	107900
	5	120000	0	75296	75925	70378	102127
	6	120000	0	77200	80300	70004	100269
	7	120000	1068	72237	75337	68915	96822
	8	120000	1079	81994	78394	70475	96006
	9	120000	0	83547	80502	72899	96870
	10	120000	0	82105	80458	75243	105721
	11	120000	53688	82726	80300	69953	107900
	12	120000	11	63473	69186	63267	96804
	13	120000	40024	110993	110341	103755	113810
	14	120000	379	92418	100150	87138	106253

Table A.3: Detected data points using MAD, KLD and CUSUM methods.

Method	Data set	Attacks (detected data points in each test data)					
		CO	RO	ICO	IRO	IROMN	ICOHS
MAD	1	119920	2094	66033	66191	65638	90037
	2	120000	912	93906	93670	92968	109821
	3	112705	11459	87969	89001	91519	115318
	4	120000	8	77718	77742	77924	102056
	5	120000	8	67239	67078	66669	95145
	6	120000	45	62856	63310	64150	95570
	7	119580	1405	63896	63785	63465	93765
	8	120000	1736	66232	66205	65897	84587
	9	119606	411	74390	74199	73940	94505
	10	119587	444	78172	77875	77805	102766
	11	90249	51004	18186	18133	18115	79005
	12	118714	1243	52617	52827	53126	88142
	13	81944	6947	110322	110337	110386	113810
	14	114922	485	85256	85842	86076	105666
KLD	1	110050	0	61350	61500	42200	91850
	2	120000	0	73750	76550	68550	104200
	3	120000	0	80450	80300	68550	107900
	4	120000	0	83950	80300	68550	107900
	5	120000	0	83950	80300	68550	107900
	6	120000	0	83950	80300	68550	107900
	7	120000	0	83950	80300	68550	107900
	8	120000	0	83950	80300	68550	107900
	9	120000	0	83950	80300	68550	107900
	10	120000	0	83950	80300	68550	107900
	11	120000	37800	83950	80300	68550	107900
	12	120000	0	83950	80300	68550	107900
	13	120000	0	92750	100150	81150	110750
	14	120000	0	92750	100150	81150	110750
CUSUM	1	119348	18097	69308	75173	69311	97212
	2	119430	0	108064	114793	108092	117212
	3	119362	118190	117674	118260	117650	118339
	4	119389	0	90203	93182	90194	111041
	5	119388	0	75296	75925	75289	102127
	6	119363	0	78026	81571	78019	100269
	7	119392	102077	72237	75337	72215	96822
	8	119360	102273	117694	78394	117666	96006
	9	119374	0	87004	87814	85027	96870
	10	119347	0	87308	99691	87317	105721
	11	119393	118044	118426	117931	118420	117757
	12	119368	0	63048	69186	63065	96804
	13	119416	118065	117521	118124	117493	118212
	14	119372	0	97018	102736	96998	106253

A.11 Statistical Properties of Training and Test Data Sets

The mean, median and standard deviation of whole training data is different from the median of each test data set. Mean and median have deviations (around 0.01) in most test data sets. The deviation of the median influences anomaly detection performance of MAD, deviations of the mean influences anomaly detection performance of CUSUM. For instance, if the the deviation of the mean is significant then CUSUM may detect anomaly in actual data or small changes due to an attack already cause detection.

Table A.4: Statistical property of whole training and test data sets.

Data	**Mean**	**Median**	**Stdev**
All training data	1.0607	1.0607	0.0046
All test data	1.0590	1.0587	0.0041

Table A.5: Statistical properties of test data per day.

Test data per day			
Day	**Mean**	**Median**	**Stdev**
1	1.0569	1.0570	0.0019
2	1.0633	1.0635	0.0015
3	1.0657	1.0663	0.0019
4	1.0599	1.0602	0.0016
5	1.0594	1.0594	0.0013
6	1.0581	1.0581	9.7630e-04
7	1.0561	1.0563	0.0020
8	1.0557	1.0555	0.0019
9	1.0578	1.0574	0.0016
10	1.0592	1.0594	0.0018
11	1.0517	1.0521	0.0035
12	1.0581	1.0581	0.0025
13	1.0636	1.0639	0.0034
14	1.0603	1.0607	0.0022

A.12 Derivation of measurement noise covariance matrix R

We assume V_z is the measured voltage magnitude, V_x is the true voltage magnitude, θ_z is the measured phase angle and θ_x is the true phase angle. There can be measurement

errors. Thus the measured voltage magnitude and phase angle are represented as

$$V_z = V_x + \tilde{V} \tag{A.20}$$

$$\theta_z = \theta_x + \tilde{\theta} \tag{A.21}$$

where $\tilde{V}$ is the voltage measurement error and $\tilde{\theta}$ is the phase measurement error.

We recall that the conversion of the measurements in rectangular coordinates from the polar coordinates below

$$V_{re,z} = V_z cos(\theta_z) \tag{A.22}$$

$$V_{im,z} = V_z sin(\theta_z) \tag{A.23}$$

where $V_{re,z}$ is the measured real voltage and $V_{im,z}$ is the measured imaginary voltage.

$$V_{re,z} = V_{re,x} + \tilde{V}_{re} = (V_x + \tilde{V})cos(\theta_z + \tilde{\theta}) \tag{A.24}$$

$$V_{im,z} = V_{im,x} + \tilde{V}_{im} = (V_x + \tilde{V})sin(\theta_z + \tilde{\theta}) \tag{A.25}$$

The measurement error of real and imaginary parts are

$$\tilde{V}_{re} = V_{re,z} - V_{re,x} \tag{A.26}$$

$$\tilde{V}_{im} = V_{im,z} - V_{im,x} \tag{A.27}$$

So the variance of real and imaginary parts (real and imaginary voltage) measurement errors are

$$\sigma^2_{re} = \sigma^2_{\tilde{V}_{re}} \tag{A.28}$$

$$\sigma^2_{im} = \sigma^2_{\tilde{V}_{im}} \tag{A.29}$$

List of Figures

List of Tables

Bibliography

[1] IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems. *IEEE Std 1547-2003*, pages 1–28, July 2003.

[2] IEC 61727 ed2.0 Photovoltaic (PV) systems - Characteristics of the utility interface. *International Electrotechnical Commission*, 2004.

[3] BDEW Generating Plants Connected to the Medium-Voltage Network. *BDEW German Association of Energy and Water Industries*, 2008.

[4] IEEE Standard for Synchrophasor Data Transfer for Power Systems. *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pages 1–53, Dec 2011.

[5] IEEE Standard for Synchrophasor Measurements for Power Systems. *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pages 1–61, Dec 2011.

[6] VDE-AR-N 4105:2011- 08 Power generation systems connected to the low-voltage distribution network. *VDE Association for Electrical, Electronic and Information Technologies*, 2011.

[7] IEEE Standard for Synchrophasor Measurements for Power Systems – Amendment 1: Modification of Selected Performance Requirements. *IEEE Std C37.118.1a-2014 (Amendment to IEEE Std C37.118.1-2011)*, pages 1–25, April 2014.

[8] C. B. A. Lee et al. Electric Sector Failure Scenarios and Impact Analyses. Technical report, 2013.

[9] M. Adamiak, D. Baigent, and R. Mackiewicz. IEC 61850 communication networks and systems in substations: An overview for users. 2009.

[10] M. Adamiak, D. Baigent, and R. Mackiewicz. IEC 61850 Communication Networks and Systems In Substations. 2010.

[11] M. Ahmad. *Power System State Estimation*. Artech House power engineering series. Artech House, 2013.

[12] N. U. Ahmed and S. M. Radaideh. Modified extended kalman filtering. *IEEE Transactions on Automatic Control*, 39(6):1322–1326, 1994.

[13] C. Alcaraz, L. Cazorla, and G. Fernandez. *Risks and Security of Internet and Systems: 9th International Conference, CRiSIS 2014*. 2014.

[14] Ali Abur, Antonio Gómez Expósito. Power System State Estimation: Theory and Implementation. 2004.

[15] A. Anwar and A. N. Mahmood. Cyber security of smart grid infrastructure. 2014.

[16] A. Arefi and M.-R. Haghifam. State estimation in smart power grids. In A. Keyhani and M. Marwali, editors, *Smart Power Grids 2011*, Power Systems, pages 439–478. Springer Berlin Heidelberg, 2012.

[17] J. Arvo. III.4 - FAST RANDOM ROTATION MATRICES. In D. KIRK, editor, *Graphics Gems III (IBM Version)*, pages 117 – 120. Morgan Kaufmann, San Francisco, 1992.

[18] A. Ashok, M. Govindarasu, and V. Ajjarapu. Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Transactions on Smart Grid*, 9(3):1636–1646, 2018.

[19] S. Barreto, M. Pignati, G. Dán, J. Le Boudec, and M. Paolone. Undetectable timing-attack on linear state-estimation by using rank-1 approximation. *IEEE Transactions on Smart Grid*, 9(4):3530–3542, 2018.

[20] S. Barreto, A. Suresh, and J. Le Boudec. Cyber-attack on packet-based time synchronization protocols: The undetectable delay box. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6, 2016.

[21] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[22] S. Basumallik, S. Eftekharnejad, N. Davis, and B. K. Johnson. Impact of false data injection attacks on pmu-based state estimation. In *2017 North American Power Symposium (NAPS)*, pages 1–6, Sep. 2017.

[23] A. Bergen and V. Vittal. *Power Systems Analysis*. Pearson/Prentice Hall, 2000.

[24] R. Bobba, K. Davis, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye. Detecting false data injection attacks on dc state estimation. 01 2010.

[25] R. B. Bobba, J. Dagle, E. Heine, H. Khurana, W. H. Sanders, P. Sauer, and T. Yardley. Enhancing Grid Measurements: Wide Area Measurement Systems, NASPInet, and Security. *IEEE Power and Energy Magazine*, 2012.

[26] E. Bompard, T. Huang, Y. Wu, and M. Cremenescu. Classification and trend analysis of threats origins to the security of power systems. *International Journal of Electrical Power and Energy Systems*, 50:50 – 64, 2013.

[27] R. G. Brown and P. Y. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. John Wiley and Sons, 2012.

[28] E. J. Byres, M. Franz, and D. Miller. The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. `https://www.ida.liu.se/labs/rtslab/iisw04/camready/SCADA-Attack-Trees-Final.pdf`, 2004. Online; accessed January 2020.

[29] V. C., A. B., and V. K. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[30] CENELEC. CEN-CENELEC-ETSI Smart Grid Coordination Group - First Set of Standards. 2012.

[31] G. Chaojun, P. Jirutitijaroen, and M. Motani. Detecting false data injection attacks in ac state estimation. *IEEE Transactions on Smart Grid*, 6(5), Sept 2015.

[32] D. Chassin, R. Carroll, and D. Bakken. NASPI Phasor Gateways and Their Relationship to Phasor Data Concentrators. 2008.

[33] H. Chen, L. Zhang, J. Mo, and K. E. Martin. Synchrophasor-based real-time state estimation and situational awareness system for power system operation. *Journal of Modern Power Systems and Clean Energy*, 4(3):370–382, 2016.

[34] P. Chen, L. Desmet, and C. Huygens. *A Study on Advanced Persistent Threats*, pages 63–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[35] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford. Petri net modeling of cyber-physical attacks on smart grid. *IEEE Transactions on Smart Grid*, 2(4):741–749, 2011.

[36] F. N. Chowdhury, J. P. Christensen, and J. L. Aravena. Power system fault detection and state estimation using kalman filter with hypothesis testing. *IEEE Transactions on Power Delivery*, 6(3):1025–1030, Jul 1991.

[37] CISCO. White paper - substation automation for the smart grid. 2010.

[38] CISCO. Snort-intrusion prevention system capable of real-time traffic analysis and packet logging. 2016.

[39] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor, and A. Tajer. Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions. *IEEE Signal Processing Magazine*, 2012.

[40] T. V. Cutsem, M. Ribbens-Pavella, and L. Mili. Bad data identification methods in power system state estimation-a comparative study. *IEEE Transactions on Power Apparatus and Systems*, PAS-104(11):3037–3049, Nov 1985.

[41] A. M. L. da Silva, M. B. D. C. Filho, and J. M. C. Cantera. An efficient dynamic state estimation algorithm including bad data processing. *IEEE Power Engineering Review*, PER-7(11):49–49, Nov 1987.

[42] G. Dan and H. Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 214–219, Oct 2010.

[43] M. Dehghani, Z. Khalafi, A. Khalili, and A. Sami. Integrity attack detection in pmu networks using static state estimation algorithm. In *PowerTech, 2015 IEEE Eindhoven*, 2015.

[44] D. Deka, R. Baldick, and S. Vishwanath. Data attacks on power grids: Leveraging detection. In *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2015.

[45] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[46] E. Drayer and T. Routtenberg. Detection of false data injection attacks in smart grids based on graph signal processing. *IEEE Systems Journal*, 14(2):1886–1896, 2020.

[47] Ecole Polytechnique Federale DE Lausanne (EPFL). Smart Grid and PMU measurements. `https://smartgrid.epfl.ch/?q=monitoring`. Online; accessed January 2020.

[48] A. Elgargouri, R. Virrankoski, and M. Elmusrati. Iec 61850 based smart grid security. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2461–2465, March 2015.

[49] M. et al. PMU Optimal Placement using sensitivity analysis for power systems fault location. In *2015 IEEE Electrical Power and Energy Conference (EPEC)*, Oct 2015.

[50] S. et al. Performance assessment of linear state estimators using synchrophasor measurements. *IEEE Transactions on Instrumentation and Measurement*, 65(3), March 2016.

[51] E. C. for Electrotechnical Standardization CENELEC. Standard EN 50160 - Voltage Characteristics in Public Distribution Systems. `http://copperalliance.org.uk/uploads/2018/03/542-standard-en-50160-voltage-characteristics-in.pdf`, 2004. Online; accessed January 2020.

[52] G. Franklin, J. Powell, and M. Workman. Digital control of dynamic systems-third edition. 01 2006.

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

[54] S. Gupta, S. Waghmare, F. Kazi, S. Wagh, and N. Singh. Blackout risk analysis in smart grid wampac system using kl divergence approach. In *2016 IEEE 6th International Conference on Power Systems (ICPS)*, March 2016.

[55] F. Hamano. Derivative of rotation matrix direct matrix derivation of well known formula. *Proceedings of Green Energy and Systems Conference*, abs/1311.6010, 2013.

[56] J. Harmouche, C. Delpha, and D. Diallo. Incipient fault detection and diagnosis based on kullback–leibler divergence using principal component analysis: Part i. *Signal Processing*, 94:278 – 287, 2014.

[57] F. Harrou, Y. Sun, and M. Madakyaru. Kullback-leibler distance-based enhanced detection of incipient anomalies. *Journal of Loss Prevention in the Process Industries*, 44:73 – 87, 2016.

[58] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han. Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis. *IEEE Systems Journal*, 10(2):532–543, 2016.

[59] M. Hutle et al. Threat analysis and risk assesment. *Smart Grid Protection Against Cyber Attacks-SPARKS European Project*, 2015.

[60] IEC-60870-5-101. IEC 60870-5-101 - Transmission protocols - Section 101: Companion standard for basic telecontrol tasks.

[61] IEC-60870-5-103. IEC 60870-5-103 - Transmission protocols –Companion standard for the informative interface of protection equipment.

[62] IEC-60870-5-104. IEC 60870-5-104 - Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles . `https://webstore.iec.ch/publication/18149`. Online; accessed January 2020.

[63] IEC-61850. IEC 61850 - Communication Networks and Systems in Substations.

[64] IEC-62351-1. Part 1: Communication network and system security – introduction to security issues. 2007.

[65] IEC-62351-6. Part 6: Security for 61850. 2007.

[66] S. IEC Geneva. IEC 62351-Part 1: Communication network and system security – Introduction to security issues. 2007.

[67] IEC-TR-61850-90-1. IEC 61850-Part 90-1: Use of IEC 61850 for the communication between substations. 2016.

[68] IEC-TR-61850-90-2. IEC 61850-Part 90-2: Using IEC 61850 for communication between substations and control centres. 2016.

[69] IEC-TR-61850-90-5. IEC 61850-Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118. 2012.

[70] IEEE-C37.118.1. IEEE Standard for Synchrophasor Measurements for Power Systems. *IEEE-Std-C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, 2011.

[71] IEEE-C37.118.2. IEEE Standard for Synchrophasor Data Transfer for Power Systems. *IEEE-Std-C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, 2011.

[72] S. Jajodia, P. Liu, V. Swarup, and C. Wang, editors. *Cyber Situational Awareness - Issues and Research.* Advances in Information Security. Springer, 2010.

[73] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64. Academic Press, Inc., 1970.

[74] X. Jiang, J. Zhang, B. J. Harding, J. J. Makela, and A. D. Dominguez-Garcõ«a. Spoofing gps receiver clock offset of phasor measurement units. *IEEE Transactions on Power Systems*, 2013.

[75] B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker, and C. Glyer. Attackers Deploy New ICS Attack Framework “TRITON" and Cause Operational Disruption to Critical Infrastructure. https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html. Online; accessed March 2018.

[76] K. D. Jones. Three-phase linear state estimation with phasor measurements. In *Masters's Thesis*, Viginia Tech, May 2011.

[77] K. D. Jones, J. S. Thorp, and R. M. Gardner. Three-phase linear state estimation using phasor measurements. In *2013 IEEE Power Energy Society General Meeting*, pages 1–5, July 2013.

[78] A. Jovicic, M. Jereminov, L. Pileggi, and G. Hug. A linear formulation for power system state estimation including rtu and pmu measurements. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–5, 2019.

[79] V. Kekatos, G. B. Giannakis, and R. Baldick. Grid topology identification using electricity prices. In *2014 IEEE PES General Meeting | Conference Exposition*, pages 1–5, July 2014.

[80] M. Kezunovic, T. Popovic, C. Muehrcke, B. Isle, S. Harp, E. Sisley, and S. Ayyorgun. NESCOR Wide Area Monitoring, Protection, and Control Systems : Standards for Cyber Security Requirements. 2012. [Online; accessed 31-January-2017].

[81] J. Kim, I. Moon, K. Lee, S. C. Suh, and I. Kim. Scalable security event aggregation for situation analysis. In *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference*, 2015.

[82] J. Kim and L. Tong. On topology attack of a smart grid: Undetectable attacks and countermeasures. *IEEE Journal on Selected Areas in Communications*, 31(7):1294–1305, July 2013.

[83] T. T. Kim and H. V. Poor. Strategic Protection Against Data Injection Attacks on Power Grids. *IEEE Transactions on Smart Grid*, 2011.

[84] F. Kong, J. Weimer, O. Sokolsky, and I. Lee. State consistencies for cyber-physical system recovery. In *2019 2nd Workshop on Cyber-Physical Systems Security and Resilience (CPS-SR)*, April 2019.

[85] F. Kong, M. Xu, J. Weimer, O. Sokolsky, and I. Lee. Cyber-physical system checkpointing and recovery. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 22–31, April 2018.

[86] G. N. Korres and N. M. Manousakis. State estimation and bad data processing for systems including pmu and scada measurements. *Electric Power Systems Research*, 81(7):1514 – 1524, 2011.

[87] G. N. Korres and N. M. Manousakis. State estimation and bad data processing for systems including pmu and scada measurements. *Electric Power Systems Research*, 81(7):1514 – 1524, 2011.

[88] O. Kosut, L. Jia, R. J. Thomas, and L. Tong. Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid*, 2(4):645–658, Dec 2011.

[89] D. Kraft. Optimal estimation with an introduction to stochastic control theory : Frank l. lewis. *Automatica*, 23:807–808, 1987.

[90] S. Kullback. Information Theory and Statistics. 22(1), 1968.

[91] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1), 1951.

[92] Y. Kwon, H. K. Kim, Y. H. Lim, and J. I. Lim. A behavior-based intrusion detection technique for smart grid infrastructure. In *PowerTech, 2015 IEEE Eindhoven*, 2015.

[93] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3):49–51, May 2011.

[94] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. Technical report, SANS ICS and E-ISAC, March 2016.

[95] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764 – 766, 2013.

[96] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, 32(4):3317–3318, July 2017.

[97] J. Liang, L. Sankar, and O. Kosut. Vulnerability analysis and consequences of false data injection attack on power system state estimation. *IEEE Transactions on Power Systems*, 31(5):3864–3872, Sep. 2016.

[98] J. Liu, Y. Xiao, S. Li, W. Liang, and C. L. P. Chen. Cyber security and privacy issues in smart grids. *IEEE Communications Surveys Tutorials*, 14(4):981–997, 2012.

[99] Y. Liu, P. Ning, and M. K. Reiter. False Data Injection Attacks Against State Estimation in Electric Power Grids. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009.

[100] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '011, 2011.

[101] J. Lueckenga, D. Engel, and R. Green. Weighted vote algorithm combination technique for anomaly based smart grid intrusion detection systems. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2738–2742, July 2016.

[102] R. A. Maronna, R. D. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. John Wiley and Sons Canada Ltd., Toronto, ON, 2006.

[103] Mathworks. Feedback connection of multiple models.

[104] Mathworks. Kalman filter design, kalman estimator.

[105] Mathworks. Kalman filtering, steady state kalman filter.

[106] Mathworks. Parallel connection of two models.

[107] G. M. Messinis and N. D. Hatziargyriou. Review of non-technical loss detection methods. *Electric Power Systems Research*, 158, 2018.

[108] Miao He, Junshan Zhang, and V. Vittal. A data mining framework for online dynamic security assessment: Decision trees, boosting, and complexity analysis. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–8, 2012.

[109] W. Miller and J. Lewis. Dynamic state estimation in power systems. *IEEE Transactions on Automatic Control*, 16(6):841–846, December 1971.

[110] MITRE. ATT and CK for Industrial Control Systems. https://collaborate.mitre.org/attackics/index.php/Main_Page. Online; accessed January 2020.

[111] Modbus. MODBUS Application Protocol Specification V1.1b3. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, 2012. Online; accessed January 2020.

[112] M. Mohammad. Smart false data injection attacks against state estimation in power grid. *ArXiv*, abs/1809.07039, 2018.

[113] J. G. Moller, M. Sorensen, H. Johansson, and J. Ostergaard. Detecting topological errors with pre-estimation filtering of bad data in wide-area measurements. In *2017 IEEE Manchester PowerTech*, pages 1–6, June 2017.

[114] A. Monticelli. Electric power system state estimation. *Proceedings of the IEEE*, 88(2):262–282, Feb 2000.

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

[116] A. P. Moore, R. J. Ellison, and R. C. Linger. Attack modeling for information security and survivability. *Carnegie Mellon University*, 2001.

[117] V. M. Morgenstern, B. R. Upadhyaya, and M. Benedetti. Signal anomaly detection using modified cusum method. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 3, pages 2340–2341, 1988.

[118] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[119] NASPI Engineering Analysis Task Team. Phase Angle Calculations: Considerations and Use Cases. *NASPI Engineering Analysis Task Team Technical Paper*, September 2016.

[120] NASPInet. Phasor gateways technical specifications for north american synchrophasor initiative network. 2009.

[121] N. E. S. C. O. R. NESCOR and N. W. T. G. . TWG1. Electric sector failure scenarios and impact analyses. http://smartgrid.epri.com/doc/NESCORfailurescenarios09-13finalc.pdf, 2013. Online; accessed January 2020.

[122] J. Ni, K. Alharbi, X. Lin, and X. Shen. Security-enhanced data aggregation against malicious gateways in smart grid. In *2015 IEEE Global Communications Conference*, 2015.

[123] NIST. Guidelines for Smart Grid Cybersecurity; Smart Grid Cybersecurity Strategy, Architecture, and High-Level Requirements. https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf. Online; accessed January 2020.

[124] NIST. Guidelines for Smart Grid Cybersecurity; Supportive Analyses and References. https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf. Online; accessed January 2020.

[125] Z. Omary and F. Mtenzi. Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning. *International Journal for Infonomics*, 3, 09 2010.

[126] S. Pal and B. Sikdar. A mechanism for detecting data manipulation attacks on pmu data. In *Communication Systems (ICCS), 2014 IEEE International Conference on*, 2014.

[127] S. Pal, B. Sikdar, and J. Chow. Real-time detection of packet drop attacks on synchrophasor data. In *Smart Grid Communications, 2014 IEEE International Conference on*, 2014.

[128] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas. A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems. *Comput. Secur.*, 64(C):92–109, Jan. 2017.

[129] S. Paudel, P. Smith, and T. Zseby. Data Integrity Attacks in Smart Grid Wide Area Monitoring. *4th International Symposium for ICS and SCADA Cyber Security Research*, 2016.

[130] S. Paudel, P. Smith, and T. Zseby. Attack models for advanced persistent threats in smart grid wide area monitoring. In *Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, pages 61–66, New York, NY, USA, 2017. ACM.

[131] S. Paudel, P. Smith, and T. Zseby. Data Attacks in Wide Area Monitoring System. *Symposium on Innovative Smart Grid Cybersecurity Solutions*, 2017.

[132] S. Paudel, P. Smith, and T. Zseby. Stealthy attacks on smart grid PMU state estimation. In *Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*, pages 16:1–16:10, 2018.

[133] S. Paudel, T. Zseby, E. Piatkowska, and P.Smith. An evaluation of methods for detecting false data injection attacks in the smart grid. *In preparation.*

[134] L. Pető and J. Botzheim. Parameter optimization of deep learning models by evolutionary algorithms. In *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 000027–000032, 2019.

[135] J. L. Peterson. Petri nets. *ACM Comput. Surv.*, 9(3):223–252, 1977.

[136] A. Phadke and J. Thorp. *Synchronized Phasor Measurements and Their Applications*. Power Electronics and Power Systems. Springer US, 2008.

[137] M. Pignati, M. Popovic, S. Barreto, R. Cherkaoui, G. Dario Flores, J. Le Boudec, M. Mohiuddin, M. Paolone, P. Romano, S. Sarri, T. Tesfay, D. Tomozei, and L. Zanni. Real-time state estimation of the epfl-campus medium-voltage grid by using pmus. In *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, Feb 2015.

[138] M. Pignati, M. Popovic, S. Barreto, R. Cherkaoui, G. D. Flores, J. Y. L. Boudec, M. Mohiuddin, M. Paolone, P. Romano, S. Sarri, T. Tesfay, D. C. Tomozei, and L. Zanni. Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs. In *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, Feb 2015.

[139] M. Pignati, L. Zanni, S. Sarri, R. Cherkaoui, J. Le Boudec, and M. Paolone. A pre-estimation filtering process of bad data for linear power systems state estimators using PMUs. In *2014 Power Systems Computation Conference*, pages 1–8, Aug 2014.

[140] M. A. Rahman, E. Al-Shaer, and P. Bera. A noninvasive threat analyzer for advanced metering infrastructure in smart grid. *IEEE Transactions on Smart Grid*, 2013.

[141] M. A. Rahman and H. Mohsenian-Rad. False data injection attacks with incomplete information against smart power grids. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 3153–3158, Dec 2012.

[142] M. Rassam, M. Maarof, and A. Zainal. Outlier detection techniques for wireless sensor networks: A survey. *American Journal of Applied Sciences*, 9(10), Second 2012.

[143] W. Reisig. *Petri Nets: An Introduction*. Springer-Verlag New York, Inc., USA, 1985.

[144] M. Reta-Hernández. 13 transmission line parameters. 2010.

[145] P. Romano and M. Paolone. Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype. *IEEE Transactions on Instrumentation and Measurement*, 63(12):2824–2836, Dec 2014.

[146] P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424), 1993.

[147] D. Ruppert and D. S. Matteson. Statistics and data analysis for financial engineering. Springer, New York, NY, 2015.

[148] H. Sandberg, A. Teixeira, and K. Johansson. On security indices for state estimators in power networks. 01 2010.

[149] S. Sarri. Methods and Performance Assessment of PMU-based Real-Time State Estimation of Active Distribution Networks. 2016.

[150] S. Sarri, L. Zanni, M. Popovic, J. Y. L. Boudec, and M. Paolone. Performance Assessment of Linear State Estimators Using Synchrophasor Measurements. *IEEE Transactions on Instrumentation and Measurement*, 65(3):535–548, March 2016.

[151] S. Savulescu. *Real-Time Stability in Power Systems*. Power Electronics and Power Systems. Springer, 2014.

[152] B. Schneier. Attack Trees. `https://www.schneier.com/academic/archives/1999/12/attack_trees.html`, 1999. Online; accessed January 2020.

[153] L. Schwartfegerand and D. Santos-Martin. Review of distributed generation interconnection standards. `https://ir.canterbury.ac.nz/bitstream/handle/10092/17537/UC-GG-14-C-LS-01_EEA_ReviewofDistributedGenerationInterconnectionStandards_LSchwartfeger_18-20June_2015_CS2.4.3.pdf?sequence=1`, 2014. Online; accessed January 2020.

[154] F. C. Schweppe. Power system static-state estimation, part iii: Implementation. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(1):130–135, Jan 1970.

[155] F. C. Schweppe and J. Wildes. Power system static-state estimation, part i: Exact model. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(1):120–125, Jan 1970.

[156] J. Searle, G. Rasche, A. Wright, and S. Dinnage. Nescor guide to penetration testing for electric utilities. `http://smartgrid.epri.com/doc/NESCORGuidetoPenetrationTestingforElectricUtilities-v3-Final.pdf`, 2016.

[157] K. Singla and S. Biswas. Machine learning explanability method for the multi-label classification model. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 337–340, 2021.

[158] F. Skopik, M. Landauer, M. Wurzenberger, G. Vormayr, J. Milosevic, J. Fabini, W. Prüggler, O. Kruschitz, B. Widmann, K. Truckenthanner, S. Rass, M. Simmer, and C. Zauner. synergy: Cross-correlation of operational and contextual data to

timely detect and mitigate attacks to cyber-physical systems. *Journal of Information Security and Applications*, 54:102544, 2020.

[159] A. Soule, K. Salamatian, and N. Taft. Combining Filtering and Statistical Methods for Anomaly Detection. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, pages 31–31, Berkeley, CA, USA, 2005. USENIX Association.

[160] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn. Guide to Industrial Control Systems (ICS) Security. `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf`, 2015. Online; accessed January 2020.

[161] K. Sun, K. Hur, and P. Zhang. A new unified scheme for controlled power system separation using synchronized phasor measurements. *IEEE Transactions on Power Systems*, 26(3):1544–1554, Aug 2011.

[162] Y. Sun, X. Guan, T. Liu, and Y. Liu. A cyber-physical monitoring system for attack detection in smart grid. In *Computer Communications Workshops, 2013 IEEE Conference*, 2013.

[163] C. B. T. Popovic et al. Electric Sector Failure Scenarios and Impact Analyses Version 3.0 (NESCOR). Technical report, 2015.

[164] A. F. Taha, J. Qi, J. Wang, and J. H. Panchal. Risk mitigation for dynamic state estimation against cyber attacks and unknown inputs. *The Computing Research Repository*, 2015.

[165] G. Tang, K. Wu, J. Lei, Z. Bi, and J. Tang. From landscape to portrait: A new approach for outlier detection in load curve data. *IEEE Transactions on Smart Grid*, 5(4):1764–1773, 2014.

[166] J. E. Tate and T. J. Overbye. Line outage detection using phasor angle measurements. *IEEE Transactions on Power Systems*, 23(4):1644–1652, Nov 2008.

[167] J. S. Thorp, A. G. Phadke, and K. J. Karimi. Real time voltage-phasor measurement for static state estimation. *IEEE Transactions on Power Apparatus and Systems*, PAS-104(11):3098–3106, 1985.

[168] W. Tu, J. Dong, and D. Zhai. Optimal ? -stealthy attack in cyber-physical systems. *Journal of the Franklin Institute*, 2019.

[169] Y. Wang, W. Li, J. Lu, and H. Liu. Evaluating multiple reliability indices of regional networks in wide area measurement system. *Electric Power Systems Research*, 2009.

[170] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, Chapel Hill, NC, USA, 1995.

[171] F. F. Wu and W. . E. Liu. Detection of topology errors by state estimation (power systems). *IEEE Transactions on Power Systems*, 4(1):176–183, Feb 1989.

[172] E. Xypolytou, T. Zseby, J. Fabini, and W. Gawlik. Detection and mitigation of cascading failures in interconnected power systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, Sep. 2017.

[173] Y. Yang, J. Shi, and J. Wang. A parameter optimization method based on relevance measurement in a prediction system. In *2015 Prognostics and System Health Management Conference (PHM)*, pages 1–4, 2015.

[174] Yi Huang, H. Li, K. A. Campbell, and Zhu Han. Defending false data injection attack on smart grid network using adaptive cusum test. In *2011 45th Annual Conference on Information Sciences and Systems*, pages 1–6, 2011.

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

[176] L. Zanni. Power System Estimation based on PMUs-Static and Dynamic Approaches from Theory to Real Implementation. 2017.

[177] L. Zanni, S. Sarri, M. Pignati, R. Cherkaoui, and M. Paolone. Probabilistic assessment of the process-noise covariance matrix of discrete Kalman filter state estimation of active distribution networks. In *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pages 1–6, July 2014.

[178] R. Zhang and P. Venkitasubramaniam. Stealthy control signal attacks in scalar lqg systems. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 240–244, 2015.

[179] Y. Zhang, N. Meratnia, and P. Havinga. *A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets*. Number Paper P-NS/TR-CTIT-07-79. Centre for Telematics and Information Technology (CTIT), Netherlands, 11 2007.

[180] Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 12(2), Second 2010.

[181] J. Zhao, A. Gómez-Expósito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A. K. Singh, J. Qi, Z. Huang, and A. P. S. Meliopoulos. Power system dynamic state estimation: Motivations, definitions, methodologies, and future work. *IEEE Transactions on Power Systems*, 34(4):3188–3198, July 2019.

[182] J. Zhao, L. Mili, and M. Wang. A generalized false data injection attacks against power system nonlinear state estimator and countermeasures. *IEEE Transactions on Power Systems*, 33(5):4868–4877, Sep. 2018.

[183] T. Zseby and J. Fabini. Security challenges for wide area monitoring in smart grids. *Elektrotechnik und Informationstechnik*, 2014.

[184] T. Zseby, J. Fabini, and D. Rani. Synchrophasor communication. *Elektrotechnik und Informationstechnik*, 2013.