ONLINE DETECTION AGAINST CYBERATTACKS IN CYBER-PHYSICAL SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Dan Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton School of Industrial and Systems Engineering

Georgia Institute of Technology

August 2021

Copyright © Dan Li 2021
ONLINE DETECTION AGAINST CYBERATTACKS IN CYBER-PHYSICAL SYSTEMS

Approved by:

Dr. Nagi Gebraeel, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jianjun (Jan) Shi
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. A.P. Sakis Meliopoulos
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Deepakraj M Divan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: June 21, 2021
To my beloved parents and my partner, Lei.
I want to express my deepest gratitude to my advisors, Professor Nagi Gebraeel and Professor Kamran Paynabar, for providing me the opportunity to start my research and for all their kind advice and support during my Ph.D. study. Finishing my Ph.D. would not have been possible without their feedback and suggestions on my research, encouragement to reach higher standards when I succeed, and support when I fail. I am sincerely thankful for their guidance not only in my research but also in life and soul. Their vision, sincerity, passion, rigorousness, and selfless care about my growth and success have been and will always be my inspiration.

I also want to express my special thanks to my committee member, Professor Jianjun Shi. I could not have met my advisors or started my Ph.D. without his help. Professor Shi has been providing me with his insightful advice throughout my study. Being the most successful faculty I have ever known, his advice always sounds simple yet hits the nail on the head. It constantly reminds me to be a pure person.

I am also extremely grateful to my other two committee members, Professor Sakis Meliopoulos and Professor Deepakraj Divan. Since we first met, Professor Meliopoulos has been devoting his time to answering all my questions and encouraging me to be involved in the cybersecurity community. Professor Divan allowed me to work with his group and gain valuable experience that enabled my research in the power system area.

A special thanks to Dr. Damon Williams, Professor Mario Berges, and Professor Burcu Arkinci from CMU for their advice and support during my job search.

My gratitude also goes to my collaborators: Ms. Ana Maria Estrada Gomez, Dr. Paritosh Ramanan, and Dr. Kavya Ashok, for their intelligence, support, and friendship. I would also like to thank my academic brothers and sisters and my friends, including but not limited to Dr. Xiaolei Fang, Dr. Murat Yildirim, Dr. Linkan Bian, Dr. Alaa Elwany, Ms. Jiachen Shi, Dr. Beste Basciftci, Mr. Benjamin Peters, Dr. Hao Yan, Dr. Mostafa
Reisi Gahrooei, Dr. Chitta Ranjan, Dr. Samaneh Ebrahimi, Dr. Zhen Zhong, Mr. Wei Yang, Mr. Qian Wang, Mr. Jinhyeun Kim, Mr. Ribhu Sengupta, Mr. Heraldo Rozas, Mr. Ayush Mohanty, Mr. Michael Ibrahim, Ms. Anjolaoluwa Popoola, Dr. Xiaowei Yue, Dr. Naipeng Li, Dr. Fangyu Li, Ms. Xiaohan Du, Ms. Yining Zhao, Dr. Zhuanan Li, Dr. Ming Tong, Ms. Yan Yan, Dr. Xu Hao, Ms. Da Fang, Dr. Yifu Shi, and Dr. Peiliang Bai. A special thanks to Dr. Xinran Shi, Dr. Jialei Chen, and Mr. Liexiao Ding for being my best company on this journey.

Last but not least, I would like to thank my family, including my partner Mr. Lei Zhang, for their unconditional love and support. They are the most precious of my life and have supported me through the rough times. I also thank my cat, Mr. Riceball, for being the loveliest creature in my family and the warmest companion during my study.
# TABLE OF CONTENTS

Acknowledgments ................................................................. iv

List of Tables ........................................................................ xi

List of Figures ......................................................................... xii

Summary .................................................................................... xiv

Chapter 1: Introduction ............................................................. 1

1.1 Background ................................................................. 1

1.2 Literature Review ......................................................... 4

1.3 Dissertation Overview .................................................... 7

  1.3.1 Differentiating Cyberattacks from Equipment Faults ........ 8

  1.3.2 Localizing Cyberattacks in Complex Networked Systems .... 10

Chapter 2: Detection and Differentiation of Replay Attack and Equipment Faults in SCADA Systems ................................................................. 12

2.1 Introduction ................................................................. 12

2.2 System Description ........................................................ 13

  2.2.1 State-Space Model .................................................... 14

  2.2.2 Kalman Filter .......................................................... 16
Chapter 3: A Degradation-Based Detection Framework Against Covert Cyber-attacks in SCADA Systems

3.1 Introduction

3.2 Problem Setting

3.2.1 System Model

3.2.2 Degradation Model

3.2.3 Covert Attack

3.2.4 Parameter Estimation
3.3 Detection Algorithm for Covert Cyberattacks
   3.3.1 Detection of Degradation
   3.3.2 Detection of Covert Attack
3.4 Numerical and Case Study
   3.4.1 Numerical Study
   3.4.2 Case Study
3.5 Conclusion

Chapter 4: An Online Approach to Cyberattack Detection and Localization in Smart Grids
   4.1 Introduction
   4.2 Problem Setup
      4.2.1 Global Model
      4.2.2 Local Model
      4.2.3 Covert Attack
   4.3 Methodology
      4.3.1 Sparse Group Lasso
      4.3.2 Linearized System
      4.3.3 Extension to Nonlinear System
      4.3.4 Threshold Selection
   4.4 Numerical Results
      4.4.1 Linear System
      4.4.2 Nonlinear System
   4.5 Conclusion
Chapter 5: Deep Learning Based Covert Attack Identification for Industrial Control Systems

5.1 Introduction .......................................................... 86
  5.1.1 Related Work .................................................. 86
5.2 System Model ...................................................... 88
  5.2.1 Covert Attack Model ......................................... 91
5.3 Proposed Detection Framework ................................. 94
5.4 Performance Evaluation ......................................... 95
  5.4.1 Data Extraction ............................................... 95
  5.4.2 Numerical Results ........................................... 97
5.5 Conclusion .......................................................... 103

Chapter 6: Conclusion and Future Research ......................... 105

6.1 Conclusion .......................................................... 105
6.2 Future Research .................................................. 108

Appendix A: SUPPLEMENTARY MATERIALS OF CHAPTER 2 ........ 111

  A.1 Proof of Theorem 2.3.1 ....................................... 111
  A.2 Proof of Theorem 2.3.2 ....................................... 114
  A.3 Proof of Theorem 2.3.4 ....................................... 115
  A.4 Proof of Theorem 2.3.3 ....................................... 118
  A.5 Proof of Theorem 2.3.5 ....................................... 119
  A.6 Simulation Results ............................................. 119
  A.7 Experimental Results ......................................... 119
Appendix B: SUPPLEMENTARY MATERIALS OF CHAPTER 3 . . . . . . . . 124

B.1 Proof of Theorem 3.3.1 . . . . . . . . . . . . . . . . . . . . . . . 124
B.2 Proof of Theorem 3.3.2 . . . . . . . . . . . . . . . . . . . . . . . 125
B.3 Derivation of $\hat{\beta}$ . . . . . . . . . . . . . . . . . . . . . . 125
B.4 Simulation Setting . . . . . . . . . . . . . . . . . . . . . . . . . 127
B.5 Simulation Results . . . . . . . . . . . . . . . . . . . . . . . . . 128
B.6 Experiment Parameters . . . . . . . . . . . . . . . . . . . . . . 129

References . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 137

Vita . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
LIST OF TABLES

2.1 Signature correspondence with types of fault/attack . . . . . . . . . . . 27
2.2 Signature comparison with $ARL_1$ for nonlinear system . . . . . . . . . 31
4.1 ARL and Accuracy under different levels of SNR . . . . . . . . . . . . . 82
5.1 Precision, Recall, and F score for DNN classification . . . . . . . . . . 100
5.2 Precision, Recall, and F score for detection and localization . . . . . . . 100
B.1 List of Parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
LIST OF FIGURES

1.1 Three categories of integrity attacks in cyber-physical systems 3

2.1 The detection framework 24

2.2 $ARL_1$ vs magnitude of change under replay attack, controller fault and plant fault 28

2.3 The experimental setup 32

2.4 The human-machine interface 32

2.5 Test statistics over time under replay attack and controller fault 33

3.1 The system model 40

3.2 The degradation signal under no attack, in Case 1, and in Case 2 47

3.3 The detection framework 49

3.4 Simulation results: Scenario 1 with $\kappa = 0$; ARL & CPE vs $m_2$; $m_a$ varying from 0 to 1 55

3.5 Simulation results: Scenario 2 with $\kappa = 0$; ARL & CPE vs $m_2$; $m_a$ varying from 0 to 1 56

3.6 System Model Justification 60

3.7 Attack Model Justification 60

3.8 Experimental results, Case 2 62

4.1 The Graphical Model of the power system (IEEE 14-bus) 67
4.2 Simulation Results ...................................................... 84

4.3 Simulation Results ...................................................... 84

5.1 Networked Industrial Control System ............................... 88

5.2 The proposed detection framework (AEN + RNN + DNN) ........ 94

5.3 The IEEE 14-Bus System ............................................. 96

5.4 The SE + DNN Structure ............................................. 98

5.5 The RNN + DNN Structure ........................................... 98

5.6 F1 score under different levels of attack for the three models ...... 102

5.7 The boxplots of accuracy under different number of training levels . 102

A.1 Replay Attack ............................................................ 120

A.2 Controller Fault .......................................................... 120

A.3 Plant Fault ................................................................. 121

A.4 Sensor Fault ............................................................... 122

A.5 Plant Degradation ........................................................ 123

A.6 Experimental results: Test statistics over time under replay attack and equipment faults. ................................. 123

B.1 Simulation results: Scenario 1 with fixed \( D; \) ARL & CPE vs \( m_1; m_a \) varying from 0 to 1 ....................................................... 128

B.2 Simulation results: Scenario 2 with fixed \( D; \) ARL & CPE vs \( m_1; m_a \) varying from 0 to 1 ....................................................... 129

xiii
Initiatives like Industry 4.0 and technological frameworks such as the Internet-of-Things have prompted a growing wave of digital transformation across numerous industrial sectors ranging from manufacturing and power generation plants to critical infrastructure systems like power networks, waste-water management, and natural gas pipeline networks. The transformation of these systems into cyber-physical systems (CPSs) has also created unique cybersecurity vulnerabilities. This thesis focuses on detecting and identifying cyberattacks that target the physical performance and reliability of these systems.

In Chapter 2, we develop an integrated data-driven framework for detecting replay cyberattacks in industrial plants and distinguishing them from naturally occurring equipment faults. We explore how to differentiate replay attacks from four types of equipment fault scenarios namely, controller fault, plant fault, sensor fault, and plant degradation. We derive unique statistical measures and a unique coding scheme for all fault/attack combinations to identify each type of fault and differentiate it from a replay attack. We evaluate our methodology through an extensive numerical studies, and demonstrate its applicability on a rotating machinery application.

Chapter 3 focuses on covert attacks, specifically attacks that target the reliability of critical industrial assets by accelerating their physical degradation. We derive the generic covert attack model under a linear time-invariant dynamic system setting represented by a state-space model and parameterize the relationship between the operating conditions of the plant and the degradation rate of the asset. We derive the mean shift of the residuals under degradation and use that as the basis to identify abnormal degradation rates caused by a covert cyberattack. We design two likelihood ratio tests that use residuals to estimate the onsets of degradation and detect a covert attack. We investigate the impact of system dynamics and severity of a covert attack on detection delay using an extensive numerical study. We also apply our detection model to a rotating machinery testbed.
Chapter 4 develops a cyberattack detection and localization framework for a power
transmission systems comprised of an Independent System operator (ISO) and multiple
Regional Control Centers (RCCs). We demonstrate a generic mechanism of covert attacks
on a regional control center under the setting of a networked power generation control
operated by the RCCs and managed by the ISO. We use the Sparse Group Lasso (SGL)
coupled with the system state estimation to extract and differentiate the impact of a covert
attack on the attacked region and its neighboring regions, which is represented by the SGL
coefficients. These coefficients are used as the basis of our attack detection and localization
scheme, where the magnitude of the coefficients is used for detection, and the sparsity of
the coefficients is used for localization. We demonstrate the effectiveness of our proposed
method through a simulation study on the IEEE 14-bus and the IEEE 118-bus system mod-
els.

In Chapter 5, we extend the power network model to a generic cyber-physical network
setting where multiple assets are connected with a control center. We develop a data-
driven framework that can be used to detect, diagnose, and localize a type of cyberattack
called covert attacks on industrial CPS networks. The framework has a hybrid design
that combines an autoencoder, a recurrent neural network (RNN) with a Long-Short-Term-
Memory (LSTM) layer, and a Deep Neural Network (DNN). This data-driven framework
considers the temporal behavior of a generic physical system that extracts features from
the time series of the sensor measurements that can be used for detecting covert attacks,
distinguishing them from equipment faults, as well as localize the attack/fault. We evaluate
the performance of the proposed method through a realistic simulation study on the IEEE
14-bus model as a typical example of CPS network. We compare the performance of the
proposed method with the traditional model-based method to show its applicability and
efficacy.
CHAPTER 1
INTRODUCTION

1.1 Background

Cyber-physical systems (CPS), or more specifically, industrial control systems (ICS), are a critical part of almost all modern industrial processes and critical infrastructure systems. They are used to control the operation of critical equipment and capital-intensive assets in things like food processing industries, water and wastewater systems, oil and natural gas networks, power grids, and hospital networks. ICSs are typically comprised of numerous control loops, Human Machine Interfaces (HMIs), and remote diagnostics and maintenance tools, which until recently used to have specialized communication and control protocols that made them relatively immune to cyberattacks. Traditional ICSs have become increasingly integrated with standard information technology (IT) components (e.g. operating systems and network protocols). This has increased their vulnerability to cyberattacks. While there is a plethora of cybersecurity tools and models that focus on IT systems (e.g. network traffic and denial of service, DoS), the integration of such tools to secure ICS components is not straightforward. This becomes especially challenging in scenarios where cyberattacks are aimed at inducing physical failures of industrial assets. Consequently, novel approaches are required to not only detect cyberattacks but also distinguish them from naturally occurring equipment faults and physical degradation processes.

Supervisory control and data acquisition (SCADA) systems are a widely used class of industrial control systems. SCADA systems are an integral part of many critical infrastructures systems, most notably electric power systems [1]. They include basic control and sensing components, such as programmable logic controllers (PLCs), networked supervisory computers, remote terminal units (RTUs), sensors, and various components of
human-machine interfaces. Ideally, SCADA systems have their own communication and network protocols that are often isolated from Internet access. However, numerous incidents of cyberattacks have been reported in recent years in which SCADA systems were directly impacted [2] causing devastating effects that ranged from economic losses to societal disruptions.

Among the three security objectives (confidentiality, integrity, and availability) defined in FIPS Pub 199 [3], information integrity has the highest impact on SCADA systems[4]. Most cyberattacks involve data manipulations that impact equipment operations, e.g. [5, 6, 7]. Cyberattacks can be generally classified into the following categories:

- **False data injection** [8, 9, 10, 11, 12, 13] shown in Figure 1.1a refers to attacks that target the manipulation of sensor readings to generate malicious control actions. They assume that the attacker has access to the sensors and has full knowledge of the system.

- **Replay attack** [14, 15] shown in Figure 1.1b refers to attacks where the attacker manipulates the control action and replays the sensor readings from normal operations. The underlying assumption is that the attacker has access to both the sensors and controller but does not have sufficient knowledge of the system.

- **Covert attack** [16, 11, 17, 18] shown in Figure 1.1c refer to scenarios where an attacker, typically a hacker, has access to sensor measurements, system controllers, and also possesses sufficient knowledge of system operations. The attacker disguises the manipulation of control actions by playing the expected sensor measurements calculated based on the knowledge of how the system operates[19]. Consequently, covert cyberattacks can remain undetected for long periods of time while repeatedly inflicting damage. Most covert cyberattacks are intended to induce physical damage of critical assets such as power generators, boilers, etc.—assets that can take down a manufacturing facility, hospitals, water distribution, as well as power grid systems.
Figure 1.1: Three categories of integrity attacks in cyber-physical systems

(a) False Data Injection

(b) Replay Attack

(c) Covert Attack
1.2 Literature Review

The literature on cybersecurity of SCADA systems can be divided into two groups. The first group focuses on vulnerability analysis of SCADA systems[20, 21] and design of adversarial models [22, 11, 8, 19, 23]. The second group, which is more relevant to this chapter, focuses on the design of detection and protection schemes against cyberattacks targeting SCADA systems.

False data injection is one of the popular kinds of cyberattacks that are very difficult to detect, especially when the attacker has full knowledge of the system [8]. Thus, most detection schemes tend to assume that a subset of sensors are protected from false data injection [9, 10]. Under this assumption, a traditional bad-data detection algorithm can be utilized to detect cyberattacks [9, 10]. In these settings, SCADA measurements are commonly modeled using a measurement function that captures the relationship between the state \( x \) and the measurements \( y \) of the form \( y = h(x) + e \), where \( e \) is the error term [8, 9, 11, 12]. The bad-data detector tests the \( \chi^2 \) statistic on measurement residuals. Measurement residuals are differences between the estimated measurements based on state estimations and the actual measurements. In other works, the measurement function is coupled with a state transition function (usually a state-space model) to model system dynamics [13, 10]. The state-space model accounts for system dynamics and control actions. Many of these models rely on statistical control charts to protect the system. For example, in [13] the attacker is assumed to have full knowledge of the detection scheme. A CUSUM control chart of measurement residuals is used for detection of the false data injection. The authors conclude that the CUSUM detection scheme limits the damage to some extent even when the attacker has full knowledge of the detection scheme.

In a replay attack, the attacker can manipulate both the sensor readings and the control actions. In this scenario, a state-space modeling approach is commonly used to describe the system. Detection schemes are based on measurement residuals generated by a state
estimator (e.g. Kalman filter). Since the measurements are manipulated by the attacker, replay attacks cannot be detected by a bad-data detector. Therefore, most works on the detection of replay attacks utilize authentication signals that are assumed to be unknown to the attacker. For example, in [14] and [15], the authors use this approach and detect the attack by checking the correlation between residuals and the authentication signal. This approach is effective but requires careful design and protection of the authentication signal, and the detection accuracy is at the expense of controller performance.

As mentioned earlier, covert attacks are considered the most challenging to detect. There are two popular approaches for detecting covert attacks. The first one is based on the correlation among the sensor measurements. The underlying assumption is that when the system is not under attack, the sensor measurements follow a known correlation structure. When part of the sensor measurements is manipulated, the original correlation structure does not hold. For example, in [18], the authors assume that a subset of the sensors is protected from covert attacks. The covert attack is detected by monitoring the correlation between the protected sensors and the unprotected ones. This correlation is captured by the residuals. The authors use a Kalman filter and a fixed-size parity approach to generate the residuals and use CUSUM statistics to monitor them. Under a covert attack, the protected sensor measurements capture the actual abnormal operation, while the unprotected sensor measurements are manipulated by the attacker to present the normal operation. Hence, the correlation between the protected and unprotected variables is different from normal conditions, which is reflected in the residuals. However, the assumption of the protected sensors may not always be effective because the sensors not only need to be protected but also correlated with those that are not protected. This may restrict the applicability of this approach to specific settings.

The second type of approach is based on analyzing the dynamics of the system. The attacker is assumed to have imperfect knowledge of the system dynamics. Thus, malicious manipulations of some sensor measurements and their control actions will not necessarily
conform to the expectations of the operator, and can, therefore, be detected by monitoring the residuals. For example, in [24], the authors utilize a confidential time-varying system dynamics to authenticate the system operations. They assume that the system dynamics represented by the state transition function changes over time. This prevents the attacker from acquiring full knowledge of the true system dynamics. In [25], the authors assume that there is an extension of the system, called an auxiliary system, which is correlated with the original system. The dynamics of the auxiliary system vary over time and are unknown to the attacker. The detection scheme is similar to [18]. An attack is detected by monitoring the correlation between the residuals generated from the auxiliary system and those generated from the original system. In [26], the detection model assumes that an operator can trigger an undisclosed system dynamic such that attackers are unable to continue undetected. An attack is detected by monitoring the residuals. All these works are based on the premise of pre-defined “secret” system dynamics that are unbeknownst to any attacker. However, the nature and characteristics of these so-called “secret system dynamics” are not well-defined. It is also not clear, whether such strategies can be standardized or whether they are designed on a case by case basis. Additionally, it is not clear how such calibrated system dynamics are protected. In many cases, through access to all sensors and controllers, a covert attacker can learn any system dynamics over time including the secret ones.

Most of the localization schemes against data integrity attacks in the literature are focusing on false data injection attacks, with very few on localization of more complex attacks involving dynamic control such as the covert attack we are focusing on in this chapter. In [27] and [28], the authors proposed an interval observer-based method to detect false data injection attacks using the interval residuals. In [29], a method based on principal component analysis was proposed to detect and localize false data injection attacks. The method proposed in [30] is designed for GPS spoofing attacks, a false data injection attack on PMU sensors. The above methods are designed for detecting and localizing false data injection
attacks, which is equivalent to identifying the contaminated sensors. Some other works focus on fault localization, such as [31]. However, these methods are not designed for cyberattack detection and face the challenge of false alarms triggered by natural faults when used for cyberattack detection. Some distributed schemes are proposed for general attack localization in power systems. In [32], the attack localization is coupled with distributed state estimation, where each region shares its belief of the attack localization. In [33], the authors used a blockchain-based framework for regions to share their local detection results.

1.3 Dissertation Overview

The objective of this thesis is addressing the following challenges in cybersecurity of industrial CPS:

- Detecting data integrity attacks in CPS. This is due to the mechanism and the sophisticated designs of the attacks. Since the attacks are based on the access to sensors and controllers, malicious manipulation of these sensitive data, especially when the attacker has some knowledge about the system operations, can be stealthy.

- Differentiating data integrity attacks from equipment faults. For the same reason mentioned above, even some anomaly detection methods can be used to detect the attacks, it is difficult to identify whether the anomaly is caused by an attack or a natural fault. In other words, data integrity attacks can be easily disguised as natural equipment faults.

- Localizing data integrity attacks in complex CPS networks. Many CPS in reality do not operate individually, but are interconnected with each other. In a complex CPS network such as a water supply system and a power transmission system, the impact of a cyberattack can propagate to other parts of the network via the physical interactions between CPSs. In this case, identifying the source of the attack is challenging,
The first challenge is the basis of this thesis and is covered in each chapter. Based on the other two challenges, this thesis can be divided into two parts: Chapters 2 and 3 focus on differentiating cyberattacks from equipment faults; Chapters 4 and 5 focus on localizing cyberattacks in complex networked systems. The motivations and the overview of each chapter are given below.

1.3.1 Differentiating Cyberattacks from Equipment Faults

The accessibility of cyber-weapons such as the *Stuxnet*[5] virus has broadened the interpretation of cyberattacks and cybersecurity to encompass industrial reliability and asset management. Indeed, attacks on SCADA systems can directly impact asset performance. The Aurora experiment conducted by Idaho National Labs in 2007 demonstrated how a cyberattack could physically destroy a generator by accessing and manipulating its circuit breaker [34]. Manifestations of cyberattacks on industrial SCADA systems can be disguised as machine faults where data signatures from equipment sensors become similar in both instances[35, 36]. Consequently, it is very important to differentiate between cyberattacks and natural equipment faults and degradation. Failure to differentiate the two generates false alarms in detection algorithms against cyberattacks as well as equipment faults. These false alarms will result in unnecessary inspections of the equipment and computer systems, which sometimes require a shut down of the operation that could have been prevented. This research topic proposes a data-driven framework that incorporates analysis of sensor measurement and residuals (post-fit residuals generated by a Kalman filter) for detection and differentiation of system anomalies including replay attack and four types of equipment faults. We derive mathematical formalisms that differentiate replay attack from equipment faults and adopt an ensembled statistical process control method to generate detection signatures of the system anomalies.

Chapters 2 and 3 of the thesis considers a single asset with dynamic control exposed to data integrity attacks. In Chapter 2, we develop a data-driven framework to detect replay
attacks and differentiate it from some common types of equipment faults, including sensor fault, controller fault, plant fault, and plant degradation. The system’s behavior is modeled as a state-space model coupled with a Kalman filter and a linear-quadratic controller. We set up a theoretical framework to derive the characteristics of the data unique to each anomalous event. We use the same framework to also derive the corresponding statistical metrics that can be used to monitor and differentiate these events. These metrics were monitored using an ensemble modeling framework with unique signatures for each anomalous event. This approach is implemented on a physical rotating machinery testbed where the rotational speed is the control variable, and vibration is the condition monitoring variable. We test a replay cyberattack on the rotational speed and to distinguish the attack from a fault in the tachometer (speed sensor), controller fault, and bearing fault and degradation.

In Chapter 3, we develop a model-based methodology that utilizes the characteristics of partial degradation in the system to detect covert attacks. We derive a generic design of covert attacks based on the linear dynamic model, which is shown to be undetectable when the attacker acquires full knowledge of the normal system operations. We assume that the characteristics of partial degradation of the equipment would be unknown to the attacker. In other words, it is reasonable to assume that the natural degradation rate of the equipment captured by condition monitoring sensors is unknown to the attacker. Consequently, covert attacks aimed at accelerating equipment failures can potentially be identified by detecting changes in equipment degradation rates. We develop two sequential likelihood ratio testing algorithms to identify the onset of equipment degradation and that of a covert attack. The algorithms are based on theoretical derivations describing the properties of the system residuals under different conditions (normal, degradation, and attack). The impact of system dynamics and the severity of the attack on the detection delay is investigated through a simulation study. We also proved the applicability of the method on the same rotating machinery setup where a thrust bearing is run to failure.
1.3.2 Localizing Cyberattacks in Complex Networked Systems

Detection of cyberattacks in a power system is a challenging task because of the large scale and the dynamic interactions between the data communication and physical operations and control. This means manipulation of sensitive data may lead to malicious control actions. As an example, the Aurora attack demonstrated by the Idaho national laboratory in 2007 has shown the potential physical damage caused by control data manipulation [34]. Hence, power systems, especially power generation plants, are more vulnerable to data integrity attacks. Data integrity attacks are defined as attacks that maliciously manipulate sensitive data in order to cause physical damage to the system.

In Chapter 4, we consider a power system network of generator and load buses that was divided into multiple regions. The generator buses in each region are controlled locally by the regional control centers (RCC), and all the buses and RCCs are monitored by the centralized independent system operator (ISO). S detection and localization framework is developed based on data dependencies that are defined by the physical connections between a generator bus and its neighbors. We study the system characteristics when a generic covert attack is applied to one of the generator buses. We assume the attacker can manipulate the sensors associated with the target generator bus only. In this case, the residuals of state estimation characterize the sparse features corresponding to the group of sensors associated with the neighbors of the targeted bus. Based on the sparsity, Sparse Group Lasso is used to estimate the potential bias in the system state induced by the attack, which is used to identify the location and severity of the cyberattack.

In Chapter 5, we extend the power network model to a generic cyber-physical network setting where multiple assets are connected with a control center. We demonstrate the application of deep learning towards the detection, diagnosis, and localization of covert attacks. We focus on generic networked industrial control systems and propose a data-driven framework that combines an autoencoder, an RNN, and a DNN. We use the RNN to characterize the system behavior under normal operations. The output of the RNN together
with the sensor measurements are fed to a DNN classifier to detect, diagnose, and localize the anomaly. We use the autoencoder to extract features that represent the system status, as well as the spatial correlation among the nodes, in an unsupervised manner. The RNN captures the temporal behavior of the features extracted by the autoencoder, and the DNN helps detect anomalies in the system as well as diagnose whether it is an attack or fault. By considering both the spatial and temporal behavior of the system, this DL framework helps reduce false alarms triggered by natural faults as well as localize the attack by extracting the features that distinguish anomalies at different locations and between attack and faults.
CHAPTER 2
DETECTION AND DIFFERENTIATION OF REPLAY ATTACK AND
EQUIPMENT FAULTS IN SCADA SYSTEMS

2.1 Introduction

This chapter builds on the existing knowledge base in cybersecurity and fault detection and studies the interplay between (natural) equipment faults and deliberate cyberattacks. The chapter develops an integrated data-driven framework for detecting and differentiating cyberattacks and equipment faults. We focus on a specific type of cyberattack, the “Replay” attack. We consider a simplified system that consists of a plant, sensors, and a controller. In this context, the plant, for example, can be a physical power plant whose operation needs to be monitored by a sensor and controlled by a controller. Sensors are used to track the performance and condition of the plant’s assets. The controller represents the control strategy or logic by which the plant operates. Our methodology is based on modeling the operation of the system using a state-space model consisting of a state transition function, representing the plant, and a measurement function, representing the sensors. Model residuals and sensor readings are used to detect replay attacks and equipment faults. The developments in the chapter are limited to simple systems that can be modeled using a linear time-invariant stochastic state-space model. This analysis allows us to derive closed-form expressions of statistical metrics that help distinguish between the faults and cyberattacks. Furthermore, the detection algorithms developed in this chapter are not intended to be used in isolation. Cyberattacks are almost always multifaceted and therefore the proposed framework is intended to be used in conjunction with other cyberattack detection algorithms. The contributions of this chapter are summarized below.

- We adopt a state-space modeling approach coupled with a Kalman filter and linear-
quadratic Gaussian (LQG) controller to develop a data-driven methodology for detecting natural equipment faults and replay cyberattacks. We utilize mean shift and changes in the covariance structure of the residuals and the sensor readings to differentiate between replay attacks and equipment faults.

- We investigate four types of equipment fault scenarios namely, controller fault, plant fault, sensor fault, and plant degradation. We derive unique statistical measures to identify each type of fault and differentiate it from a replay attack. We develop a unique coding scheme for all fault/attack combinations.

- We perform extensive simulation studies to highlight the average run lengths (ARLs) for each statistic, which allows practitioners to utilize to evaluate the expected detection delay time. We also implement our data-driven cybersecurity model on a rotating machinery application to demonstrate its applicability to real-world settings.

As mentioned earlier, cyberattacks can manifest themselves as equipment faults. In fact, there is a growing interest in developing integrated methods that consider the interplay between cybersecurity, and asset management and reliability. This chapter builds on existing literature and focuses on the differentiation between cyberattacks and equipment faults. To the best of our knowledge, there is no systematic framework that aims to distinguish between these two. Specifically, our work focuses on detecting replay attacks and differentiating them from conventional equipment faults. In line with the literature in this space, we leverage the state-space modeling framework to develop our methodology.

2.2 System Description

Our problem setting involves a system $\mathcal{S}$ that consists of the following components: A plant comprised of physical equipment and assets, sensors for measuring and monitoring the performance of the plant assets, a controller that controls the system based on the sensor
measurements, a Kalman filter (KF) module for learning the system dynamics over time, and a detection scheme for detecting system anomalies (cyberattacks and equipment faults).

In this chapter, we have the following explicit assumptions regarding the system model and the attack model:

1. We assume a linear time-invariant (LTI) system. This assumption is adopted widely in literature including [13, 22, 18, 15, 14, 37]. In the cases where the actual dynamics is nonlinear, such as power systems, the system model is often linearized about the operating point or equilibrium point to reach a simplified system model as well as controller design. Although the LTI assumption might not be practical in some cases, studies based on LTI systems lays the foundation for future studies on more complex and realistic settings.

2. The system is assumed to have two kinds of variables, controllable and uncontrollable, hereafter denoted as $S_{C|U}$. Controllable variables, denoted by the set $C$, are used to determine appropriate control actions. They may include speeds, temperatures, pressures, loads, etc. Uncontrollable variables, denoted by the set $U$, refer to measurements associated with the health and performance of plant assets. Examples of such variables include vibration signals and other measurements that are not part of the control system.

3. Due to the mechanism of replay attacks of manipulating the control actions, i.e., replay attacks target PLCs, which only deal with controllable variables, it is reasonable to assume that the controllable variables are the most vulnerable to cyberattacks. Consequently, we assume that the uncontrollable variables cannot be manipulated by the attacker since they are usually not connected to any equipment drivers and are typically read-only.

2.2.1 State-Space Model

We use a linear time-invariant stochastic state-space model to represent our system. Let $x_k$ denote the state at time $k$ such that $x_k \in \mathbb{R}^m$, where $m$ is the number of state variables, and $q$ of them are controllable variables. Next, let $y_k$ denote the sensor measurements at time $k$ such that $y_k \in \mathbb{R}^n$, where $n$ is the number of sensors. Let $u_k$ denote the control actions at
time \( k \) that are determined by controllable variables, that \( u_k \in \mathbb{R}^p \), where \( p \) is the dimension of the control action. \( v_k \) and \( w_k \) are the process noise and measurement noise terms at time \( k \), which are independent from all the other variables and are assumed to be not affected by any system anomaly. Both \( v_k \) and \( w_k \) follow multivariate normal distributions with zero mean. i.e., \( v_k \sim N(0, Q) \) and \( w_k \sim N(0, R) \), where \( Q \in \mathbb{R}^{m \times m} \) and \( R \in \mathbb{R}^{n \times n} \) are the covariance matrices. The state-transition function and the measurement function are given below:

\[
\begin{align*}
x_{k+1} &= Ax_k + Bu_k + v_k, \\
y_k &= Cx_k + w_k,
\end{align*}
\]

We assume the matrices \( A, B, C, Q \) and \( R \) are known and can be estimated using data pertaining to baseline/normal operating conditions, i.e., a system under no attacks or faults [38]. For simplicity, we assume that \( m = n = p \). The matrix \( C \) is assumed to be identity, which implies that the measurements are noisy realizations of the system states.

To simplify the representation, we rearrange our matrices in terms of controllable and uncontrollable variables. For example, matrix \( A \) can be represented by a \( 2 \times 2 \) block matrix where \( A_c \) and \( A_u \) are submatrices related to the controllable and uncontrollable variables, respectively, and \( A_1 \) and \( A_2 \) are submatrices representing how the controllable variables affect the uncontrollable variables, and vice versa. Hence, we have:

\[
A = \begin{bmatrix} A_c & A_2 \\ A_1 & A_u \end{bmatrix}.
\]

In our setting, we assume that only \( A \) describes the relationship between the controllable variables and uncontrollable variable (\( B_1, B_2 \) and \( B_u \) are 0). We consider two forms for the matrix \( A \) as follows:

1) \( A \) is diagonal implies that there is no interaction between the controllable and un-
controllable variables.

2) A is sparse \((A_2 = 0)\) implies that the mean of controllable variables is not affected by uncontrollable variables, but the converse is not true.

We do not consider the case where \(A\) is dense. A dense matrix \(A\) means all the variables are correlated. Specifically, the controllable and uncontrollable variables temporally depend on each other. In this case, the attack can be easily detected by the traditional detection schemes if the attacker only manipulates the controllable variables. In fact, this case contradicts with our definition of uncontrollable variables, because when all the variables are correlated, all the measurements would be processed by the controller (PLC), and the attacker would easily get access to all the measurements.

2.2.2 Kalman Filter

We use a steady-state Kalman filter (KF) [39] to estimate the system state. KF is an optimal state estimator [40] for the stochastic linear state-space model. The steady-state Kalman gain \(K\) is given by \(K = PC^k(CPC^k + R)^{-1}\), where \(P\) is the steady-state estimation of the error covariance matrix and is given by solving the following Riccati equation:

\[
P = APA^k - APC^k(CPC^k + R)^{-1}CPA^k + Q \tag{2.3}
\]

The KF algorithm is given below:

\[
\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{2.4}
\]

\[
\hat{x}_k = \hat{x}_{k|k-1} + Kr_k \tag{2.5}
\]

\[
r_k = y_k - C\hat{x}_{k|k-1} \tag{2.6}
\]

where \(\hat{x}_{k|k-1}\) denotes the predicted system state given the measurement at time \(k - 1\), \(y_{k-1}\), \(\hat{x}_k\) denotes the updated state estimation given the measurement at time \(k\), \(y_k\), and \(r_k\) denotes the residual at time \(k\), where \(r_k\) is the difference between predicted and actual
measurements as shown in Eq.(2.6).

### 2.2.3 LQG Controller

A linear-quadratic Gaussian (LQG) controller is used to calculate the control action based on the state estimation ($\hat{x}_{k|k}$) given by the KF. The LQG controller minimizes the following objective function: $J = \mathbb{E}[x_k^TWx_k + u_k^TUu_k]$, where $W$ and $U$ are both $q \times q$ positive semidefinite weight matrices ($q$ is the number of controllable variables) that determines the aggressive regulation of the state variables $x_k$ and the cost of control $u_k$, respectively.

For the system presented in Eq.(2.1-2.2), the control action $u_k$ at time $k$ is calculated according to the following equation:

$$u_k = L\hat{x}_{k|k}$$

where $L_1 = L_2 = L_u = 0$, $L_c = -(B_c^TSA_c + U)^{-1}B_cSA_c$, ($L_1, L_2, L_c, L_u$ are submatrices of $L$), and $S$ is the solution to the Riccati equation [41, 42]

$$S = A_c^TSA_c + W - A_c^TSA_c(B_c^TSA_c + U)^{-1}B_c^TSA_c.$$

### 2.3 Theoretical Analysis

This section derives the system characteristics under normal and anomalous conditions. We use two state-space models, $S_{[C|U]}$ and $S_{[U]}$. As is mentioned in Section III, $S_{[C|U]}$ includes a KF and the LQG controller to model the entire system including the controllable and the uncontrollable variables. The second linear time-invariant stochastic state-space model $S_u$ used to model the uncontrollable variables is given below:

$$x_{k+1,u} = A_u x_{k,u} + v_{k,u},$$

$$y_{k,u} = C_u x_{k,u} + w_{k,u},$$
where $x_{k,u}$ and $y_{k,u}$ are the state and measurement of uncontrollable variables at time $k$, respectively, $A_u$ and $C_u$ are submatrices of $A$ and $C$ in Eq. (2.1-2.2), and $v_{k,u}$ and $u_{k,u}$ are the noise terms of the process and measurement functions. A Kalman filter is also used to estimate the state of uncontrollable variables:

\[
\hat{s}_{k|k-1} = A_u \hat{s}_{k-1|k-1}, \quad (2.10)
\]
\[
\hat{s}_{k|k} = \hat{s}_{k|k-1} + \tilde{K} \gamma_k, \quad (2.11)
\]
\[
\gamma_k = y_{k,u} - C_u \hat{s}_{k|k-1}, \quad (2.12)
\]

where $\hat{s}_{k|k-1}$ and $\hat{s}_{k|k}$ are the predicted and the estimated states of uncontrollable variables at time $k$, respectively, $\gamma_k$ is the residual at time $k$, and $\tilde{K}$ is the steady-state Kalman gain of the above KF.

Let $x_k, y_k, u_k, r_k,$ and $\gamma_k$ denote the vectors of the states, measurements, control actions, and residuals generated by the two KFs at time $k$, respectively, when the system is operating under normal conditions. Additionally, let $x'_k, y'_k, u'_k, r'_k$ and $\gamma'_k$ denote the corresponding values under any of the three anomaly cases. Denote the covariance matrix of $r_k, r'_k, y_k,$ and $y_k$ at time $k$ as $\Sigma_r, \Sigma_r', \Sigma_y, \Sigma_y'$, respectively. Recall that $Q_1$ is the submatrix of $Q$ corresponding to the covariance matrix $Cov(v_{k,u}, v_{k,c})$. Assuming that a system anomaly occurs at time $t$, we investigate the characteristics of the mean shifts of the residuals $r_k$ and $\gamma_k$, and the changes in the covariance matrices $r_k$ and $y_k$ for the three system anomalies considered in this study: Replay attack, controller fault, plant fault, sensor fault, and plant degradation.

2.3.1 Replay attack

In a replay attack, we assume that the attacker manipulates the control action $u_k$ by introducing a bias $a_k$ at time $k$, while replaying previous measurements representing normal
operating conditions. The manipulated control action \( u'_k \) is given as:

\[
u'_k = u_k + a_k.
\]  

(2.13)

We also assume that the uncontrollable variables are not affected by the replayed data. Hence, we have,

\[
y'_{k,c} = \tilde{y}_{k,c}
\]

(2.14)

\[
y'_{k,u} = y_{k,u}
\]

(2.15)

for \( k > t \), where \( \tilde{y}_k \) is a previous measurement or a false measurement that is being generated by the attacker. We assume that \( \tilde{y}_k \) and \( y_k \) are independent draws from the same Gaussian distribution.

**Theorem 2.3.1** Suppose a system can be modeled as an \( S_{[c,u]} \), and that \( Q_1 \neq 0 \) and \( a_k \neq 0 \), and a replay attack occurs at time \( t \), the following is true:

- If matrices \( A \) and \( K \) are both diagonal, \( \mathbb{E}[\Delta r_k] = 0 \), \( \mathbb{E}[\Delta \gamma_k] = 0 \), \( \Sigma_{r'}^{k'} \neq \Sigma_r^k \) and \( \Sigma_y^{k'} \neq \Sigma_y^k \) for \( k > t \).

- If matrix \( A \) is sparse where \( A_2 = 0 \) and \( A_1 \neq 0 \), \( \mathbb{E}[\Delta r_{k,u}] \neq 0 \), \( \mathbb{E}[\Delta \gamma_k] \neq 0 \), \( \mathbb{E}[\Delta r_{k,c}] = 0 \), \( \Sigma_{r'}^{k'} \neq \Sigma_r^k \), and \( \Sigma_y^{k'} \neq \Sigma_y^k \) for \( k > t \).

**Proof.** See Appendix A.

Theorem 2.3.1 provides the basis for detecting and differentiating replay attacks from controller and plant faults. It shows that in case 1, a replay attack can be detected by monitoring the covariance matrices \( \Sigma_r^k \) and \( \Sigma_y^k \). On the other hand, the mean values \( \mathbb{E}[r_k] \) and \( \mathbb{E}[\gamma_k] \) remain unchanged. Using this unique character, if an anomaly is detected by monitoring \( \mathbb{E}[r_k] \) and \( \mathbb{E}[\gamma_k] \), we are able to differentiate if from a replay attack. Similar rules apply to case 2.
2.3.2 Controller fault

We demonstrate a controller fault by introducing an error term $a_k$ to the original control action at time $k$ for $k \geq t$. The modified controller action, $u'_{t+i}$, is now defined as follows:

$$u'_{t+i} = L\hat{x}'_{t+i|t+i} + a_{t+i}, \quad i = 0, 1, \ldots$$  \hspace{1cm} (2.16)

By the definition of uncontrollable variables, $a_{t+i, u} = 0$ for all $i \in \{0, 1, 2, \ldots\}$.

**Theorem 2.3.2** Suppose a system can be modeled as an $S[C|U]$, assuming that $Q_1 \neq 0$ and $\text{Cov}(a_k) \neq 0$ (or $a_k = a \neq 0$), and a controller fault occurs at time $t$, the following is true:

- If matrices $A$ and $K$ are both diagonal, $\mathbb{E}[\Delta r_k] = 0$, $\mathbb{E}[\Delta \gamma_k] = 0$, $\Sigma_r' = \Sigma_r$, and $\Sigma_y' = \Sigma_y$ (or $\mathbb{E}[\Delta y_k] \neq 0$) for $k > t$.

- If matrix $A$ is sparse where $A_2 = 0$ and $A_1 \neq 0$, $\mathbb{E}[\Delta r_k] = 0$, $\mathbb{E}[\Delta \gamma_k] \neq 0$, $\Sigma_r' = \Sigma_r$, and $\Sigma_y' \neq \Sigma_y$ (or $\mathbb{E}[\Delta y_k] \neq 0$) for $k > t$.

**Proof.** See Appendix A. \hfill \blacksquare

Theorem 2.3.2 implies the following: In case 1, a controller fault can be detected by monitoring the covariance matrix $\Sigma_y$: In case 2, it can be detected by monitoring the covariance matrix $\Sigma_y$ and the mean value $\mathbb{E}[\gamma_k]$. Theorems 2.3.1 and 2.3.2 show the difference between a replay attack and a controller fault: The residual $r_k$ stays unchanged under a controller fault in case 1 and case 2, while a replay attack causes a change in $\Sigma_r$ in both cases as well as a change in $\mathbb{E}[r_k, u]$ in case 2. Therefore, by monitoring the covariance matrix $\Sigma_r$, we can differentiate a replay attack from a controller fault.
2.3.3 Plant fault

A plant fault introduces an error term $a$ to the states $x_t$ only at time $t$. After time $t$, the system operates according to Eq.(2.1-2.6). In a plant fault, we have

$$x'_t = x_t + a$$

(2.17)

where $t$ is the time when the plant fault occurs.

**Theorem 2.3.3** Suppose a system can be modeled as an $S_{[C|U]}$, assuming that $[A(I - KC)]^m$ converges to 0 in $m$ steps, and a plant fault occurs at time $t$, the following is true:

- If matrices $A$ and $K$ are both diagonal, and the fault occurs in a controllable (uncontrollable) variable, $E[\Delta r_{k,c}] \neq 0$, $E[\Delta r_{k,u}] = 0$, $E[\Delta \gamma_k] = 0$ ($E[\Delta r_{k,c}] = 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta \gamma_k] \neq 0$), and $E[\Delta y_k] \neq 0$ for $t < k < t + m$.

- If matrix $A$ is sparse where $A_2 = 0$ and $A_1 \neq 0$, and the fault occurs in a controllable variable, $E[\Delta r_{k,c}] \neq 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta y_k] \neq 0$, and $E[\Delta \gamma_k] \neq 0$ for $t < k < t + m$; If the fault occurs in an uncontrollable variable, $E[\Delta r_{k,c}] = 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta y_k] \neq 0$, and $E[\Delta \gamma_k] \neq 0$ for $t < k < t + m$.

**Proof.** See Appendix A. □

Theorem 2.3.3 shows that a plant fault can be detected by monitoring the mean value $E[r_k]$ and the covariance matrices $\Sigma^k_r$ and $\Sigma^k_y$. When the plant fault occurs in an uncontrollable variable, it can be detected by monitoring the mean value, $E[\gamma_k]$. Notice that the mean shift in $r_k$ converges to 0 over time. Therefore, the mean shift can only be detected within limited time steps, which can be taken as an identifiable feature of the plant fault.

Theorem 2.3.1 and 2.3.4 show the following: In case 1, we can differentiate a plant fault from a replay attack by monitoring the mean value $E[r_k]$. In case 2, if a plant fault occurs
in a controllable variable, we can differentiate it from a replay attack by monitoring $E[r_k]$, and if a plant fault occurs in an uncontrollable variable, we can differentiate it from a replay attack by identifying whether the mean shift in $E[r_k]$ converges to 0.

### 2.3.4 Sensor fault

A sensor fault refers to when one or more sensors are generating inaccurate (biased or noisy) measurements. In a sensor fault that occurs at time $t$, we have

$$y_{t+i} = y_{t+i} + a_{t+i} \quad \forall i = 0, 1, ... \quad (2.18)$$

**Theorem 2.3.4** Suppose a system can be modeled as an $S_{c|u}$, and that $Q_1 \neq 0$ and $a_k \neq 0$, and a sensor fault occurs at time $t$, the following is true:

- If matrices $A$ and $K$ are both diagonal, and the fault occurs in a controllable variable, $E[\Delta r_{k,c}] \neq 0$, $E[\Delta r_{k,u}] = 0$, $E[\Delta \gamma_k] = 0$, $\Sigma_r^{k'} \neq \Sigma_r^k$, and $\Sigma_y^{k'} \neq \Sigma_y^k$ for $k > t$; If the fault occurs in an uncontrollable variable, $E[\Delta r_{k,c}] = 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta \gamma_k] \neq 0$, $\Sigma_r^{k'} \neq \Sigma_r^k$, and $\Sigma_y^{k'} \neq \Sigma_y^k$ for $k > t$.

- If matrix $A$ is sparse where $A_2 = 0$ and $A_1 \neq 0$, and the fault occurs in a controllable variable, $E[\Delta r_{k,c}] \neq 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta \gamma_k] = 0$, $\Sigma_r^{k'} \neq \Sigma_r^k$, and $\Sigma_y^{k'} \neq \Sigma_y^k$ for $k > t$; If the fault occurs in an uncontrollable variable, $E[\Delta r_{k,c}] \neq 0$, $E[\Delta r_{k,u}] \neq 0$, $E[\Delta \gamma_k] \neq 0$, $\Sigma_r^{k'} \neq \Sigma_r^k$, and $\Sigma_y^{k'} \neq \Sigma_y^k$ for $k > t$.

**Proof** See Appendix A. ■

Theorem 2.3.4 shows that: in case 1 where $A$ is diagonal, a sensor fault can be detected by monitoring the covariance matrices $\Sigma_r^k$ and $\Sigma_y^k$, and whether it is in a controllable or an uncontrollable variable can be diagnosed by monitoring the mean shift in the residuals $r_{k,c}$, $r_{k,u}$ and $\gamma_k$. In case 2, a sensor fault can be detected by monitoring the covariance matrices...
\( \Sigma^k_r \) and \( \Sigma^k_y \), as well as the mean shift in the residual \( r_k \), and whether it is in a controllable or an uncontrollable variable can be diagnosed by monitoring the mean shift in the residual \( \gamma_k \). Theorems 2.3.1 and 2.3.3 show that: in case 1, we can differentiate a replay attack from a sensor fault by monitoring the mean shift in the residuals, and in case 2, we can differentiate a replay attack from a sensor fault by monitoring the mean shift in the residual \( r_{k,c} \).

### 2.3.5 Plant degradation

We model a plant degradation as a constant bias \( a \) introduced to the uncontrollable variables at each time step. Hence, for a plant degradation starting at time \( t \), we have

\[
x_{t+i} = x_{t+i} + a \quad \forall i = 0, 1, \ldots
\]

where \( a_c = 0 \) by definition.

**Theorem 2.3.5** Suppose a system can be modeled as an \( S_{\text{c|u}} \), and a plant degradation occurs at time \( t \), then following is true: \( \mathbb{E}[\Delta r_{k,c}] \neq 0 \), \( \mathbb{E}[\Delta r_{k,u}] \neq 0 \), \( \mathbb{E}[\Delta \gamma_k] \neq 0 \), \( \Sigma^k_r \neq \Sigma^k_r \), and \( \Sigma^k_y \neq \Sigma^k_y \) for \( k > t \)

**Proof** See Appendix A.

Theorem 2.3.5 shows that we can detect a plant degradation by monitoring the mean shift in the residuals \( r_k \) and \( \gamma_k \), as well as the covariance matrices \( \Sigma^k_r \) and \( \Sigma^k_y \). Hence, by Theorems 2.3.1 and 2.3.5, in case 1, a replay attack can be distinguished from a plant degradation by monitoring the mean shift in the residuals \( r_k \) and \( \gamma_k \), and in case 2, this can be done by monitoring the mean shift in \( r_{k,c} \).

### 2.4 Detection and Differentiation Framework

In this section, we propose a data-driven methodology for detecting and differentiating cyberattacks and equipment faults. As shown in the previous section, different system
The detection characteristics are expressed in the mean values of residuals $r_k$ and $\gamma_k$, and the covariance matrices $\Sigma_{r_k}$ and $\Sigma_{y_k}$ under normal and the considered anomalous conditions. This makes it possible for us to detect equipment faults and distinguish their signatures from a replay cyberattack.

The proposed framework is illustrated in Fig. 2.1. We utilize an ensemble control chart modeling framework to generate unique signatures for detecting and distinguishing replay attacks, controller faults, and plant faults. We monitor the mean shifts in residuals $r_k$ and $\gamma_k$ and the change in covariance matrices of $r_k$ and $y_k$.

### 2.4.1 Monitoring the Mean of the Residuals

Based on Theorem 2.3.4, the mean shifts in the residuals $r_k$ and $\gamma_k$ can be used to differentiate a replay attack from a plant fault, as well as to identify the affected variables in a plant fault. Hence, we use the (multivariate) exponentially weighted moving average (EWMA or MEWMA) control chart to detect mean shifts in the residuals generated by the two state-space models mentioned earlier, $r_k$ and $\gamma_k$. When the system is not under replay attack and there are no equipment faults, $r_k$ and $\gamma_k$ both follow Gaussian distributions given by $r_k \sim N(0, CPC_k + R)$, and $\gamma_k \sim N(\mu, C_u \tilde{P}C_u^k + R_u)$ [43]. The mean $\mu$ of the residuals $\gamma_k$ depends on the structure of the matrix $A$ and the target state of the controller, $x_{trg}$. As
mentioned earlier, we consider two scenarios: when $A$ is diagonal, $\mu$ is equal to 0; and when $A$ is sparse, and $\mu$ is given by the result in [44]. If $x_{trg} = 0$, then $\mu = 0$.

An MEWMA control chart is used to monitor the mean of $\gamma_k$. For $r_k$, we use two MEWMA control charts to monitor $r_{k,c}$ and $r_{k,u}$ separately. The test statistic used for the MEWMA chart is given by, $T^2_k = [z_k - \mu_0]^{T}\Sigma^{-1}_z [z_k - \mu_0]$, where $z_k = \lambda l_k + (1 - \lambda) z_{k-1}$, and $\Sigma_z = \frac{\lambda}{2 - \lambda} [1 - (1 - \lambda)^2] \Sigma$, where $\lambda \in [0,1)$ in (M)EWMA is the coefficient that determines the depth of memory of the control chart with smaller values meaning the control chart has a deeper memory of the history, and $l_k$ is the testes residual at time $k$ and $\Sigma$ is the covariance matrix of the tested residual. The alarm is triggered whenever $T^2_i$ is above the $1 - \alpha$ quantile of its empirical distribution under normal operation, and $\alpha$ is the desired type-I error rate. For the univariate case (when there is only one controllable/uncontrollable variable), we monitor $z_k$ directly and the alarm occurs whenever $|z_k - \mu_0| > \Phi^{-1}(1 - \frac{\alpha}{2}) \sqrt{\frac{\lambda}{2 - \lambda} [1 - (1 - \lambda)^2] \sigma_0}$, where $\sigma_0$ is the variance of the corresponding residual.

### 2.4.2 Monitoring of the Covariance Matrices

Based on Theorems 2.3.1 and 2.3.2, in case 1 the controller fault can be detected by monitoring the covariance matrix $\Sigma_{y_k}$, and the replay attack can be differentiated from a controller fault by monitoring $\Sigma_{r_k}$. Therefore, we use $PCA + \chi^2$ detection scheme to monitor changes in the covariance matrices of the residual $r_k$ and the measurement $y_k$. We apply PCA on $r_k$: $\Sigma_0 = \Sigma_e \Lambda \Sigma_e^{-1}$, $\zeta_k = \Sigma^T_e r_k \Lambda^{-1/2}$, where $\Sigma_0 = CPC^T + R$ is the covariance matrix of $r_k$, and $\Sigma_e \Lambda \Sigma_e^{-1}$ is the eigendecomposition of $\Sigma_0$. $\zeta_k$ is the standardized vector of PC scores for residual at time $k$. We standardize the PC scores $\Sigma^T_e r_k$ according to their variance (eigenvalues), so each of the standardized PC scores should follow a standard normal distribution and independent from each other. The $\chi^2$ test statistic is, $T_{\chi^2,k} = \zeta_k^{T} \zeta_k$. Under no replay attack or equipment fault, $T_{\chi^2,k} \sim \chi^2_m$ where $m$ is the number of selected variables. In this chapter, we use all the PC scores ($m = p$). For cases involving a large number of variables, a selected subset of PC scores based on the proportion of variation
explained can be used. An alarm is raised whenever $T_{\chi^2,k}^2 > \chi^2_{m,\alpha}$, where $\chi^2_{p,\alpha}$ is the $\alpha$ level upper-quantile of $\chi^2$ distribution with $p$ degrees of freedom. We apply the same hypothesis testing on the measurement $y_k$ to detect change in covariance matrix of $y_k$. The covariance matrix of $y_k$ is estimated based on historical data obtained from normal operations.

2.4.3 Detection Signatures

In this chapter, we consider different scenarios for different types of anomalies. Specifically, for the replay attack (RA) and the controller fault (CF), we consider two scenarios: 1) The error term $a_k$ is in the form of a constant bias, i.e., $a_k = a$. These scenarios are designated by RA-B and CF-B; and 2) The error term is in the form of white noises, i.e., $a_k \sim N(0, \Sigma)$ where $\Sigma$ is a diagonal matrix. This scenario is designated by RA-N and CF-N. For the plant fault (PF), we consider whether the plant fault occurs in the controllable variables (designated by PF-C) or the uncontrollable variables (designated by PF-U). We consider all the foregoing scenarios for Cases 1 and 2 defined earlier based on the structure of matrix $A$. According to Theorems 2.3.1-2.3.4 and as described in the detection framework, we can use a subset of the monitoring statistics to detect each anomalous event. We define each subset by a series of five binary numbers that respectively correspond to the monitoring methods, (M)EWMA for $r_{k,c}$, (M)EWMA for $r_{k,u}$, PCA+$\chi^2$ for $r_k$, PCA+$\chi^2$ for $y_k$, and (M)EWMA for $\gamma_k$. In this series, “1” indicates that the corresponding monitoring method is capable of detecting the anomaly and “0” indicates otherwise. These series are referred to as “signatures” in this chapter.

Tab.2.1 shows all scenarios with the corresponding signatures. For example, scenario 1.1 RA-B represents the scenario where the system is under a replay attack where the error term $a_k = a$ for all $k$, and the $A$ matrix is diagonal. The signature “00110” represents that PCA+$\chi^2$ for $r_k$ and PCA+$\chi^2$ for $y_k$ are able to detect the replay attack. It is worth noticing that in case 2, the signatures of “PF-U” coincides with the signature of “RA-B/N”. We have proved in the previous section that plant fault causes a mean shift in $r_k$ that converges to 0.
which presents a “spike” in the control chart. This typical feature of the plant fault helps us differentiate it from a replay attack. Also notice that the $\chi^2$ detectors for monitoring the covariance matrices are also sensitive to mean shifts. Therefore, by Theorem 2.3.4, a plant fault can be detected by the $\chi^2$ detectors for monitoring the covariance matrices $\Sigma^k_r$ and $\Sigma^k_y$.

Table 2.1: Signature correspondence with types of fault/attack

<table>
<thead>
<tr>
<th>#Scenario (Case 1)</th>
<th>Anomaly</th>
<th>Signature</th>
<th>#Scenario (Case 2)</th>
<th>Anomaly</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>RA-B</td>
<td>00110</td>
<td>2.1</td>
<td>RA-B</td>
<td>01111</td>
</tr>
<tr>
<td>1.2</td>
<td>RA-N</td>
<td>00110</td>
<td>2.2</td>
<td>RA-N</td>
<td>01111</td>
</tr>
<tr>
<td>1.3</td>
<td>CF-B</td>
<td>00010</td>
<td>2.3</td>
<td>CF-B</td>
<td>00011</td>
</tr>
<tr>
<td>1.4</td>
<td>CF-N</td>
<td>00010</td>
<td>2.4</td>
<td>CF-N</td>
<td>00011</td>
</tr>
<tr>
<td>1.5</td>
<td>PF-C</td>
<td>10110</td>
<td>2.5</td>
<td>PF-C</td>
<td>10110</td>
</tr>
<tr>
<td>1.6</td>
<td>PF-U</td>
<td>01111</td>
<td>2.6</td>
<td>PF-U</td>
<td>01111</td>
</tr>
<tr>
<td>1.7</td>
<td>SF-C-B</td>
<td>10110</td>
<td>2.7</td>
<td>SF-C-B</td>
<td>11110</td>
</tr>
<tr>
<td>1.8</td>
<td>SF-C-N</td>
<td>10110</td>
<td>2.8</td>
<td>SF-C-N</td>
<td>11110</td>
</tr>
<tr>
<td>1.9</td>
<td>SF-U-B</td>
<td>01111</td>
<td>2.9</td>
<td>SF-U-B</td>
<td>11111</td>
</tr>
<tr>
<td>1.10</td>
<td>SF-U-N</td>
<td>01111</td>
<td>2.10</td>
<td>SF-U-N</td>
<td>11111</td>
</tr>
<tr>
<td>1.11</td>
<td>PD-U</td>
<td>01111</td>
<td>2.11</td>
<td>PD-U</td>
<td>11111</td>
</tr>
</tbody>
</table>

2.5 Numerical and Case Study

We conduct a numerical study to evaluate the performance of the detection and distinction framework, as well as its sensitivity regarding the magnitude of attack or faults. In a case study, we conduct physical experiments on a rotating machinery setup to study the applicability of our methodology.

2.5.1 Numerical Study

In the numerical study, without loss of generality, we consider a system with two controllable variables ($q = 2$) and an uncontrollable variable ($p = 3$). We assume that $W = U = B_c = I_2$, $C = I_3$, $R = 0.1I_3$, where $I_k$ is the $k$-dimensional identity ma-
Figure 2.2: $ARL_1$ vs magnitude of change under replay attack, controller fault and plant fault

$$Q = [q_{ij}] = [\rho^{\mid i-j \mid}] \text{ with } \rho = 0.5, \text{ and } A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \text{ and } \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.1 & 0.1 \end{bmatrix} \text{ in cases 1 and 2.}$$

For each scenario, we simulate different magnitudes $m$ (ranging from 0.1-6). The magnitude $m$ is an indicator of the severity of the anomaly and it is defined as follows: When the error term $a_k$ is a bias, $a_k = a = m\sigma_{x_i}e_i$ where $\sigma_{x_i}$ is the standard deviation of the affected state variable $x_i$, and $e_i$ is a indicator vector of $i$ (a $p \times 1$ vector with the $i^{th}$ element equal to 1 and the rest equal to 0); when the error term $a_k$ is a white noise, $a_k$ is drawn from a multivariate normal distribution with the mean of zero and the covariance of $\Sigma$ that is a diagonal matrix with its diagonal elements equal to $m^2\sigma_{x_i}^2 e_i$. The control limits of all control charts are chosen based on the empirical distributions of their corresponding test statistics. The type-I error rate, $\alpha$, is set to 0.005 so that the in-control average run length $ARL_0 = 200$. 1000 replications are simulated for each case and the out-of-control average run length $ARL_1$ (the expected number of observations before the first alarm raised when the system changes) for scenarios 1.1, 1.3, 1.5, 2.1, 2.3, and 2.5 are shown in Fig.2.2.
Results for other cases are in the supplementary file. In each subplot, the $ARL_1$ (vertical axis) for each monitoring statistic is plotted against the magnitude $m$ (horizontal axis). As you can see from the plots, when $m$ increases, the $ARL_1$ values decrease for the monitoring statistics that can detect the anomalous event, while the $ARL_1$ values stay around 200 for those statistics that are not sensitive to the anomaly. For example, in the first plot (1.1 RA-B Case 1), the $ARL$’s for MEWMA, EWMA, and gamma $(\gamma_k)$ stay around 200 as $m$ increases, while the $ARL$’s for PCA-r and PCA-y are significantly lower than 200. This means that, in case 1, the PCA+$\chi^2$ detectors for $r_k$ and $y_k$ are sensitive to the replay attack where $a_k$ is a constant, while the other control charts are not sensitive to the replay attack. This simulation result is consistent with the derived signature (00110) for scenario 1.1.

The simulation result validates the detection signatures presented in Tab.2.1. This numerically proves that the monitoring statistics derived from our theoretical results form signatures that differentiate the replay attack from equipment faults. It also shows that as the severity of equipment faults or a replay attack increases, the expected detection delay time decreases.

To demonstrate the performance of the proposed method when the linearity assumption of the system dynamics does not stand, we apply the method to a quadratic system, where the state transition function is in the following form:

$$x_{k+1} = Ax_k + (Fx_k) \odot x_k + Bu_k + v_k.$$ 

In the above equation, $F$ is the weight matrix for quadratic terms, and $\odot$ represents the Hadamard (element-wise) product. In the simulation, we use extended Kalman filters for state estimation for both the complete state $x_k$ and the uncontrollable variables $x_{k,u}$, in replacement of the two Kalman filters used for the linear system. As for control, we keep the LQG controller by linearizing the system at each time step according to the state estimation $\hat{x}_{k\mid k}$ given by the extended Kalman filter and calculate the control action $u_k$ according to
Section III.C. The parameter values used in the simulation are given in Appendix B. We show the comparison of the derived signature and the observed signatures inferred from the out-of-control average run length ($ARL_1$) for all the cases in Table 2.2. The magnitude of attack/fault in the simulation is set to $m = 2$, and the in-control ARL is set to 200, so the $ARL_1$’s significantly less than 200 are inferred as “1”’s in the observed signature, and “0” otherwise. In general, the observed signatures match the derived signatures. However, there are a few exceptions (labeled with * in the table): As shown in Scenarios 2.9-2.11, when the fault occurs on the uncontrollable variables, the MEWMA control chart, which is used to monitor the mean shift in the controllable variables, fails to detect the anomaly. This result is different from when the system is linear. Intuitively speaking, this is caused by the nonlinear temporal correlation in this case. Since the mean shift of controllable variables is no longer a linear function of the mean shift of the uncontrollable variables, it could be close to 0 under some parameter settings, like the one we use in this simulation. Similarly, in Scenarios 2.7 and 2.8, when the sensor fault occurs on the controllable variables, the EWMA control chart for the uncontrollable variable also fails to detect it. In summary, our proposed method still successfully detects the replay attack and distinguishes it from the considered faults in Case 1. While in Case 2, it successfully detects all anomalies, but might fail to distinguish a replay attack from a plant degradation or a sensor fault in the uncontrollable variables.

2.5.2 Case Study

We also demonstrate our methodology by experimenting with a rotating machinery setup shown in Fig.2.3. In the experiment, a thrust bearing is supporting a rotating shaft in the vertical direction. The shaft is belt-driven by a motor. The voltage of the motor can be adjusted to control the rotating speed at the target level. A hydraulic system provides pressure on the shaft from the top to simulate the loading condition of the bearing. A tachometer is monitoring the rotating speed, and a vibration sensor is measuring the vibration signal...
Table 2.2: Signature comparison with $ARL_1$ for nonlinear system

<table>
<thead>
<tr>
<th># Scenario</th>
<th>Anomaly Type</th>
<th>Theoretical Signature</th>
<th>Observed Signature</th>
<th>MEWMA</th>
<th>EWMA</th>
<th>PCA-γ</th>
<th>PCA-y</th>
<th>EWMA for $\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>RA-B</td>
<td>00110</td>
<td>00110</td>
<td>217.300</td>
<td>223.510</td>
<td>79.02</td>
<td>91.84</td>
<td>219.18</td>
</tr>
<tr>
<td>1.2</td>
<td>RA-N</td>
<td>00110</td>
<td>00110</td>
<td>231.385</td>
<td>218.135</td>
<td>77.08</td>
<td>87.06</td>
<td>193.1</td>
</tr>
<tr>
<td>1.3</td>
<td>CF-B</td>
<td>00010</td>
<td>00010</td>
<td>210.85</td>
<td>207.86</td>
<td>236.66</td>
<td>45.63</td>
<td>180.92</td>
</tr>
<tr>
<td>1.4</td>
<td>CF-N</td>
<td>00010</td>
<td>00010</td>
<td>176.24</td>
<td>192.51</td>
<td>180.68</td>
<td>15.80</td>
<td>207.32</td>
</tr>
<tr>
<td>1.5</td>
<td>PF-C</td>
<td>10110</td>
<td>10110</td>
<td>217.300</td>
<td>223.510</td>
<td>79.02</td>
<td>91.84</td>
<td>219.18</td>
</tr>
<tr>
<td>1.6</td>
<td>PF-U</td>
<td>01111</td>
<td>01111</td>
<td>198.19</td>
<td>80.41</td>
<td>141.24</td>
<td>145.46</td>
<td>73.71</td>
</tr>
<tr>
<td>1.7</td>
<td>SF-C-B</td>
<td>01110</td>
<td>01110</td>
<td>20.68</td>
<td>187.57</td>
<td>44.69</td>
<td>43.57</td>
<td>251.24</td>
</tr>
<tr>
<td>1.8</td>
<td>SF-C-N</td>
<td>10110</td>
<td>10110</td>
<td>5.32</td>
<td>187.70</td>
<td>5.98</td>
<td>4.59</td>
<td>207.02</td>
</tr>
<tr>
<td>1.9</td>
<td>SF-U-B</td>
<td>01111</td>
<td>01111</td>
<td>174.58</td>
<td>1.86</td>
<td>3.29</td>
<td>2.29</td>
<td>3.31</td>
</tr>
<tr>
<td>1.10</td>
<td>SF-U-N</td>
<td>01111</td>
<td>01111</td>
<td>154.39</td>
<td>6.94</td>
<td>9.84</td>
<td>3.36</td>
<td>5.75</td>
</tr>
<tr>
<td>1.11</td>
<td>PD-U</td>
<td>01111</td>
<td>01111</td>
<td>183.88</td>
<td>5.91</td>
<td>10.90</td>
<td>17.15</td>
<td>10.70</td>
</tr>
<tr>
<td>2.1</td>
<td>RA-B</td>
<td>01111</td>
<td>01111</td>
<td>225.095</td>
<td>47.695</td>
<td>21.115</td>
<td>58.635</td>
<td>29.80</td>
</tr>
<tr>
<td>2.2</td>
<td>RA-N</td>
<td>01111</td>
<td>01111</td>
<td>215.900</td>
<td>11.130</td>
<td>11.130</td>
<td>11.520</td>
<td>12.12</td>
</tr>
<tr>
<td>2.3</td>
<td>CF-B</td>
<td>00011</td>
<td>00011</td>
<td>190.48</td>
<td>198.75</td>
<td>219.43</td>
<td>62.18</td>
<td>146.06</td>
</tr>
<tr>
<td>2.4</td>
<td>CF-N</td>
<td>00011</td>
<td>00011</td>
<td>168.98</td>
<td>178.83</td>
<td>192.02</td>
<td>15.98</td>
<td>76.18</td>
</tr>
<tr>
<td>2.5</td>
<td>PF-C</td>
<td>10110</td>
<td>10110</td>
<td>77.85</td>
<td>191.35</td>
<td>86.68</td>
<td>52.45</td>
<td>248.78</td>
</tr>
<tr>
<td>2.6</td>
<td>PF-U</td>
<td>01111</td>
<td>01111</td>
<td>195.39</td>
<td>80.81</td>
<td>137.12</td>
<td>89.83</td>
<td>100.62</td>
</tr>
<tr>
<td>2.7</td>
<td>SF-C-B</td>
<td>11110</td>
<td>10110*</td>
<td>69.22</td>
<td>308.82</td>
<td>138.47</td>
<td>113.22</td>
<td>289.35</td>
</tr>
<tr>
<td>2.8</td>
<td>SF-C-N</td>
<td>11110</td>
<td>10110*</td>
<td>5.10</td>
<td>246.87</td>
<td>5.81</td>
<td>4.11</td>
<td>253.55</td>
</tr>
<tr>
<td>2.9</td>
<td>SF-U-B</td>
<td>11111</td>
<td>01111*</td>
<td>186.43</td>
<td>2.83</td>
<td>6.53</td>
<td>3.57</td>
<td>1.23</td>
</tr>
<tr>
<td>2.10</td>
<td>SF-U-N</td>
<td>11111</td>
<td>01111*</td>
<td>166.50</td>
<td>7.67</td>
<td>8.61</td>
<td>3.13</td>
<td>6.81</td>
</tr>
<tr>
<td>2.11</td>
<td>PD-U</td>
<td>11111</td>
<td>01111*</td>
<td>193.71</td>
<td>4.84</td>
<td>6.85</td>
<td>5.60</td>
<td>3.16</td>
</tr>
</tbody>
</table>

to monitor the performance of the system. The measurements of speed and vibration are sent to a PC as a supervisory computer. A DAQ (data acquisition) board is used for data transformation between analog and digital to communicate between the equipment and the PC. An LQG control strategy as well as the human-machine interface (shown in Fig.2.4) is coded on the supervisory computer using the software package Labview. Under this setting, the rotating speed is the controllable variable, and the vibration is the uncontrollable variable.

We use the historical data of the bearing to estimate the states-space model and degradation parameters using the method in [45]. To estimate these parameters, we monitor the following variables under normal operations: rotating speed, vibration (rms values), the voltage of the driving motor, and control action. The size of the dataset we use to estimate the state-space model is 4 variables $\times$ 1000 observations $\times$ 8 replications. Under this setting, the rotating speed (rpm) is assumed to be the controllable variable and the vibration (the root mean square of the vibration signal) is the uncontrollable variable. The
sampling frequency is 1 per 5 seconds (i.e., 0.2 Hz). A 2-variable state-space model where 
\[ x^T = [\text{speed (rpm)}, \text{vibration}] \] is learned from historical data to model this system, and the 
estimated parameters are given below. Based on a goodness-of-fit test, the dynamics of the 
system follows a linear state-space model where the \( A \) matrix fits the form presented in 
case 2. The detailed model is given in Appendix B.

In detection phase, the two Kalman filters are designed based on the state-space model 
above, which calculates the residuals \( \{r_k\} \) and \( \{\gamma_k\} \) in real time. We monitor the above 
parameters as well as the manipulated sensor readings and the residuals for speed and 
vibration. The size of the raw dataset for monitoring is 8 variables \( \times \) 213 observations for 
RA, and 8 variables \( \times \) 227 observations for CF. The first 20 and the last 3 observations 
of the data are removed for both cases to eliminate the cold start phase and the instability 
when the machine is turned off.

![Figure 2.3: The experimental setup](image)

![Figure 2.4: The human-machine interface](image)
Figure 2.5: Test statistics over time under replay attack and controller fault.

We simulate the anomalous conditions by changing the logic of the controller or the sensor on the supervisory computer. The residuals $r_k$, $\gamma_k$ and the measurement $y_k$ are monitored in real-time. The control limits of the control charts are estimated from the data collected under normal operations. In the experiment, each data collection is started with a one-minute normal operation before the system anomaly occurs. The test statistics overtime under a replay attack and a controller fault are shown in Fig.2.5. The red vertical line indicates the time point when the event occurs. An alarm is raised when the test statistic is outside the bandwidth between the two horizontal lines. In the first plot, the frequency of alarm does not increase after the occurrence of the replay attack, which means the MEWMA for $r_{k_e}$ failed to detect the replay attack. This result is identical to the corresponding “0” in the signature. Comparing the detection signatures with experimental results, the performance of the test statistics under these scenarios is identical to that of theoretical analysis. The experimental results show that the designed framework is applicable to real-world settings.

2.6 Conclusion

In this chapter, we consider a practical problem of detection and differentiation of replay attack and equipment faults in SCADA systems. We develop a data-driven methodology based on a modeling framework consisting of a state-space model, a Kalman filter, and an
LQG controller. The measurement and the residuals generated by the two models build the basis of our detection and differentiation framework. We design an ensemble control chart methodology using different statistical metrics that generate unique signatures for the considered anomalies including replay attack, controller fault, plant fault, sensor fault, and plant degradation. We derive the characteristics of the mean shift in the residuals $r_k$ and $\gamma_k$ and covariance change in the residual $r_k$ and the measurement $y_k$, which can be utilized to derive the signatures for each anomaly under different scenarios. A simulation study shows that the signatures derived from the theoretical analysis have a one-to-one correspondence with the types of anomalies. The detection power of the proposed increases as the severity of the anomaly increases. An experiment on the rotating machinery setup proves the applicability of our methodology in real-world settings.

Under the assumptions of this chapter, since the system dynamics is represented by a state-space model, the modeling parameters include matrices $A$, $B$, $C$, $Q$, and $R$. Therefore, a larger system requires more model parameters to be estimated with the number of parameters growing in the order of $n^2$. However, since we are dividing the system variables into two types and monitor the residuals and measurements as a whole, the number of control charts does not grow with the number of parameters. As for the computation of the test statistics, both (M)EWMA control charts and PCA+$\chi^2$ requires inversion of the covariance matrix of the residuals or the measurements, which is of size $n \times n$ or $m \times m$. However, the covariance matrix does not change over time, and hence, this inversion only needs to be calculated once and stored. For nonlinear systems, a simplified linear model can be established by linearizing the system dynamics around the operating point. For networked systems where the complete system can be decomposed into small subsystems where the operations are independent, our method can be applied to each subsystem separately. The detection results from each subsystem can be gathered in a centralized or distributed manner to decide whether the complete system is under a coordinated attack.

Our study serves as one of the fundamental building blocks of more complex settings.
For future study, a possible direction is to incorporate machine learning techniques to deal with more complex and nonlinear systems (e.g., power systems). Specifically, recurrent neural networks are very flexible in modeling temporal correlation. It also well couples with other deep neural networks in architecture, which could be trained to recognize specific patterns that helps identify anomalous events such as cyberattacks and faults. This chapter provides insights on how these machine-learning based methods could be applied to solve this problem. Another future direction is on how to detect smarter cyberattacks such as covert attack, where the attacker has acquired much system knowledge to design attacks to fully decouple the feedback control in the plant and the surveillance of the operator. Under such type of attack, all the data that is observed by the operator is normal, which makes the attack extremely hard to detect.
CHAPTER 3
A DEGRADATION-BASED DETECTION FRAMEWORK AGAINST COVERT CYBERATTACKS IN SCADA SYSTEMS

3.1 Introduction

This chapter focuses on long-term covert cyberattacks designed to diminish asset reliability by accelerating physical degradation and increasing failure rates. There are two reasons why we focus on the long-term attack that accelerates the degradation. First, abrupt damages can be achieved by less sophisticated attacks (such as false data injections and replay attacks). Most ICSs are also equipped with comprehensive systems to notice and analyze abrupt failures. In this situation, the abrupt faults could be easily detected and further prevented. Considering the difficulty of acquiring the system knowledge, it is not beneficial for a covert attacker to do so. The second reason is the greed of the attacker. Compared to unexpected faults, a decline in the system reliability could hardly be noticed until a large number of failures have been analyzed, which means a long-term covert attack could be far more damaging to the system than short-term attacks. The well-known virus Stuxnet is a typical example that shows how much damage a long-term covert attack can bring to the system: The virus was designed to raise the rotating speed of the centrifuges to a higher level that accelerates their degradation. Only after it had caused more than 1000 failures of the centrifuges did the virus caught people’s attention and got revealed [46]. Since covert attacks are difficult to detect, their effects often manifest as classic reliability challenges. To an unassuming plant operator, repeated equipment failures will almost always be attributed to a poor batch quality, installation faults, poor maintenance practices, etc.—the common causes of poor reliability.

Most of the existing models used for detecting covert cyberattacks can be grouped
into two categories. The first group of literature assumes that a subset of sensors is protected from any manipulation by the attacker [18]. The relationship between this subset of immune sensors and the rest of the plant variables becomes the basis of the detection algorithm. The second category assumes that the attacker has only partial knowledge of the system dynamics. These partial knowledge is utilized for developing detection models in a similar fashion to the first category [25, 24, 26]. Some of these assumptions may require redesigning elements of an existing system or simply do not apply.

Our approach is based on a data-driven methodology that models the natural degradation and reliability characteristics of industrial assets to detect covert attacks on SCADA systems. Our proposed detection algorithm is not intended to work as a stand-alone algorithm. Instead, it is designed to work in conjunction with existing intrusion detection algorithms. However, it acts as a final line of defense of all else fail. The contributions of this chapter can be summarized as follows: We derive the generic covert attack model under a linear time-invariant dynamic system setting represented by a state-space model. We parameterize the relationship between the operating conditions of the plant and the degradation rate of the asset using a state-space model. We derive the distribution of the residuals under degradation and use that as the basis to identify abnormal degradation rates caused by a covert cyberattack. We design two likelihood ratio tests that use residuals to estimate the onsets of degradation and detect a covert attack. We investigate the impact of system dynamics and the severity of a covert attack on detection delay using an extensive numerical study. We also apply our detection model to a rotating equipment and simulate a covert attack to demonstrate the applicability of our model.

This chapter is based on the premise of detecting covert cyberattacks by monitoring anomalies in the system dynamics. However, we extend the concept of using predefined system dynamics to encompass equipment degradation. That is, rather than defining secret system dynamics that can also be observed by a potential attacker, we utilize anomalies in the natural degradation rate of critical assets. This premise is supported by the fact
that in many real-world cases of cyberattacks on SCADA systems are intended to damage critical assets of the system. The advantage of our approach is that it addresses some of the restrictive assumptions commonly used in the literature. Additionally, it can be standardized across different industrial application domains.

### 3.2 Problem Setting

This chapter focuses on identifying the underlying system properties and requirements necessary for detecting potential covert attacks on industrial SCADA systems. We consider a system consisting of four components: a plant, sensors, a Kalman filter, and a linear controller. A schematic diagram of the system is given in Fig. 3.1. The dynamics of the system are characterized by a state-space model. The state of the plant, $x$, is modeled by a state-transition function. We assume in practice the system undergoes degradation (or partial degradation). The sensor measurements, $y$, that describe the state of the system variables are modeled as a function of the system state and measurement noise. A Kalman filter estimates the state based on the measurements and generates the residuals ($r$) as the difference between predicted and actual measurements. We assume that a linear controller generates a control action, $u$, based on the estimated state obtained by the Kalman filter.

Some of the assumptions used in our modeling framework may not necessarily capture the complexities of many real-world systems. However, our intent is to use a framework that facilitates the development and derivation of system characteristics in a formal and tractable manner with the goal of providing insights that will enable the development of more practical system models in the future. In this chapter, we have the following explicit assumptions regarding the system model and the attack model:

1. We assume a linear time-invariant (LTI) system. Specifically, we assume the state-transition function and the measurement function are both linear. This assumption is adopted widely in literature including [13, 22, 18, 15, 14, 37]. Although this assumption might not be practical in some cases, studies based on LTI systems lays the foundation
for future studies on more complex and realistic settings.

2. We assume there are two types of state variables: namely controllable and uncontrollable variables [47]. The controllable variables determine appropriate control actions for the system. They may include speeds, temperatures, pressures, loads, etc. The measurements obtained from uncontrollable variables are used for monitoring the performance (or health) of the system (and its assets) but not to decide the control actions. Examples of such variables include vibration signals, temperatures, and other measurements that are not part of the control system. We denote the set of controllable and uncontrollable variables as $C$ and $U$, respectively.

3. We assume that the state-transition function under degradation (or partial degradation) is different from that under normal operating conditions. In particular, we use a Brownian motion with linear drift to model the degradation dynamics, which is commonly used and studied in [48, 49, 50, 51, 52, 53]. We assume the degradation rate is a linear function of the system state $x_k$. This assumption is practical because intuitively, a higher load would accelerate the degradation process. The correlation between the operating condition and the degradation process is studied in [49, 50].

4. We assume the attacker has full knowledge of the system operations before degradation. As we mentioned in Section 1, existing methods are built on the assumptions that attacker only has partial knowledge of the system [18, 25, 24, 26]. In terms of detection, the assumption of the attacker’s full knowledge is more progressive – with the attacker’s full knowledge, the covert attack is designed to be undetectable before degradation.

5. We assume the attacker does not know the system dynamics under degradation. This is a valid assumption because the degradation models are learned from massive historical failures. Thus, it is difficult for an attacker to observe enough failures and to learn the degradation model.
A time-invariant linear state-space model is used to characterize system dynamics. The state-space model consists of a state-transition function and a measurement function given by equations 2.1 and 2.2. The matrices $A$, $B$, and $C$ are defined according to the following structure:

$$
A = \begin{bmatrix}
    A_c & A_{c,u} \\
    A_{u,c} & A_u
\end{bmatrix},
$$

(3.1)

where $C$ and $U$ represent the sets of controllable and uncontrollable variables. Recall that we assume the state transition of the controllable variables does not depend on the state of uncontrollable variables. Therefore, by definitions of controllable and uncontrollable variables, it is true that $A_{c,u} = B_{c,u} = 0$. As there is no control action for the uncontrollable variables, $B_{u,c} = B_u = 0$. Hence, we have,

$$
x_{c,k+1} = A_c x_{c,k} + B_c u_{c,k} + v_{c,k},
$$

(3.2)

$$
x_{u,k+1} = A_{u,c} x_{c,k} + A_u x_{u,k} + v_{u,k}.
$$

(3.3)

The state-space model is coupled with a control strategy comprised of a Kalman filter and a controller. The Kalman filter and the linear-quadratic Gaussian (LQG) controller considered herein are the optimal state estimator and controller when the state-space model precisely represents the system[40, 41]. The Kalman filter is used to estimate the current
state $\hat{x}_{k|k}$ based on the measurements $y_1, \ldots, y_k$, predict the next state $\hat{x}_{k+1|k}$, and calculate the residuals $r_k$. The linear controller calculates the control action $u_k$.

The steady-state KF and the LQG controller are also defined in sections 2.2.2 and 2.2.3.

In summary, we define a generic LTI system $S$ by the following conditions: (A1) The system dynamics can be characterized as a state-space model in Eq.(4.19-4.20), with the error terms identically and independently following normal distributions with zero-mean; (A2) The system consists of a Kalman filter represented by Eq.(2.4-2.6); (A3) The system consists of a linear controller represented by Eq.(2.7).

### 3.2.2 Degradation Model

The measurements of the uncontrollable variables are used to monitor system health and/or performance degradation. In this work, we assume that the vector of uncontrollable variables, $x_u = [x_i]; i \in U$, are used to represent system degradation. Furthermore, we assume that an uncontrollable variable $x_i$ where $i \in U$ is modeled as a Wiener process with a constant drift $\theta$, which is widely used to model degradation processes [51, 54, 55, 56, 57]. That is,

$$x_{i,k+1} = x_{i,k} + \theta_i + e_{i,k},$$  \hspace{1cm} (3.4)

where the noise term $e_k \sim N(0, \sigma^2)$ for $k = 0, 1, 2, \ldots$, and the drift $\theta_i$ is the degradation rate of variable $x_i$. In reality, the degradation rate depends on the operating condition, which means the degradation rate could change over time. Generally, degradation rates tend to increase as the operating/environmental conditions become more severe (e.g. higher loads, higher temperatures, faster speeds, etc.). Thus, we assume that $\theta_{i,k}$ depends on the operating condition of the system, which is governed by the controllable state variables $x_c$.

We assume that the degradation rate $\theta_{i,k}$ of the variable $x_i, i \in U$, at time $k$ is a linear function of $x_{c,k}$, i.e., $\theta_{i,k} = \kappa_i + \beta_i^T x_{c,k}$. Hence, we have:

$$x_{i,k+1} = x_{i,k} + \kappa_i + \beta_i^T x_{c,k} + e_{i,k},$$  \hspace{1cm} (3.5)
where $\kappa_i$ denotes the constant drift in variable $x_i; i \in U$, $\beta_i$ is the linear coefficient of $x_{c,k}$, and $e_{i,k}$ is the error term. The linear coefficient $\beta_i$ represents the correlation between the controllable variables $x_c$ and the rate of change in the uncontrollable variable given by the first order difference, i.e., $x_{i,k+1} - x_{i,k}$. The magnitude of $\beta_i$ represents the magnitude by which the degradation rate increases for a unit increase in $x_c$. Eq.(3.5) can be rewritten in terms of the set of uncontrollable variables $U$ as follows:

$$x_{u,k+1} = Gx_{c,k} + Ix_{u,k} + \kappa_u + \epsilon_k,$$

(3.6)

where $I$ is $|U|$-dimensional identity matrix, $G^T = [\beta_1, \beta_2, ...], \kappa_u^T = [\kappa_1, \kappa_2, ...], \epsilon_k^T = [e_{1,k}, e_{2,k}, ...]$. That is, under degradation, the coefficients of $x_{c,k}$ and $x_{u,k}$ are $G$ and $I$, respectively. Compared with Eq.(3.3), before degradation, the coefficients of $x_{c,k}$ and $x_{u,k}$ are $A_{u,c}$ and $A_u$, respectively. Therefore, by emphasizing this change in the coefficients, Eq.(3.6) can be rewritten as:

$$x_{u,k+1} = [A_{u,c} + (G - A_{u,c})]x_{c,k} + [A_u + (I - A_u)]x_{u,k} + v_{u,k}.$$

(3.7)

For simplicity and without loss of generality, we assume that the error term $\epsilon_k$ is generated from the same distribution as $v_{u,k}$. Since $\{v_k\}$ are independent and identically distributed, we can still write the error term as $v_{u,k}$.

Usually, the aging of equipment changes the dynamic behavior of the system. Thus, the degradation process may also affect the dynamics of the controllable variables. In this chapter, we assume that the state-transition coefficient during degradation, denoted by $A'_c$, is different from the coefficient $A_c$ before degradation in Eq.(3.2), and this difference is defined as $J := A'_c - A_c$. Consequently, the state transition of the controllable variables during degradation can be written as:

$$x_{c,k+1} = A'_c x_{c,k} + B_c u_{c,k} + v_{c,k} = (A_c + J)x_{c,k} + B_c u_{c,k} + v_{c,k}.$$

(3.8)
During degradation, the state transition functions of the controllable and uncontrollable variables are represented by Eq.(3.8) and Eq.(3.7), respectively. Thus, the state-transition function of the complete state $x = [x_c^T, x_u^T]^T$ during degradation has the following form:

$$x_{k+1} = (A + D)x_k + Bu_k + \kappa + v_k,$$

(3.9)

where $D = \begin{bmatrix} J & 0 \\ G - A_{u,c} & I - A_u \end{bmatrix}$, and $\kappa = [0, \kappa_u]$.

The state-transition functions of the system before and after the onset of degradation are given by Eq.(4.19) and Eq.(3.9), respectively. The difference between the two functions represents the change in system dynamics introduced by degradation, which is characterized by the degradation parameters $D$ and $\kappa$. Therefore, we generalize the two functions by adding a degradation term $s_k$ to the state-transition function to account for this change in system dynamics. $s_k$ before degradation is 0, and after the onset of degradation, $s_k = \kappa + Dx_k$. Hence, for any $k$, we have:

$$x_{k+1} = Ax_k + Bu_k + v_k + s_k,$$

(3.10)

where $s_k := \begin{cases} 0, & k < \tau \\ \kappa + Dx_k, & k \geq \tau \end{cases}$, and $\tau$ is the onset of the degradation process. The above equation is taken as the model of system $S$ with degradation.

In summary, we define a system $S_D$ subject to degradation by the following conditions: (B1) The system dynamics before degradation can be characterized as a generic system $S$ defined in Section 3.1; (B2) The system consists of controllable and uncontrollable variables, such that the submatrices $A_{c,u}$ and $B_{c,u}$ are 0; (B3) The state-transition function during degradation can be characterized by Eq.(3.10), with $\tau$ being the onset of degradation, and $\kappa_c$ and $D_{c,u}$ both equal to 0.
3.2.3 Covert Attack

The formulation of our covert attack is an extension of that in [19]. In [19], the author considered a linear steady-state system where the measurement $y$ is a linear function of the control action $u$ and disturbance $w$ with noise $n$:

$$ y = \Pi_u u + \Pi_w w + n. $$

The covert attacker first injects a bias $\delta$ to the control action:

$$ u' = u + \delta. $$

Then, the attacker manipulates the measurement $y$ by injecting a compensation $\gamma$:

$$ y' = y' + \gamma $$

In order for the attacker to stay undetected, the manipulated measurement should be equivalent to the original measurement ($y' = y$). Hence, $\gamma$ is derived as the difference between the measurements corresponding to the original and the manipulated control actions $u$ and $u'$, which is a linear function of $\delta$:

$$ \gamma = \Pi u - \Pi u' = -\Pi \delta. $$

However, this formulation only applies to the steady-state and does not account for system dynamics. In this chapter, we expand the above formulation to the dynamic system setting described in section 3.1.

To see this, consider a covert attack that takes place at time $\tau_a$. Next, assume that at time $\tau_{a+k}$, the covert agent performs the following actions:

- Step 1: read the system control action $u_k$ and manipulate $u_k$ by injecting a bias term
\( \delta_k \) as follows:

\[
\begin{align*}
    u'_{\tau_a+k} &= u_{\tau_a+k} + \delta_k;
\end{align*}
\]  

where \( \delta_{k,u} = 0 \) by definition of the controllable and uncontrollable variables.

- Step 2: read the measurement \( y'_{\tau_a+k+1} \) and manipulate \( y'_{\tau_a+k+1} \) by injecting the compensation term \( \gamma_{k+1} \) as follows:

\[
\begin{align*}
    y^a_{\tau_a+k+1} &:= y'_{\tau_a+k+1} - \gamma_{k+1},
\end{align*}
\]  

where \( \gamma_{k+1} \) is a function of \( \delta_0, \ldots, \delta_k \), such that

\[
\begin{align*}
    y^a_{\tau_a+k+1} &= y_{\tau_a+k+1}.
\end{align*}
\]  

Eq.(3.13) implies the covertness that the manipulated sensor measurement \( y^a_{\tau_a+k+1} \) perfectly recovers the normal sensor measurement without attack \( y_{\tau_a+k+1} \).

Now, we solve \( \gamma_{k+1} \) subject to the constraint Eq.(3.13). As mentioned earlier, we assume that the attacker has acquired all the system parameters \( A, B, \) and \( C \). This leads us to the following proposition:

**Proposition 1** For a system \( S \) defined in Section 3.1, a covert attack defined by \( \{\delta_k, \gamma_{k+1}\} \) has the following formulation:

\[
\begin{align*}
    \gamma_{k+1} = \sum_{i=0}^{k} CA^{k-i} B \delta_i.
\end{align*}
\]  

**Proof** According to Lemma 1 in [47], if at time \( \tau_a \), a bias term \( \delta_0 \) is added to the original control action \( u_{\tau_a} \), the deviation in the measurement \( y_{\tau_a+k+1} \) at time \( \tau_{a+k} \) induced by \( \delta_0 \) is equal to \( CA^{k-1} B \delta_0 \) for all \( k = 1, 2, 3, \ldots \). Therefore, the deviation in \( y_{\tau_a+k+1} \) induced by
the attack is the total deviation induced by $\delta_0, \ldots, \delta_k$. That is,

$$y_{\tau_a+k+1}^{a} + \gamma_{k+1} = y_{\tau_a+k+1}^{'} = y_{\tau_a+k+1} + \sum_{i=0}^{k} CA^{k-i} B \delta_i = y_{\tau_a+k+1}^a + \sum_{i=0}^{k} CA^{k-i} B \delta_i$$

Hence,

$$\gamma_{k+1} = \sum_{i=0}^{k} CA^{k-i} B \delta_i$$

**Corollary 3.2.1** A covert attack given by Proposition 1 on a system $S$ defined in Section 3.1 is undetectable by monitoring $\{r_k, y_k, u_k\}$ before degradation.

**Proof** Before degradation, the attacker’s knowledge of the system $(A, B, C)$ is perfect. Hence, under the formulation of covert attack given by Eq.(3.14), we have $y_k^a = y_k$ for all $k$. Since $r_k$ and $u_k$ are functions of $\{y_1, \ldots, y_k\}$, we have the residuals under attack $r_k^a = r_k$ and the control actions under attack $u_k^a = u_k$ for all $k$. This means the monitored data under covert attack $\{r_k^a, y_k^a, u_k^a\}$ is equivalent to the data without covert attack $\{r_k, y_k, u_k\}$. Therefore, the covert attack is undetectable by monitoring $\{r_k, u_k, y_k\}$ before degradation.

Corollary 3.2.1 states that the attack is undetectable before degradation. However, a covert attack can be detected only after the onset of degradation, $\tau$. In this chapter, we consider two scenarios: In case 1, the onset of covert attack $\tau_a$ occurs before the onset of degradation $\tau$ ($\tau_a < \tau$). In this case, the attack is detected after the degradation onset $\tau$. In Case 2, the covert attack occurs after $\tau$ ($\tau_a \geq \tau$). In this case, the attack is detected after the attack onset $\tau_a$. This is illustrated in Fig.3.2. Our proposed algorithm detects the attack by identifying the accelerated degradation rate (dotted and dashed lines) after the attack, and we assume the natural degradation rate (solid line) is known *a priori*. Recall in Section 3.2, we modeled the degradation term $s_k$ as a function of $x_k$. By manipulating the control action, the covert attack alters the system state to $x_k^a = x_k + B \delta_k$. When $k > \tau$ (after the
onset of degradation), this will lead to the abnormal degradation rate $s_k^c = s_k + DB\delta_k$, where $DB\delta_k$ is the acceleration caused by the covert attack.

$$DB\delta_k$$

Figure 3.2: The degradation signal under no attack, in Case 1, and in Case 2

3.2.4 Parameter Estimation

The parameters in the state-space model, i.e., matrices $A$, $B$ and $C$, can be estimated from historical sensor data using the methods in [58, 59, 60, 45]. The degradation parameters, including matrix $D$ and $\kappa$, can be estimated as follows: The degradation rate $G$ and $\kappa$ can be estimated by first estimating the degradation model for all the uncontrollable variables shown in Eq.(3.4) separately, and then aggregated according to the definitions of $G$ and $\kappa$ as defined in Eq.(3.6). Then, since $G$ and $\kappa$ are given, the submatrix of $D$, $J$, can be estimated by then fixing the remaining part of $A$ matrix and re-estimating the new state-space model using data in degradation phase. $J$ can be calculated as the difference between the new transition matrix $A'$ under degradation and the old one $A$ before degradation.
3.3 Detection Algorithm for Covert Cyberattacks

The basis of our detection methodology is to leverage abnormalities in the natural degradation rate of an asset. That is, covert attacks that occur before degradation will remain undetected until the degradation process begins. Thus, the first step of our methodology is to determine the onset of degradation, and then use our algorithm to detect abnormal degradation rates. This work assumes the natural degradation rate is known. Specifically, the degradation model expressed in Eq.(3.10) has been estimated a priori.

As shown in Fig.3.3, our detection algorithm is comprised of two likelihood ratio tests, one for detecting degradation, and the other for detecting covert attacks. The inputs to the detection algorithm are the state estimation $\hat{x}$ and residuals $r$ (Eq. 2.5,2.6). Recall that when there is no attack or degradation, the distribution of $r$, $F_0$, is known. To detect degradation, we use a likelihood ratio test LRT-I, which tests whether $r$ follows the pre-degradation distribution $F_0$ or the post-degradation distribution $F_1$. We also use LRT-I to specify the onset of degradation $\hat{\tau}$ after which $r$ follows $F_1$ if the system is not under attack. Once $\hat{\tau}$ is determined, we use a second likelihood ratio LRT-II to test whether $r$ follows the natural post-degradation distribution $F_1$ or some different distribution, $F_2$, generated by a cyber attack. The output of our algorithm is an alarm raised by the LRT-II and estimation of the onset of the covert cyberattack $\hat{\tau}_a$. In the following section, we derive $F_1$, $F_2$, and derive the likelihood ratio testing algorithms LRT-I and LRT-II. There are two reasons we use likelihood ratio tests to detect covert attack in our proposed method: 1) The likelihood ratio tests have built-in change point detection. The main purpose of detection of degradation is to specify the onset of degradation and attack. This function is not achieved by classical control charts; and 2) The likelihood ratio test has better detection power. As we will show in the rest of this section, we can explicitly derive or estimate the post-change distributions $F_1$ and $F_2$, and the two likelihood ratio tests LRT-I and LRT-II only test whether the residuals $\{r_k, r_{k+1}, \ldots, r_{k+w}\}$ follow these specific distributions. Therefore,
they have more statistical power in detection of such changes compared to general control charts such as Shewhart or EWMA.

3.3.1 Detection of Degradation

Degradation is detected by difference in the distributions of $r_k, F_0$ and $F_1$, before and after the onset of degradation $\tau$. Before degradation (i.e., when $k < \tau$), the residuals $r_k$ follow a Gaussian distribution $F_0 = \mathcal{N}(0, R + CPC^T)$, where $P$ is the steady-state covariance of the prediction error given by the Kalman filter (Eq.(2.3)). After $\tau$, the state transition function of $x_k$ changes due to an added degradation term $s_k$. However, the Kalman filter still estimates the state $x_k$ and generates the residual $r_k$ using the model estimated prior to degradation. This leads to a different distribution of $r_k$. i.e., for $k > \tau$, $r_k \sim F_1, F_1 \neq F_0$.

This is the basis of Theorem 3.3.1, which derives the distribution $F_1$.

**Theorem 3.3.1** For a system $S_D$ defined in Section 3.3.2 with $\tau$ as the onset of degradation, the residuals $r_{\tau+i}, i = 1, 2, 3, ...$ follow a Gaussian distribution, $F_1 = \mathcal{N}(\mu_{\tau+i}, \Sigma_{\tau+i})$, where,

$$\mu_{\tau+i} = C(I - M)^{-1}(I - M^i)\kappa + C \sum_{j=1}^{i} M^{i-j}z_j,$$

$$\Sigma_{\tau+i} = R + CPC^T + C\Phi_iC^T + \Psi_iC^T + C\Psi^T_i,$$

Figure 3.3: The detection framework
with \( \Phi_i \) and \( \Psi_i \) calculated recursively using the following equations:

\[
\Phi_{i+1} = M \Phi_i M^T + D \tilde{P} D^T + D \sum_{k=1}^{i} (A^k \tilde{P} (M^{k-1})^T + M^{k-1} \tilde{P} (A^k)^T) D^T,
\]

\[
\Psi_i = C \sum_{j=0}^{i-1} A^j \tilde{P} (M^j)^T D^T,
\]

where \( M = A(I - KC) \), \( \hat{z}_j = D\hat{x}_{\tau+j|\tau+j} \), and \( \tilde{P} \) is the steady-state updated estimate covariance given by the Kalman filter, i.e. \( \tilde{P} = (I - KC) P = \text{cov}(x_k - \hat{x}_{k|k}) \).

**Proof** See Appendix B.

Theorem 3.3.1 describes the distribution of the residuals \( r \) during degradation. Since the distributions of \( r \) before and during degradation are both known, the onset of degradation can be detected by testing the likelihood ratio \( g = \frac{\text{likelihood}(r \sim \mathcal{F}_1)}{\text{likelihood}(r \sim \mathcal{F}_0)} \). A high likelihood ratio, \( g \), means it is more likely that \( r \sim \mathcal{F}_1 \) and the degradation has already started. To determine the onset of degradation, \( \tau \), in real-time, we implement the likelihood ratio test based on a moving window with length, \( \omega \). At each time step \( k \), we consider the data within the window \( \{r_{k-\omega+1}, ..., r_k\} \). We iterate over all the possible values of \( \hat{\tau} = i \) within the window \( (i = k - \omega + 1, ..., k) \) and calculate \( g_i \) as follows:

\[
g_i = \frac{\text{likelihood}(r_i, ..., r_k \sim \mathcal{F}_1)}{\text{likelihood}(r_i, ..., r_k \sim \mathcal{F}_0)}.
\]

We monitor \( g_k = \max_i(g_i) \) and \( \hat{\tau}_k = \arg\max_i(g_i) \). An alarm is triggered whenever \( g_k \) crosses a pre-specified threshold \( \lambda_1 \), and the corresponding \( \hat{\tau}_k \) is specified as the estimated onset of degradation.

The threshold \( \lambda_1 \) is determined based on the empirical distribution of \( g_k \) and the specified type-I error rate \( \alpha \), where \( \lambda_1 \) is the upper \( \alpha \)-quantile of the empirical distribution \( F_g \), i.e., \( P(g_k > \lambda_1) = \alpha \) when \( g_k \) is drawn from the process before degradation \((k < \tau)\). The above process is formalized in Algorithm 1. Algorithm 1 is used to detect degradation and estimate the degradation onset, \( \hat{\tau} \), which is taken as an input of the likelihood ratio test to
Algorithm 1 Likelihood ratio test for degradation (LRT-I)

Input: \{r_k\}, \{\hat{z}_k\}, \lambda_1
1: \text{alarm} = 0;
2: \textbf{for} t = 1 \textbf{to} T \textbf{do}
3: \quad \textbf{for} i = 1 \textbf{to} \omega - 1 \textbf{do}
4: \quad \quad l_0 = \sum_{j=i}^{\omega} f(r_{t-\omega+j}; 0, R + CPC^T), \text{where } f(x; \mu, \Sigma) \text{is the log likelihood function of the multivariate Gaussian distribution with mean } \mu \text{ and covariance } \Sigma \text{ for } x;
5: \quad \quad l_1 = \sum_{j=i+1}^{\omega} f(r_{t-\omega+j}; r_{t-\omega+i} + \mu_{t+j-i}, R + CPC^T + \Sigma_{t+j-i}), \text{where } \mu_{t+j-i} \text{ and } \Sigma_{t+j-i} \text{are calculated according to Eq.(3.15) and Eq.(3.16), respectively;}
6: \quad \quad g_i = l_1 - l_0;
7: \quad \textbf{end for}
8: \quad g_t = \max_i(g_i);
9: \quad \hat{\tau}_t = t - \omega + \text{argmin}_i(g_i);
10: \quad \textbf{if} (g_t > \lambda_1) \textbf{then}
11: \quad \quad \text{alarm} = 1, \hat{\tau} = \hat{\tau}_t;
12: \quad \quad \textbf{break;}
13: \quad \textbf{end if}
14: \textbf{end for}
15: \textbf{Return } \hat{\tau}

3.3.2 Detection of Covert Attack

As shown in Corollary 3.2.1, covert attacks are undetectable prior to the degradation phase. Under covert attack, the system state changes to \(x_k^a \neq x_k\). During degradation, an attacker capable of manipulating system measurements and compensating for the new system state, \(x_k^a\), cannot compensate for the unforeseen degradation characteristics denoted by \(s_k^a \neq s_k\). Recall the covert attack model defined by Eq.(3.11,3.12), the attack is represented by the control bias \(\delta_k\) and the measurement compensation \(\gamma_k\), where \(\gamma_k\) is a function of \(\{\delta_0, \ldots, \delta_k\}\). Assuming that a covert attack induces a constant bias denoted by \(\delta_k = \delta\) for all \(k > \tau_a\), the residuals under attack during the degradation phase, \(r_k^a\), will follow a distribution \(F_2 \neq F_1\).

Theorem 3.3.2 Given the control bias \(\delta\), for a system \(S_D\), defined in Section 3.3.2, under
covert attack, defined in Proposition 1, at time $\tau_a$, the residuals $r_{\tau_a+i}$, $i = 1, 2, 3, \ldots$ follow a Gaussian distribution $F_2 = \mathcal{N}(\mu_i^a, \Sigma_i^a)$, where $\Sigma_i^a = \Sigma_i$, and $(\mu_i^a - \mu_i) = i\beta$, where $\beta$ is a constant, and $\beta \propto \delta$.

**Proof** See Appendix B.

Since the value of $\delta$ is unknown, the covert attack detection problem is formulated as a change point detection of a normal process mean with an unknown linear trend $\beta$. Therefore, we adopt the method proposed in [61] to detect the attack: We first estimate the slope of the linear trend $\hat{\beta}$ using Eq. (3.17) and then use a likelihood ratio test to find $\hat{\tau}_a = i$ within the window $i = k - \omega + 1, \ldots, k - 1$.

$$\hat{\beta} = \frac{6(\sum_{j=i+1}^{\omega}(j-i)\zeta_j - 0.5(\omega - i)(\omega - i + 1)\mu_0)}{(\omega - i)(\omega - i + 1)(2\omega - 2i + 1)}$$

(3.17)

where $\mu_0 = r_{t-\omega+i}$, $\zeta_j = r_{k-\omega+j} - \mu_{\tau+j}$, and $\Sigma_i$ and $\mu_{\tau+j}$ are given by Theorem 3.3.1. The derivation of $\hat{\beta}$ is given in Appendix B.

Next, the following likelihood ratio is used to test for a change in slope,

$$g_i^a = \frac{\text{likelihood}(r_i, \ldots, r_k \sim F_2)}{\text{likelihood}(r_i, \ldots, r_k \sim F_1)}$$

for all $i = k - \omega + 1, \ldots, k - 1$.

We monitor $g_k^a = \max_i(g_i^a)$ and if an attack is detected, the change point is determined by $\hat{\tau}_a^a = \arg\max_i(g_i^a)$. An alarm is triggered when $g_k^a$ exceeds the threshold $\lambda_2$. As in Algorithm 1, $\lambda_2$ is based on the empirical distribution of $g_k^a$ and the specified type-I error rate $\alpha'$. The above process is formalized in Algorithm 2.

Note that Theorem 3.3.2 holds when $\delta_k = \delta$ for all $k > \tau_a$. In the case where $\delta_k$ is a random variable, we only have $\mu_i^a \neq \mu_i$ one can estimate the mean $\mu_i^a$ using maximum likelihood estimation, and a generalized likelihood ratio test can be used.

Note that Algorithm 2 detects the covert attack and also specifies the estimated onset of the attack $\hat{\tau}_a$. This information can be useful in further investigations of the attack. For
Algorithm 2 Likelihood ratio test for covert attack (LRT-II)

Input: \( \hat{\tau}, \{r_\tau, r_{\tau+1}, \ldots\}, \{\hat{z}_\tau, \hat{z}_{\tau+1}, \ldots\}, \lambda_2 \)

1: \( \text{alarm} = 0 \);
2: \( \text{for } t = \hat{\tau} \text{ to } T \text{ do} \)
3: \( \zeta_t = r_t - \mu_{\tau+t}, \text{ where } \mu_{\tau+t} \text{ is given by Theorem 3.3.1;} \)
4: \( \text{for } i = 1 \text{ to } \omega - 1 \text{ do} \)
5: \( \text{Calculate } \hat{\beta} \text{ using Eq.(3.17), where } \mu_0 = r_t - \omega + i; \)
6: \( g_i^a = -0.5(\omega - i)(\omega - i + 1)\hat{\beta}^T \Sigma_1^{-1}[\mu_0 + \frac{1}{6}\hat{\beta}(2\omega - 2i + 1)] - \hat{\beta}^T \Sigma_1^{-1} \sum_{j=i+1}^{\omega} (j - i)\zeta_{t-\omega+j}, \text{ where } \Sigma_1 = 2(R + CPC^T) + \lim_{i \rightarrow \infty} \Sigma_{\tau+i}, \text{ and } \Sigma_{\tau+i} \text{ is calculated using Eq.(3.16);} \)
7: \( \text{end for} \)
8: \( g_i^a = \max_i(g_i^a); \)
9: \( \hat{\tau}_i = t - \omega + \arg\min_i(g_i^a); \)
10: \( \text{if } (g_i^a > \lambda_2) \text{ then} \)
11: \( \text{alarm} = 1, \hat{\tau}_a = \hat{\tau}_i; \)
12: \( \text{break;} \)
13: \( \text{end if} \)
14: \( \text{end for} \)
15: \( \text{Return } \hat{\tau}_a \)

example, when \( \hat{\tau}_a \) and \( \hat{\tau} \) are close to each other, it can be inferred that the covert attack occurred prior to the degradation phase.

3.4 Numerical and Case Study

We conduct a simulation study aimed at understanding the effects of the degradation parameters \( \kappa \) and \( D \) as well as the magnitude of the attack \( m_a \) on the detection delay of our detection algorithm. We also evaluate the performance of our methodology through an experimental implementation on a rotating machinery testbed to demonstrate the applicability of our method in practice.

3.4.1 Numerical Study

We evaluate the performance of the detection framework using two statistical metrics: 1) the average run length (ARL), which refers to the expected detection delay; and 2) the
average change-point error (CPE), which refers to the difference between the estimated and true change points, i.e. $|\hat{\tau} - \tau|$ or $|\hat{\tau}_a - \tau_a|$. The performance of Algorithm 1 and 2 are tested both individually and jointly (i.e. the combination of the two algorithms). When Algorithm 2 is tested individually, we use the true degradation onset $\tau$ as its input. When Algorithms 1 and 2 are tested jointly, we use the estimated degradation onset $\hat{\tau}$ from Algorithm 1 as the input to Algorithm 2. The control limits $\lambda_1$ and $\lambda_2$ are selected based on empirical distribution such that the in-control ARL is 500 for both algorithms.

We consider a system with three state variables. The values of parameters used in our simulations are given in Appendix B. We consider two scenarios: In scenario 1, the system variables are comprised of two controllable variables and one uncontrollable variable. In scenario 2, all three variables are controllable (i.e., $D = J$). For each scenario, we consider two cases: In Case 1, the attack occurs before the onset of degradation, namely, $\tau_a = \tau - 200$. In this case, the attack is undetectable until $\tau$ when the degradation starts. Therefore, we expect an accumulated detection delay of the joint algorithm because the input $\hat{\tau}$ to Algorithm 2 is different from the true $\tau$. In Case 2, the attack occurs in the degradation phase, namely, $\tau_a = \tau + 200$. In this case, we expect a shorter detection delay than Case 1.

We investigate the sensitivity of our detection algorithm to the degradation rate and the attack severity. The degradation rate is represented by $\kappa$ and $D$, where $\kappa$ represents the constant drift of the Brownian motion (which is used to model degradation), and $D$ represents the association between the system state $x_k$ and the degradation term $s_k$. Increases in the magnitudes of $\kappa$ and $D$ both result in an increase in the degradation rate. We define the magnitude of $\kappa$ and $D$ as follows:

$$m_1 = \kappa^T \Sigma_r^{-1} \kappa,$$

$$m_2 = \|D\|_F,$$

where $\Sigma_r = R + CPC^T$ is the covariance matrix of the residuals $r_k$, and $\|\cdot\|_F$ is the
Frobenius norm. In our simulations, we consider two settings: $m_1$ is fixed at 0, while $m_2$ is varied ($m_2 = 0, 0.25, 0.5, 0.75, 1$), and $m_2$ is fixed at 0.25, while $m_1$ is varied ($m_1 = 0, 0.5, 1, 1.5, 2$).

The magnitude of the attack, denoted as $m_a$, measures the non-centrality of the constant bias $\delta$ and is defined as:

$$m_a = \delta^T \Sigma^{-1} \delta.$$  

In the simulations, six levels of $m_a$, namely, $m_a = 0, 0.2, 0.4, 0.6, 0.8, 1$, are considered to study the sensitivity of our algorithm to $m_a$. We repeat the simulations 200 times and record both the ARL and the average CPE over the simulation replications.

![Simulation results](image)

Figure 3.4: Simulation results: Scenario 1 with $\kappa = 0$; ARL & CPE vs $m_2$; $m_a$ varying from 0 to 1

The simulation results are shown in Figures 3.4 and 3.5. In these figures, the ARL and average CPE values are plotted against $m_2$ for Algorithm 1, Algorithm 2, and the joint algorithm under various magnitudes of attack, $m_a$ for the different cases and scenarios presented earlier. For example, in Fig.3.4-(b), when there is no attack, i.e., $m_a = 0$, the ARL slightly changes (from around 500 to 450) as $m_2$ increases from 0 to 0.5. However, when the system is under an attack with the magnitude of $m_a = 0.4$, the ARL decreases...
Figure 3.5: Simulation results: Scenario 2 with $\kappa = 0$; ARL & CPE vs $m_2$; $m_a$ varying from 0 to 1

from approximately 500 to slightly less than 100 as $m_2$ increases from 0 to 0.5. In Appendix B (Fig. B.1-B.2), we also show the simulation results when $m_2$ is fixed and $m_1$ varies. Based on the simulation results, we make the following remarks:

1. Fig.3.4(a) and Fig.3.4(f) indicate that in scenario 1, both the ARL and CPE values for Algorithm 1 decrease as $m_2$ increases. This is because as the degradation rate increases, the difference between the mean values of $F_1$ and $F_0$ increases (as proved...
by Theorem 3.3.1). Consequently, the likelihood of \( r_k \sim F_0, k > \tau \), decreases. Thus, our algorithm detects degradation faster and more accurately. The ARL and CPE values also decrease as \( m_a \) increases because an increase in the severity of the attack results in a higher degradation rate. This is evident from Theorem 3.3.2, where the linear drift \( \beta \) is shown to be a function of \( D \) and \( \delta \) (See Appendix B for more details).

2. As shown in Fig.3.4, in scenario 1, the ARL and CPE for Algorithm 2 in both Cases 1 and 2 decrease as \( m_a \) increases. This is proved by Theorem 3.3.2, where the difference in the mean values between \( F_1 \) and \( F_2 \) is shown to be proportional to the value of \( \delta \). When \( m_a \neq 0 \), the ARL and CPE also decrease as \( m_2 \) increases, this is true because the linear drift \( \beta \) is a function of \( D \). Consequently, the difference between \( \mu_i \) and \( \mu_i^a, i > \tau \) increases as the magnitude of \( D \) increases.

3. As shown in Fig.3.4, in scenario 1, the ARL and CPE for the joint algorithm in both cases 1 and 2 decrease as \( m_a \) increases. While the ARL values are similar, the CPE values are generally lower in Case 2 compared to Case 1. Recall that in Case 1, the attack occurs before \( \tau \), and hence the distribution of \( r_k \) for \( k > \tau \) is different from \( F_1 \). Although this does not have a significant impact on the detection delay, it affects the accuracy of the estimated degradation onset, \( \hat{\tau} \), which in turn affects the performance of Algorithm 2. Therefore, in Case 1, the CPE for the joint algorithm is cumulative of the estimation errors for the two change points. On the other hand, as long as \( \hat{\tau} < \tau^a \), the performance of Algorithm 2 is not affected in Case 2, and the CPE values for the joint algorithm are not accumulated.

4. As can be seen in Fig.3.5, the ARL and CPE patterns described for Scenario 1 are similar to those in Scenario 2. However, the ARL and CPE values for Algorithm 2 are both larger than those reported for Scenario 1. This is because the linearity of the degradation trend in Theorem 3.3.2 does not hold in Scenario 2. Therefore, Algorithm 2, which is designed to detect a linear drift, is not suitable in this scenario.
However, when \( m_2 \) and \( m_a \) are large, our algorithm can still successfully detect the attack with low ARL and CPE values. This is true because we are using maximum likelihood estimator \( \hat{\beta} \) based on the data. Even when the data does not follow a linear trend, the likelihood of \( r_k \sim \mathcal{F}_2 \) is high for severe attacks. Additionally, when \( m_a \) is large, \( r_k \sim \mathcal{F}_1 \) decreases faster. Therefore the likelihood ratio increases as \( m_a \) increases, resulting in a better performance of Algorithm 2.

5. From Fig.B.1-B.2, we can see that similar to \( m_2 \) in both scenarios, the ARL and CPE values for Algorithm 1 and the joint algorithm decrease as \( m_1 \) increases. However, unlike \( m_2 \), as \( m_1 \) increases the ARL and CPE values for Algorithm 2 approximately remain constants. This is because \( \beta \) is not a function of \( \kappa \) as shown in the proof of Theorem 3.3.2. Hence, as \( m_1 \) varies, the difference between the mean values of \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \), \( \mu^a_i - \mu_i \), remains constant. Note that in Fig.B.1, the ARL value for Algorithm 1 when \( m_1 = 0 \) is 50. This is because we fix \( m_2 = 0.25 \) in this setting, which results in a positive degradation rate. Therefore, Algorithm 1 can detect degradation with relatively lower ARLs.

3.4.2 Case Study

Our detection algorithm is implemented on a rotating machinery setup shown in Fig.2.3. We study the degradation of a thrust bearing which is shaft-driven by a motor. The voltage of the motor can be adjusted to control the rotating speed (rpm) at the prespecified target level. Where the adjustment of motor voltage is the control action denoted as \( u_1 \), and the rotating speed is denoted as \( x_1 \). The loading condition is controlled by a hydraulic system. A tachometer, measuring the speed (denoted as \( y_1 \)), is used as part of a feedback control loop to control the speed. A vibration sensor is used to measure the vibration level, monitoring the degradation of the system. The actual vibration level is denoted as \( x_2 \), and the measurement of vibration level is denoted as \( y_2 \). Data acquisition is prosecuted on a DAQ (data acquisition) board. An LQG control strategy as well as the human-machine
interface (shown in Fig.2.4) is coded on the computer using Labview. Under this setting, the rotating speed (rpm) is assumed to be the controllable variable and the vibration (the root mean square of the vibration signal) is the uncontrollable variable. For simplicity, we assume the measurements are the accurate representations of the actual state under normal conditions, i.e. $y_k = x_k$. The corresponding state-space model is given below:

$$y_{k+1} = Ay_k + Bu_k + v_k + s_k$$ \hspace{1cm} (3.18)

$$s_k = \begin{cases} 0 & k < \tau \\ \kappa + Dy_k & k \geq \tau \end{cases}$$ \hspace{1cm} (3.19)

where $y_k = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k$, $u_k = \begin{bmatrix} u_1 \\ 0 \end{bmatrix}_k$, and $v_k \sim \mathcal{N}(0, Q)$.

We use historical data of the bearing to estimate the states-space model and degradation parameters, including matrices $A$, $B$, $D$, and $Q$, using the vars [62] package in R. To estimate these parameters, we monitor the following variables: $k$, $y_1$, $y_2$, $u_1$, and the voltage of the driving motor as a reference for $u_1$. We first estimate $A$, $B$, and $Q$ using non-degradation data, and then estimate $\kappa$ and $D$ using degradation data. The size of the non-degradation dataset we use in the first step is $5$ variables $\times 1000$ observations $\times 8$ replications, and the size of the degradation dataset is $5$ variables $\times 1400$ observations $\times 3$ replications. After estimating the model parameters, the LQG controller and the Kalman filter are designed following equations in Section 3.1. We also use the degradation data to estimate the control limit $\lambda_2$. The estimated parameters, the Kalman gain, and the $L$ matrix that defines the controller are given in Appendix B.

To justify the validity of the model assumptions, we show the KF residuals for speed and vibration as well as their quantile-quantile plot against normal distribution in Fig.3.6. It can be seen that the values of residuals are small compared to the actual scales of speed and vibration (shown in Fig.3.7-(a,b)), which means that the state-space model can effectively
represent the system dynamics. The results also show that the residuals approximately follow normal distributions centered around 0, and there is no significant autocorrelation in the residuals.

![Figure 3.6: System Model Justification](image)  

![Figure 3.7: Attack Model Justification](image)
To justify the validity of our design of the covert attack, we implement a covert attack before the degradation onset. Notice that we expect the designed covert attack to be undetectable in this scenario. The actual and observed (manipulated) measurements of speed and vibration are plotted over time in Fig.3.7-(a-b). We also show the estimated vibration state based on the state-space model. In these figures, the vertical lines indicate the onset of the covert attack, and the horizontal lines indicate the control limits. Recall that $\omega = 20$, so the time index of Fig.3.7-(d) is off by 20. The results show that the observed measurements and the monitoring statistic do not show any indication of the attack, and the detection method fails to detect the attack in this case, which means the attack model is effective. Besides, Fig.3.7-(c) shows that the mode-estimation (crosses) of the vibration value and the observed measurements (solid line) are very close, which means the attacker can successfully conceal the increase in the vibration level caused by the increase in the actual speed.

In detection phase, we monitor the control action ($u_1$), the actual speed and vibration ($y$), the manipulated measurements ($y^a$ length 2 vector), the KF state estimation ($\hat{x}_{t|t}$, length 2 vector), and the residuals for speed and vibration ($r$, length 2 vector). The size of the detection dataset is 9 variables $\times$ 183 observations. Notice that the actual speed and vibration are assumed unobserved in practice, and here monitored only for analysis but not used for detection. Since it is not easy to predict the onset of degradation, and the covert attack cannot be executed for too long for safety issues, we focus on detecting the covert attack during the degradation phase of the bearing. This scenario is represented by Case 2 in the previous discussions. We first calculate $z_{r+t}$ and $\mu_{t+\infty}$ using Theorem 3.3.1, and then implement Algorithm 2. As mentioned in Section 4.2, the input to Algorithm 2 includes $\{r_1, ..., r_{183}\}$, $\{\hat{z}_1, ..., \hat{z}_{183}\}$ and $\lambda_2$. Notice the calculation of Algorithm 2 requires Theorem 3.3.1 to calculate $\mu_t$. Since the degradation has already started, we assume $\mu_t = \mu_{r+\infty}$ for all $t$. The window size we use in the case study is $\omega = 20$. The first 35 and the last 3 observations of the data are removed to guarantee the stability of the processed
data. The detection results are shown in Fig. 3.8. As shown in Fig. 3.8-(b), although the change in the rotating speed is successfully concealed, the observed vibration is far from the expected vibration signal according to the degradation model. In contrast with Fig. 3.7-(b), where the attack is successful, the result in Fig. 3.8-(b) shows that the attacker cannot stay undetected in this case. On the other hand, before the onset of attack, the model-estimation of the vibration level during degradation is very close to the actual value, which justifies the validity of our degradation model. The difference between the residual for vibration and its estimation based on Theorem 3.3.1, which is calculated using $\zeta_t = r_t - \mu_{\tau+t}$, is shown in Fig. 3.8-(c). One can see a significant linear trend in $\zeta$ after the onset of the attack, indicating an increase in the degradation rate. It also proves the correctness of Theorem 3.3.2. The monitoring statistics in Fig. 3.8-(d) show that the proposed method successfully detects the covert attack. The results of this case study show that our proposed method can effectively detect covert attacks in real applications.

Figure 3.8: Experimental results, Case 2
3.5 Conclusion

In this chapter, we proposed a detection framework against covert attacks on SCADA systems represented by a state-space model, a Kalman filter, and an LQG controller. We built our method based on the premise that a covert attack accelerates the degradation process of an asset. To detect the attack, we characterized the difference in the system dynamics before and after the onset of degradation and studied how a covert attack affects such dynamics. Assuming that the natural degradation rate can be learned \textit{a priori} from historical data, which is unknown to the attacker, we derived the distributions of residuals before and during degradation as well as under the covert attacks and used them to develop two likelihood ratio tests to detect degradation and covert attack. In a numerical study, we showed the sensitivity of our method to the degradation rate and the magnitude of the attack. The results showed that our proposed methodology has an increasing detection power as the degradation rate and/or the magnitude of the attack increases. In a case study on a rotating machinery setup, we showed the validity and applicability of our methodology in real-world settings.
4.1 Introduction

In recent years, cyberattacks have raised increasing concern in the cybersecurity of power transmission and distribution systems. The cybersecurity vulnerability of power systems is increasing from two aspects: the accelerated implementation of wireless data communication that needs to be properly secured, and the increasing capability and motivation of the hackers to access sensitive information and even take down the control system while bypassing the traditional detection schemes. Accordingly, research in this area is focused on addressing the cybersecurity issue via protective and detective measures. The protective measures include data encryption and protocol design [63], as well as distributed and robust state estimation strategies proposed in [64, 65, 66, 67]. On the other hand, the detective methods are focused on detecting cyberattacks by monitoring either the network traffic data [68] or the sensor data [18, 37, 69]. While the implementation of the protective methods such as new network security protocols may require many of the industrial control systems to be redesigned, the detective methods can be much more flexible and are necessary for securing the system in practice.

Traditionally, detection of data integrity attacks has relied on bad-data-detection algorithms that involve performing statistical tests on the residuals of the state estimation [8, 70, 71]. In power networks, the majority of the cyberattack literature are focused on detecting FDI attacks. The detection methods proposed in [22, 72, 73, 10] still rely on testing the residuals or the difference between, but dynamic (as opposed to steady-state) state estimation/prediction is performed by considering the control actions or historical states.
[10] considers the difference between the observed and predicted sensor data distributions, rather than the data points. Of these methods, [29, 74, 30, 75] investigates localization of FDIs. [29] and [74] localizes FDIs by identifying the patterns in the sensor measurements through matrix decomposition. [30] considers GPS spoofing on PMU sensors and identifies the attacked sensors using a probing technique. The methods proposed in [75, 32] are based on distributed state estimation.

The methods proposed in the literature for detecting covert attacks are based two types of assumptions: either 1) some of the sensors in the system are immune to the attack [18] or 2) the operations of some part of the system is unknown to the attacker [25]. In [25] and [26], the authors assume an auxiliary system whose dynamics correlated with system under attack but is unknown to the attacker. The manipulation of control action will impact the behavior of the auxiliary system, where state estimation can be used to capture the attack. In [76], the authors assume the system dynamics during degradation is unknown to the attacker and detects the covert attack in a similar manner. Among the studies on covert attack detection, only [77] uses a deep learning framework to identify the location of covert attacks through supervised classification.

In this chapter, we aim at simultaneously detecting and localizing covert attacks in a power grid. Specifically, we consider a transmission system consisting of multiple regional control centers (RCCs) and an independent system operator (ISO), where the RCCs are vulnerable to covert attacks. The ISO acts as a neutral organization that collects all the data in the network and manages the power transmission among regions. The RCCs are controlling power generation locally according to the demand of their designated regions considering the planned power transmission between regions. We assume RCCs represent different power utility companies and no data communication between them. Our goal is to establish a detection and localization method to be implemented at the ISO in near real-time. We aim at extracting the data feature from the sensor data that utilizes the interconnectivity of the system to detect and localize covert attacks.
Our main contributions are summarized as follows:

• First, we build an integrated algorithm that detects and localize covert attacks in real-time. Previous studies on covert attacks mainly focus on detection. Our proposed method detects and localizes covert attacks simultaneously. The proposed method is based on a realistic combination of the two basic assumptions (protected sensors and auxiliary system) used in previous studies on covert attacks. We do not assume any sensors are protected from manipulation, but assume the attacker’s resources is limited to a single region in the grid. Following that, we use the correlation between different regions to detect and localize a covert attack. When one region is under attack, the other regions can be taken as the auxiliary system.

• Second, we demonstrate a generic mechanism of covert attacks on a regional control center under the setting of a networked power generation control operated by multiple RCCs and an ISO as described above. The proposed method detects and localizes covert attacks on one of the RCCs by analyzing the residuals from the ISO state estimation.

• Third, we use the Sparse Group Lasso (SGL) to extract and differentiate the impact of a covert attack on the attacked region and its neighboring regions, which is represented by the SGL coefficients. These coefficients are used as the basis of our attack detection and localization scheme, where the magnitude of the coefficients is used for detection, and the sparsity of the coefficients is used for localization. We lay the theoretical foundation of formulating the SGL problem under a linearized setting and later extend to the nonlinear settings.

Although not a contribution, we demonstrate the effectiveness of our proposed method through a simulation study on an IEEE 14-bus system and an IEEE 118-bus system.
4.2 Problem Setup

We consider an \( N \)-bus power transmission system comprised of power plants and substations that are grouped in \( L \) different regions (an example of \( N = 14 \) and \( L = 3 \) is shown in Figure 4.1). An ISO acts as a centralized coordinator which manages and controls the electric transmission of the power network. The power generation plants are operating under the control of RCCs. The communication between RCCs is realized by ISO, which calculates the power generation setpoints for each generator and sends that information to the corresponding RCCs.

The global system state \( \mathbf{x} \in \mathbb{R}^n \) (\( n = 2N - 1 \) under the \( N \)-bus setting) is defined as 
\[ \mathbf{x}^T = [x_1^T, \ldots, x_N^T], \]
where \( x_i \) represents the state of bus \( i \). In most cases, \( x_i \) is defined as the voltage and phase angle \( (x_k = [v_k, \theta_k]) \) for load bus \( k \), and voltage and active power \( (x_k = [v_k, p_k]) \) for generator bus \( k \). Hence, \( x_k \in \mathbb{R}^2 \) for \( k = 2, \ldots, N \). We represent the decomposition of the network into \( L \) regions as the sets of buses in each region, where \( k \in R_i \) if bus \( k \) is in region \( i \). We further denote the set of generator buses in region \( i \) as \( G_i \) (i.e., \( k \in G_i \) if bus \( k \) is a generator bus in region \( i \)). Without loss of generality, the state vector \( \mathbf{x} \) can be rewritten as \( \mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_L] \), where \( \mathbf{x}_i \) is the state vector of region \( i \). We further assume there are \( m \) \((m > n)\) sensors in the global system that guarantee the observability of the system, and we denote the vector of measurements as \( \mathbf{z} \in \mathbb{R}^m \). \( z_j \) may be the line power flows, the bus voltages and/or currents, loads of all the load buses, and
the active powers of all generator buses. We denote the set of sensors in region $i$ as $N_i$, and that $z_j \in N_i$ if sensor $j$ is measuring the voltage, power, current, or power flow in/out of bus $k$, where $k \in R_i$.

### 4.2.1 Global Model

The measurement model is represented by a (nonlinear) measurement function

$$
z = h(x) + e, \quad (4.1)$$

where $e$ is the measurement noise. The state estimation is given by solving the following optimization problem:

$$
\min_x (z - h(x))^T \Sigma^{-1} (z - h(x)) \quad (4.2)
$$

where $\Sigma$ is the diagonal matrix of the sensor measurement precision.

For a nonlinear model, the above problem is solved by Newton’s method, where $J$ is the Jacobian matrix of $h(\cdot)$, and the state estimation $\hat{x}$ is given by

$$
\hat{x}^{\nu+1} = \hat{x}^\nu - (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (z - h(\hat{x}^\nu)) \quad (4.3)
$$

However, if the system operates around a state $x_0$, the model can be properly linearized around $x_0$ [78]. The state $x_0$ can be obtained from a recent state estimation, which remains valid for multiple observations.

### 4.2.2 Local Model

We assume each RCC controls the local power generation through a SCADA system. The SCADA system uses the received power generation setpoints and the nominal voltage levels as the local control targets. The RCC then calculates the vector of local control actions $u_i$ for generators buses $G_i$. We use a state-space model to represent the dynamics of the local
where the function \( f_i \) defines the dynamics of region \( i \). It is commonly represented by a linear dynamic model, where \( f_i(x_i, u_i) = A_i x_i + B_i u_i + w_i \), where the matrices \( A_i \) and \( B_i \) depend on the corresponding generator and actuator configurations, and \( w_i \sim \mathcal{N}(0, P_i) \) represents the process noise. Based on the system model, the control action is calculated as

\[
    u_i = f_c^i(z_i, \hat{x}_i, x^s_i)
\]

where \( \hat{x}_i \) is the local state estimation based on a (extended) Kalman filter. The function \( f_c^i \) is defined such that \( u_i \) minimizes the discrepancy between the actual state and the setpoints \( (x^s_i) \) represented by some discrepancy function \( D(x_i, x^s_i) \). One example is \( D(x_i, x^s_i) = (x_i - x^s_i)^T \Sigma^{-1} (x_i - x^s_i) \) where \( \Sigma \) is the diagonal weight matrix.

4.2.3 Covert Attack

The covert attack is defined as the cyberattack that maliciously manipulates the system control and disguises itself by rewriting the sensor measurements to compensate for such manipulation. The mechanism of a covert attack on a steady-state linear system was first proposed in [19], where the manipulation of sensor data is a linear function of the manipulation of the control action. In [76], the authors proposed the covert attack against a linear dynamic system. The proposed attacks were proved to be undetectable when all the sensors in the system are vulnerable to manipulation. However, there has not been a generalized definition of how a covert attack can be implemented on a nonlinear system.

We provide a more generalized definition of covert attack that applies to both linear and nonlinear systems. For both types of systems, the mechanism of the covert attack can be taken as the attacker has an accurate system simulator, that takes the control action as the
input and generates the expected sensor readings as outputs. With this simulator and the ability to access the controller, the attacker could manipulate the control action arbitrarily, and in the meantime, replace the intermediate sensor data with the expected output from the simulator.

We assume a covert attacker has limited resources and can access no more than one RCC. This assumption is more realistic for larger and complex systems such as the smart grid. On the other hand, for simplicity, we relax the assumption on the attacker’s system knowledge by assuming the attacker can simulate sensor data based on the full knowledge of the network \( h(\cdot) \). During the attack, the attacker follows the procedure below:

1. Read and manipulate the control of the generator such that the state (power and/or voltage) of the generator is altered. That is,

\[
\begin{align*}
\mathbf{u}_i^a & \neq \mathbf{u}_i \\
\Rightarrow \mathbf{x}_i^a & = \mathbf{x}_i + \delta_i, 
\end{align*}
\]  

where \( \delta_i \) is the shift of the state caused by a covert attack, and \( \mathbf{x}_i^a \) is the state of region \( i \) under attack.

2. Simulate the correct sensor measurements corresponding to the normal state of the generator based on acquired system knowledge. That is, calculate

\[
\tilde{\mathbf{z}} = h(\mathbf{x})
\]

3. Manipulate the corresponding sensor measurements of the generator \( i \) with the simulated sensor measurements. That is,

\[
\mathbf{z}_k^a = \tilde{\mathbf{z}}_k \quad \forall k \in N_i.
\]

By following the above procedure, the attacker has a very high chance of bypassing
the local detection schemes. This is because all the sensors in $M_i$ that would show the abnormality in state $x_i$ are manipulated with simulated normal readings. The covert attack is proven to be undetectable for linear systems when the attacker has access to all the sensors and full knowledge of the system dynamics[19]. In [18], the authors designed a detection method for the scenario when the attacker has limited access to the sensors, which is similar to our assumption here. However, the limited access in [18] relies on the assumption that a subset of the sensors is always protected from manipulation. In this chapter, we assume none of the sensors is immune from manipulation. Instead, we assume the attacker could only access those sensors that are related to the generator bus being attacked, which is more realistic in the sense that it is almost impossible to acquire access to all the sensors in the system. By manipulating only the related sensors, the attack can theoretically be detected by the traditional $\chi^2$ detection scheme. However, due to the sparse connection between the state of the attacked generator and the sensors outside the attacked region, as well as the measurement noise, the performance of the $\chi^2$ detection scheme can be largely discounted. This is also true in practice, especially when the system scale and the number of sensors are large.

4.3 Methodology

In this section, we propose an online approach to tackle the problem of detecting a covert attack as described in Section 4.2.3 on one of the RCCs, as well as locating where the attack occurs. The problem we consider is under the scenario where a power transmission system consists of multiple regions, where in each region, there are generator buses that are vulnerable to covert attacks. An ISO collects all the sensor data from all the regions and estimates the state. Our algorithm is designed to be implemented at the ISO, where all the sensor measurements are processed, and the complete network topology is known. We first introduce the formulation of the Sparse Group Lasso (SGL). Then, we show how to formulate the detection and localization problem as an SGL problem based on a simplified
linear model, in order to build the theoretical basis and develop the detection algorithm. Finally, we extend the methodology to the nonlinear system setting.

4.3.1 Sparse Group Lasso

The formulation of the sparse group Lasso was proposed in [79] as an advancement of the group Lasso problem that selects the group(s), among \( L \) groups of predictors \( X_1, \ldots, X_L \), that explain the variation in the data \( y \)

\[
\min(\|y - \sum_{l=1}^{L} X_l \beta_l\|^2_2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|\beta_l\|_2),
\]

(4.9)

where \( p_l \) is the group size, and \( \| \cdot \|_2 \) is the \( L_2 \) (Euclidean) norm. The penalty term \( \lambda \sum_{l=1}^{L} \sqrt{p_l} \|\beta_l\|_2 \) yields sparsity at the group level. The sparse group Lasso considers within-group sparsity in addition to the group level sparsity, which solves the following optimization problem:

\[
\min(\|y - \sum_{l=1}^{L} X_l \beta_l\|^2_2 + \lambda_1 \|\beta\|_1 + \lambda_2 \sum_{l=1}^{L} \|\beta_l\|_2),
\]

(4.10)

where the \( L_1 \) penalty term \( \lambda_1 \|\beta\|_1 \) yields the element-wise (within-group) penalty. The SGL problem can be solved by block coordinate descent. The algorithm is given in [79].

4.3.2 Linearized System

We first hypothesize a simplified linearized model of the system, where the system operates around some state \( x_0 \), and the measurement function is linearized in the form of

\[
z = h(x) = H_0 x,
\]

(4.11)

where \( H_0 = h(x_0) \) is a known constant, which is derived from the steady operating point \( x_0 \). This approach is commonly used in the literature. For example, a state-space model
is used in [37], and a linear regression model is used in [80]. Under this setting, the state estimation is simplified as a weighted least squares estimator given below:

\[
\hat{x} = (H^T \Sigma^{-1} H)^{-1} H^T \Sigma^{-1} z. \tag{4.12}
\]

During the attack, as given in Section 4.2.3,

\[
z' = H(x + \delta) + e = z + H\delta + e, \tag{4.13}
\]

where \(z'\) is the original measurements under covert attack before the attacker manipulates the sensor data, and \(\delta\) is sparse, such that \(\delta_j = 0\) for all \(j \notin N_i\), because the attack only changes the state of the generator buses in region \(i\). The measurements (\(z^a\)) after the attack manipulates the sensor data, according to (4.8), is

\[
z^a_j = z_j, \quad \forall j \in N_i \tag{4.14}
\]

\[
z^a_j = z'_j, \quad \forall j \in N_i^C \tag{4.15}
\]

where \(\delta_i = \delta[S_i]\), where \(S_i\) denotes the set of elements corresponding to the state of region \(i\), \(x_i\), in the state vector \(x\). \(N_i\) represents the set of sensors in region \(i\), and \(N_i^C\) is the complement of the set \(N_i\). From (4.13-4.15), we have

\[
z^a = z + B_i \delta_i, \tag{4.16}
\]

where \(B_i \in \mathbb{R}^{G_i \times 2}\) is defined as follows:

\[
B_i[N_i^C, \cdot ] = H[N_i^C, S_i], \\
B_i[N_i, \cdot ] = 0,
\]
with $B[S, \cdot]$ representing the rows in set $S$ of matrix $B$, and $B[\cdot, S]$ representing the columns in set $S$ of matrix $B$. In general, the matrix $B_i$ shows the relation between the state of bus $i$ and the measurements of the sensors that are not directly connected to bus $i$.

When the system is complex, each state is correlated with (different) multiple sensors. Notice that $B_i$ is obtained by taking some columns of $H$ and setting a subset of elements to zero. This transformation is nonlinear since it is an element-wise sparse operation, meaning there is a very high chance that $B_i$ does not fall into the column space of the matrix $H$. Here we assume columns of $B_i$ do not fall in the column space of $H$. In practice, if there exists a column that does, we remove the corresponding column, and the alteration of the corresponding state element is undetectable.

Based on the above derivation, when we know that region $i$ is under attack, we could estimate $\delta$ in the following way:

1. Project the measurement onto the column space of matrix $H$ to estimate the state $\hat{x}$ (i.e., solve the weighted least square estimation using Eq. (4.12)).

2. Project the residuals $r = z - H\hat{x}$ onto the column space of matrix $B_i$, and the solution is $\delta_i$, which can be expressed as:

$$\delta_i = (B_i^T B_i)^{-1} B_i^T r.$$  

The above solution is only valid when it is known that region $i$ is under attack. In reality, the region under attack is unknown and needs to be localized. Note that in the above solution, $B_i$ can be treated as the basis for region $i$. Therefore, one can find all the bases for all the regions, and the problem can be formulated as finding the basis among $B_i$ for all $i = 1, ..., L$ that best explains the residuals $r$. Since it is likely that only a subset of the states of region $i$ is altered (e.g. when there are multiple generators in one power station, the attacker might only attack a subset of the generators, or the attacker only changes the power without changing the voltage), $\delta_i$ can be also sparse. Therefore, if we divide the
elements into groups according to the states of each node $i$, the estimated $\delta$ should be: 1) between-group sparse, meaning there should be only one basis $B_i$ that properly explains the residuals and 2) within-group sparse, meaning it is very likely that only a subset of the elements in $x_i$ is altered, which also means only a subset of the columns in basis $B_i$ is important in explaining the residuals variation.

The above problem can then be formulated as a Sparse Group Lasso (SGL) with linear constraint in the following form:

$$
\min_{\beta_1, ..., \beta_L, \hat{x}} ||\mathbf{r} - \sum_{i=1}^L B_i \beta_i||_2^2 + \lambda_1 ||\beta_i||_1 + \lambda_2 \sum_{i=1}^L ||\beta_i||_2
$$

(4.17)

such that $H \hat{x} + \mathbf{r} = \mathbf{z}$, (4.18)

where $\lambda_1 ||\beta||_1$ is the $L_1$ penalty term that encourages within-group sparsity, and the $L_2$ penalty term $\lambda_2 \sum_{i=1}^L ||\beta_i||_2$ encourages the between-group sparsity. Under our assumption that only one region is under attack, there should be only one of all $||\beta_i||$’s that is significantly greater than 0. Notice the SGL is different from the original Lasso where only element-wise sparsity is encouraged, which may lead to multiple non-zero elements in the solution belonging to different regions, and further inference needs to be conducted to identify the attacked region. On the other hand, the SGL automatically forces group-wise sparsity, which selects the region that is most skeptical of being attacked.

Since we assume that the basis $B_i$ does not lie in the column space of $H$, the above optimization problem can be solved by first solving the linear regression problem and then the SGL without the constraint, which is demonstrated in Algorithm 0. The solution $\beta = [\beta_1^T, ..., \beta_L^T]$ can be taken as a representation of $\delta$. For region $i$ under attack, $||\beta_i|| > 0$, and for region $j$ not under attack, we expect to get $\beta_j \approx 0$. Within the detected generator, the non-zero elements would correspond to the altered state variables. When there is no attack, $||\beta_i||$ would be close to 0 for all $i$. The online detection mechanism is built based on the maximum magnitude of L1 norm $||\beta_i||_1$: the alarm is triggered when $\max ||\beta_i||_1$ is
greater than the pre-specified threshold $\theta$. The selection of threshold is described in Section 4.3.4

**Algorithm 3** SGL-based attack detection and localization for linear system

**Input:** $tol, z, H, \{M_1, \ldots, M_K\}, \{S_1, \ldots, S_N\}, \theta$

1. $alarm = 0$, $converge = 0$
2. **for** $t = 1, 2, \ldots$ **do**
3. \hspace{1em} $z \leftarrow z(t)$
4. \hspace{1em} $\hat{x} = (H^T \Sigma^{-1} H)^{-1} H^T \Sigma^{-1} z$
5. \hspace{1em} $r = z - H \hat{x}$
6. \hspace{1em} Solve (4.17);
7. \hspace{1em} **if** ($\max ||\beta_i||_1 > \theta$) **then**
8. \hspace{1em} \hspace{1em} $alarm = 1$ (Detection);
9. \hspace{1em} \hspace{1em} $location = \arg\max ||\beta_i||_1$ (Localization);
10. \hspace{1em} \hspace{1em} **break**;
11. \hspace{1em} **end if**
12. **end for**
13. **Return** $alarm, location$

### 4.3.3 Extension to Nonlinear System

We now extend the formulation in the previous subsection to the nonlinear system setting as given by (4.1). The optimization problem formulation for the nonlinear system is as follows:

$$\min_{\beta_i, \ldots, \beta_L, \hat{x}} ||r - \sum_{i=1}^{L} B_i \beta_i||_2^2 + \lambda_1 \sum_{i=1}^{L} ||\beta_i||_1 + \lambda_2 \sum_{i=1}^{L} ||\beta_i||_2$$

such that

$$h(\hat{x}) + r = z$$

$$B_i[N_i^C, \cdot] = H(\hat{x})[N_i^C, S_i]$$

$$B_i[N_i, \cdot] = 0$$

In the above optimization problem, $H(x)$ is the Jacobian matrix of $h(x)$. Similar to the linear setting, when region $i$ is under attack, the corresponding $\beta_i$ should be large, otherwise, we expect $\beta_i \approx 0$.  

76
Algorithm 4: SGL-based attack detection and localization for nonlinear system

| Input: tol, z, H, \{M_1, ..., M_K\}, \{S_1, ..., S_N\}, λ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1: alarm = 0, converge = 0; |
| 2: for \(t = 1, 2, \ldots\) do |
| 3: \(z = z_{new} \leftarrow z(t)\); |
| 4: \(\hat{x}_{old} = x_0\); |
| 5: while \(!converge\) do |
| 6: Solve \(\min_x (z - h(x))^T \Sigma^{-1} (z - h(x))\); |
| 7: \(r = z - H\hat{x}\); |
| 8: Solve (4.17); |
| 9: \(z_{new} \leftarrow z - \sum_{i=1}^L B_i \hat{\beta}_i\); |
| 10: if \(||\hat{x}_{new} - \hat{x}_{old}|| < tol\) then |
| 11: converge = 1; |
| 12: end if |
| 13: \(\hat{x}_{old} \leftarrow \hat{x}_{new}\); |
| 14: end while |
| 15: if \((\max ||\beta_i||_1 > \lambda)\) then |
| 16: alarm = 1; |
| 17: location = \(\arg\max ||\beta_i||_1\) |
| 18: break; |
| 19: end if |
| 20: end for |
| 21: Return alarm, location |

Note that due to the nonlinear constraint, the above optimization problem has no closed-form solution or iterative algorithm with guaranteed convergence. The Jacobian matrix \(H(x)\) is not a constant, but a function of \(x\). This means the solutions of SGL and SE rely on each other. Therefore, we need to iterate between the SGL and the SE problems and re-approximate \(H\) and the basis \(B_i\) at every iteration, based on the new state estimation \(\hat{x}\). At each iteration, we first solve the SE problem using Newton’s method and get the residual \(r\). Then, we solve the SGL using block coordinate descent and get estimates of \(\beta_i, i = 1, \ldots, L\). In the next iteration, the state estimation is solved by correcting \(z\) using \(\hat{\beta}_i, i = 1, \ldots, L\); i.e., \(z' = z - \sum_{i=1}^L B_i \hat{\beta}_i\). At each time step, this procedure is iterated until convergence. This procedure is shown in Algorithm 0.
4.3.4 Threshold Selection

Since the $\max ||\beta_i||_1$ values, when the system is not under attack, depend on both the system configuration ($H$ matrix and the topology) and the penalty parameters $\lambda_1$ and $\lambda_2$, the theoretical distribution of $\max ||\beta_i||_1$ is unknown. However, based on our previous derivations, we expect that the resulting $\max ||\beta_i||_1$ value under attack tends to be higher than its value under normal conditions. Therefore, we choose the threshold $\theta$ offline based on the empirical distribution of $\max ||\beta_i||_1$ under normal conditions. This can be taken as a training process where the algorithm is implemented on the historical data that contains only normal data (i.e., we know that the data is not contaminated by cyberattacks). The input to the model is a collection of measurements $z$ and the system configuration. The output from the algorithm would be the collection of $\{\max ||\beta_i||_1\}$ values constituting the empirical distribution of the monitoring statistic. We then select the upper-$\alpha$ quantile of the historical statistics such that

$$Pr(\max ||\beta_i|| > \theta) = \alpha$$

Here $\alpha$ means the desired false alarm rate, which is related to the desired in-control average run length $ARL_0$ (number of observations between false alarms), i.e.,

$$\alpha \approx 1/ARL_0.$$ 

Either $ARL_0$ or $\alpha$ can be specified to identify the threshold. In practice, there is a trade-off between the false alarm rate and the sensitivity of the monitoring scheme. Intuitively, a higher threshold leads to a lower false alarm rate and lower sensitivity (higher missing report rate). On the other hand, a lower threshold increases the sensitivity of the detection scheme but also increases the false alarm rate. A typical engineering specification of $\alpha$ is between 0.01 and 0.001, such that the $ARL_0$ between 100 and 1000.
The exact value of $\theta$ could be different depending on the system configuration, the penalty parameters $\lambda_1$ and $\lambda_2$, and the desired false alarm rate. However, the above process of threshold selection applies to any general system.

4.4 Numerical Results

We validate the proposed detection algorithm on both linear and nonlinear systems. For the linear system setting, we model the complete system with a 20-variable linear time-invariant state-space model composed of 4 regions. For the nonlinear system setting, we use the IEEE 14-bus model and decompose it into 4 regions, and IEEE 118-bus model decomposed into 3 regions.

4.4.1 Linear System

For simulation on the linear system, we use the following discrete-time state-space model to represent the system operations:

$$x(t + 1) = Ax(t) + Bu(t) + e(t)$$  \hspace{1cm} (4.19)

$$z(t) = Hx(t)$$  \hspace{1cm} (4.20)

where $x$ and $z$ are the system state and sensor measurement, respectively, as defined earlier, $u$ is the control action that is calculated by the controller to keep the system state at target. (4.19) is the state-transition function, and (4.20) is the measurement function. We generate the state-transition matrix $A$ randomly as a near-block diagonal positive-definite matrix with block submatrices on diagonal representing the within-region strong connections, and smaller values on the off-diagonal representing the lose connections across regions. The control action $u$ is calculated by a coupled linear-quadratic regulator[41]. Note that $H$ is a sparse matrix where each state variable only affects a subset of the sensors. The non-zero elements are generated from a uniform distribution between 0 and 1. The sensors connected
to region \( i \) are defined by the strong correlation between the sensor \( j \) and the state \( x_i \), where \( ||H_{[j,S]}||_\infty > 0.5 \) means sensor \( j \) is directly connected to region \( i \). The state-transition function is taken as a “black box” which is assumed unknown, and the steady-state estimation is implemented only based on (4.20) using (4.12). In this simulation, we have 20 state variables (i.e., \( x \in \mathbb{R}^{20 \times 1} \)) and 30 sensors (i.e., \( z \in \mathbb{R}^{30 \times 1} \)). We run the detection algorithm for \( N = 500 \) replications to evaluate the detection delay and localization performance on average. In each replication, the attacked region \( i_{\text{attack}} \) is chosen randomly, and the thresholds remain unchanged.

The magnitude of attack is defined by the signal-to-noise ratio (SNR):

\[
SNR = \sqrt{\delta_i^2 \Sigma_i^{-1} \delta_i},
\]

where \( \Sigma_i \) is the covariance matrix of the state variables of region, \( x_i \).

The attack detection performance of our proposed method is evaluated by the in-control and out-of-control average run length, i.e. \( ALR_0 \) and \( ALR_1 \). The average run length is defined as the average number of observations before an alarm is raised:

\[
ARL = \mathbb{E}[\min\{t : \max(||\beta_i(t)||_1 > \lambda\}]
\]

(4.21)

It has the following relation with the type-I and type-II error rates:

\[
ARL_0 \approx \frac{1}{Pr(\text{type-I error})} = \frac{1}{Pr(\text{false positive})}
\]

(4.22)

\[
ARL_1 \approx \frac{1}{1 - Pr(\text{type-II error})} = \frac{1}{1 - Pr(\text{false negative})}
\]

(4.23)

The out-of-control \( ARL_1 \)'s along with their standard deviations under different SNR's are shown in Table 4.1. For comparison, we use the traditional \( \chi^2 \) detector as a baseline, whose ARL is also given in the table.

Recall that in our proposed algorithm, the attack localization is identified as \( \text{argmax} ||\beta_i||_1 \).
The attack localization performance of the proposed method is evaluated by the identification accuracy, precision, recall, and the $F$ score of the proposed attack localization approach, which are calculated using the following equations:

$$\text{Accuracy} = Pr(\text{argmax}||\beta_i||_1 = i_{\text{attack}}),$$  \hspace{1cm} (4.24)

$$\text{Precision} = \frac{TP}{TP + FP},$$  \hspace{1cm} (4.25)

$$\text{Recall} = \frac{TP}{TP + FN},$$  \hspace{1cm} (4.26)

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$  \hspace{1cm} (4.27)

where $TP$, $TN$, $FP$, $FN$ are the number of true positives, true negatives, false positives, and false negatives, respectively. The precision values in Table 4.1 are obtained by taking the average of the precision values over all three regions, and the same applies to the recall and $F$ score values. The accuracy represents the probability that the algorithm correctly identifies the attacked region at the time of detection. Precision represents the proportion of correct alarms among all the alarms, recall represents the proportion of correct alarms among all the cases where region $i$ is indeed under attack, and $F$ score is the harmonic mean of precision and recall.

The accuracy of attack localization of the proposed method is compared with a modification of the hypothesis testing technique used in the literature [71], where we test the group of sensors that are related to region $i$. More specifically, for each region $i$, we remove the sensors in the set $N_i$ and re-estimate the state using the remaining sensor measurements. The new $\chi^2_i$ statistic is calculated accordingly. After we go through all the regions, the new $\chi^2_i$ statistics are compared, and the attacked region is identified as the region $i$ that minimizes $\chi^2_i$. This is because a low $\chi^2_i$ value means the removed sensors best explain the abnormality. The accuracy of the proposed method and the hypothesis testing method is given in Table 4.1.

The results in Table 4.1 show that the proposed method has a higher detection power.
and a higher localization accuracy than the traditional $\chi^2$ detector. For example, when the SNR is 1, the localization accuracy of SGL is 69.8%, which is greater than the accuracy of the $\chi^2$ detector, 48.8%. When the SNR is 6, the accuracy of both the methods increase, where the $\chi^2$ detector reaches a 86.6% accuracy, and SGL reaches a 99.4% accuracy, which is also better than $\chi^2$. Table 4.1 shows as SNR increases, the localization accuracy for both methods increase. However, the proposed method has a higher accuracy than the hypothesis tests under all the tested SNR levels. More importantly, the proposed method reaches a reasonably high accuracy (more than 85%) at a relatively low SNR level (SNR=2), while the hypothesis test reaches similar accuracy at a much higher SNR level (SNR=6). This means the proposed method is more sensitive to covert attacks.

Table 4.1: ARL and Accuracy under different levels of SNR

<table>
<thead>
<tr>
<th>SNR</th>
<th>$\chi^2$ ARL (Std.Dev.)</th>
<th>Precision</th>
<th>Recall</th>
<th>F score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\chi^2$ SGL</td>
<td>$\chi^2$ SGL</td>
<td>$\chi^2$ SGL</td>
<td>$\chi^2$ SGL</td>
</tr>
<tr>
<td>0</td>
<td>203.46 (8.04) 200.84 (9.96)</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>1</td>
<td>171.68 (7.26) 153.34 (6.80)</td>
<td>48.80% 69.80%</td>
<td>52.74% 72.63%</td>
<td>48.80% 69.80%</td>
</tr>
<tr>
<td>2</td>
<td>97.40 (4.48) 82.78 (3.88)</td>
<td>60.80% 85.80%</td>
<td>63.13% 86.77%</td>
<td>59.45% 85.37%</td>
</tr>
<tr>
<td>3</td>
<td>50.20 (2.22) 41.47 (1.95)</td>
<td>68.20% 92.60%</td>
<td>73.86% 92.79%</td>
<td>68.27% 92.60%</td>
</tr>
<tr>
<td>4</td>
<td>23.32 (1.00) 16.18 (0.66)</td>
<td>72.40% 97.00%</td>
<td>75.97% 97.13%</td>
<td>71.37% 97.05%</td>
</tr>
<tr>
<td>5</td>
<td>15.54 (0.63) 13.11 (0.51)</td>
<td>80.40% 96.40%</td>
<td>82.90% 96.77%</td>
<td>80.91% 96.38%</td>
</tr>
<tr>
<td>6</td>
<td>12.13 (0.44) 8.38 (0.27)</td>
<td>86.60% 99.40%</td>
<td>87.99% 99.39%</td>
<td>86.36% 99.41%</td>
</tr>
</tbody>
</table>

4.4.2 Nonlinear System

To validate the performance of the method on nonlinear systems, we simulate the attack using both the IEEE 14-bus model and the larger-scale IEEE 118-bus model to investigate the scalability. The inputs to the simulation are the load profiles of the load buses, the generation plan of the generator buses, and the phase angles and voltages at each bus. The load profiles are generated from the real data from Pecan Street dataset. The generation plan is generated based on the load by solving the mixed-integer unit commitment problem [81].

We simulate the system for 250 observations and monitor the $l_1$ norm of the $\beta_i$ vectors. The threshold is selected based on the 0.995 quantile of the monitoring statistic ($\max ||\beta_i||$),
such that the in-control average run length is around 100. Two examples of attack on #3 for
the IEEE 14-bus experiment and R2 for the IEEE 118-bus experiment are shown in Figure
4.2. In the IEEE 14-bus example, the network is divided into 4 regions, with each region
containing a generator bus, and the attack reduces the voltage level of the attacked generator
bus by 20%, which is considered a significant attack. In the IEEE 118-bus example, the
network is divided into 3 regions based on connectivity. The attack reduces the power
generation level by 5%, which can be considered a less significant attack. In both cases,
the onset of the attack is at \( t = 50 \). Before the onset of the attack, the magnitudes of \( ||\beta_i|| \)
for all generator buses are low, and false alarms are triggered with a low possibility. On the
contrary, after the onset of the attack, and the \( \max ||\beta_i|| \) values are above the threshold such
that alarms are frequently triggered, and the \( \max ||\beta_i|| \) is reached by the region under attack
with high probability. The result indicates the proposed algorithm successfully detects and
localizes the attack at the same time. We compare the proposed method with the group-
wise \( \chi^2 \) test as in the previous subsection. The sensors in each region are grouped and a
corresponding \( \chi^2 \) statistic is calculated by

\[
\chi^2_i = \sum_{k \in N_i} \frac{r^2_k}{\sigma^2_k}
\]

for region \( i \). The \( \chi^2_i \) for all the regions are monitored individually in parallel. The monitoring results are shown in Figure 4.3. Comparing Figure 4.2a with 4.3a, it is observed that when the attack is significant, it can be captured by both the proposed method and the traditional method. However, the traditional method raises false alarms for the regions not under attack, making it unable to accurately identify the attacked region. On the other hand, by comparing the results in Figure 4.2b with Figure 4.3b, it is shown that, using the traditional method, the covert attacks on one region raise false alarms in other regions (when the attack is on region 2, region 1 raised alarms more frequently), which can be more significant than the attacked region. However, this problem is well-addressed by the proposed method due
to the penalization terms in SGL.

As for the run time of the proposed method, the average run time for each iteration to converge under the IEEE 118-bus setting is 3.581 seconds, expectedly longer than the 0.391 seconds for state estimation due to the iterative algorithm. However, this also means the algorithm is capable of converging within several iterations (with the average number of iterations to convergence being 8.12), and the run time is in the acceptable range for real world cybersecurity implementations. The run time of the proposed algorithm under the IEEE 14-bus setting is 0.067 seconds, which is similar to the state estimation run time 0.051 seconds under the same setting, and the average number of iterations to convergence is 5.26.

Figure 4.2: Simulation Results

(a) IEEE 14-bus, SGL-based detection of attack on bus #2
(b) IEEE 118-bus, SGL-based detection of attack on region 2

Figure 4.3: Simulation Results

(a) IEEE 14-bus, group hypothesis testing-based detection of attack on bus #2
(b) IEEE 118-bus, group hypothesis testing-based detection of attack on region 2
4.5 Conclusion

We proposed an online approach to detect and localize covert attacks in power transmission grids. Our detection approach is based on the SGL formulation. We showed the theoretical foundation of applying SGL to a linearized system and extending the method to a nonlinear system setting with relaxation. We conducted a simulation study to evaluate the performance of our proposed method by highlighting the average run length, which indicates the expected detection delay, as well as the localization identification accuracy. The results showed that, as the severity (SNR) of attack increases, both the detection power and the localization accuracy of the proposed method increase. The results also showed the proposed method is more sensitive than $\chi^2$ tests, especially when the system scale is large. Furthermore, we implemented a case study on the IEEE 14-bus and IEEE 118-bus systems as a representative of the more practical and large-scale nonlinear systems. The results showed that the proposed method applies to nonlinear systems and able to reach shorter detection delay and higher localization accuracy as the attack severity increases. As for future work, we will investigate the scalability and robustness of the proposed method. We can also extend the method to detection, localization, and identification of other types of cyberattacks and faults in power transmission systems. Another direction is to incorporate the deep-learning-based methods in order to reach a high accuracy and still preserving interpretability. Presently, our optimization algorithm for the nonlinear setting is an extension of the linear setting. We plan to further develop the optimization algorithm that better solves the SGL with nonlinear constraints.
CHAPTER 5
DEEP LEARNING BASED COVERT ATTACK IDENTIFICATION FOR
INDUSTRIAL CONTROL SYSTEMS

5.1 Introduction

In this chapter, we demonstrate the application of deep learning towards the detection, diagnosis, and localization of covert attacks. We focus on generic networked industrial control systems and propose a data-driven framework that combines an autoencoder, an RNN, and a DNN. We use the RNN to characterize the system behavior under normal operations. The output of the RNN together with the sensor measurements are fed to a DNN classifier to detect, diagnose, and localize the anomaly. We use the autoencoder to extract features that represent the system status, as well as the spatial correlation among the nodes, in an unsupervised manner. The RNN captures the temporal behavior of the features extracted by the autoencoder, and the DNN helps detect anomalies in the system as well as diagnose whether it is an attack or fault. By considering both the spatial and temporal behavior of the system, this DL framework helps reduce false alarms triggered by natural faults as well as localize the attack by extracting the features that distinguish anomalies at different locations and between attack and faults.

5.1.1 Related Work

The literature on cyberattack detection for industrial control systems can be divided into two groups: model-based detection and data-driven detection.

Model-based methods rely on the engineering knowledge of the physical rules to establish a parameterized model of the normal sensor measurements [8, 18, 13, 37, 69]. In [18], [13], and [37], the observed sensor measurements are compared with the estimated
sensor measurements from these models, and the residuals (i.e., the difference between the observations and model estimations) are monitored to detect the anomalies. The models are often established such that the expectation of the residuals is approximately 0 under normal operations. During monitoring, a large residual means a large discrepancy between the model estimation and the observations, which indicates the anomalous behavior of the system. The $\chi^2$ detector, for example, tests the sum of squared residuals (SSR) and triggers alarms whenever the SSR is above the threshold defined by a $\chi^2$ distribution. The drawback of the model-based methods is the complexity in establishing an accurate physical model, especially when the system consists of multiple subsystems with complex connections and correlations, both spatially and temporally. For complex systems such as power systems, the system model is usually represented by the steady-state power flow equations, which takes the observations as independent inputs and does not consider the system dynamics over time.

Compared to model-based methods, data-driven methods are more flexible. A number of applications of machine learning techniques have been developed for detecting cyber-attacks. However, most of them focus on designing intrusion detection systems based on the network traffic data [82, 83, 84]. If such intrusion detection systems fail, there is a lack of backup detection scheme that detects cyberattacks based on the physical process. On the other hand, unlike the behavior of network traffic, the physical process has a relatively stable and consistent dynamic behavior that does not change over time. Therefore, a recurrent neural network can be used to characterize the system behavior in a non-parametric manner, especially when the complexity of the system obstructs the establishment of a physics-based model. To take into consideration network traffic data and the sensor data, in [85], the authors proposed a multi-model data fusion and adaptive deep learning method based on a convolutional neural network to characterize the normal system behavior. The framework then detects cyberattacks as well as physical intrusions in a single ICS pertaining to water infrastructure. However, due to the complexity of the proposed method, it
would not be applicable to a network of infrastructures, where a utility operates multiple pieces of equipment (subsystems) with interactions at the same time. Meanwhile, the establishment of such a model requires a thorough understanding of the system, and it is hard to generalize it to other systems.

5.2 System Model

![Networked Industrial Control System](image)

Figure 5.1: Networked Industrial Control System

We consider a networked control system consisting of $k$ subsystems as shown in Fig. 5.1, where the subsystems are not necessarily homogeneous, meaning there is not a single model that can represent the behaviors of all the subsystems. For example, a smart grid is a network of generation buses, representing the power plants, and load buses, representing the substations, and different models should be used to represent the power plants and the substations. We denote the state of the subsystem $i$ at time $t$ by a vector $x_i^t$, where $x_i^t \in \mathbb{R}^{n_i}$, and $n_i$ is the minimum number of variables needed to uniquely define the system state of site $i$. Denote the history of state of subsystem $i$ till time $t$ as $x_i^t = \{x_i^0, x_i^1, ..., x_i^t\}$. The full system state vector at time $t$, $x_t$, is given by a concatenation of all the subsystem states. i.e., $x_t^T = [x_1^{1T}, ..., x_k^{kT}] \in \mathbb{R}^N$ where $N = n_1 + n_2 + ... + n_k$. Note that $x$ is not directly observed, but inferred from the measurements from the $M$ ($M > N$ for system observability) sensors distributed throughout the system. Similarly, we have $y_t^T =
$[y^T, \ldots, y^{kT}] \in R^M$ where $M = m_1 + m_2 + \ldots + m_k$, and $m_i$ is the number of sensors in subsystem $i$. Denote the control action as $u_t^T = [u_1^T, \ldots, u_k^T] \in R^P$, where $u_i^T \in R^{p_i}$ and $P = p_1 + \ldots + p_k$. Similarly, we denote the history of control actions and measurements till time $t$ as $u_{\{t\}} = \{u_0, u_1, \ldots, u_t\}$ and $y_{\{t\}} = \{y_0, y_1, \ldots, y_t\}$, respectively.

In general, the dynamics of the networked system can be represented by

$$x_t = F(x_{t-1}, u_{\{t-1\}}),$$ \hspace{1cm} (5.1)
$$y_t = G_i(x_t),$$ \hspace{1cm} (5.2)

and a single subsystem $i$ can be represented by a recursive function of $x_i^t$ as follows:

$$x_i^t = f_i(x_{t-1}, u_{\{t-1\}})$$ \hspace{1cm} (5.3)
$$y_i^t = g_i(x_t)$$ \hspace{1cm} (5.4)

Systems with linear dynamics where the Markovian property holds, can be represented by a linear state-space model. For an individual subsystem $i$ which operates independently, we have:

$$x_i^t = A_i x_{i-1}^t + B_i u_{i-1}^t$$ \hspace{1cm} (5.5)
$$y_i^t = C_i x_i^t$$ \hspace{1cm} (5.6)

When the subsystems are correlated with each other, the full system can be modeled by

$$x_t = A x_{t-1} + B u_{t-1},$$ \hspace{1cm} (5.7)
$$y_t = C x_t,$$ \hspace{1cm} (5.8)

where $A$, $B$, and $C$ are functions of $A_i$’s, $B_i$’s, and $C_i$’s. In model-based methods, typically the above model is used to build a state estimator (e.g., a Kalman filter) that estimates $\hat{x}_t$. 

89
based on the observed measurement $y_t$ and the previous estimation $\hat{x}_{t-1}$. The estimated $\hat{x}$ is then substituted back into the system model to predict the next measurement $\hat{y}_{t+1}$, which is compared with the observed $y_{t+1}$ to obtain the residuals:

$$r_t = y_t - \hat{y}_t.$$

When accurately parameterized, the above model is robust and works well for simple systems. However, one disadvantage of these physics-based models is the estimation of all the parameters requires a large amount of data, and might cause the identifiability issues. Therefore, We assume the interconnectivity of these systems are complex to represent using a physics-based model, this is true especially when the system is nonlinear and the subsystems are interconnected.

Another disadvantage of the physics-based models is that sometimes they do not consider the dynamic behavior of the system or the temporal correlation in the data. In most systems, the control actions are calculated based on a period of historical data, or on a state estimation from the previous time step. In this case, the temporal correlation is implicit, and a steady-state estimation does not capture this type of correlation. For example, in a power system, the state vector $x$ contains the control actions $u_t$ as well.

We note that the control actions can be represented by the power generation setpoints. These setpoints can be determined by solving an operational planning problem subject to the system demand profile. Therefore, in this chapter, we utilize the well known Mixed Integer Unit Commitment (MIUC) [81] as a means to compute our operational setpoints. The MIUC is widely used in the power industry for operational planning and can be formulated
as follows (5.9):

$$\min_{\alpha, \theta} \quad c^T \alpha + d^T \theta \quad (5.9a)$$

subject to

$$Q \theta + R \alpha = E \quad (5.9b)$$

$$F \theta = H \quad (5.9c)$$

In Problem (5.9) $\alpha$ represents a binary vector of length $|\mathcal{G}| \times T$ indicating if generators are turned on or turned off across each time epoch for each generator in the network, where $|\mathcal{G}|$ is the number of generators. Similarly, $\theta$ represents a real-valued vector of dispatch variables specifying the level of production on generators as well as the electric phase angles on separate buses of length $(|\mathcal{B}| + |\mathcal{G}|) \times T$, where $|\mathcal{B}|$ is the number of buses.

Constraint (5.9b) ensures that the commitment and production decisions are coupled along with ramp-up (Q) and ramp-down (R) constraints found in unit commitment. Constraint (5.9c) enforces flow constraints (F) subject to the phase angle values as well as the transmission line capacities (H).

As mentioned earlier, the implicit temporal correlation generated by calculating the control actions in a history-dependent manner cannot be easily represented by any physics-based model. On the other hand, the occurrence of an attack might be easily captured by analyzing the temporal behavior of the system, while the steady-state does not show any anomaly. Therefore, a data-driven method, specifically RNN, well fits this situation where temporal correlation needs to be captured via data-driven techniques.

### 5.2.1 Covert Attack Model

The covert cyberattack was first proposed in [19] for a single linear system as represented by functions 5.5 and 5.6. Specifically, the attacker knows the values of $B_i$ and $C_i$. Recall that a covert attacker is assumed to possess access to control action and the sensor measurements as well. Under these assumptions, the attacker implements the covert attack by
the following steps:

• First, the attacker manipulates the control actions using the following equation

\[ \tilde{u}_t^i = u_t^i + a_t, \]

(5.10)

where \( \tilde{u}_t^i \) is the manipulated control action at time \( t \), \( a_t \) is the attack signal added to the original control action \( u_t^i \). According to 5.5, this manipulation will alter the system state at \( t + 1 \) by \( B_ia_t \). That is,

\[ \tilde{x}_{t+1}^i = x_t^i + B_ia_t \]  

(5.11)

The consequent sensor measurements will be biased by \( C_iB_ia_t \). That is,

\[ \tilde{y}_{t+1}^i = y_{t+1}^i + C_iB_ia_t. \]  

(5.12)

• Then, the attacker manipulates the sensor measurements by subtracting the above bias. i.e.,

\[ \tilde{y}_{t+1}^i = \tilde{y}_{t+1}^i - \gamma, \]

(5.13)

where \( \gamma = C_iB_ia_t \). In this way, the manipulated measurement \( \tilde{y}_{t+1}^i \) is equal to the expected measurement \( y_{t+1}^i \) without attack. Hence, the covert attack can be successfully disguised, as the measurement is the only output of the system, which means all the data inference is conducted based on \( y \).

In this work, we generalize the covert attack to nonlinear systems represented by Equations (5.3) and (5.4). We assume the attacker gains knowledge of the dynamics of subsystem \( i \). This could be taken as the attacker obtains an estimation of the local functions \( \hat{f}_i \) and \( \hat{g}_i \) in Equations (5.3-5.4), which can serve as the simulator of system \( i \). With this knowledge as well as the access to the control actions and sensor measurements, the attacker conducts
a covert attack using the following steps:

- First, the attacker reads the original control action $u_t$ and simulates the expected sensor measurements using the knowledge of subsystem $i$. That is,

$$\hat{x}_{t+1}^i = \hat{f}_i(x_t^i, u_t^i), \quad (5.14)$$

$$\hat{y}_{t+1}^i = \hat{g}_i(\hat{x}_{t+1}^i). \quad (5.15)$$

- Then, the attacker manipulates the control actions as in Eq. (5.10). According to Eq. (5.3), this manipulation will alter the system state at $t+1$ as well as the sensor measurements. That is,

$$\tilde{x}_{t+1}^i = f_i(x_t^i, \tilde{u}_t^i) \neq \hat{x}_{t+1}^i \quad (5.16)$$

$$\tilde{y}_{t+1}^i = g_i(\tilde{x}_{t+1}^i) \neq \hat{y}_{t+1}^i. \quad (5.17)$$

- Finally, the attacker replaces the consequent sensor measurements with the simulated one. i.e.,

$$\tilde{y}_{t+1}^i \leftarrow \hat{y}_{t+1}^i. \quad (5.18)$$

Notice that we assume the attacker’s access to the sensors is limited to subsystem $i$. This means the attacker is not capable of compensating for the impact of attacking subsystem $i$ on other subsystems. This fact lays the foundation for our detection and localization framework. However, detecting the attack, in this case, is nontrivial because the sensors that are most informative of the attack are manipulated, while the attack’s impact on other sensors does not have a clear indication of the occurrence of an attack as well as its location. In the scenario where the attacker can compensate for the impact on the neighbors as well, the detection method might fail because of the insignificance of the attack’s impact on the “neighbors of neighbors”.

93
5.3 Proposed Detection Framework

The structure of the proposed framework is shown in Fig. 5.2. The autoencoder is used for unsupervised feature extraction. Essentially, it projects the sensor measurements to a lower-dimensional layer that filters out the noise and better represent the system status, which corresponds to the steady-state state estimation in traditional frameworks. We use an RNN with an LSTM [86] layer to capture the nonlinear temporal dependency and the correlation among all the variables and extract the residuals. The RNN is used as a predictor, which corresponds to the particle filters or Kalman filters in the traditional frameworks. The residuals together with the original sensor measurements are fed into a DNN for detection as well as diagnosis, and the DNN corresponds to the detection and diagnosis schemes in the traditional frameworks.

Since the measurement $y \in R^m$ is obtained by the measurement function $G(\cdot)$, which is a mapping from $x$ in the lower dimensional space $R^n$. The correlation among the sensor measurements is actually defined by the correlation among the state variables. Therefore, we use an autoencoder (AEN) to reduce the dimension of the sensor measurements, and for
generality, the code size (output size of the encoder) is chosen as the dimension of the state vector, $n$. In our experiments, the autoencoder (and decoder) consists of 2 dense hidden layers with leaky ReLU activation functions. The encoded sensor measurements is taken as the input to the RNN (in our experiments, consisting of an LSTM layer followed by a dense layer). The output of the RNN is then decoded to reconstruct the predicted sensor measurements. The residuals are calculated as the difference between the observed measurements and the predicted ones. Then, we concatenate the residuals with the observed measurements, and the concatenated data are fed into a DNN. The DNN used in our experiments contains 3 dense hidden layers with ReLU, Sigmoid, and Softmax activation functions in sequence. The output of the DNN determines the type and location of the attack, designated by “normal”, “attack+location”, and “fault+location”.

By the nature of covert attacks, the attacker has to obtain accurate system knowledge to conduct a successful attack. However, in reality, the complex subsystems are often geographically far apart from each other which makes it very unlikely that an attacker can obtain the knowledge necessary for the attack, and get access to more than one subsystem. Therefore, in this work, we assume that the attacker only attacks one node, and it is less likely that faults on different nodes happen to occur at the same time. Hence, there are at most $2k + 1$ independent possible conditions of the system, including ”normal”, {“attack on node $i$, $i = 1, ..., k$"}, and {“fault on node $i$, $i = 1, ..., k$"}.

The autoencoder and the LSTM are trained using only the data from normal operations. The DNN can be trained in a supervised manner using labeled data from simulation or historical data.

5.4 Performance Evaluation

5.4.1 Data Extraction

\footnote{https://icseg.iti.illinois.edu/ieee-14-bus-system/}
We generate data via a simulation study on a smart grid. We use the IEEE 14-bus (Fig. 5.3) to represent the smart grid at the transmission level to generate the time series of sensor measurements under different conditions. The model contains 9 load buses (buses 4, 5, 7, and 9-14), representing 9 substations; 4 generation buses (buses 2, 3, 6, and 8), representing 4 power generation plants; and 1 slack bus (bus 1), which is used to balance the active and reactive power in the system and also serves as a reference for all other buses. The model has 20 edges, representing the 20 transmission lines connecting the load and generator buses. The input to the simulation is the load profiles of all the load buses and the power generation plans of all the generation buses. We obtain the load profile of each substation by aggregating the load profiles of a random number of hourly residential power consumption profiles extracted from Pecan Street [87]. The Pecan Street dataset is collected from real-world residential energy and water usage. To eliminate noise and preserve correlation in the data, we select the data corresponding to residents in Austin, TX in the year of 2017. The generation plan is constructed using the method mentioned in Section 5.2. The simulation in Matlab uses Matpower [88] 7.0 to solve the power flow equation and add measurement noise to guarantee a realistic simulation based on the IEEE
14-bus topology. The output of the simulation is the hourly time series of the 39 sensor measurements for the simulated period. The 39 sensor measurements include the active power flow on each of the 20 transmission lines, the power generation of each generator bus as well as the slack bus, and the voltage of all the 14 buses.

Recall that the mechanism of a covert attack is to alter the system state by manipulating the control actions. Since in the transmission system we are considering in this simulation, most of the control happens in the power generation plants, we only consider the covert attacks on the generator buses. During the attack, the attacker decreases the generation level by a specific portion. The reason we choose to decrease the generation is from the attacker’s objective: compared to generating more power than needed, decreasing the generation will cause possible blackouts and overloading of other generators, which is likely to cause more damage to the system. We simulate the covert attacks on each of the 4 generator buses at 5 levels of severity, where the attacker decreases the power generation by level 1: 20%, level 2: 40%, level 3: 60%, level 4: 80%, and level 5: 100% of the planned generation. We assume the attacker obtains access to all the sensors related to the attacked generator and manipulating the sensor measurements by replacing the original values with the ones obtained from simulation such that it shows the attacked generator bus generates the same amount of power as planned. For comparison, we simulate the faults as the decrease of power generation by the same amount caused by equipment malfunctioning. The biggest difference between a fault and a covert attack is there is no sensor data manipulation.

5.4.2 Numerical Results

We compare our proposed method with two benchmarks. The first one uses the traditional state estimation (SE) residuals for detection (Fig. 5.4). The state estimation is implemented by solving the (nonlinear) static power flow equations using Newton’s method [89], which is implemented in Matpower 7.0. The second method uses the same RNN model as in the proposed model but does not include the autoencoder (Fig. 5.5).
For the proposed model and the “RNN + DNN” model mentioned above, we first calculate the first-order difference of observations to ensure the stationarity and then standardize the differences based on the standard deviation for each sensor. In this case, we use $n = 14$ as the code size of the autoencoder. Usually for an AC $k$-bus system the state space has dimension $2(k - 1)$, consisting of the magnitudes and phase angles of voltages at each bus. Since here we only care about the active power and do not consider the phase angle, we use $k - 1 = 13$ as the code size. The stateful LSTM uses 10 lags to predict the next encoded observation. We train the autoencoder and the RNN with 80% of the normal data and keep them fixed when generating the residuals for attack and fault series. Therefore, the autoencoder and the RNN are only exposed to normal data, which can be used to represent the system behavior under normal operations. The autoencoder and the RNN can be viewed as a non-parametric substitute for the physics-based models.

Since there are 39 sensors, the concatenation of residual and observation data is of
length 78. Since there are 4 generator buses considered, the output of DNN is a multi-class classification of 9 labels listed in Tab 5.1. The numbers refer to the bus number where the attack/fault occurs.

We train the model with 80% of data and show the classification performance of the DNN by testing the model on the rest 20% of data. The performance of the method is evaluated by precision, recall, and the F1-score, which are shown in Tab.5.1 and Tab.5.2, where Tab.5.1 shows the classification performance of DNN among the 9 labels (normal, 4 attacks, and 4 faults), and Tab.5.2 shows the classification among the 3 classes (normal, attack, and fault) as well as the localization performance within the attack and fault classes. The precision, recall, and $F_1$-score are calculated based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) using the following equations:

\[
\text{Precision} = \frac{TP}{TP + FP} \\
\text{Recall} = \frac{TP}{TP + FN} \\
F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

Tab.5.1 and Tab.5.2 show that for most of the cases, the proposed method outperforms the other two methods in terms of detection as well as localization. In general, the SE+DNN model has a similar but slightly better performance than the RNN+DNN model. However, both of them have a very low recall for the attack on bus #2. This is because, in the network topology, bus #2 has a high degree – it has four neighbors, which is the highest among the four generator buses. According to the proposed generalization of the covert attack in Section 5.2, the attacker manipulates all the sensors measuring the power flow on the four edges. Intuitively, in this case, the attacker has the highest coverage of sensor accesses, which best covers the attack. Since the state estimation only refers to the data at the current step and does not consider temporal correlation, it does not give a good estimation of the
underlying truth. On the other hand, since the RNN is taking all the sensor measurements as input, the prediction of RNN is relatively inaccurate due to the noise in the sensor data. However, since the RNN considers the temporal correlation, the performance of RNN + DNN, in this case, is slightly better than SE + DNN. In contrast, the proposed method uses an autoencoder for unsupervised feature extraction, which helps filter out the noise in the data when training the RNN. In this case, the RNN gives a more precise prediction of the data. Therefore, the residuals in this case could better capture the anomaly caused by the attack.

Table 5.1: Precision, Recall, and F score for DNN classification

<table>
<thead>
<tr>
<th>Model</th>
<th>State Estimation + DNN</th>
<th>RNN + DNN</th>
<th>Autoencoder + RNN + DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F Score</td>
</tr>
<tr>
<td>Normal</td>
<td>0.7330</td>
<td>0.9699</td>
<td>0.8350</td>
</tr>
<tr>
<td>Attack #2</td>
<td>0.8761</td>
<td>0.1943</td>
<td>0.3180</td>
</tr>
<tr>
<td>Attack #3</td>
<td><strong>0.9985</strong></td>
<td>0.9983</td>
<td><strong>0.9984</strong></td>
</tr>
<tr>
<td>Attack #6</td>
<td>0.9998</td>
<td>0.9920</td>
<td><strong>0.9959</strong></td>
</tr>
<tr>
<td>Attack #8</td>
<td>0.8645</td>
<td>0.9771</td>
<td>0.9173</td>
</tr>
<tr>
<td>Fault #2</td>
<td><strong>0.9999</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
<tr>
<td>Fault #3</td>
<td>0.9925</td>
<td>0.9994</td>
<td>0.9959</td>
</tr>
<tr>
<td>Fault #6</td>
<td><strong>0.9308</strong></td>
<td>0.9806</td>
<td><strong>0.9551</strong></td>
</tr>
<tr>
<td>Fault #8</td>
<td>0.9572</td>
<td>0.9005</td>
<td><strong>0.9280</strong></td>
</tr>
</tbody>
</table>

Table 5.2: Precision, Recall, and F score for detection and localization

<table>
<thead>
<tr>
<th>Model</th>
<th>State Estimation + DNN</th>
<th>RNN + DNN</th>
<th>Autoencoder + RNN + DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F Score</td>
</tr>
<tr>
<td>Normal</td>
<td>0.7330</td>
<td>0.9699</td>
<td>0.8350</td>
</tr>
<tr>
<td>Attack</td>
<td>0.9778</td>
<td>0.7199</td>
<td>0.8292</td>
</tr>
<tr>
<td>Fault</td>
<td><strong>0.9776</strong></td>
<td>0.9885</td>
<td><strong>0.9830</strong></td>
</tr>
<tr>
<td>Attack #2</td>
<td>0.9962</td>
<td>0.9468</td>
<td>0.9709</td>
</tr>
<tr>
<td>Attack #3</td>
<td><strong>0.9999</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
<tr>
<td>Attack #6</td>
<td>0.9999</td>
<td><strong>1.0000</strong></td>
<td>1.0000</td>
</tr>
<tr>
<td>Attack #8</td>
<td>0.9460</td>
<td>0.9961</td>
<td>0.9704</td>
</tr>
<tr>
<td>Fault #2</td>
<td>0.9999</td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
<tr>
<td>Fault #3</td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
<tr>
<td>Fault #6</td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
<tr>
<td>Fault #8</td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
</tbody>
</table>
We also show the $F_1$ scores of the three methods under different levels of attacks and faults in Fig. 5.6. In general, the proposed method has a better performance than the other two, especially for attacks on bus #2. Another inspection is that the $F_1$ score increases as the level (severity) of the attack increases. This is a validation that the covertness (the ability to stay undetected) of the attack decreases as the severity of attack increases, meaning the distinction between normal data and the data under attack becomes clearer as the severity of attack increases, leading to a higher detection power and diagnosis accuracy. Moreover, it can be seen that the performance of all three methods generally depends on the connectivity of the attacked node – Attacks on bus #2 and #6 has lower detection rates because they have more neighbors. Since we designed the framework based on a steady-state simulation, the temporal correlation in the data only relies on the temporal correlation in the load profile. In the scenario where the attacker slowly changes the state, the method is able to detect the attack with similar accuracy once it reaches the tested severity level. This is because, during the attack, the attacker’s manipulation of the sensor readings are always based on the normal input (Eq. (5.15), rather than the strategy adopted to alter the state (either slowly or abruptly).

To evaluate the performance of the proposed method when encountering attack levels that are not in the training set, we use a subset of the attack levels to train the DNN and test it on the other levels. We test the method by selecting $l$ ($l = 1, 2, 3, 4$) levels among the five levels simulated. For each $l$, we replicate the training and testing 20 times. Within each replication, the attack levels for training are randomly selected. The boxplot of testing accuracy is shown in Fig. 5.7. The result shows that the proposed method has a relatively high accuracy when the model is trained on more than 2 levels of attacks. Meanwhile, as more attack levels are used in training the model, the testing accuracy increases, and the variance of the accuracy decreases. When the testing levels are out of the range of the training levels, the performance is worse (e.g., the outliers for $l = 4$ are corresponding to the replications where the model is trained with levels 2-5 and tested on level 1). This is
because if the data is trained on a stronger attack, the data from a weaker attack would lie in between the clusters of normal data and the attack data, and hence the DNN would have difficulty classifying these data.

Figure 5.6: F1 score under different levels of attack for the three models

Figure 5.7: The boxplots of accuracy under different number of training levels
5.5 Conclusion

In this chapter, we proposed a generic data-driven framework for detecting, diagnosing, and localizing covert attacks on industrial control systems. The proposed framework uses an autoencoder for unsupervised feature extraction from sensor measurements and then uses an RNN to capture the temporal correlations among the encoded sensor measurements. The prediction of the RNN is decoded and compared with the input sensor measurements to get the residuals, which help detect the anomaly. The residuals and the sensor measurements are processed with a DNN to determine whether an observation is representing normal conditions, an attack, or a fault, and to identify the location of the attack/fault. The proposed framework was compared with the model-based state estimation technique, as well as a modification of itself by removing the autoencoder. The results showed that the autoencoder helps extract the important features from the data, as well as reduce the dimension of the input to the RNN. This significantly helps improve the classification accuracy of the DNN. It is worth noticing that the RNN does not provide a more accurate estimation of the state compared to the model-based state estimation. However, since the RNN considers the temporal behavior of the system, which is not considered by the model-based SE, the residuals obtained from the decoded RNN prediction could better capture the anomalous characteristics of the data when the system is under the attacks/faults. The reason model-based SE does not perform well under covert attack is because the objective of SE is to minimize the residuals. This leverages the estimation of the state to normal conditions, which does not represent the underlying truth, especially when the attacker has access to more sensors. The simulation study and performance evaluation validated the proposed method. Since this method is model-free, it is easily generalizable to other networked industrial control systems.

In this work, we only trained and tested the model on the known attack/fault types. A future direction is to extend the method to anomaly-based detection, which can detect novel
attacks and faults. It is also interesting to consider the scenarios of more complex attack strategies (e.g., when the attacker knows the detection network) and study the robustness of the proposed method. Another direction is to combine the method with graphical network topology, as well as the correlation structure of the data.
CHAPTER 6
CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

This thesis focuses on fundamental developments aimed at creating new data-driven methodologies and algorithms that are specifically tailored for securing modern ICSs in IoT-enabled industrial ecosystems. In chapters 2 and 3, we developed system models for idealized industrial settings and studied the theoretical properties of the system under cyber-attacks, equipment faults, controller faults, and degradation. This was achieved by investigating and formally characterizing the properties of the data that was generated by the system’s control variables and condition monitoring sensors. In chapters 4 and 5, we expanded this investigation to encompass more complex settings involving large networked systems like the power grid.

In chapter 2, the objective was to detect replay attacks on a SCADA system and distinguish it from natural equipment faults. In a replay attack, the attacker alters the control actions and disguises this manipulation by replaying the sensor measurements from normal operations. The manipulations of both the control actions and the sensor measurements can successfully bypass many conventional detection schemes. Furthermore, malicious behavior is intentionally disguised as an equipment fault due to the replay of sensor measurements. To address these challenges, we developed a data-driven framework to detect replay attacks and differentiate it from some common types of equipment faults, including sensor fault, controller fault, plant fault, and plant degradation. The system’s behavior was modeled as a state-space model coupled with a Kalman filter and a linear-quadratic controller. We set up a theoretical framework to derive the characteristics of the data unique to each anomalous event and used the same framework to also derive the corresponding statistical
metrics that can be used to monitor and differentiate these events. These metrics were monitored using an ensemble modeling framework with unique signatures for each anomalous event. This approach was implemented on a physical rotating machinery testbed where the rotational speed was the control variable, and vibration was the condition monitoring variable. We tested a replay cyberattack on the rotational speed and successfully distinguished the attack from a fault in the tachometer (speed sensor), controller fault, and bearing fault and degradation.

In chapter 3, we focused on covert attacks. In contrast to replay attacks, covert attacks are much more sophisticated and difficult to detect. In a covert attack, the attacker is assumed to have sufficient knowledge of the system behavior, aside from full access to the sensors and the controller. This enables attackers to fully disguise the manipulation of control actions by generating the expected sensor measurements calculated based on their knowledge of normal system operations. Covert attacks are, therefore, practically undetectable. On this topic, we developed a model-based methodology that utilizes the characteristics of partial degradation in the system to detect covert attacks. We derived a generic design of covert attacks based on the linear dynamic model, which was shown to be undetectable when the attacker acquires full knowledge of the normal system operations. We assumed that the characteristics of partial degradation of the equipment would be unknown to the attacker. Consequently, covert attacks aimed at accelerating equipment failures can potentially be identified by detecting changes in equipment degradation rates. We developed two sequential likelihood ratio testing algorithms to identify the onset of equipment degradation and that of a covert attack. The algorithms were based on theoretical derivations describing the properties of the system residuals under different conditions (normal, degradation, and attack). The impact of system dynamics and the severity of the attack on the detection delay was investigated through a simulation study. We also proved the applicability of the method on the same rotating machinery setup where a thrust bearing was run to failure.
In Chapter 4, we considered a power system network of generator and load buses that was divided into multiple regions. The generator buses in each region are controlled locally by the regional control centers (RCC), and all the buses and RCCs are monitored by the centralized independent system operator (ISO). We developed a detection and localization framework based on data dependencies that are defined by the physical connections between a generator bus and its neighbors. We studied the system characteristics when a generic covert attack is applied to one of the generator buses. We assumed the attacker can manipulate the sensors associated with the target generator bus only. In this case, the residuals of state estimation characterize the sparse features corresponding to the group of sensors associated with the neighbors of the targeted bus. Based on the sparsity, Sparse Group Lasso was used to estimate the potential bias in the system state induced by the attack, which was used to identify the location and severity of the cyberattack. The proposed method demonstrated shorter detection delay and around 20% higher localization accuracy than the baseline bad-data-detection and grouped hypothesis testing.

We then extended this problem to large-scale power networks in Chapter 5. Machine Learning methods provide a very strong alternative in large-scale systems where physics-based models can no longer be a viable option for modeling the network. In this chapter, we used a deep learning approach to detect and localize cover attacks. we used an autoencoder for unsupervised dimension reduction, which makes the framework free from physical models, and useful for other applications. The dynamic behavior of the system was modeled by a recurrent neural network (RNN) with a long-short term memory layer based on the coded measurements. The residuals (differences between the RNN predictions and the extracted features) were then decoded and fed to a feedforward perceptron, where the output is a classification of the type of anomaly (normal/attack/fault) along with its location information. The proposed deep learning framework improved the detection precision from 73% to 99% compared to a physics-based benchmark method.
6.2 Future Research

In the long-term, I would like to expand the scope of my CPS cybersecurity framework to encompass other critical applications, such as smart manufacturing, healthcare, and national infrastructure systems (water, gas, oil, chemicals). These systems have their own modeling challenges and unique cyber-vulnerabilities. For example, the smart process manufacturing industries (including oil refinery, water supply, food processing, etc.) comprise a high level of automation where a cyberattack at any part of the process can propagate very quickly and have very disruptive consequences. In healthcare systems, cyberattacks on hospital infrastructures (such as energy supply, HVAC, gas and vacuum) can seriously affect the normal operations of important facilities such as ICU, surgical rooms, blood product refrigerators, and medical laboratories. Without these facilities functioning normally, the patients’ health and lives are put at risk. Therefore, detecting cyberattacks on the hospital infrastructure networks is critical. My current research has a prospective in these applications.

My future research plan will focus on the generalization, automation, and computational scalability of models and algorithms for real-time cybersecurity of the CPS applications discussed above. This plan requires understanding the domain of each CPS application, identifying and developing appropriate system modeling frameworks, and integrating that with fundamental statistical Machine Learning and Artificial Intelligence tools. The overarching goal of my plan is to automatically comprehend system behaviors, screen for emerging/anomalous events, and optimize control actions and decisions. Achieving this plan will involve fundamental research in modeling, data analytics and data privacy, and computational scalability.

I am also interested in studying the strategic response to cyberattacks, specifically automated recovery and mitigation. Based on my PhD research in localization of cyberattacks in power networks, data analytic approaches can be utilized to enhance system recovery.
Specifically, data analytics can be used to induce suitable corrections in the control actions to mitigate the impact of the cyberattack. This approach applies to systems with well-developed physics-based models. In more complex scenarios where physical models cannot be used to retrieve the change in the system state, I plan to explore using reinforcement learning to design fault-tolerant control models to mitigate the effects of cyberattacks.
Appendices
APPENDIX A
SUPPLEMENTARY MATERIALS OF CHAPTER 2

A.1 Proof of Theorem 2.3.1

Lemma A.1.1 The residuals $\gamma_k$ generated by the second model is a function of $\gamma_{k-1}$ and measurements $y_{k,u}$ and $y_{k-1,u}$ in the following form:

$$\gamma_k = C_u A_u C_u^{-1} (I - C_u \tilde{K}) \gamma_{k-1} + y_{k,u} - C_u A_u C_u^{-1} y_{k-1,u} \quad (A.1)$$

Proof. By Eq.(2.12), we have

$$\hat{s}_{k|k-1} = C_u^{-1} (y_{k,u} - \gamma_k) \quad (A.2)$$

Substitute Eq.(2.5) into Eq.(2.10), and we get

$$\hat{s}_{k|k-1} = A_u \hat{s}_{k-1|k-2} + A_u \tilde{K} \gamma_{k-1}. \quad (A.3)$$

Substitute Eq.(A.2) for $\hat{s}_{k-1|k-2}$ into Eq.(A.3), and we get

$$\hat{s}_{k|k-1} = A_u C_u^{-1} (y_{k-1,u} - \gamma_{k-1}) + A_u \tilde{K} \gamma_{k-1}. \quad (A.4)$$

Finally, substituting Eq.(A.4) into Eq.(2.12), we have

$$\gamma_k = y_{k,u} - C_u A_u C_u^{-1} (y_{k-1,u} - \gamma_{k-1}) - C_u A_u \tilde{K} \gamma_{k-1}$$

$$= C_u A_u C_u^{-1} (I - C_u \tilde{K}) \gamma_{k-1} + y_{k,u} - C_u A_u C_u^{-1} y_{k-1,u} \quad \square$$
Under normal conditions, based on the assumption that $x_{trg} = 0$ and the properties of Kalman filter, we have $E[r_k] = 0$ and $E[\gamma_k] = 0$, where $x_{trg}$ is the vector of the target states of the controllable variables. Based on our assumption of time-invariant system dynamics, the covariance matrices, $\Sigma^k_r$ and $\Sigma^k_\gamma$ are constants.

Define $\delta_k \equiv \tilde{y}_k - y_k$, then $E[\delta_k] = 0$. Denote the elements of a vector $v$ corresponding to controllable and uncontrollable variables as $[v|\cdot]$ and $[\cdot|v]$, respectively. Define the $\Delta$ operator as the difference between the value under an anomaly and the value under normal operations, i.e. $\Delta z_k = z'_k - z_k$ for any value $z_k$.

Now assume that a replay attack occurs at time $t$. Based on Eq.(2.13-2.15), we have:

$$
\Delta x_{t+1} = Ba_t,
$$

$$
\Delta y_{t+1} = \Delta r_{t+1} = [\delta_{t+1}|CBa_t] = [\delta_{t+1}|0].
$$  \tag{A.5}

Note that $(CBa)_u = 0$ for any vector $a$ since $B_1 = B_2 = B_u = 0$, and matrix $C$ is diagonal.

It can be shown by induction that for $i \geq 1$ that,

$$
\Delta r_{t+i+1} = [\delta_{t+i+1} - CBL \sum_{j=1}^{i}(A + BL)^{i-j}K\Delta r_{t+j}] C \sum_{j=0}^{i-1} A^{i-j}Ba_{t+j} + C \sum_{m=1}^{i-1} A^{m}BL \sum_{j=1}^{i-1-m} (A + BL)^{i-m-j}K\Delta r_{t+j}
$$

$$
\sum_{j=1}^{i} (A + BL)^{i-j}K\Delta r_{t+j}] - CA \sum_{j=1}^{i} (A + BL)^{i-j}K\Delta r_{t+j}
$$  \tag{A.6}

$$
\Delta y_{t+i+1} = [\delta_{t+i+1}|C \sum_{j=0}^{i-1} A^{i-j}Ba_{t+j} + C \sum_{m=1}^{i-1} A^{m}BL \sum_{j=1}^{i-1-m} (A + BL)^{i-m-j}K\Delta r_{t+j}]
$$

\tag{A.7}
• Case 1: Using Eq.(A.5-A.6), the following is true:

\[
\mathbb{E}[\Delta r_{t+1,C}] = \mathbb{E}[\Delta r_{t+1}] = 0,
\]

\[
\mathbb{E}[\Delta r_{t+i+1,e}] = -C_e(A_e + B_e L_e) \cdot \sum_{j=1}^{i} (A_e + B_e L_e)^{i-j} K_e \mathbb{E}[\Delta r_{t+j,e}]
\]

\[
\mathbb{E}[\Delta r_{t+i+1,u}] = C_u A_u \sum_{j=1}^{i} [(A + BL)^{i-j}]_u K_u \Delta r_{t+j,u}.
\]

Therefore, \(\mathbb{E}[\Delta r_{t+i,C}] = 0\), and \(\mathbb{E}[\Delta r_{t+i,u}] = 0\) \(\forall i > 0\). Using Eq.(A.7) we can show that \(\mathbb{E}[\Delta y_{t+i,u}] = 0\) \(\forall i > 0\). By Lemma A.1.1, \(\mathbb{E}[\Delta \gamma_{t+i}] = 0\) \(\forall i > 0\).

As mentioned earlier, we assume independence between \(y_k\) and \(\tilde{y}_k\). Therefore, from Eq.(A.5), we have \(Cov(y'_{t+1,e},y'_{t+1,u}) = Cov(\tilde{y}_{t+1,e}, y_{t+1,u}) = 0\). On the other hand, \(Cov(y_{t+1,e}, y_{t+1,u}) \neq 0\) since \(Q_1 \neq 0\). Therefore, we have \(Cov(y'_{t+1,e}) \neq Cov(y_{t+1,e})\).

Similarly, we can show \(Cov(y'_{t+i}) \neq Cov(y_{t+i})\) for \(i > 1\). Since \(r'_{t+1} = r_{t+1} + [\delta_{t+1}]_0\), we have

\[
Cov(r'_{t+1,e}, r'_{t+1,u}) = Cov(r_{t+1,e}, r_{t+1,u}) + Cov(\tilde{y}_{t+1,e}, r_{t+1,u}) - Cov(y_{t+1,e}, r_{t+1,u}).
\]

By the independence between \(\tilde{y}_k\) and \(y_k\),

\[
Cov(\tilde{y}_{t+1,e}, r_{t+1,u}) = 0.
\]

By Eq.(2.6), \(Cov(y_{t+1,e}, r_{t+1,u}) \neq 0\).

Hence, \(Cov(r'_{t+1}) \neq Cov(r_{t+1})\). Similarly, we can show \(Cov(r'_{t+i}) \neq Cov(r_{t+i})\) for \(i > 1\).

• Case 2: As is shown above, \(\mathbb{E}[\Delta r_{t+i,C}] = 0\) \(\forall i > 0\). On the other hand, since \(A_1 \neq 0\) and \(\mathbb{E}[a_{k,e}] \neq 0\), we have \(\mathbb{E}[(A^{i-j} B a_{t+j})_u] \neq 0\). Therefore, \(\Delta r_{k,u} \neq 0\). By Eq.(A.7), \(\mathbb{E}[\Delta y_{k,u}] \neq 0\). Hence, by Lemma A.1.1, we have \(\mathbb{E}[\Delta \gamma_{k+i}] \neq 0\). Similar to case 1,
we can show that $\text{Cov}(y'_k) \neq \text{Cov}(y_k)$, and $\text{Cov}(r'_k) \neq \text{Cov}(r_k)$ for $k > t$.

### A.2 Proof of Theorem 2.3.2

Using Eq.(2.16), at time $t$ we have $u'_t = u_t + a_t$.

Substitute the above to Eq.(2.4), we get

$$\hat{x}'_{t+1|t} = \hat{x}_{t+1|t} + B a_t. \tag{A.8}$$

From Eq.(A.8) and Eq.(2.6) we get

$$\hat{x}'_{t+1|t+1} = \hat{x}_{t+1|t+1} + B a_t. \tag{A.9}$$

Substitute Eq.(A.9) into Eq.(2.16), we arrive at

$$u'_{t+1} = u_{t+1} + L B a_t + a_{t+1}.$$  

By induction, we can show the following:

$$\Delta x_{t+i} = \Delta \hat{x}_{t+i|t+i-1} = \sum_{j=0}^{i-1} (A + BL)^{i-j-1} B a_{t+j},$$

$$\Delta r_{t+i} = 0. \tag{A.10}$$

$$\Delta y_{t+i} = \sum_{j=0}^{i-1} C(A + BL)^{i-j-1} B a_{t+j}. \tag{A.11}$$

- **Case 1:** By Eq.(A.10), we have $\mathbb{E}[\Delta r'_k] = 0$ and $\text{Cov}[r'_k] = \text{Cov}[r_k]$. Since $\mathbb{E}[a_{k,u}] = 0$, by Eq.(23), we have $\mathbb{E}[\Delta y_{k,u}] = 0$. Therefore, by Lemma A.1.1, $\mathbb{E}[\Delta \gamma_k] = 0$. By
Eq.(A.11), the following is true:

\[
\text{Cov}(y'_{t+i}) = \text{Cov}(y_{t+i}) + \text{Cov}(\sum_{j=0}^{i-1} C(A + BL)^{i-j-1} Ba_{t+j}),
\]

\[
E[\Delta y_{t+i}] = \sum_{j=0}^{i-1} C(A + BL)^{i-j-1} B E[a_k].
\]

If Cov(a_k) ≠ 0, Cov(y'_{t+i}) ≠ Cov(y_{t+i}) for i > 0. If Cov(a_k) = 0, a_k = a ≠ 0, then E[\Delta y_{t+i}] ≠ 0, the \(\chi^2\) detector also detects the mean shift in \(y_k\).

- Case 2: As is shown above, we have \(E[r'_{k}] = E[r_k], \text{Cov}[r'_{k}] = \text{Cov}[r_k]\), and \(\text{Cov}(y'_k) \neq \text{Cov}(y_k)\) (or \(E[y'_k] \neq E[y_k]\)) for \(k > t\). By Eq.(A.11), \(E[\Delta y_{k,u}] = C_u \sum_{j=0}^{i-1} ((A + BL)^{i-j-1})_1 B E[a_{k+j,c}]\). Since \(A_1 \neq 0\), \(E[\Delta y_{k,u}] \neq 0\). Therefore, by Lemma A.1.1, \(E[\Delta \gamma_k] \neq 0\).

### A.3 Proof of Theorem 2.3.4

**Lemma A.3.1** Given that the noise terms \(v_k\) and \(w_k\) are unaffected by any system anomaly, if there is no further bias term introduced by the anomaly, the change in residuals \(\Delta r_{k+1}\) can be expressed recursively as:

\[
\Delta r_{k+1} = CA(I - KC)C^{-1}\Delta r_k. \tag{A.12}
\]

**Proof.** From Eq.(5.5-5.6) we get

\[
\Delta y_{k+1} = C \Delta x_{k+1} = CA \Delta x_k + CB \Delta u_k. \tag{A.13}
\]
From Eq.(2.4-2.5) we get

\[ \Delta \hat{x}_{k+1|k} = A \Delta \hat{x}_{k|k} + B \Delta u_k, \]
\[ = A(\Delta \hat{x}_{k|k-1} + K \Delta r_k) + B \Delta u_k. \quad (A.14) \]

From Eq.(2.6) we get

\[ \Delta r_{k+1} = \Delta y_{k+1} - C \Delta \hat{x}_{k+1|k}. \quad (A.15) \]

Substitute Eq.(A.13) and Eq.(A.14) into Eq.(A.15), we get

\[ \Delta r_{k+1} = CA \Delta (\Delta x_k - \Delta \hat{x}_{k|k-1}) - CAK \Delta r_k \]
\[ = CA C^{-1} (\Delta y_k - C \Delta \hat{x}_{k|k-1}) - CAK \Delta r_k \]
\[ = CA C^{-1} \Delta r_k - CAK \Delta r_k \]
\[ = CA(I - KC)C^{-1} \Delta r_k. \]

Lemma A.3.2 Given parameters of the state-space model \( A, C \) and Kalman gain \( K \), each eigenvalue \( \lambda_i \) of matrix \( A(I - KC) \) satisfies \( |\lambda_i| < 1 \), i.e., for any vector \( v \), \( \lim_{k \to \infty} [A(I - KC)]^k v = 0 \).

Proof. Proof of Lemma A.3.2 can be found in [90].

Note that the shift in the system state only occurs at time \( t \), which means there is no difference in the state estimation of KF, i.e., \( \Delta \hat{x}_{t|t-1} = 0 \). Therefore, by Eq.(2.6) and Eq.(2.17) we have

\[ \Delta r_t = \Delta y_t = C \Delta x_t = Ca. \quad (A.16) \]

Since every part of the system is functioning normally, by Lemma A.3.1, we have

\[ \Delta r_{t+i} = C[A(I - KC)]^i a \quad \forall i = 0, 1, ... \]
and by induction, we have:

\[
\Delta r_{t+i} = C[A(I - KC)]^i a \tag{A.17}
\]

\[
\Delta y_{t+i} = C[A(I - KC)]^i a + \sum_{j=1}^{i} C(A + BL)^j KC[A(I - KC)]^{i-j} a \tag{A.18}
\]

- Case 1: By Eq.(A.17) and Lemma A.3.2, we have

\[
\lim_{i \to \infty} E[\Delta r_{t+i}] = \lim_{i \to \infty} C[A(I - KC)]^i a = 0.
\]

i.e., \(E[\Delta r_k] \neq 0\) for limited time steps. Similarly, by Eq.(A.18), we have \(E[\Delta y_k] \neq 0\) before the mean shift converges to a constant. Before the mean shift in \(r_k\) converges to 0, it is detected in the corresponding variables since \(A\) and \(K\) are diagonal. That is, if \(a_u = 0, a_c \neq 0\), then \(E[\Delta r_{k,u}] = 0, E[\Delta r_{k,c}] \neq 0\), and vice versa. If \(E[a_u] \neq 0\), by Eq.(A.18), \(E[\Delta y_u] \neq 0\), and by Lemma A.1.1 we have \(E[\Delta \gamma_k] \neq 0\).

- Case 2: Similarly, by Lemma A.3.2 and Eq.(A.17-A.18), we have \(E[\Delta r_k] \neq 0\) and \(E[\Delta y_k] \neq 0\) for limited time steps. The \(\chi^2\) detection schemes for monitoring \(Cov(r_k)\) and \(Cov(y_k)\) both detect this change. When the fault occurs in a controllable variable, i.e. when \(a_u = 0, a_c \neq 0\), we have

\[
E[\Delta r_{t+i,u}] = C_u A_1((I - KC)a)_c \neq 0,
\]

since \(A_1 \neq 0\). Similarly, \(E[\Delta y_{t+i,u}] \neq 0\), and by Lemma A.1.1, \(E[\gamma_{t+i}] \neq 0\). When the fault occurs in an uncontrollable variable, as is shown above, we have \(E[r_{k,c}] = 0\), \(E[r_{k,u}] \neq 0\) and \(E[\gamma_k] \neq 0\).
A.4 Proof of Theorem 2.3.3

Consider the case when only \( a_t \) is added to measurement \( y_t \) and no additional error is added for the following steps. Notice that \( a_t \) impacts the dynamics of the system at step \( t \). By Eq.(2.18) and Eq.(2.12) we get \( \Delta r_t = a_t \).

For the following steps, if the system operation and control remains normal, the bias term at time \( t + i \) caused by the initial fault can be calculated using Lemma A.3.1:

\[
\Delta r_{t+i} = CA(I - KC)C^{-1}\Delta r_{t+i-1} \\
= [CA(I - KC)C^{-1}]^{i-1}\Delta r_t \\
= C[A(I - KC)]^{i-1}AKa_t
\]

If the system is under attack for more steps, the bias introduced from all the previous steps remain the same. The actual bias introduced by the sensor fault is the summation of the biases from all the previous steps, which can be calculated as follows

\[
\Delta r_{t+i} = a_{t+i} - \sum_{j=0}^{i-1} C[A(I - KC)]^{i-j-1}AKa_{t+j}
\]  
(A.19)

and one can show by induction that

\[
\Delta y_{t+i} = a_{t+i} + \sum_{j=0}^{i-1} \sum_{m=0}^{i-j-1} C(A + BL)^m. \\
BL[(I - KC)A]^{i-j-m-1}Ka_{t+j}
\]

Hence, when \( a_k = a \) for all \( k > t \), \( \Delta y_k \neq 0 \) for all \( k > t \); and when \( a_k \overset{iid}{\sim} N(0, \Sigma_a) \), then the covariance matrix \( \Sigma_y \neq \Sigma_y' \). Similarly, \( \Sigma_r' \neq \Sigma_r \).

- Case 1: When \( A \) and \( K \) are both diagonal, by (26), \( \Delta r_{k,c} \neq 0 \) when \( a_{k,c} \neq 0 \) and \( \Delta r_{k,u} \neq 0 \) when \( a_{k,u} \neq 0 \). Similarly, by (27), \( \Delta y_{k,c} \neq 0 \) when \( a_{k,c} \neq 0 \). Hence, by
Lemma A.1.1, if the sensor fault is in an uncontrollable variable, both $E[\Delta r_{k,u}]$ and $E[\Delta \gamma_k] \neq 0$.

- Case 2: When $A_1 \neq 0$, this means when $a_{k,c} \neq 0$, then $r_{k,c}$ and $r_{k,u} \neq 0$. By (27), since $B_2 = 0$, we have $y_{k,u} = 0$ when $a_{k,u} = 0$. Hence, by Lemma A.1.1, $E[\Delta \gamma_k] = 0$ for $k > t$. On the other hand, when $a_{k,u} = 0, a_{k,c} \neq 0$, then $y_{k,u} \neq 0$ and $E[\Delta \gamma_k] \neq 0$ for $k > t$

A.5 Proof of Theorem 2.3.5

We showed in proof of Theorem 2.3.3 that the change of residual at time $t + i$ is the summation of the change introduced each error from step $k$ to $t + i$. The coefficient of $a_{t+j}$ in $r_{t+i}'$ is equal to the coefficient of $a_k$ in $r_{t+i-j}$. By Lemma A.3.1, we have

$$\Delta r_{t+i} = C \sum_{j=0}^{i} [A(I - KC)]^{i-j}a$$

(A.20)

where $a_u \neq 0$. Therefore, the plant degradation can be detected by monitoring the mean shift in the residual $r_{k,u}$ and the covariance matrix $\Sigma_r$. Similarly, we have:

$$\Delta y_{t+i} = \sum_{j=0}^{i} C[A(I - KC)]^{i}a + \sum_{j=0}^{i} \sum_{m=1}^{i-j} C(A + BL)^m KC[A(I - KC)]^{i-j-m}a$$

Hence, the plant degradation can be detected by monitoring the mean shift in the residual $\gamma_k$ and the covariance matrix $\Sigma_y$. When $A$ and $K$ are both diagonal, by (28) we have $E[\Delta r_{k,c}] = 0$; and when $A_1 \neq 0, E[\Delta r_{k,c}] \neq 0$.

A.6 Simulation Results

A.7 Experimental Results
Figure A.1: Replay Attack

RA-B Case1

RA-N Case1

RA-B Case2

RA-N Case2

Figure A.2: Controller Fault

CF-B Case1

CF-N Case1

CF-B Case2

CF-N Case2
Figure A.3: Plant Fault
Figure A.4: Sensor Fault
Figure A.5: Plant Degradation

<table>
<thead>
<tr>
<th>Signature</th>
<th>MEWMA for $r_{k,c}$</th>
<th>EWMA for $r_{k,u}$</th>
<th>PCA-z for $r_k$</th>
<th>PCA-x for $y_k$</th>
<th>EWMA for $r_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 BA-B</td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>2.2 CE-B</td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>2.3 FF-U</td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>2.4 SC-E-B</td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>

Figure A.6: Experimental results: Test statistics over time under replay attack and equipment faults.
APPENDIX B
SUPPLEMENTARY MATERIALS OF CHAPTER 3

B.1 Proof of Theorem 3.3.1

Proof of Theorem 1

Let $M = A(I - KC)$, $\psi_i = \psi_i(M)$, $\hat{z}_j = D\hat{x}_{r+j|\tau+j}$. Under degradation (starting at $\tau$), the residuals $r_k$ satisfies:

\[
r'_{\tau+i} = r_{\tau+i} + C \sum_{j=1}^{i} M^{i-j}(\kappa + z_j) = r_{\tau+i} + C(I - M)^{-1}(I - M^i)\kappa + C \sum_{j=1}^{i} M^{i-j}z_j
\]

(B.1)

where $z_j = Dx_{r+j} \sim \mathcal{N}(D\hat{x}_{r+j|\tau+j}, DP_{r+j|\tau+j}D^T)$, and $w_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma_w)$ is white sequence.

When $x_{r+j}, j = 1, ..., i$ are estimated using $\hat{x}_{r+j|\tau+j}$, the residuals $r_{\tau+i} \sim \mathcal{N}(\delta_{\tau+i}, \Sigma_{\tau+i})$, where

\[
\delta_{\tau+i} = C(I - M)^{-1}(I - M^i)\kappa + C \sum_{j=1}^{i} M^{i-j}\hat{z}_j
\]

\[
\Sigma_{\tau+i} = R + CPC^T + CV\text{ar}\left(\sum_{j=1}^{i} M^{i-j}z_j\right)C^T +
\]

\[
\text{Cov}(r_{\tau+i}, \sum_{j=1}^{i} M^{i-j}z_j) + CC\text{ov}\left(\sum_{j=1}^{i} M^{i-j}z_j, r_{\tau+i}\right)
\]

\[
\Phi_i = \text{Var}\left(\sum_{j=1}^{i+1} M^{i+1-j}z_j\right) = M\Phi_{i-1}M^T + \text{Var}(z_{i+1}) + \sum_{j=1}^{i} \text{Cov}(z_{i+1}, M^{i-j}z_j) +
\]

\[
\text{Cov}(M^{i-j}z_j, z_{i+1})
\]

Since $\text{Cov}(r_{\tau+i}, z_{i-k}) = CA^k\tilde{P}D^T$,
we get $\text{Cov}(r_{\tau+i}, \sum_{j=1}^{i} M^{i-j} z_j) = C \sum_{j=1}^{i-1} A^j \tilde{P}(M^j)^T D^T = \Psi_i.$

### B.2 Proof of Theorem 3.3.2

The expression of $r_{\tau+i}$ is given by simply replacing the term $z_j$ in Eq.(3.15) with $z_j'$. In this study, we assume that $\delta_k = \delta$ for all $k$. Therefore,

$$E[r_{\tau+i} - r'_{\tau+i}] = E[C \sum_{j=1}^{i} M^{i-j}(z_j' - z_j')] = \sum_{j=1}^{i} M^{i-j} D \sum_{k=0}^{j-1} A^{j-k-1} B \delta$$

Let $\eta_i = E[r_{\tau+i+1} - r'_{\tau+i+1}] - E[r_{\tau+i} - r'_{\tau+i}]$, then we have

$$\lim_{i \to \infty} \eta_i = \lim_{i \to \infty} \sum_{j=1}^{i} M^{i-j}(I - M)D(I - A)^{-1} A^j B \delta = \text{const.} \neq 0 \quad \text{(B.2)}$$

Hence, given $\delta$, $E[r_{\tau+i}'] - E[r_{\tau+i}]$ forms a linear trend with its slope proportional to $\delta$.

### B.3 Derivation of $\hat{\beta}$

Based on our problem setting, the linear trend starting at $\tau$ in $n$ sequential $p$-dimensional vectors $\{x_i\}, i = 1, 2, ..., n$ is defined as $x_i \sim (\mu_i, \Sigma_x)$, where $\mu_i = \mu$ for $i \leq \tau$, and $\mu_i = \mu + (i - \tau)\beta$ for $i > \tau$. We use the maximum likelihood estimator $\hat{\beta}_1$ which is derived below. In our case, we calculate the estimation on the orthogonal space and then project it back to the original space. We first standardize $\zeta_i$, where $x_i = \Sigma_i^{-1/2} \zeta_i$ so $\Sigma_x = I$. Similarly, $\mu = \Sigma_i^{-1/2} \mu_0$, and $\beta_1 = \Sigma_i^{-1/2} \beta$. The derivation for one-dimensional case is given by [61]. Here, the derivation is very similar. The log-likelihood function is given by,
\[
\log L(\tau, \beta_1 | \{x_i\}) = \text{const.} - \frac{1}{2} \left( \sum_{i=1}^{n} x_i^T x_i + \mu^T \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} x_i^T \mu + n \mu^T \mu \right)
+ \frac{1}{2} (n - \tau)(n - \tau + 1)(\beta_1^T \mu + \mu^T \beta_1)
+ \frac{1}{6} (n - \tau)(n - \tau + 1)(2n - 2\tau - 1)\beta_1^T \beta_1
- (\beta_1^T \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} x_i^T \beta_1)).
\]

The MLE is given by taking the partial derivative of the above function with respect to \( \beta_1 \), and the solution is given by,

\[
\hat{\beta}_1 = \frac{6(\sum_{j=i+1}^{\omega} (j - i)x_j - 0.5(\omega - i)(\omega - i + 1)\mu)}{(\omega - i)(\omega - i + 1)(2\omega - 2i + 1)},
\]

and the estimation in the original space is

\[
\hat{\beta} = \frac{6(\sum_{j=i+1}^{\omega} (j - i)\zeta_j - 0.5(\omega - i)(\omega - i + 1)\mu_0)}{(\omega - i)(\omega - i + 1)(2\omega - 2i + 1)}.
\]
### B.4 Simulation Setting

Table B.1: List of Parameters

<table>
<thead>
<tr>
<th>Scenario</th>
<th>3</th>
<th>( {3} )</th>
<th>( \phi )</th>
<th>( \sqrt{m_2} )</th>
<th>( \sqrt{m_1} )</th>
<th>( \sqrt{m_a} )</th>
</tr>
</thead>
</table>
| 1        |   | \[
\begin{bmatrix}
0.14 & 0.35 & 0 \\
0.21 & 0.56 & 0 \\
0.35 & 0.14 & 1-D_{(3,3)}
\end{bmatrix}
\] | \[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.12 & 0.30 & 0.06 \\
0.06 & 0.48 & 0.06 \\
0.30 & 0.12 & 0.54
\end{bmatrix}
\] | \[
\begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\] | 0.336 |
| 2        |   | \[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0.058 & 0.058 & 0.996
\end{bmatrix}
\] | \[
\begin{bmatrix}
0 \\
0.521 \\
0
\end{bmatrix}
\] | \[
\begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\] | 0.329 |

### Simulation Setting

\( \beta \) = \frac{1}{\sqrt{2}} \left( \begin{array}{ccc}
 \frac{\lambda_0}{3} & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \\
 \end{array} \right) \times \frac{\beta}{\sqrt{2}} \left( \begin{array}{ccc}
 \frac{\lambda_0}{3} & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \\
 \end{array} \right)
B.5 Simulation Results

Figure B.1: Simulation results: Scenario 1 with fixed $D$; ARL & CPE vs $m_1$; $m_a$ varying from 0 to 1
Figure B.2: Simulation results: Scenario 2 with fixed $D$; ARL & CPE vs $m_1$; $m_\alpha$ varying from 0 to 1

B.6 Experiment Parameters

$$A = \begin{bmatrix} 0.996 & 0 \\ 2.29 \times 10^{-6} & 0.165 \end{bmatrix}$$

$$B = \begin{bmatrix} 304 \\ 0 \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 0 \\ 7.762 \times 10^{-5} \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 1.036 \times 10^{-7} & 0.435 \end{bmatrix}$$

$$Q = \begin{bmatrix} 3.985 & -1.332 \times 10^{-4} \\ -1.332 \times 10^{-4} & 6.592 \times 10^{-7} \end{bmatrix}$$

$$L = \begin{bmatrix} -2.929 \times 10^{-3} & 0 \\ 0 & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$
REFERENCES


[87] P. Street, “Dataport: The world’s largest energy data resource”, *Pecan Street Inc*, 2015.


VITA

Dan Li was born and raised in Xinjiang, China. She received her B.S. degree in Mechanical Engineering from Tsinghua University, Beijing, China, in 2015, and her M.S. degree in Statistics from Georgia Institute of Technology, Atlanta, in 2020. She joined the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology in 2015 as a masters student, and started her doctoral study in 2016.

Dan’s research interests focuses on cybersecurity for Cyber-Physical Systems and/or Industrial Control Systems. She is also interested in machine learning, sensor-based anomaly detection, complex system modeling, and reliability engineering.

Dan is the recipient of the best student paper award in Energy Systems and the best paper award in Data Analytics and Information Systems (DAIS) at the ISEE Annual Meeting. She is also a finalist of the QCRE and DAIS best student paper competitions at ISEE annual meeting. Dan is an NYU Tandon Faculty First-Look Fellow.

Dan served as a session chair at the 2018 ISEE Annual Conference, and the 2020 and 2021 INFORMS Annual Conference. She also served as a reviewer for the IEEE Transactions on Automation Science and Engineering, Journal of Quality Technology, and the IISE Transactions. Dan is a student member of IEEE, INFORMS, IISE, and ASA.