11-13-2020

Defense by Deception against Stealthy Attacks in Power Grids

Md Hasan Shahriar
Florida International University, mshah068@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

DEFENSE BY DECEPTION AGAINST STEALTHY ATTACKS IN POWER GRIDS

A thesis submitted in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Md Hasan Shahriar

2020

To: Dean John Volakis
College of Engineering and Computing

This thesis, written by Md Hasan Shahriar, and entitled Defense by Deception against Stealthy Attacks in Power Grids, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

A. Selcuk Uluagac

Sumit Paudyal

Mohammad Ashiqur Rahman, Major Professor

Date of Defense: November 13, 2020

The thesis of Md Hasan Shahriar is approved.

Dean John Volakis
College of Engineering and Computing

Andres G. Gil
Vice-President for Research and Economic Development
and Dean of University of Graduate School

Florida International University, 2020

DEDICATION

To my wife, Farah Ulfat, who energized me with unconditional love

and

my brother, A. S. M. Ibnul Hasan Even,

who always set the bars too high and helped me to beat them all.

ACKNOWLEDGMENTS

First of all, I would like to thank my major professor and advisor, Prof. Dr. Mohammad Ashiqur Rahman, for his invaluable guidance, mentorship, and motivation during my graduate studies. I would also like to express gratitude to my thesis committee members, Dr. A. Selcuk Uluagac and Dr. Sumit Paudyal, for their insightful comments, guidance, encouragement, and generous support. I would also like to thank Dr. Badrul Chowdhury from the University of North Carolina at Charlotte for his valuable comments and suggestions on this work. I want to thank my colleagues of Analytic for Cyber Defense (ACyD) Lab for their constant help and encouragement. I also want to acknowledge the support provided by the National Science Foundation, FIU University Graduate School, and the Department of Electrical and Computer Engineering at Florida International University. This thesis is mostly based upon the work supported by the National Science Foundation under Award Number NSF-CRII CNS 1929183. Last but not least, I am immensely grateful to my parents, friends, and family for encouraging me throughout my graduate studies. My wife deserves an accolade for being so cool when I needed her to be. I can thank enough my brother, who provided me with education, inspiration, and guidance.

ABSTRACT OF THE THESIS

DEFENSE BY DECEPTION AGAINST STEALTHY ATTACKS IN POWER GRIDS

by

Md Hasan Shahriar

Florida International University, 2020

Miami, Florida

Professor Mohammad Ashiqur Rahman, Major Professor

Cyber-physical Systems (CPSs) and the Internet of Things (IoT) are converging towards a hybrid platform that is becoming ubiquitous in all modern infrastructures. The integration of the complex and heterogeneous systems creates enormous space for the adversaries to get into the network and inject cleverly crafted false data into measurements, misleading the control center to make erroneous decisions. Besides, the attacker can make a critical part of the system unavailable by compromising the sensor data availability. To obfuscate and mislead the attackers, we propose DDAF, a deceptive data acquisition framework for CPSs' hierarchical communication network. Each switch in the hierarchical communication network generates a random pattern of addresses/IDs by shuffling the original sensor IDs reported through it. During the data acquisition from remotely located sensors to the central controller, the switches craft the network packets by replacing a few sensors' associated addresses/IDs with the generated deceptive IDs and by adding decoy data for the rest.

While misleading the attackers, the control center must retrieve the actual data to operate the system correctly. We propose three remapping mechanisms (*e.g.,* seed-based, prediction-based, and hybrid) and compare their robustness against different stealthy attacks. Due to the deception, artfully altered measurements turn into random data injections, making it easy to remove them as outliers. As the outliers and

the estimated residuals contain the potential attack vectors, DDAF can detect and localize the attack points and the targeted sensors by analyzing this information. DDAF is generic and scalable to be implemented in any hierarchical CPSs network. Experimental results on the standard IEEE 14, 57, and 300 bus power systems show that DDAF can detect, mitigate, and localize up-to 100% of the stealthy cyberattacks. To the best of our knowledge, this is the first framework that implements complete randomization in the data acquisition of the hierarchical CPSs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Cyber-physical systems (CPSs) consolidate sensing, communication, processing, and control of both cyber and physical domains [GPGV14]. They are omnipresent in critical infrastructure such as transportation systems, smart grids, smart health care, sewage/water management, etc. The Internet of things (IoT) has opened up a new dimension to the CPS through phenomenal unification enabling real-time monitoring, data exchange, and optimal control. State estimation (SE) is one of the core parts of the CPSs control system that plays a vital role in secure and appropriate control decisions.

1.1 General Statement of the Problem

CPSs are getting larger with heterogeneous sensor interaction due to widespread acceptance, creating a massive attack space for the adversaries. The contemporary cyberattacks are sophisticated enough to overcome legacy defense tactics like intrusion or anomaly detection systems. Utilizing the targeted system's knowledge and states, the adversaries can launch influential cyber attacks like false data injection (FDI) attacks, covert attacks, zero dynamics attacks, replay attacks, and denial of service (DoS) attacks [YHK$^+$12, RSM19].

Cyber-attacks in CPS are performed in several ways. By eavesdropping on the communication channels, a powerful intruder gains access to the sensor data. Depending on the attack goal, he/she injects some malicious stealthy data into the sensor reading, misleading the SE of CPS to his/her aspired direction. The attacker can also flood the targeted network devices with superfluous requests and overloads

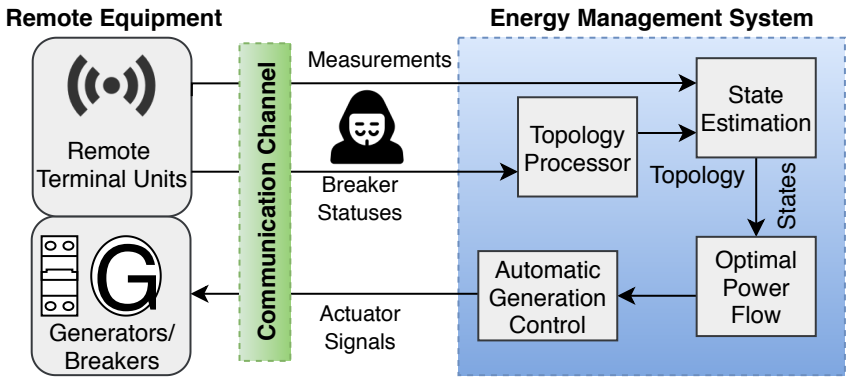


Figure 1.1: A stealthy attack in power grids SCADA network, where an attacker compromises the communication channel to inject malicious data.

the system, which prevents legitimate requests from the control center and makes the overall system unobservable! [RSJM19, NHS+20].

Cyber-attacks not only hamper the services' reliability but also introduce substantial financial losses. Stuxnet, a 500-kilobyte computer worm, attacked over fifteen Iranian facilities in 2010, destroying over a thousand uranium enriching centrifuges [Lan11]. A recent report states that the US government could lose $1 trillion if a Stuxnet-like attack would be carried out in the US smart grid [Dri15]. Again, BlackEnergy Malware, a denial-of-service (DoS) attack on supervisory control and data acquisition (SCADA) system, is responsible for a massive power outage in Ukraine [Kov16]. More recently, some utility companies in Utah, Wyoming, and California power grids also suffered from cyber-attacks that caused power outages in those areas [Sob19].

1.2 Research Purpose and Objectives

In most of the CPSs, the devices in the SCADA networks have limited computational power to perform high-end encryption on the sensor data streams [AQS+15, SPA+18, AFA+20]. Besides, some critical infrastructures (e.g., power systems) have a time

constraint on the data acquisition process. For example, IEC 61850 standard enforces a maximum end-to-end time delay of 4 ms for generic object-oriented substation event (GOOSE) messages within a substation [IEC]. Thus, in some cases, despite having sufficient computational capability, high-end security may not be applied due to the time-restriction. Hence, several studies showed, once a sophisticated attacker gains adequate knowledge about the targeted system's topology and states, he/she can easily evade the existing defense mechanisms by utilizing the resources in the generation of the attack data [LNR11, NSRU20].

"All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near.— Sun tzu, The Art of War [TTS+71]". As the author said, deception also plays a vital role in misleading attackers in cyberwar, stopping them from accomplishing their intent. The fundamental goal of our proposed framework is to provide a deceptive, secure, and robust data acquisition framework. As the smart grid is a perfect example of modern CPSs, we consider a smart grid hierarchical communication network as our testbed.

Our design objectives are to ensure the CIA triad (confidentiality, integrity, and availability) for the CPSs data acquisition process, which are described as follows:

- To maintain confidentiality and preserve the systems' privacy from the adversary, we hide the true information of the system and show the artfully crafted sensors data. Such a move misleads the attacker and prevents reconnaissance attacks in the system states.
- To maintain the integrity and continue unperturbed system operation, any malicious data injections need to be removed from the system control loop. Thus, we

aim to mitigate the stealthy FDI injection attacks by identifying the compromised sensors and eliminate them from the SE procedure.

- To ensure the availability of the sensor data and optimize the dynamic behavior of the system in run-time, the system must be observable to EMS. However, in the DoS attacks, some parts of the system might go offline, making the whole system unobservable. Thus, the goal is to predict the missing data and keep the system running around the optimal operation point.
- To detect, mitigate, and localize the attacks, and estimate the states accurately, EMS needs to remap the randomized sensors data. The objective is to design a remapping mechanism for EMS to reproduce the original data sequence while filtering out the compromised sensors as outliers and impute missing data if necessary. Moreover, due to the deception, any stealthy FDI attack attempt turns into a random data injection, which is easily detected and eliminated by the existing bad data detection techniques. As the attacked sensors become outliers, analyzing their residual values reveals the injection data, the targeted sensors, and the attacker's position in the network.

In general, the proposed deceptive framework makes the system resilient to different stealthy attacks and reveals the existence of the attacker.

1.3 Contributions

Cyberattacks are evolving and becoming persistent in modern control systems. We implemented deception as a defense in the networked control systems to defend against stealthy attacks. We designed a deceptive data acquisition framework, called DDAF, for any hierarchical communication system of CPSs that misleads the stealthiest cyberattacks. DDAF includes modeling the hierarchical network as a tree, where the

switches are modeled as nodes. To maximize the deception, we designed a tree-based randomization algorithm that ensures the dispersing and impairing of the targeted stealthy attacks. To reallocate the randomized sensors, we presented three different mechanisms: seed-based, prediction-based, and hybrid. For seed-based remapping, we considered implicit sharing of randomization information using hardware/software tokens. Besides, we presented the attack detection and localization algorithm utilizing the residual data from state estimation. We modeled and trained a long-short term memory (LSTM) model for prediction-based remapping. Finally, we designed a hybrid model combining the first two mechanisms and evaluate all of them on standard IEEE bus systems.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows: We introduce the problems and our contribution in Chapter 1. We present the system models in Chapter 2. The related works and background information are discussed in Chapter 3. We introduce our proposed DDAF in Chapter 4. Chapter 5 discusses the technical details of the deception mechanism. Chapter 6 explains three types of remapping mechanisms along with example case studies. We describe the implementation of DDAF in Chapter 7. The evaluation setup and result analysis are formulated in Chapter 8. At last, we conclude the thesis and discuss the future work in Chapter 9.

CHAPTER 2

SYSTEM AND THREAT MODELS

This chapter describes the system and threat models considered for the implementation and evaluation of DDAF.

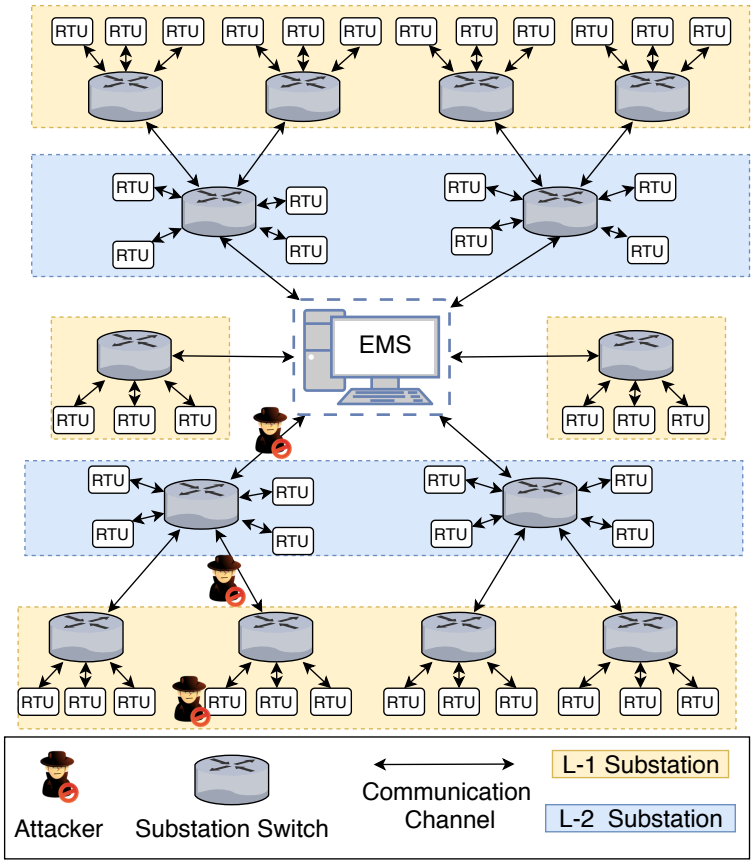


Figure 2.1: A two-level hierarchical communication network in power grids.

2.1 System Model

Future CPS networks are supposed to maintain a hierarchical communication structure as it lessens the communication overhead and assures the system's stability, reliability, and efficiency [WZYC16]. With the increasing trend in distributed generation, renewable energy, and electricity demands, smart grid hierarchical communication

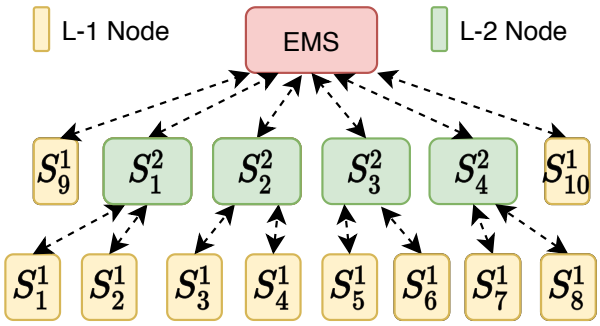


Figure 2.2: Tree diagram of a two-level hierarchical communication network.

network is becoming a preferable option day by day [LLWC16]. Figure 2.1 shows our considered the hierarchical communication network in a smart grid: a model with two layers of substations switches in the hierarchy.

The sensors in the substations can be remote terminal units (RTUs), intelligent electronic devices (IEDs), phasor measurement units (PMUs), etc. The sensors are located within the substations and directly report the measurement data to their substation switches. Thus, we define the sensors as the layer-0 (L-0) elements of the network. A Layer-1 (L-1) substation only receives measurement data from its own sensors, whereas a Layer-2 (L-2) substation receives data both from its sensors and the underneath L-1 substations. In a two-layered network, L-2 switches directly report data to the EMS. After collecting remote sensor data, the EMS runs the SE algorithm to estimate the system's states and take appropriate control decisions. Such hierarchy forms a tree structure, where the sensors are the leaf nodes, substation switches as internal nodes, and EMS as the root. Figure 2.2 shows the tree structure of the hierarchical network shown in Figure 2.1. Moreover, we define a communication channel based on the layer of the outgoing switch, when forwarding data toward the EMS. Thus, a channel between L-1 and L-2 switches is defined as L-1 channel and the one between L-2 switch and EMS is defined as L-2 channel.

2.2 Threat Model

In this section, we define the threat model according to the attacker's capabilities and goals. Figure 2.3 illustrates the attack tree that we consider in this work. During the data acquisition process, there remain multiple vulnerable points, which an adversary can compromise. The attacker's accessibility/position into the network plays a very crucial role in the attack's success. In general, we classify an adversary's position into two groups. Firstly, the attacker can compromise the individual targeted sensors [BAF20, ALD13, VVP+06]. In this case, the attacker is sophisticated enough to be distributed to the network's edge nodes. Sensors are usually located within a substation, which are highly secured to be physically accessed. Since this type of attack comes with a great cost, it is less common in reality. As the sensors are the leaf nodes network, we define such sensor attacks as L-0 attacks. The second type is compro-

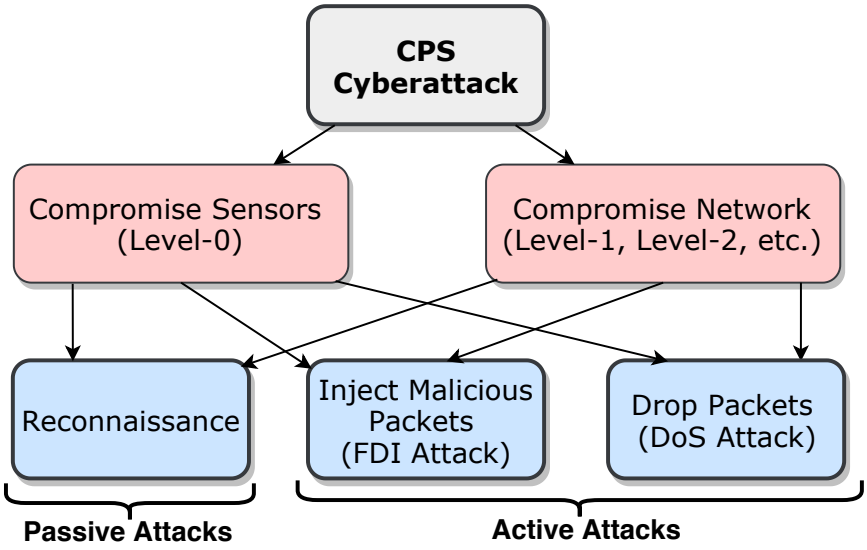


Figure 2.3: Considered attack tree in CPSs.

mising network devices, such as routers, switches, channels, etc. As these elements span the overall SCADA network, physical security is fragile at some points. Moreover, high-end encryption may not be implemented in all the network switches due

to low computational capabilities. Thus, compromising the communication network is more prevalent in the CPSs cyberattacks. However, network switches or routers are mostly in secured locations as they also belong to the substations/local control center. Thus, communication channels remain the most vulnerable points. Therefore, when the attacker compromises the L-1 communication channels, we consider them as L-1 attacks; in L-2 channels, we define them as L-2 attacks.

On the other hand, depending on the attacker's intend, we again categorize the cyberattacks into two classes— active attacks and passive attacks [SSVE04]. Passive attacks are the process of reconnaissance of the system states, where the attacker gets into the network, sniffs, and analyzes the packets without obstructing the system's normal operation. Such an attack aims to study the parameters of the physical system and determine the optimal attack tactics without creating any attention of the defender. A passive attack is dangerous for the confidentiality of the system.

Active attacks are the injections of malicious data into the sensor measurements that help achieve the attacker's goal. Active attacks exploit the integrity as well as the availability of sensor data. The effectiveness/stealthiness of the active attacks depends on the success of the passive attacks. We consider two influential cyberattacks, e.g., FDI attacks and DoS attacks, as the active attacks. Both of them fall within the class of man in the middle attacks. In the first case, the attacker injects malicious data into the network packets to mislead the system in a hazardous direction. In DoS attacks, the attacker drops the targeted packets, leaving the sensors from a critical part unavailable. As a result, the total system becomes unobservable and may collapse due to the delayed response. In this thesis, we evaluate the performance of DDAF, considering all the combinations of cyberattacks as discussed.

The proposed framework is resilient to single-level attacks as well as attacks compromising the sensors at multiple levels. However, we focus only on single-level at-

tacks in the evaluation to limit the number of combinations. The mixed-level attack's impact will be somewhat in between the impact extents of these levels. Another assumption is that there is only one kind of attack at any given time. Thus, the nodes at a stage of the attack tree are considered as OR-nodes. Moreover, we assume that the attacker may have the necessary (complete or partial) knowledge of the bus topology and the transmission lines' electrical properties to compute stealthy attacks. It is also assumed that the attacker does not have the historical time-series sensor data; however, (s)he can access a few of the system nodes to read and compromise the measurement data.

CHAPTER 3

BACKGROUND AND LITERATURE REVIEW

In this chapter, we present a brief background of the terminologies used throughout the paper.

3.1 CPSs Hierarchical Network

The hierarchical network model of CPSs is a way of building a reliable, efficient, collaborative, and economic distributed network [WZYC16]. There exists a significantly large number of nodes and branches at the bottom compared to the upper layers. The control center is located at the root of the network tree and communicates with the edge nodes through the branches. CPSs communication networks mostly comprise a bi-directional communication channel, where nodes of any level can send data to their children and vice-versa. Throughout this paper, we use the terms layer and level interchangeably.

3.2 State Estimation and Bad Data Detection

State estimation (SE) is one of the core parts of the SCADA system of CPSs, and bad data detection (BDD) is one of the main functions of SE. SE refers to a mathematical algorithm that processes raw measurement data collected from the remote sensors and estimates the system states [JH20, SSU16]. Usually, a power system is an overdetermined system with redundant measurements. Thus, SE and BDD help find the best states by considering only the compliant measurement data and eliminating outliers. In CPSs, if the measurement vector is z and the states vector is x, then their relationship can be presented as

$$z = h(x) + e$$

where, $h(x)$ represent the measurement functions that relate the measurement set to the states vector and e is the set of measurement errors, which are assumed to be uncorrelated and follow normal distribution [AE04]. A general approach to estimate state is Weighted Least Square (WLS) when performing SE based on available measurements. To achieve an optimal solution, WLS formulates SE as a residual minimization problem [WXH$^+$17]. Linear estimation is done with the following equation:

$$\hat{x} = (H^T WH)^{-1} H^T Wz \tag{3.1}$$

In the above equation, $\hat{x}$, H, W represent the most probable system states vector, the Jacobian matrix, and a diagonal weighting matrix, respectively. After system sates are estimated, residuals between measurements and estimated states are used to identify bad data. So, the estimation of measurements, $\hat{z} = h(\hat{x})$ and the measurement residuals set, $r = |z - \hat{z}|$. If we consider that threshold of residual as τ, any measurement z_i is regarded as bad data and removed from the SE procedure if $r_i > \tau$.

In power grids, the measurements are real and reactive line power flows, real and reactive bus power injections, phasor measurement unit (PMU), etc. The states are the voltage magnitudes and angles of the buses. DC power flow is used in contingency analysis due to its simplicity, robustness, and high computation speed, and it simplifies full power flow by looking only at active power [VVP$^+$06]. In DC approximation, all voltage magnitudes are considered to be 1.0 (unity), and reactive power flow is zero. So, the state vector only contains the phase angles of the bus voltages i.e. $x = [\theta]$, where θ is the voltage angles vector. We use $SE - BDD$ to indicate the SE and BDD algorithms together for the rest of the thesis.

3.3 False Data Injection Attacks

In FDI attacks, sensor readings are compromised so that the malicious data comply with the system topology and remain undetected by the BDD. An attack vector is defined as a set of malicious data to be injected into a set of sensor measurements. A random attack vector creates an anomaly, and the compromised sensors become outliers, leaving the system safe. However, if the attack vector is calculated intelligently considering the targeted system's information, it can bypass the BDD [LNR09, XWG+17]. In FDI attacks, the goal of the attacker is to change the state variable $\hat{x}$ to $\hat{x_f}$ by modifying it with a malicious amount of c, where

$$\hat{x_f} = \hat{x} + c \quad \text{and} \quad z_f = z + a$$

To remain stealthy, false data a is maliciously injected into the measurement set z and the observed measuring is z_f. Now if the condition $a = h(c)$ holds, this false data a will be hidden from the traditional BDD, because the residual:

$$r = ||z_f - h(\hat{x_f})|| = ||(z + a) - h(\hat{x} + c)|| = ||z - h(\hat{x})||$$

Thus, the data injection disappears from the residual, and the attack remains stealthy. This attack requires full knowledge of the targeted system's topology, parameters, and measurement configuration. For the rest of the paper, we use the term FDI to refer to the stealthy FDI attacks.

3.4 Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural networks that can learn the temporal behavior of the dataset [DBDJH14]. The nodes in the RNN are connected as a directed graph forming a temporal sequence. RNN can be applied to sequential data prediction such as handwriting recognition, time series anomaly

detection, speech recognition, music composition, etc. Figure 3.1 shows the interconnection of RNN network architecture, where the output of one block becomes the input of the another. An RNN model is formed with a chain of repeating modules of neural networks. The building block of RNN is shown in Figure 3.2. For each time step t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g_1\left(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a\right) \tag{3.2}$$

and

$$y^{<t>} = g_2\left(W_{ya}a^{<t>} + b_y\right) \tag{3.3}$$

where W_a, W_{aa}, W_{ya}, b_a, b_y are weights/coefficients that are shared temporally and g_1, g_2 activation functions. Usually, sigmoid is used as the activation function as followed:

$$g(z) = \frac{1}{1+e^{-z}} \tag{3.4}$$

There are different types of RNN models, depending on the dimension of input and output sequence. In this project, we consider the many to one version, where multiple time-steps of immediate historical data are provided to predict the next batch of data. Despite different benefits, RNN suffers from 'short term memory' and 'vanishing gradient' problems. To handle these issues, we use the version of gated RNN, Long short-term memory.

3.5 Long Short-term Memory

Long short-term memory (LSTM) is an RNN architecture with the capability of learning long-term dependencies of time-series data. In 1997, Hochreiter & Schmidhuber first introduced them, and later many people refined them [HS97,GES02,SM19].

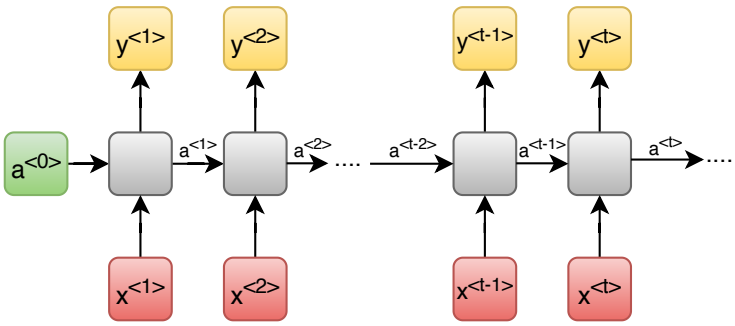


Figure 3.1: Many to many RNN chain

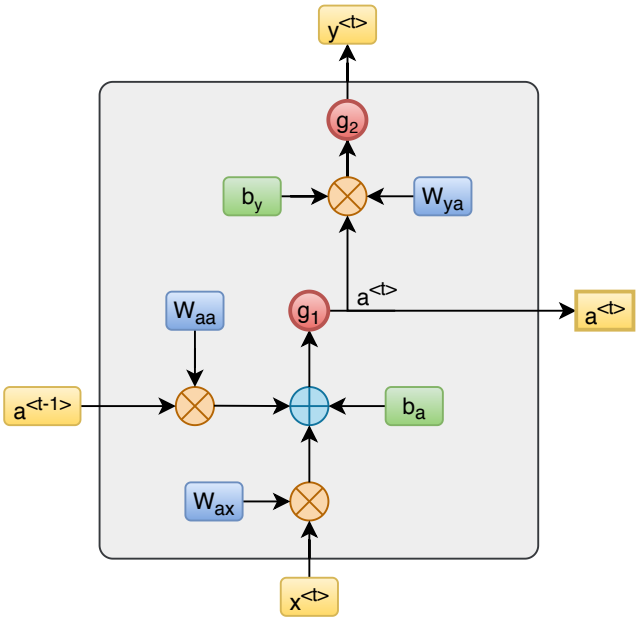


Figure 3.2: A block of RNN chain architecture

LSTM solves the typical RNNs' limitations, where they are designed to hold long term memories. Figure 3.3 shows the flow diagram of a LSTM module. Multiple gates control the information propagation through the cell state, which is the horizontal line running through the top of the diagram. Cell state contains the information c_t, which is controlled by the gates. Gates control the operation of each module, controlling which information should pass and which should not. They contain sigmoid neural network layers and are used in element-wise multiplication operation to regulate the information flow.

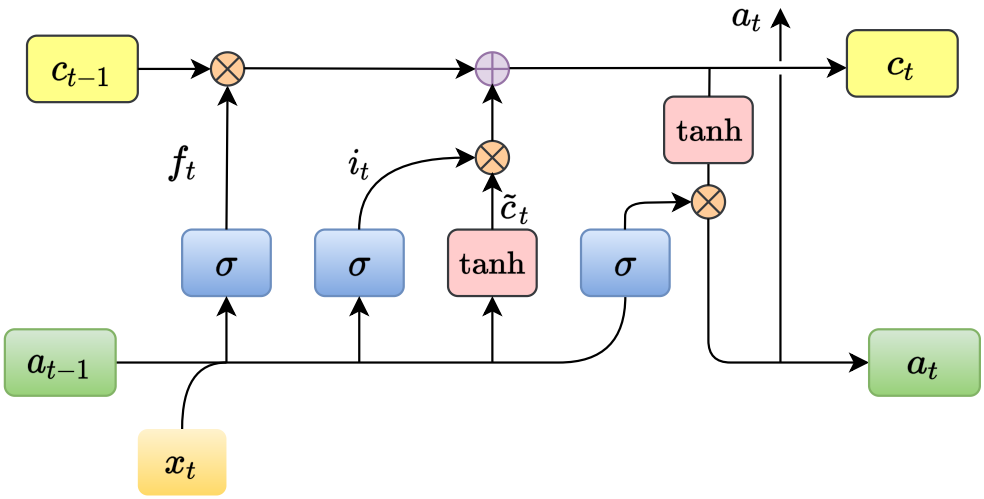


Figure 3.3: A block of LSTM architecture

Among the steps a module takes during the information processing, the first one is to decide what information should be removed from the cell state. This is controlled by the 'forget gate layer.' As shown in the following equation:

$$f_t = \sigma\left(W_f \cdot [a_{t-1}, x_t] + b_f\right)$$

Forget gate f_t outputs a number between 0 to 1 for each of the elements in the cell state c_{t-1}, where 1 means completely allow to keep propagating, and 0 means complete removal. Similarly, the 'input gate layer' controls which elements in the cell state should be amended with information. Besides, a tanh layer outputs $\tilde{c}_t$, the updated cell state vector added to the previous information. The output of 'input gate layer' i_t and new candidate $\tilde{c}_t$ are calculated as follows:

$$i_t = \sigma\left(W_i \cdot [a_{t-1}, x_t] + b_i\right)$$

$$\tilde{c}_t = \tanh\left(W_c \cdot [a_{t-1}, x_t] + b_c\right)$$

Finally, the cell state updated with a portion of the previous information c_{t-1}, and new information $\tilde{c}_t$, controlled by f_t and i_t, respectively.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

Finally, the module's output is also controlled by a layer called the 'output gate layer'. The outcome depends on the updated cell state c_t. First, a sigmoid layer controls which elements of the cell state should go to output. Secondly, a tanh layer determines what the output should be. The following equations determine a_t, the output of the module.

$$o_t = \sigma\left(W_o\left[a_{t-1}, x_t\right] + b_o\right)$$

$$a_t = o_t * \tanh\left(c_t\right)$$

3.6 Literature Review

This section presents the related works that are closely related to the research work presented in this thesis. Moving target defense (MTD) techniques is extensively proposed to defend against different cyberattacks on the sensory channels.

There are several works introducing uncertainty/randomness in the CPSs control loop. Griffioen et al. proposed an MTD introducing stochastic and time-varying parameters in the control loop of the CPS [GWS20]. Giraldo presented MTD by randomly changing the availability of the telemetry sensor data for detecting and minimizing the impacts of FDI attacks [GCS19]. Rahman et al. reduced the window of the attack by adding uncertainty in the sensor being considered in the state estimation process [RASB14]. Zhang et al. showed an MTD mechanism could fail to detect FDI attacks if the number of lines is less than twice the number of states in the power system [ZDY+19]. Hu et al. proposed a stealthy attack detection strategy leveraging skewness coefficients to distinguish the forged residuals from the attack-

free residuals [HLY+19]. Markov model-based games are also useful in the modeling of MTD. Maleki et al. modeled a Markov chain for the interaction between a defender and an attacker [MVK+16]. In some works, authors examined to perturb the considered systems' physical properties to make the attack vectors invalid. Lakshminarayana et al. proposed a formal model for reactance perturbation-based MTD using D-FACTS [LY18]. Tian et al. proposed a hidden MTD using D-FACTs in smart grids to defend structured FDI attacks [TTGL18]. In [LW20], Liu et al. considered the reactance of D-FACTS lines as decision variables that affect the trade-off between the system loss and the MTD effectiveness.

Some researches focused on deceiving the attacker in the MTD with crafty network packets. Li et al. used CPSMorph creating several fake network sessions along with the actual ones to hide them from attackers [LDZ14]. Pappa et al. proposed a seed-based end-to-end IP hopping among trusted peers of a SCADA system [PAG17]. They used the seed to generate random IP and share them through a pre-installed public-private encryption channel. Groat et al. introduced MT6D as a secured IPv6 based smart grid communication system [GDU+12].

Some other research works are also performed for deceptive defense in CPS. McQueen et al. explored deception defenses in control system cybersecurity and categorized them into several dimensions [MB09]. Lin et al. proposed a randomized data acquisition into multiple rounds [LKI18]. The software-defined network-based framework controls the network flows and collects real measurements from randomly selected online sensors and spoof measurements for the rest. However, as only a few sensors send original data at each session, an intelligent attacker may inject false data into the online devices. Even analyzing the pattern of the sensor measurements may give the attacker insight into the correct measurement. In another work, they proposed a physical function virtualization technique along with randomizing and craft-

ing the decoy data to disrupt the reconnaissance attack in power grids [LZHZ20]. However, an attacker can inject FDI attacks into a part of the system, ignoring the rest of the system information. Thus, their proposed approach only secures the system where virtual nodes are placed, leaving some parts unsecured. Moslemi et al. propose a maximum likelihood estimation-based decentralized cyber-attack detection [MMV17]. Ma et al. proposed a zero-sum Markov game to model dynamic interactions between attacker and defender and demonstrated a possible way of defense using deception [MYLR12].

Several works consider data-driven approaches to detect cyberattacks and the imputation of missing data [HSD+20, SHRAJ20, NSRU19]. Ayad et al. present an RNN-based FDI attack detector [AFYES18]. In [YZL+20], Yang et al. proposed a time series analysis method to detect FDI attacks. Li et al. proposed autoencoder and LSTM in FDI attack detection [LHQZ20]. To predict the stability of the power system, Alazab et al. proposed multidirectional LSTM [AKK+20]. LSTM models are also popular in missing data prediction. Verma et al. proposed an LSTM based missing data predictor for health care [VK19]. In [KKC18], Kim presented an LSTM-based daily load forecasting mechanism.

Even though, as mentioned above, there exist a few works on partial randomization of the data acquisition process. However, our proposed framework introduces complete randomization that secures the overall system. Besides, we propose an intelligent remapping that avoids the overhead of an additional communication channel or seed-based sharing. The existing data-driven defense techniques are mostly attack specified and do not provide a complete immunity. However, DDAF is generic and offers a complete solution for the system's robustness against different cyberattacks.

CHAPTER 4

PROPOSED FRAMEWORK

DDAF applies to CPS's hierarchical networked control system. In a SCADA network, the sensors send measurement data from the remote locations to the control center through the intermediate access points/switches. Based on the connectivity, the nodes in the system form a tree, where each node receives the data packets from its child nodes and forwards them to its parent node. Figure 4.1 shows the key features of DDAF.

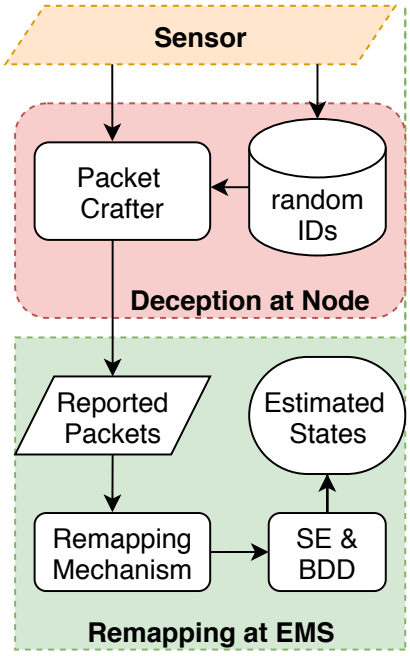


Figure 4.1: Keys features of DDAF.

As the nodes need to analyze and modify the network packets, the "deceiving" nodes are equipped with software-defined networking (SDN) controllers [DKG+14]. An SDN controller can read, edit, and assemble packets in run-time. In modern communication networks, SDN is widely implemented. In case of the unavailability of SDN switches, similar capabilities can also be implemented on the conventional switches by VMware and Nicira [Doh16].

The deception mechanism is implemented at the SDN controller of the nodes. EMS assigns each sensors in any of the three categories: *randomized*, *decoyed*, and *fixed* and share the information with the nodes. Nodes keep track of the assigned sensor types and utilize them during the deception process. As a part of the deception, each node does two routine tasks: i) $recID$-$randID$ pair generation, and ii) packet crafting. A received ID - random ID or $recID$-$randID$ pair contains two sets of IDs. The first one is the received IDs from the child nodes. The received IDs are the set of sensor IDs whose data packets are reported to the node by its child nodes. At a specific interval, the received IDs are shuffled using a randomization algorithm to generate the random IDs. Such a pair of IDs are used to craft the packets during the data acquisition process, explained in the following chapter.

On the other hand, another routine task that a node executes is forwarding the packets. Whereas the conventional nodes forward the packets to the next devices, in DDAF, the SDN controller crafts the packets before forwarding. The packet crafting is done in two steps. Firstly, the nodes replace the received IDs of the *randomized* sensors with the deceived IDs using the node's $recID$-$randID$ pair. Secondly, to support such randomization, the nodes calculate and add decoy data for the sensors in *decoyed*. The process continues at each node until the packets reach EMS. Whereas packet crafting is almost a continuous process, $recID$-$randID$ generation is carried out whenever the deception patterns need to be updated, and it depends on the defender's choice. Thus, to maintain the defense's dynamicity, EMS asks the nodes to update their $recID$-$randID$ pairs at a regular interval.

We propose three reallocation mechanisms to recover the original IDs from the reported packets. They are prediction-based remapping, seed-based remapping, and hybrid remapping. In the prediction-based remapping, a recurrent neural network model is trained on the historical time-series sensor data and used to predict the

expected measurement data. However, as EMS receives randomly shuffled sensor data, we propose a combinatorial optimization algorithm that finds the optimum sequence of recovery to find the correct pattern of the reported data. A measurement is considered in recovered data only if within a specific range of the predicted value in the remapping algorithm. Hence, in the case of stealthy FDI attacks, any sudden change in any measurement would be easily detected by the deviation between the predicted and reported measurement and removed from the recovered data. Such removal eliminates the compromised sensors and makes the system immune to the stealthy FDI attacks.

For the seed-based mechanism, we propose that the randomization algorithm uses a seed value generated using the hardware/software token-based RSA SecurID authentication mechanism. This technique generates a secure authentication code (used as seed) for each node using the built-in clock and the node's factory encoded secret key [sec]. Thus, each node generates a different seed simultaneously and uses it while running the randomization algorithm. Therefore, seed-based remapping does not consider any prediction model; instead, EMS generates the same seeds as the nodes do. As all the nodes are interconnected through physical lines, they can maintain the clock synchronization with EMS. Thus, whenever the nodes generate the seeds, EMS follows the same steps to generate the same seeds using the nodes' secret keys, which are already shared with the EMS during the installation stage. Using the seed values, EMS runs the same randomization algorithm for all the nodes to generate the $recID$-$randID$ pairs and build a stack, called ID-Stack. Using the ID-Stack containing all the randomization information, EMS recovers the original IDs from the collected crafted packets by running a remapping algorithm.

The third remapping mechanism integrates the first two, thus called hybrid remapping. The features of both seed-based and prediction-based algorithms are present

in the hybrid remapping. Seed-based remapping is considered the preferred mechanism as they provide the exact information of the randomization generating the ID-Stack. However, in case of the unavailability of the ID-Stack, the framework goes for prediction-based remapping. In both cases, the predicted measurement values are used to pre-filter the data or impute them if needed. The predictive filtering mitigates the impacts of FDI attacks, and imputation tackles the DoS attacks.

Finally, once the remapping is done, EMS executes the necessary routine tasks (i.e., $SE - BDD$) on the remapped original data and takes control decisions for the system's optimal operation. As the sensor data packets are sent with deceptive IDs, if an attacker injects stealthy false data into some targeted sensors, the injection will happen to the wrong measurements and result in bad data. In the prediction-based technique, most compromised sensors are eliminated during remapping as they do not follow the trend. Later, $SE - BDD$ further removes outlier and makes the system robust from stealthy FDI attacks. On the other hand, in the seed-based technique, nothing is removed during the remapping step; instead, the $SE - BDD$ procedure does the cleansing task. Thus, by making the compromised sensors outlier, DDAF removes FDI attacks that would remain undetected without the randomization.

Whereas the prediction-based technique can only mitigate the attacks, the seed-based one can also localize them. The seed-based DDAF analyzes the residual data and the outlier sensors to detect stealthy FDI attacks. If there is an attack, the framework provides the attacker's location in the communication network and the targeted sensors. In detecting and localizing the attack, the residual vector and the outliers are reshuffled using the $recID$-$randID$ pairs of each level in the ID-Stack to observe them from an attacker's perspective. As an FDI attack vector follows the system topology ($a = h(c)$) if there is an attack at any network level, the shuffling

finds a topological pattern into the residual data and the outlier senors' locations. The following section provides a detailed overview of different parts of the framework.

CHAPTER 5

DECEPTION MECHANISM

This chapter shows the details of each part of the deception mechanism of DDAF. Figure 5.1 shows the tasks of the nodes for the deception mechanism. We model the nodes with some common attributes, as explained in Table 5.1. The deception mechanism is divided into two steps. EMS initiates the deception by sending instructions to the sensors. The nodes store the instructions and run the randomization algorithm to determine the randomization plan. On the other hand, during the data acquisition steps, the nodes randomize the IDs and add decoy data as the core part of the deception. The following sections lay out the details of each step.

Table 5.1: Attributes of a node in the hierarchical network.

Attributes	**Type**	**Description**	**Example**
nodeLevel	string	Level of the node, 'l'	'2'
nodeID	object	ID of the node, $\mathcal{S}_i^l$	$\mathcal{S}_1^2$
leaf	boolean	leaf node or intermediate node	False
parent	object	ID of the parent node, $\mathcal{S}_i^l$	$\mathcal{S}_1^3$
nchild	integer	Number of child nodes	4
childs	list	Address of child nodes	[$\mathcal{S}_1^1$, $\mathcal{S}_2^1$, ..]
sensors	set	Sensors under the node	{1, 2, ..., m}
fixed	set	Sensors with fixed IDs	{1, 2, ..., m-2}
randomized	set	Sensors for ID randomization	{3, 4, ..., m-1}
decoyed	set	Sensors for decoyed data	{5, 6, ..., m}
deceptive	set	Deceiving sensors to the node	{1, 2, ..., m}
totDec	integer	Number of sensors in deception	12
decoyed	set	Decoyed sensors to the node	{1, 2, ..., m}
totRem	integer	Number of sensors remaining to be deceived	8
reported	set	Reporting sensors to the node	{1, 2, ..., m}
values	list	sensor measurements	[10, 20, ..., -15]

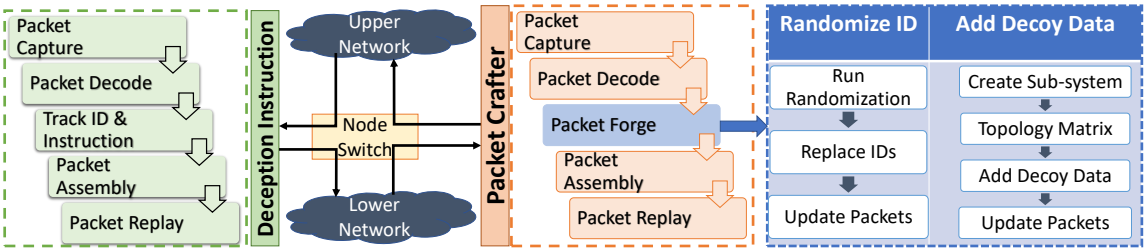


Figure 5.1: Deception mechanism in the nodes of the hierarchical network.

5.1 Deception Instruction

Firstly, EMS sends the deception instructions to the sensors, and the nodes keep track of the instructions. The instruction contains a type flag where 0, 1, and 2 indicates *fixed*, *randomized*, and *decoyed*, respectively. Thus, each node constrains an m dimensional array $\mathcal{T}$ to store the sensor-wise deception instructions, where m is the number of sensors.

5.1.1 $recIDs$-$randIDs$ Pair Generation

$recIDs$-$randIDs$ pair generation is the first step of the deception process. The received IDs of type-1 (*random*) sensors $recIDs$ are shuffled among themselves to generate the deceptive random IDs $randIDs$. The set $randIDs$ is generated following procedure **randIDGen**, as shown in Algorithm 1 using $recIDs$, and the generated seed. We propose a tree-based approach for the randomization. Each node generates a seed value and utilizes it to generate the random IDs. The pair $recIDs$-$randIDs$ contains the lists of IDs that define the randomization pattern. The $recIDs$-$randIDs$ pairs of the nodes are updated at a regular interval with EMS's request.

The sub-task of random ID generation for a particular node is done by a recursive algorithm, **Rand-Child**, as shown in Algorithm 2, which ensures the uniform distribution of the deceptive sensors. **Rand-Child** takes two arguments- the ID of the node and number of sensors, $totDec$ of the considered child node to be randomized.

Algorithm 1: randIDGen($nodeX$)

initialization $recIDs, randIDs = [] ;$
$recIDs \leftarrow nodeX.sensors - nodeX.randomized;$
$randIDs \leftarrow nodeX.sensors - nodeX.randomized;$
$recIDs$.append($nodeX.randomized$);
for *for each child* $nodeC \in nodeX.childs$ **do**
    $randIDs$.append(Rand-Child($nodeX, nodeC.randomized$));
Return $recIDs, randIDs$

Thus, each **Rand-Child** call for the child nodes in the list *childs* returns the set of deceptive IDs for that child. The merged output IDs of these *nchilds* calls is the set of random IDs for that node. The same process is performed for all the nodes in the network other than EMS to generate and store their $recIDs$-$randIDs$ pair.

Algorithm 2: Rand-Child($nodeX, n$)

initialization;
if $nodeX.leaf = True$ **then**
    $nodeX.totRem \leftarrow 0;$
    Return [$nodeX.nodeID$];
for *for each child* $nodeC \in nodeX.childs$ **do**
    $remSensorsChild$ append ($nodeC.totRem$);
$ndist \leftarrow wdist(n, remSensorsChild);$
for *each* $(nodeC, nc) \in (nodeX.childs, ndist)$ **do**
    if $nc > 0$ **then**
        $randIDs$.append(Rand-Child($nodeC, nc$));
        $nodeX.update();$
Return $randIDs$;

The primary goal of random ID generation is to shuffle the sensors of one child node as the sensors coming from all the child nodes. Hence, if an adversary attacks the sensors coming from that child node or a critical part of the system, the injected malicious data will be distributed to the sensors coming from different child nodes, which makes the stealthy attacks invalid also keeps the system observable.

5.1.2 Generation of Subsystem

To support the ID randomization, the nodes add the decoy data for the type-2 sensors. Usually, the nodes are lightweight devices, thus to minimize the computational overhead in calculating the decoy data, each node creates a virtual sub-system that spans the sensors reported through that node and the substations where those sensors are located. We define H^i as the topology matrix of the i-th subsystem, where

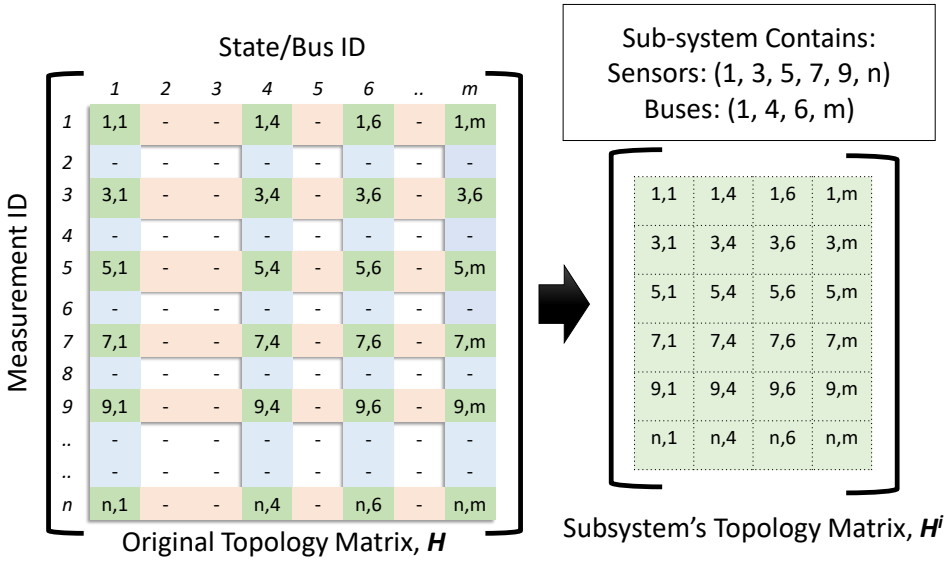


Figure 5.2: Generation of topology matrix of the subsystem.

the rows represent the sensors, and the columns are for the substations. Figure 5.2 shows a case of generating a subsystem's topology matrix, H^i from the system's full topology matrix H, where the subsystem contains the sensors 1, 3, 5, 7, and n, which are located at the substations 1, 4, 6, and m.

5.2 Packet Crafting

This section explains the steps for packet crafting, which has two steps, as discussed below:

Algorithm 3: RanID($\mathcal{I}_{rec}, recIDs, randIDs, \mathcal{T}$)

```
initialize I_rand = I_rec;
for i = 1 to len(I_rec) do
    for j = 1 to len(recID) do
        if I_rec[i] = recID[j] and T[i] == 1 then
            I_rand[i] = randIDs[j];
            break;
```

5.2.1 Randomizing IDs

During the data acquisition steps, the IDs of type-1 sensors are replaced utilizing the $recIDs$-$randIDs$ pair. Algorithm 3 shows the **RandID** procedure, which replaces the type-1(*random*) sensors' received IDs $\mathcal{I}_{rec}$ with deceptive random IDs, $\mathcal{I}_{rand}$.

As the data packets contain randomized IDs, if the attacker is not aware of the deception, s(he) will be injecting the false data to the deceived locations. Let us assume that the $\mathcal{I}_{org} = \{o_1, o_2, ...o_{m-1}, o_m\}$ is the original sequence of m IDs for the measurement data at one layer, where the shuffled IDs $\mathcal{I}_{rand} = \{r_1, r_2, ...r_{m-1}, r_m\}$ are used with that measurement set. Thus, the probability that $\mathcal{I}_{org}$ and $\mathcal{I}_{rand}$ are exactly the same is $\frac{1}{k!}$, where k is the number of *randomized* sensors and $k \leq m$. For $k = 5$ the probability is 0.008 and $k = 10$ the probability is 2.75×10^{-7}. Thus, for a node with a little higher number of sensors m, such probability converges towards zero. If an attacker tries to launch a reconnaissance attack, he will end up with different state estimation. On the other hand, if the attacker launches a targeted active attack, he/she will be attacking the wrong set of sensors. Usually, an FDI attack vector contains a critical set of sensors. Due to this randomization, the attacker attacks sensors with the critical IDs, but they contain the sensors' measurement data coming from different parts of the system. Thus, removing those sensors does not create any issue for the observability of the system. Moreover, the tree-based randomization

algorithm ensures that the two pairs will not be the same if the node has more than one sensor to randomize. Thus, we can assume with high confidence that $\mathcal{I}_{org} \neq \mathcal{I}_{rand}$.

5.2.2 Adding Decoy Data

ID randomization makes the deception more visible to the attacker as the random IDs may not follow the topological pattern. Thus, if the adversary runs the **SE-BDD** on the deceptive data that only contains the random IDs, the sensors with the random IDs will become the outliers, and he/she might end up with the actual state estimation. Thus, to support the random IDs, we propose to add decoy data for the type-2 (*decoyed*) sensors, which supports the random IDs to be good data in the attacker's state estimation, making the original type-0 (*fixed*) data outlier.

Algorithm 4: FindDecoy ($\mathbf{z}_{fix}$, $\mathbf{z}_{rand}$, $\mathbf{z}_{dec}$, H, α)

```
1 initialize the decoy data with the existing data, z_decoy ← z_dec;
2 [r_fix, r_rand, r_decoy] = SE([z_fix, z_rand, z_decoy], H);
  /* run state estimation and amend decoy data                    */
  /* repeat until the decoy data is harmonized                    */
3 while ||r_decoy|| > α do
4 |   [r_fix, r_rand, r_decoy] = SE([z_fix, z_rand, z_decoy], H);
5 |   z_decoy ← z_decoy - r_decoy;
6 return z_decoy
```

Let's assume that $\mathbf{z}^i$ ($\in \mathbf{z}$) is the measurement vector consisting of only the sensors of i-th node. Thus, we can define $\mathbf{z}^i$ as $[\mathbf{z}^i_{fix}, \mathbf{z}^i_{rand}, \mathbf{z}^i_{dec}]$, where $\mathbf{z}^i_{fix}$, $\mathbf{z}^i_{rand}$, and $\mathbf{z}^i_{dec}$ contains the fixed, randomized, and decoyed measurement data. Algorithm 6 shows the technique to calculate the decoy data $\mathbf{z}^i_{dec}$. The procedure takes the sub-system's randomized data, topology matrix H^i, and a threshold α.

CHAPTER 6

REMAPPING MECHANISM

As the reported sensor data contain decoy data and randomized IDs, EMS needs to remap the shuffled measurement data to the original sequence before utilizing them control decisions. Remapping is exactly the opposite of the deception process. Depending on the defender's resource and choice, we present three types of remapping mechanisms.

6.1 Seed-based Remapping

Like the deception process, it has two parts. Firstly, it generates the ID-Stack and then remaps the crafted data reported by nodes. The primary features of the seed-based DDAF are shown in Figure 6.1.

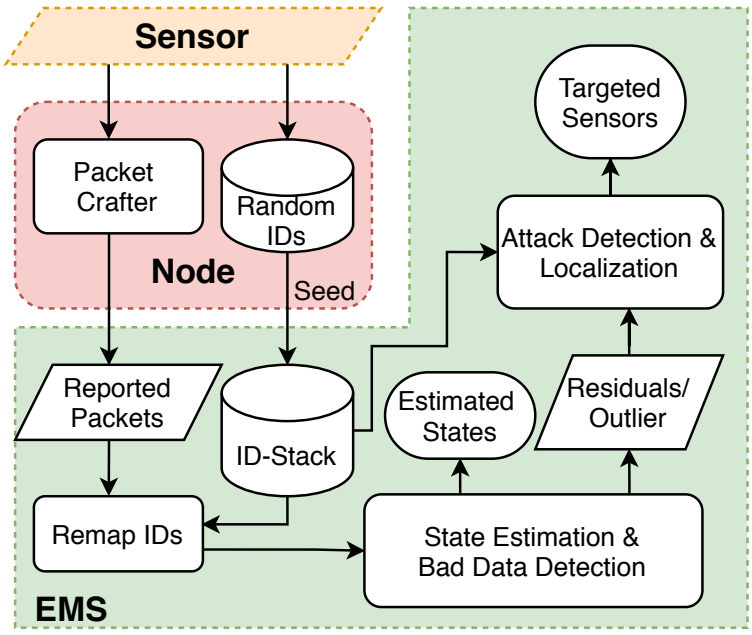


Figure 6.1: Block diagram of seed-based DDAF.

6.1.1 Building ID-Stack

During the *recIDs-randIDs* pair generation process as discussed in Section 5.1.1, EMS also generates the seeds using the nodes' secret keys. Using the seed values, EMS regenerates the same *recIDs-randIDs* pairs that each node has generated and builds an ID-Stack with all the randomization patterns for different levels and nodes. EMS can use the ID-Stack to trace back to the original sensor IDs from the reported deceptive IDs.

During the ID-Stack formation, the *recIDs-randIDs* pairs of same level nodes are concatenated vertically, which gives the total view of the deception at that level. Each level adds additional randomization into the system. *recIDs-randIDs* pairs of different levels (i.e., levels 1, 2, 3, etc.) are stacked sequentially so that the height of the ID-Stack is equal to the size of the network hierarchy. Figure 6.2 shows a sample ID-Stack for a three-layer hierarchical network. Here, the blue, green, and orange boxes represent the randIDGen pairs for different nodes at level 1, 2, and 3.

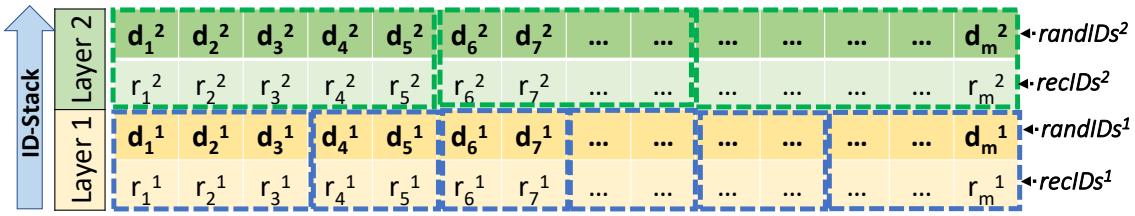


Figure 6.2: An example ID-Stack.

6.1.2 Remapping IDs

The second and core part of the remapping process is ID remapping that recovers original IDs from the deceived IDs. First, the crafted packets are collected and decoded at the EMS switch. The data packets contain the shuffled sensor IDs, along with some decoyed data after multiple randomizations. Thus, the first step is to drop

the packets/data reported with the type-2 sensor IDs. The remaining packets contain original data but randomized IDs. As the exact $recIDs$-$randIDs$ are generated in ID-Stack, multiple remaps of the reported IDs in reverse order of the levels provide the original IDs associated with the measurement values. We use the same **randID** algorithm, but now the shuffling direction is changed. Thus, $randIDs$ are replaced with the $recIDs$. In this case, EMS calls the algorithm for all the layers, and the last call for level 1 returns the original ID sequence of the measurement data.

Algorithm 5: State Estimation and Bad Data Detection
$SE - BDD(\mathbf{H_{init}}, \mathbf{z_{msr}})$

$\mathbf{H} \leftarrow \mathbf{H_{init}}$;
$outlier = []$;
while H *has fullrank* **do**
    estimated states, $\mathbf{x_{est}} \leftarrow (\mathbf{H^T H})^{-1}\mathbf{H^T z_{msr}}$;
    estimated measurements, $\mathbf{z_{est}} \leftarrow \mathbf{H x_{est}}$;
    estimated residual , $\mathbf{r_{est}} \leftarrow \mathbf{z_{msr}} - \mathbf{z_{est}}$;
    if $max(|\mathbf{r_{est}}|) > r_{threshold}$ **then**
        $i \leftarrow argmax(|\mathbf{r_{est}}|$;
        remove i-th measuremnt from $\mathbf{z_{msr}}$;
        $outlier$.append(i-th ID);
        update $\mathbf{H}$ excluding i-th measurement;
    else
        Return $\mathbf{x_{est}}, \mathbf{z_{est}}, \mathbf{r}, outlier$
print("System unobservable")

6.1.3 Attack Detection, Mitigation, and Localization

This section explains how the stealthy attacks are detected, mitigated, and localized after remapping to the original IDs and running the **SE-BDD** algorithm.

Attack mitigation:

This subsection explains how the $\mathbf{SE-BDD}$ algorithm mitigates FDI attacks from a system executing DDAF. EMS runs the $\mathbf{SE-BDD}$ procedure, as shown in Algo-

rithm 5, on the remapped data packets. The topology matrix H and measurement vector $\mathbf{z_{msr}}$ are generated using the original sensors IDs and measurement values, respectively. The $SE-BDD$ algorithm returns the estimated states, estimated measurements, residual vector, and outlier sensors.

Now, let us assume, there is an attacker trying to inject stealthy data $\mathbf{a_{org}}$ into the set of sensors $\mathcal{I}^c_{org}$ of that node, where $\mathbf{a_{org}} = [a_{o_1}, a_{o_2}, ..., a_{o_{m-1}}, a_{o_m}]$. The attack will be successful (stealthy) if $\mathbf{a_{org}} = \mathbf{Hc}$, where $\mathbf{c}$ is the targeted malicious state. Due to the randomization, the injection ends up as a randomized attack data, $\mathbf{a_{rand}}$ into the sensors $\mathcal{I}^c_{rand}$ where, $\mathbf{a_{rand}} = [a_{r_1}, a_{r_2}, ..., a_{r_{m-1}}, a_{r_m}]$. We already show that with sufficient randomization, $\mathcal{I}^c_{org} \neq \mathcal{I}^c_{rand}$, thus, $\mathbf{a_{org}} \neq \mathbf{a_{rand}}$ and $\mathbf{a_{rand}} \neq \mathbf{Hc}$. Therefore, the attack loses it stealthiness and with proper randomization, the BDD process at EMS will find all the compromised sensors as outliers. The process is explained in the next subsection.

Let us assume, and the original measurement vector is $\mathbf{z_{org}}$, which is randomized as $\mathbf{z_{rand}}$ at the compromised node. Thus, the attacker injects $\mathbf{a_{org}}$ to the randomized measurement vector $\mathbf{z_{rand}}$. The compromised measurement vector that EMS receives is $\mathbf{z^a_{rand}}$, where $\mathbf{z^a_{rand}} = \mathbf{z_{rand}} + \mathbf{a_{org}}$. However, EMS remaps the deceptive IDs $\mathcal{I}_{rand}$ to original IDs $\mathcal{I}_{org}$ (also $\mathcal{I}^c_{rand}$ to $\mathcal{I}^c_{org}$) using the ID-Stack. Thus, the compromised measurement data $\mathbf{z^a_{rand}}$ also changes to the original sequence as $\mathbf{z^a_{org}}$. As remapping and shuffling are two exact opposite tasks, during this remapping process, the original injected attack vector gets shuffled from $\mathbf{a_{org}}$ to $\mathbf{a_{rand}}$, where, $\mathbf{z^a_{org}} = \mathbf{z_{org}} + \mathbf{a_{rand}}$. Finally, EMS runs the $\mathbf{SE-BDD}$ algorithm on the $\mathbf{z^a_{org}}$ data that makes the remapped compromised sensors in the set $\mathcal{I}^c_{org}$ outliers as the attack vector, $\mathbf{a_{rand}} \neq Hc$.

Moreover, in the case of a ideal noiseless system with sufficient redundancy, as the compromised sensors in $\mathcal{I}^c_{org}$ become outliers and get excluded from the estimation process, the rest of the clean measurement data give an estimated measurement

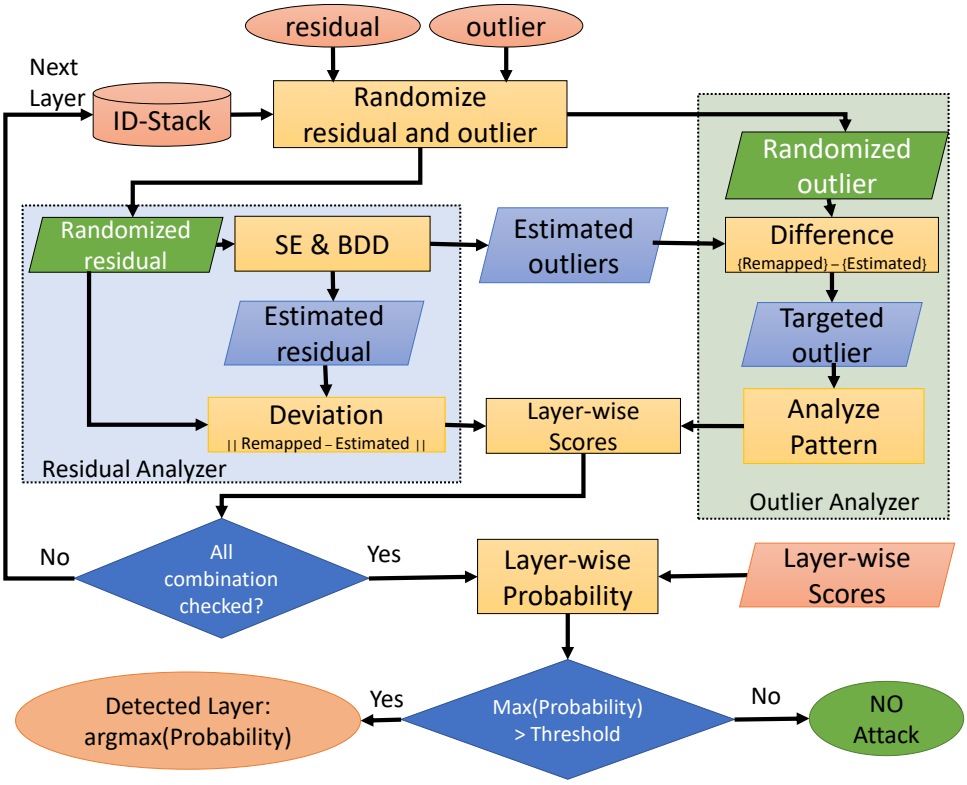


Figure 6.3: Block diagram for attack detection and localization.

vector, $\mathbf{z^a_{est}}$ which is the same as $\mathbf{z_{org}}$. The residual vector, $\mathbf{r_{org}} = \mathbf{z^a_{org}} - \mathbf{z^a_{est}} = \mathbf{z^a_{org}} - \mathbf{z_{org}} = \mathbf{a_{rand}}$. Thus, the residual vector is actually the randomized version of the original attack vector $\mathbf{a_{org}}$. The set of outlier sensors, $\mathbf{out_{org}} = \mathcal{I}^c_{org}$, is the randomized version of the compromised sensors $\mathcal{I}^c_{rand}$.

Attack detection and localization:

Here we explain how the residual/outlier data are used to detect and localize the attacks. The residual and outlier data are leveraged to detect attacks and reveal the attacker's target and location. As discussed in the attack mitigation process, proper randomization makes all the compromised sensors outlier. The process is shown in Figure 6.3. If we again randomize the residual vector and the outliers with

the nodes' $recIDs$-$randIDs$ pairs, the randomized residual vector, $\mathbf{r_{rand}}$ goes back to the actual attack vector $\mathbf{a_{org}}$ and the randomized outliers $\mathbf{out_{rand}}$ becomes $\mathcal{I}^c_{rand}$. Now, as $\mathbf{r_{rand}} = \mathbf{a_{org}} = Hc$, if we run the $\mathbf{SE-BDD}$ algorithm on $\mathbf{r_{rand}}$ data, we get the same estimated attack vector and no outlier. This condition proves that the deceived residual is not just random system noises but a precisely calculated FDI attack vector.

However, along with the attacked sensors, if there are some other outliers $\mathbf{out_{noise}}$ due to random system noise vector $\mathbf{n}$ ($\mathbf{n} \neq Hc$), $\mathbf{SE-BDD}$ on the residual data considers $\mathbf{out_{noise}}$ as outliers due to their noncompliance values. Hence, in that case, the set $\mathbf{out_{sus}} = \mathbf{out_{rand}}$ - $\mathbf{out_{noise}}$ contains the list of suspected/targeted sensors in the attack. Similarly, there is a perfect pattern in the sensors' location in the sensors in the $\mathbf{out_{sus}}$ as an FDI attack vector selects the sensors following a topological pattern in their locations. DDAF considers all the layers one by one and checks if any layer provides a pattern in the randomized residual/outlier data to detect the attacker's position in the network. If random system noises cause the residuals/outliers, none of the layers will have a high likelihood.

6.1.4 A 5 Bus Case Study

In this section, we present an example case study explaining the overall process of DDAF for the IEEE 5 bus system. Figure 6.4 shows the electrical and communication network on the considered approach. Generally, each transmission line contains two sensors reporting forward and backward line power flows, and each bus has one sensor reporting the power consumption. Thus, IEEE 5 bus system consists of 5 substations (buses), 7 lines and $(2 \times lines + buses) = 19$ measurement sensors in total. The two-layer hierarchical network has five level-1 nodes at the substations and two level-2 nodes that communicate directly with EMS.

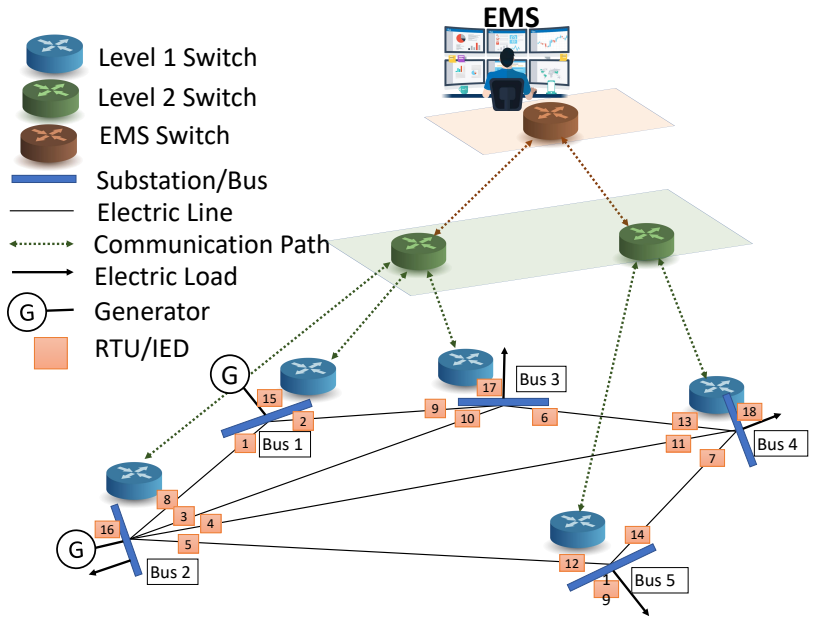


Figure 6.4: SCADA Network of IEEE 5 bus system.

recIDs – randIDs pairs and ID-Stack generation

Figure 6.5 demonstrates the *recIDs-randIDs* pair at each node generated by the randIDGen. Here, substation 1 has three sensors with IDs $\{1, 2, 15\}$. Thus, switch $\mathcal{S}_1^1$ generates a seed and runs Algorithm 1 to generate the set of random IDs $\{15, 2, 1\}$, as shown with the vivid color boxes. Similarly, the rest of the switches follow the

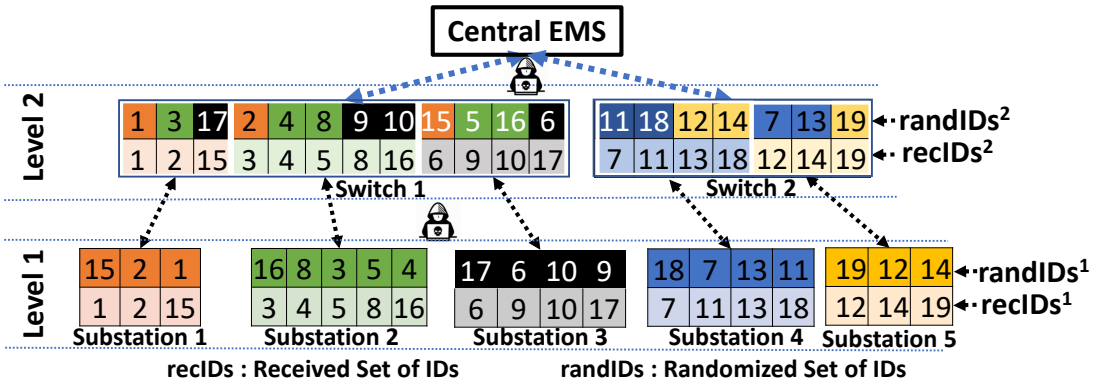


Figure 6.5: ID-Stack of IEEE 5 Bus System.

same procedure to generate the deceptive IDs. Thus, the set of all the received and randomized IDs at level 1 are presented by $recIDs^1$ and $randIDs^1$, respectively. While randomizing at level 1, all the deceptive IDs are selected within the same substation as indicated by their colors. Switch $\mathcal{S}_1^1$, $\mathcal{S}_2^1$, and $\mathcal{S}_3^1$ forward the packets to their parent node $\mathcal{S}_1^2$, which is a level 2 switch. Similarly, $\mathcal{S}_4^1$, and $\mathcal{S}_5^1$ report their data to $\mathcal{S}_2^2$. As shown in the figure, the set of received IDs at level 2 switch, $\mathcal{S}_2^1$ are 1, 2, 15, 3, 4, 5, 8, 16, 6, 9, 10, 17. The randomized pattern of this received IDs is 1, 3, 17, 2, 4, 8, 9, 10, 15, 5, 16, 6. According to Figure 6.5, the sensors 1, 2, and 15 are reported by $\mathcal{S}_1^1$. However, $\mathcal{S}_1^2$ randomizes the pattern as 1, 3, and 17. The colors imply that these three IDs come from three different substations. Similarly, the rest of the sensors received from a single child node are distributed to multiple child nodes. Thus, the randomization at level 2 is more effective than level 1 and so on. Simultaneously, EMS generates the same seed and runs the randomization algorithm for each of the nodes to build the complete ID-Stack, as shown in Figure 6.5.

Packet crafting and remapping

As the system executes DDAF, the deceptive sensors' packets are crafted at the nodes that belong to its path to the EMS. With the generated $recIDs$-$randIDs$ pairs, the received packets are crafted with the deceptive IDs. That means, all the level 1 switches send the original measurement data of sensor {1, 2, 15, 3,...., 18, 12, 14, 19} with deceptive IDs {15, 2, 1, 16, ..., 11, 19, 12, 14}. Similarly, the level 2 switches send the received data with sensor IDs {1, 2, 15, 3, ..., 18, 12, 14, 19} with the deceptive IDs {1, 3, 17, 2,...., 14, 7, 13, 19} and report to EMS. After collecting the crafted packets, EMS performs the remapping for all the layers in a backward direction to get back to the original IDs. The remapping starts with the highest layer,

which is layer 2 in this case, propagates to the first layer, and recovers the original IDs.

Table 6.1: Attack Scenario for IEEE 5 Bus System for Seed-based Remapping.

Original IDs, $\mathcal{I}_{org}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Original Sensor Data, $\mathbf{z_{org}}$	83	41	19	29	54	15	5	-83	-41	-19	-29	-54	-15	-5	125	20	-45	-40	-60
Randomized IDs, $\mathcal{I}_{rand}$	17	3	10	9	2	6	14	8	15	16	11	19	12	7	1	4	5	18	13
Actual Attack Vector, $\mathbf{a_{org}}$	**17**	**4**	0	0	0	0	0	**-17**	**-4**	0	0	0	0	0	**21**	**-17**	**-4**	0	0
Randomized Attack Vector, $\mathbf{a_{rand}}$	**-4**	0	0	**-4**	**4**	0	0	**-17**	**21**	**-17**	0	0	0	0	**17**	0	0	0	0
Remapped Sensor Data, $\mathbf{z^c_{org}}$	**79**	41	19	**25**	**58**	15	5	**-100**	**-20**	**-36**	-29	-54	-15	-5	**142**	20	-45	-40	-60
Estimated Sensor Data, $\mathbf{z_{est}}$	**83**	41	**19**	29	**54**	**15**	5	**-83**	-41	**-19**	-29	-54	-15	-5	**124**	20	-44	-39	-59
Residual Vector, $\mathbf{r_{org}}$	**-4**	0	0	**-4**	**4**	0	0	**-17**	**21**	**-17**	0	0	0	0	**17**	0	0	0	0

Impact of an FDI attack

Let us assume that an intruder launches an FDI attack on the communication path between layer- 2 and EMS of the IEEE 5 bus system. Table 6.1 demonstrates an attack, where the original and deceived IDs are shown as $\mathcal{I}_{org}$ and $\mathcal{I}_{rand}$. The attacker plans to launch the attack by injecting 17, 4, -17, -4, 21, -17, and -4 MW to the measurement data of sensors in set $\mathcal{I}^c_{rand}$ which are 1, 2, 8, 9, 15, 16, and 17, respectively. However, due to deception at node $\mathcal{S}_1^2$, IDs in $\mathcal{I}^c_{rand}$ are used to send the original measurement data of sensors in $\mathcal{I}^c_{org} = \{15,\ 5,\ 8,\ 4,\ 9,\ 10,\ 1\}$. Thus, whenever the attacker injects into the sensors in $\mathcal{I}^c_{rand}$, the injections take place at the measurement data of sensors in $\mathcal{I}^c_{org}$.

As shown in Table 6.1, $\mathbf{z^c_{org}}$ represents the remapped version of the compromised measurement vector. The estimation process on $\mathbf{z^c_{org}}$ makes the compromised sensors outliers, and only the good measurements prevail. Thus, the estimated measurement vector $\mathbf{z_{est}}$ remains almost the same as the original measurement vector $\mathbf{z_{org}}$, leaving no deviation in the state estimation due to the attack. Similarly, the residual vector $\mathbf{r_{org}}$ follows the pattern of the attack vector $\mathbf{a_{rand}}$, revealing the intent of the attack. The following part shows the detection of the stealthy attack, analyzing the residual vector and outlier sensors.

- **Analyzing the Residual Vector:** EMS inspects $\mathbf{r_{org}}$ and $\mathbf{out_{org}}$ from an attacker's point of view at each level, starting from level 1. The shuffled residual vector $\mathbf{r_{rand}}$ and its estimated version for each of the layers are shown in Figure 6.8. For level 1, the shuffled residual vector does not follow any topological pattern. Thus the estimated residual becomes almost zero, indicating a high deviation. On the other hand, for level 2, the estimation perfectly follows the provided residual vector, and the deviation is almost zero. A low deviation proves that the shuffled residual vector is not random noises rather a perfectly synthesized FDI attack vector injected at the communication medium between level 2 and EMS switches.

- **Analyzing the Outliers:** Similar to analyzing the residual vector task shown in the previous subsection, the actual outliers in $\mathbf{out_{org}}$ are shuffled back to different levels. The shuffled outliers, $\mathbf{out_{rand}}$ for level 1 is {1, 3, 5, 6, 8, 10, 15} and the noisy outlier during the residual estimation,$\mathbf{out_{noise}}$ is also {1, 3, 5, 6, 8, 10, 15}. As shown in the residual analyzer part, level-1 does not find any pattern in the residual vector; it considers all of the sensors with nonzero residuals as outliers. Thus, the ultimate suspected outliers, $\mathbf{out_{sus}}$ is $\mathbf{out_{rand}}$ - $\mathbf{out_{noise}}$ = {}. All the shuffled outliers are random, and there is no suspected outliers. On the other hand, for level 2, $\mathbf{out_{rand}}$ = {1, 2, 8, 9, 15, 16, 17}. As SE-BDD finds a perfect pattern in the shuffled residual data, noisy outlier $\mathbf{out_{noise}}$ is {}. In this case, $\mathbf{out_{sus}}$ is {1, 2, 8, 9, 15, 16, 17}, which is the same as $\mathbf{out_{rand}}$. The location of these suspected outliers in the electrical network is shown in Figure 6.7. Whereas for level 1, there is no suspected sensor, for level 2, we find all the initial outliers as suspects, and they follow a clear topological pattern in their location.

Analyzing the suspected sensors' locations and connectedness, we assign a similarity score/probability for each level. Thus, the detection algorithm finds a high probability that there is an FDI attack at level 2 (after node $\mathcal{S}_1^2$), and the attacker intended to compromise sensors 1, 2, 8, 9, 15, 16, and 17.

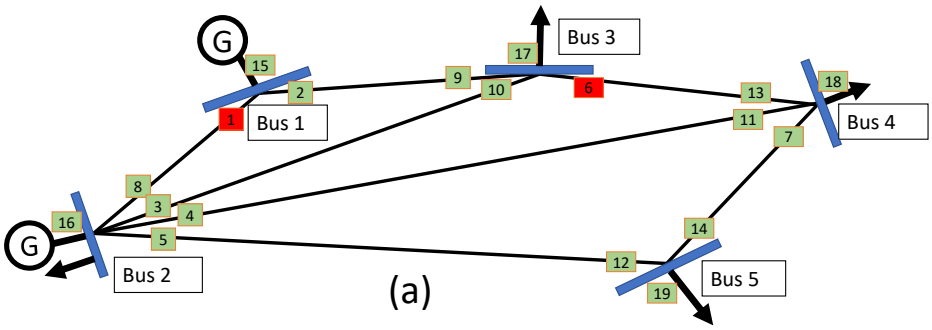


Figure 6.6: Distribution of the suspected sensors 1, and 6 for an FDI attack at Level 1, which indicates a lower probability of attack.

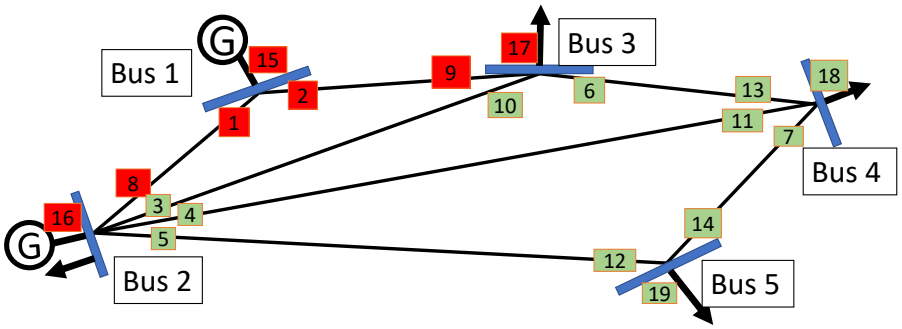


Figure 6.7: Distribution of the suspected sensors 1, 2, 8, 9, 15, 16, and 17 for an FDI attack at Level 2, which indicates a higher probability of attack.

Impact of random noise

Now, we analyze another case when the outliers occur due to random system noises. In this case, we inject the same attack vector, which we consider in the previous case, but at random locations, to mimic the pattern of random noise. The set of compromised sensors at layer 2 is {3, 4, 7, 9, 14, 16}, which makes the sensors {2, 4,

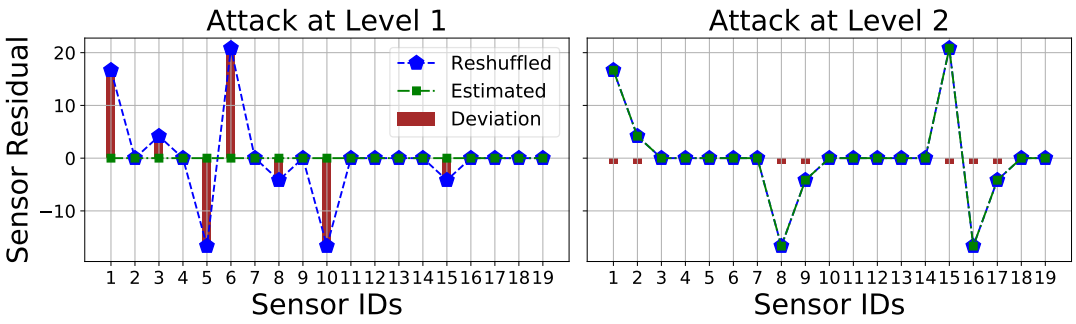


Figure 6.8: Residual estimation under an FDI attack. Level 1 shows high deviations in the residual estimation, presenting a lower chance of FDI attack. Whereas Level 2 shows almost zero deviations indicating a higher chance of FDI attack.

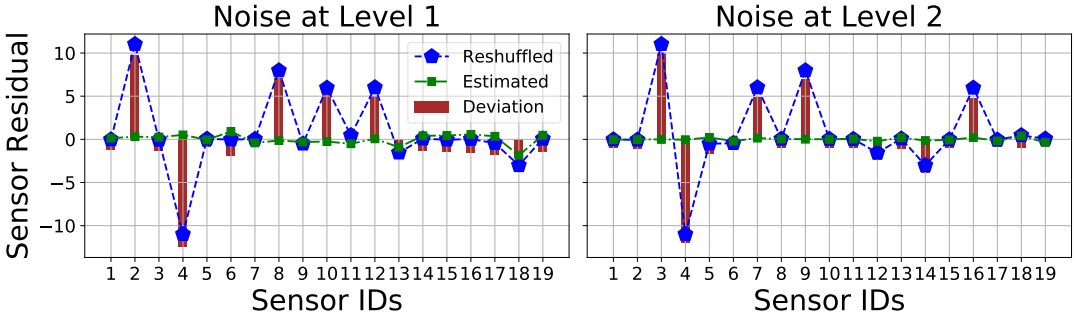


Figure 6.9: Residual estimation with random noise, where both the levels shows high deviations and lower probabilities of FDI attacks.

7, 10, 14, 16} outliers at EMS. Now, we analyze both the residual and outliers data as before. Figure 6.9 shows both the levels observe high deviations in the residual estimations. Similarly, as the injections are done randomly, all the sensors with non zero residual become an outlier and return no suspected sensors for any levels.

6.2 Prediction-based Remapping

Even though the seed-based remapping mechanism can successfully mitigate, detect, and localize the FDI attacks, it has some limitations. The time synchronization is mandatory in generating the same seed at the EMS side. Failure to maintain such synchronization can lead to different seed values; thus, different $recIDs$-$randIDs$ pairs

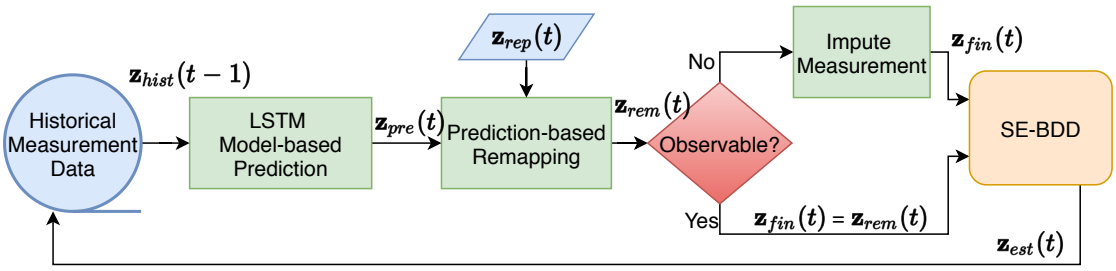


Figure 6.10: Prediction-based remapping mechanism.

are very dangerous for the system. Therefore, we propose a data prediction based remapping mechanism, where seed sharing is not necessary. However, even though the remapping mechanism is different, we consider the same deception mechanism as described in chapter 5.

This section introduces prediction-based remapping, where remapping is done based on the predicted measurement vector. An LSTM model is trained on the historical sensor data and used to predict the next measurement vector in real-time based on the previous historical data. Figure 6.10 shows the process of remapping using the LSTM's prediction. In the following sections, we explain different modules of the prediction-based remapping.

6.2.1 Historical Sensor Data

The time-series sensor data is used to train the LSTM model. The measurement vector has a dimension of m, where m is the number of sensors in the system. Thus the dataset has m features representing different sensors.

6.2.2 LSTM Model

We consider a many-to-one LSTM model in this application. In the input of the LSTM, we provide the past k samples of the measurement vector and predict the

next set of measurement data. Thus, the LSTM takes an input of $m \times k$ dimensional data and outputs a $m \times 1$ dimensional measurement data. In the input we provide the measurement data of $t-1$, $t-2$, $t-3$, ... , $t-k$ time step and ask to generate the data from t. All the last k time steps of data is shows as $\mathbf{z}_{hist}(t-1)$ and the prediction of the LSTM is expressed as $\mathbf{z}_{pre}(t)$, as shown in Figure 6.10.

6.2.3 Prediction of LSTM

The predicted measurement data $\mathbf{z}_{pre}(t)$ contains the expected values for each sensor. We provide an $m \times k$ dimensional measurement data as input to the LSTM's, and generate $m \times 1$ dimensional data. Therefore, $\mathbf{z}_{hist}(t-1) = [\mathbf{z}_{est}(t-1), \mathbf{z}_{est}(t-2),$ ..... , $\mathbf{z}_{est}(t-k+1), \mathbf{z}_{est}(t-k)]$. We train the LSTM model offline using the historical time series data. The dimension of the LSTM model depends on the considered system's topology.

6.2.4 Remapping Mechanism

Due to the deception, the sensor's measurement is crafted with randomized IDs, and the reported measurement vector, $\mathbf{z}_{rep}(t)$, needs to be reshuffled to get back to the original pattern. To do this reallocation process, we design an optimization algorithm to assign the best set of sensors IDs to $\mathbf{z}_{rep}(t)$ considering the predicted data $\mathbf{z}_{pre}(t)$.

Remapping algorithm

This part explains the repairing algorithm. As we deal with different combinations of the data, we use mixed integer programming (MIP) to implement the combinatorial optimization problem. Table 6.2 shows the notations used in defining the constraints.

Table 6.2: Modeling Parameters of Repairing Algorithm.

Notation	**Type, Dimension**	**Definition**
$\mathbf{z}_{pre}$	1-D Array, $n_p \times 1$	Set of predicted measurements
n_p	Integer	Num of predicted measurements
$\mathbf{z}_{rep}$	1-D Array, $n_s \times 1$	Set of reported measurements
n_r	Integer	Num of reported measurements
$\mathcal{M}$	2-D Array, $n_p \times n_s$	Recovery mapping matrix
$\mathcal{F}$	2-D Array, $n_p \times n_s$	Fixed sensor mapping matrix
$\mathcal{D}$	1-D Array, $n_r \times 1$	Decoy mapping array
$\mathcal{C}$	2-D Array, $n_p \times n_s$	Recovery cost matrix
$\mathcal{O}_{Reco}^{Cost}$	Integer	Measurement recovery cost
$\mathcal{O}_{Assi}^{Prof}$	Integer	Measurement assignment profit
η	Integer	Recovery threshold

Our goal is to find the $(n_p \times n_r)$ dimensional binary matrix $\mathcal{M}$, where the positions of the ones represent the successful recovery of reported measurements. The rows and columns of the ones in $\mathcal{M}$ represent the IDs of predicted and reported measurement, respectively. For example, a one in the position (i, j) of $\mathcal{M}$ represents that the j-th reported reading in $\mathbf{z}_{rep}$ is the actual measurement of i-th sensor. Thus, the total number of ones in $\mathcal{M}$ indicates the number of recovered measurements.

fixed sensors do not participate in the randomization process during deception. Their IDs remain the same during the whole process and we explicitly define them during the repairing process. $\mathcal{F}$ is the 2-D binary mapping matrix, where the rows and columns represent the already known remapping of *fixed* sensors. On the other hand, $\mathcal{D}$ presents the 1-D binary mapping of the *decoyed* sensors. As shown in (6.1) and (6.2), each recovery of the randomized sensor comes with a cost, defines as the difference between the predicted and the reported value. In the case of a fixed ID, the cost is explicitly defined as zero. As shown in (6.3), the *decoyed* measurements are not assigned to any of the sensors.

$$\forall_{1,1 \leq i,j \leq n_s,n_r} \quad \mathcal{F}_{i,j} == 0 \implies \mathcal{C}_{i,j} == |\mathbf{z}_{pre}^{i} - \mathbf{z}_{rep}^{j}| \tag{6.1}$$

$$\forall_{1,1\leq i,j\leq n_s,n_r} \;\; \mathcal{F}_{i,j} == 1 \implies (\mathcal{M}_{i,j} == 1) \;\wedge\; (\mathcal{C}_{i,j} == 0) \tag{6.2}$$

$$\forall_{1\leq j\leq n_r} \;\; \mathcal{D}_j == 1 \implies \sum_{i=1}^{n_s} \;\; \mathcal{M}_{i,j} == 0 \tag{6.3}$$

The recovery data must be within a specific range of the predicted data. Thus, as shown in (6.4), a recovery is valid only if the associated cost is within $\eta\%$ of the expected data. However, if there is no such reported value within that range, that sensor remains unassigned.

$$\forall_{1\leq i,j\leq n_s,n_r} \;\; \mathcal{C}_{i,j} > \mid \mathbf{z}^i_{pre} \times \eta \mid \implies \mathcal{M}_{i,j} == 0 \tag{6.4}$$

The constraints in (6.5) mandate that any measurement can be assigned to almost one sensor and vice-versa.

$$\forall_{1\leq i\leq n_s} \sum_{j=1}^{n_r} \;\; \mathcal{M}_{i,j} \leq 1 \quad \text{and} \quad \forall_{1\leq j\leq n_r} \sum_{i=1}^{n_s} \;\; \mathcal{M}_{i,j} \leq 1 \tag{6.5}$$

As shown in (6.7) and (6.8), the ultimate goal is to assign as much measurements as possible (maximize assignment profit), while keeping the recovery cost to minimum. This two objective functions are merged together in (6.6). Thus, the optimization maximizes the assignment profit by allocating as many sensors as possible and minimizes the recovery cost by assigning to the closest prediction. To ensure the maximum number of recovered sensors, we multiply the assignment profit by k and emphasize it than the recovery cost. Hence, there will always be a solution; however, no sensor will be recovered in the worst-case scenario. EMS will rely only on the fixed sensors to run the state estimation in such a rare case. Thus, it will be useful to select a set of fixed sensors that spans the system's critical parts and ensures observability.

$$\mathcal{O}^{Cost}_{Reco} = \sum_{i=1}^{n_s} \sum_{j=1}^{n_r} \;\; \mathcal{M}_{i,j} \times \mathcal{C}_{i,j} \tag{6.6}$$

$$\mathcal{O}_{Assi}^{Prof} = \sum_{i=1}^{n_s} \sum_{j=1}^{n_r} \mathcal{M}_{i,j} \tag{6.7}$$

$$\min \ (\mathcal{O}_{Reco}^{Cost} - k \times \mathcal{O}_{Assi}^{Prof}) \tag{6.8}$$

The successful execution of the program returns the matrix $\mathcal{M}$ from where we find the original pattern of $\mathbf{z}_{rep}$. Let us assume that $\mathbf{z}_{rem}$ is the recovered measurement vector used in $SE - BDD$. If the system is observable, it finds the state vectors and takes the necessary control decision. However, if the system is unobservable, we use an imputation algorithm to fill up the missing data.

6.2.5 Data Imputation

If there is an L-1/L-2 FDI attack in the system, the recovery algorithm can remove the compromised measurement and may keep the system observable under sufficient randomization. However, in the case of L-0 attacks, dropping the compromised measurement during the repairing process may lead the system to unobservability. Besides, instead of an FDI attack, there can be a DoS attack, where the attacker's goal is to make some targeted critical measurement missing. In both cases, DDAF replaces the missing data with the predicted measurements from $\mathbf{z}_{pre}(t)$.

6.2.6 A 14 Bus Case Study

In this section, we provide a case study of prediction-based remapping in the IEEE 14 bus system [IEEa], as shown in Figure 6.11. For simplicity, we consider a case where 100% of sensors are reporting measurement data to EMS, and all of them are considered for randomization.

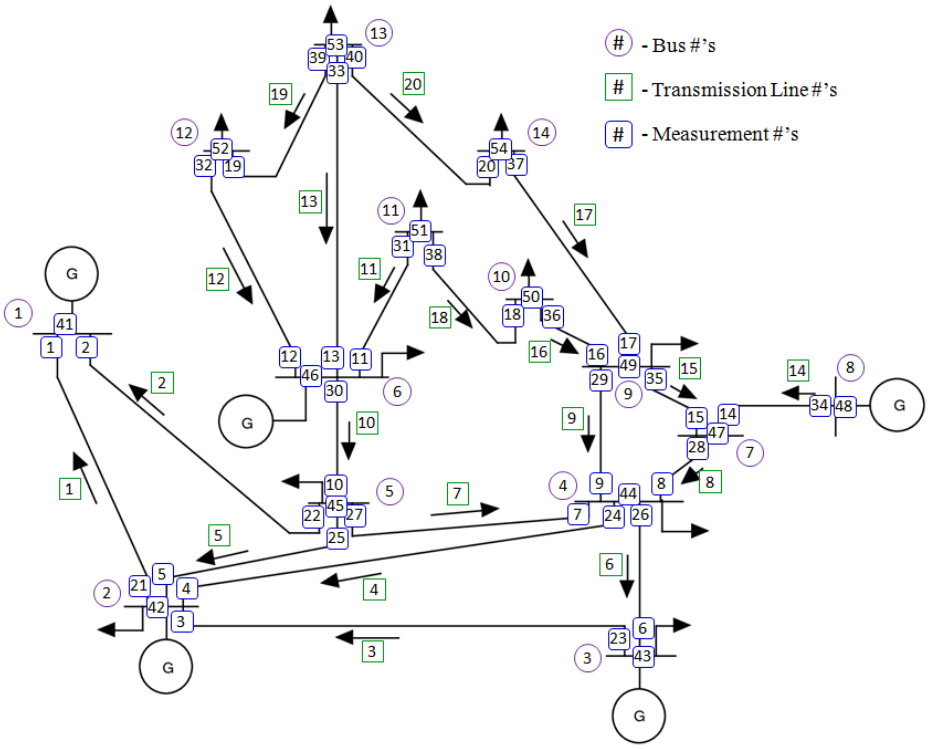


Figure 6.11: IEEE 14-bus test system [RASK14].

Remapping under normal condition

Figure 6.12 shows the randomized reported measurement data, sensor-wise prediction, and the recovered data under normal operating conditions. The reported data contains random IDs; thus, the reported measurement vector's shape does not follow the predicted points. However, once the repairing algorithm assigns the random measurement values to the right IDs, the recovered data precisely follow the LSTM's predicted data. However, sensors 8, 43, 44, 49, and 53 are not assigned to any measurement as the model cannot find any possible candidate for them. Those sensors may contain noises that deviate them from the predicted values. However, among 54 measurements, 49 of them are assigned to the right IDs, which is enough to make the system observable.

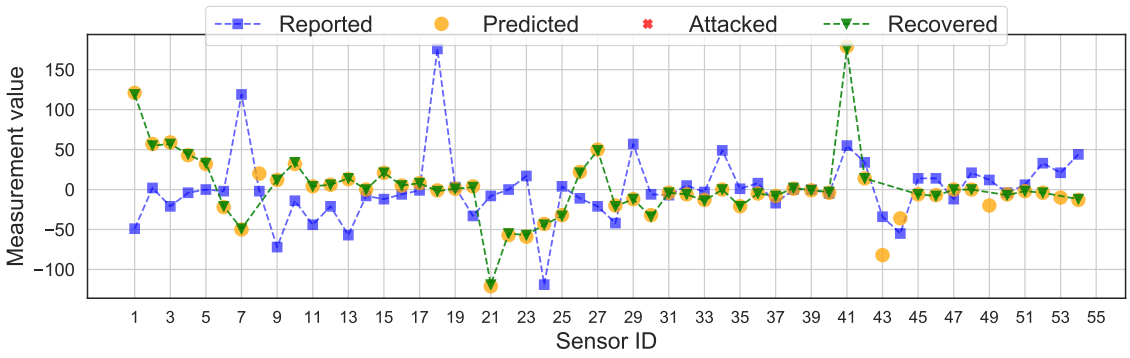


Figure 6.12: Prediction-based remapping mechanism under normal condition, where almost 91% (49 out of 54) of the randomized sensors are remapping accurately.

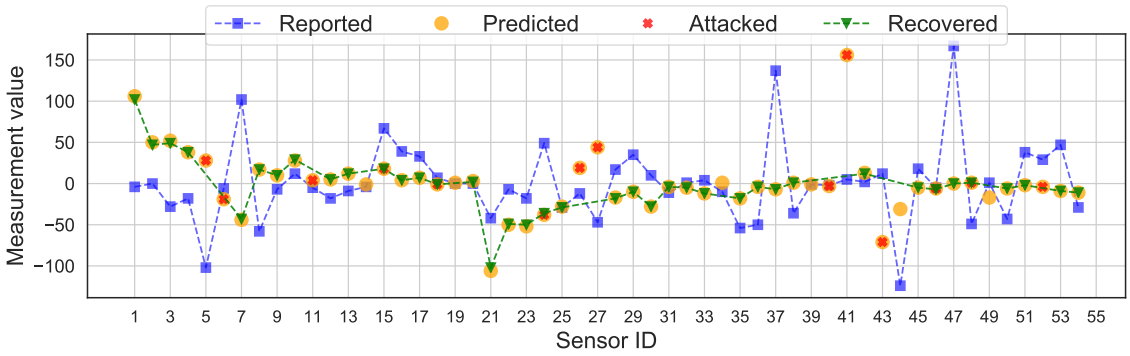


Figure 6.13: Prediction-based remapping mechanism under FDI attack, where the attacked sensors are omitted during the remapping and keeps the state estimation resilient to the FDI attacks.

Remapping under an FDI attack

Figure 6.13 shows how prediction-based remapping eliminates the impact of the FDI attacks. In this case, we consider an FDI attack, where the targeted sensors are 8, 9, 15, 16, 17, 28, 29, 35, 36, 37, 44, 47, 50, and 54. However, due to the deception, these sensor IDs are used to send the data of 5, 6, 11, 15, 18, 24, 26, 27, 40, 41, 43, 46, 48, and 52, respectively. Hence, even though the attacker expects to be stealthy and bypass the BDD, the remapping mechanism mitigates the attack by declining the compromised sensors. Thus, among the attacked sensors 5, 6, 11, 26, 27, 40, 41, 43, and 52 are unassigned due to their suspicious values, which alleviates the impact

of the FDI attack. Sensors 15, 18, 24, and 48 are not removed as their injection amounts are very small, within 5% of the predicted values. Further processing on this measurement data in the state estimation process removes the remaining outliers and makes the system effectively immune to the attack.

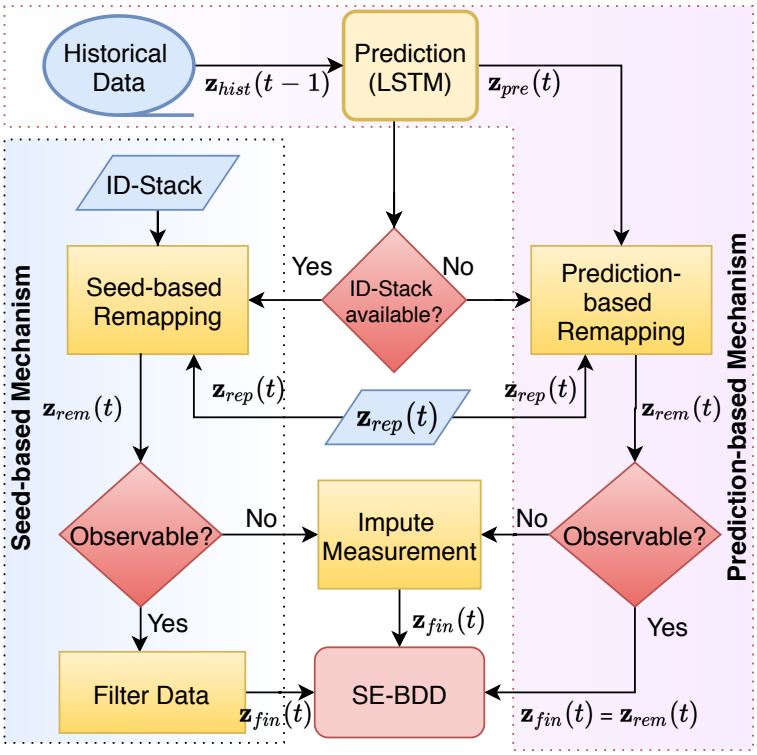


Figure 6.14: Hybrid remapping mechanism.

6.3 Hybrid Remapping

We provide a hybrid remapping mechanism utilizing both seed-based and prediction-based approaches. An amalgamation of these two mechanisms provides the best performance compared with the individual mechanism regarding the accuracy and robustness of the remapping. Figure 6.14 shows the block diagram of the hybrid

remapping mechanism. In the following subsections, we discuss different features of hybrid remapping.

6.3.1 Seed-based Approach

Seed-based remapping is the preferred remapping approach in a hybrid mechanism. Seed-based remapping is only possible when the ID-Stack is already generated. The generation process of ID-Stack is already explained in Section 6.1. Once the ID-Stack is ready, we use the ID-Stack to remap back to the original IDs to the deceptive reported measurement data $\mathbf{z}_{rep}(t)$. The remapped measurement data $\mathbf{z}_{rem}(t)$ contains the measurement vector with the original IDs and measurement pairs that is ready to use in the $SE-BDD$ algorithm. If the $SE-BDD$ mechanism finds the $\mathbf{z}_{rem}(t)$ sufficient to give the system observability, we pass the vector to the filtering data block to further eliminate compromised sensors.

Filtering Data

This block does a straightforward operation by comparing the measurement data in $\mathbf{z}_{rem}(t)$ and $\mathbf{z}_{pre}(t)$ for individual sensors. It does remove any sensor i if the remapped data is not within the η percentage of the predicted value. The following constraint formalizes the task of this block.

$$\forall_{i \in SensorList} |\mathbf{z}^i_{rem} - \mathbf{z}^i_{pre}| > \mathbf{z}_{pre} \times \eta \implies remove\ \mathbf{z}^i_{rem}\ from\ \mathbf{z}_{rem}$$

After filtering the $\mathbf{z}_{rem}$, it only contains the measurement that is remapped to the original data through ID-Stack and also follows the predicted trend.

Imputing Data

In case of limited redundancy of the sensors, filtering out a few of them may create an unobservability in the system. Besides, any attack on the availability of the measurements may create a similar unexpected scenario. Thus, to handle the data unavailability issues, we implement an imputation algorithm that replaces the missing sensor data with the LSTM's predicted sensor data to ensure the system's observability. The core task of the imputation block is shown in the following:

$$\forall_{i \in SensorList} \;\; \mathbf{z}^i_{rem} \; is \; unavailable \implies \mathbf{z}^i_{rem} = \mathbf{z}^i_{pre}$$

$\mathbf{z}_{fin}(t)$ is the final measurement vector to use in the $SE-BDD$ algorithm. $SE-BDD$ algorithm is explained in Algorithm5, and the attack detection and mitigation part are the same as explained in Section 6.1.3.

6.3.2 Prediction-based Approach

In case of losing synchronization with the nodes, EMS may fail to build the ID-Stack. In such cases, we propose prediction-based remapping as the back-up option. The core idea of prediction-based remapping is already explained in Section 6.2. The concept is still the same. The predicted measurement $\mathbf{z}_{pre}(t)$ is used to run the remapping optimization algorithm that reshuffles the $\mathbf{z}_{rep}(t)$ and provides the remapped vector $\mathbf{z}_{rem}(t)$. The remaining steps are similar to the seed-based approach. The only exception is that we do not need to run the filtering block as the optimization by itself makes sure that the remapped data are within the allowed range of the predicted values. The only part that might be crucial is data imputation. Optimization does not consider checking the observability of the system. Thus, in case of such data unavailability problems, we need to pass the remapped measurement $\mathbf{z}_{rem}(t)$ to add

Algorithm 6: Hybrid Remap($\mathbf{z}_{rep}, \mathbf{z}_{pre}, ID - Stack$)

if $Id - Stack$ *available* **then**
  $\mathbf{z}_{rem} = seed - based - remap(\mathbf{z}_{rep}, ID - Stack)$;
else
  $\mathbf{z}_{rem} = prediction - based - remap(\mathbf{z}_{rep}, \mathbf{z}_{pre})$;
//Checking if the system is observable
if $\mathbf{z}_{rem}$ *observable* **then**
  $\mathbf{z}_{fin} = filter - data(\mathbf{z}_{rem})$;
else
  $\mathbf{z}_{fin} = impute - data(\mathbf{z}_{rem})$;

the missing data from the predicted values and finalize $\mathbf{z}_{fin}(t)$. Algorithm 6 shows the pseudo-code of the hybrid remapping process.

CHAPTER 7

IMPLEMENTATION

This chapter explains the implementation of DDAF in a power grid hierarchical network. The framework is virtually implemented in Python using Jupyter Notebook [jup]. We use different libraries and also design numerous functions for the implantation. Moreover, DDAF's performance is verified with the *PowerWorld [Pow21]* simulator.

7.1 Data Pre-processing

To start with implementing DDAF, we consider standard IEEE bus systems as the test case' [ieeb]. The following steps are executed to generate the necessary data for the implementation.

- Collection and generation of IEEE bus systems' data using *pandas*, and *NumPy*.
- Generation of time series sensor data with OPF using *PYPOWER*.
- Designing communication network topology and hierarchy.
- Implementing **SE-BDD** algorithm using *pandas*, and *NumPy*.

7.2 Implementing Deception

This part explains the steps for the implementation of deception at the nodes. For the following steps we use *pandas*, and *NumPy* libraries.

- Designing tree-based randomization algorithm using a seed value
- Creating subsystem and generating the topology matrix
- Designing the algorithm for decoy data generator

7.3 Implementing Remapping

This part explains the steps for the implementation of different remapping mechanism at EMS.

- Implementing seed at the nodes and sharing the nodes' secret keys with EMS using *pandas*, and *NumPy*.
- Generation of seeds and creating the ID-Stack using *pandas*, and *NumPy*.
- Designing and training the LSTM model with the time-series data using *Keras*, *TensorFlow*, and *PyTorch*.
- Designing the MIP optimization algorithm using *Google OR-Tools*

7.4 Implementing Stealthy Attacks

The FDI attack vectors are generated using formal modeling. The entire constraints, as well as the system configuration, are encoded into SMT [dMB09]. *Z3 .Net API* [Z3S] is used for encoding the formalization of the proposed FDI model. The formalization is mainly encoded using Boolean (i.e., for logical constraints) and Real (e.g., for the relation between power flows or consumption with states) terms. A text file (*input* file) is used to import the system configurations and constraints. As the FDI attacks consider a critical set of the measurements, launching DoS attacks on these sensors would make the system unobservable. Thus, we implement DoS attacks considering the same FDI attacks but instead of injecting malicious data, we consider dropping the packets.

CHAPTER 8

EVALUATION

This section evaluates the performance of three different reallocation mechanisms along with the common deception mechanism. We assess them for the different matrices discussed in the following subsection.

8.1 Evaluation Methodology

We run the evaluation on IEEE 14, 57, 300 bus systems. For each test system, we generate 250 FDI attack vectors targeting different parts of the system. For the evaluation, we consider that the attacker can compromise a maximum of five buses at a time, which is statistically a pragmatic assumption. On average, a bus has a connectivity degree close to 3 [HBSB10], i.e., it is connected to approximately three other buses. Thus, it includes around four measurements/sensors [HBSB10]. Hence, the attacks limited to at most five buses can compromise up to 20 sensors, which is more than enough to make the system unobservable.

We model a two-layer communication network for each of the systems and evaluate DDAF's performance against the attacks at any of these layers, including L-0. Thus, we assess the robustness of DDAF against three different attacks: i) Reconnaissance, ii) FDI, and iii) DoS attacks. We also consider each of these attacks in three different levels: i) L-0, L-1, and L-2. We use the 365 days of synthetic IEEE bus system time-series data [idd] to train the LSTM model. To evaluate the benefit of the ID randomization, consider the percentage of *randomized* sensors from 0% to 100%. Among the rest of the sensors, 50% are reported with *fixed* IDs and the rest as *decoyed*. An attack can be launched when the control center collects the measurements and executes control routines. The period of this operation is often several seconds to a

few minutes [ZDGK10]. We implement data acquisition at 10 seconds intervals and choose an arbitrary time (i.e., five cycles) to apply different attacks.

8.1.1 Environmental Setup and Methodology

We conduct our experiments on a Dell Precision 7920 Tower workstation with Intel Xeon Silver 4110 CPU @3.0GHz, 32 GB memory, 4 GB NVIDIA Quadro P1000 GPU.

8.2 Evaluation Metrics

To assess the performance of our proposed framework, we consider the following evaluation metrics. The evaluation metrics determine the impact of randomization to deceive the adversaries against data alteration attacks.

Reconnaissance Deviation: It is defined as the l-2 norm of the difference between the actual and attacker's estimated measurement vector. A higher deviation indicates DDAF's success in misleading the attacker. Considering z_{est}^{org}, and z_{est}^{dec} as the actual, and attacker's estimated measurement vector,

$$Reconnaissance\ Deviation = ||z_{est}^{org} - z_{est}^{dec}||$$

Estimation Deviation: It is defined as the l-2 norm of measurement deviation vector at the defender's side due to the attacks. Higher estimation deviation indicates the attacker's success in misleading the defender's SE. If z_{est}^{attack} is the defender's estimated measurement vector,

$$EstimationDeviation = \left|\left|z_{est}^{org} - z_{est}^{attack}\right|\right|$$

Percentage of Stealthy Attacks (PSA): We consider an attack to be stealthy if none of the compromised sensors are detected and eliminated by the BDD during the

state estimation process. PSA is defined as the percentage of the attack that bypasses the BDD without creating any alarm.

$$PSA = \frac{\#\ of\ stealthy\ attacks}{\#\ of\ total\ attacks} \times 100$$

Percentage of Exposed Attacks (PEA): We consider an attack vector to be exposed if any of the compromised sensors becomes outlier and creates alarm at BDD. PEA is defined as:

$$PEA = \frac{\#\ of\ exposed\ attacks}{\#\ of\ total\ attacks} \times 100$$

Percentage of Unobservable Cases (PUC): PUC is defined as the percentage of attacks that create unobservability in the defender's estimation. The primary goal of DoS attacks is to increase the PUC as much as possible. PUC is defined as:

$$PUC = \frac{\#\ of\ unobservale\ cases}{\#\ of\ total\ attacks} \times 100$$

Compromised Sensor Elimination Rate (CSER): CSER is defined as the percentage of compromised sensors detected and eliminated as outliers by the BDD from all the compromised sensors during an attack. Even though an attack is exposed, it may impact the state estimation depending on the number of removed compromised sensors. Thus, a completely stealthy attack has 0% CSER, whereas an exposed attack has non zero CSER. Only an attack with 100% CSER is exposed and has no impact on the state estimation. An attack with a CSER between 0 and 100% is considered exposed but still impacts the state estimation. CSER is expressed as:

$$CSER = \frac{\#\ of\ eliminated\ compromised\ sensors}{\#\ of\ compromised\ sensors} \times 100$$

Precision: Precision, also known as the positive predictive value of a model, is an exactness quality measure metric that expresses the percentage of the data points the model says was relevant among the relevant data points. [BBR$^+$02].

Receiver Operating Characteristic (ROC) Curve: ROC curve is a way of measuring the performance of classification problems for different thresholds. The area under the ROC curve (AUC) determines separability measures for distinguishing among classes [HSG+95].

8.3 Impact of Deception against Reconnaissance Attack

This part shows how DDAF ensures the system's privacy by randomizing the IDs and then adding decoy data. In this case, we consider 50% of the sensors as *randomized*.

For the rest 50% sensors, we study the impact of treating them as *decoyed* sensors and observe how the attacker's state estimation gets deviated from the original estimation Figure 8.1 shows the reconnaissance deviation for different cases.

As decoy data are added in the nodes, not the sensors, in L-0 passive attacks, the reconnaissance deviation is zero Thus, DDAF fails to hide the system states from the attacker. Such attacks are highly expensive and infeasible as the attacker needs to compromise all the sensors individually. A more practical approach for reconnaissance is to attack the networks (L-1/L-2). In those cases, adding more decoy data increases the reconnaissance deviation and misleads the attacker. Although DDAF fails to prevent L-0 passive attacks, the later analysis shows it can completely mitigate the L-0 active attacks; and thus, such successful passive attacks become ineffective.

8.4 Evaluation of Seed-based Remapping

This section shows the robustness of seed-based remapping against FDI and DoS attacks. We consider another level as 'Random,' where the attacks are randomly implemented one of the levels. We analyze the attack impacts for different amounts

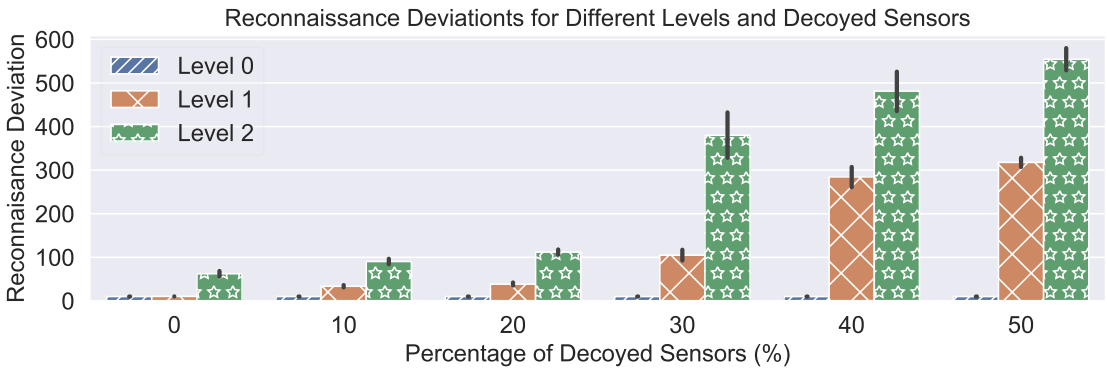


Figure 8.1: Impacts of decoy data on reconnaissance deviation for different levels. Reconnaissance deviation increases as more sensors are used to add decoy data supporting the ID randomization. A higher level considers more sensors in the node; thus, the decoy data become more effective.

(0 to 100%) of sensors used in the deception process, which we define as a percentage of deception.

8.4.1 Evaluation of FDI Attack Mitigation

We evaluate the performance of our proposed DDAF based on the FDI attack mitigation capability. First, we evaluate DDAF's performance for the 57 bus system, considering different levels as the attack points. Figure 8.2(a) shows, for L-1 and L-2 FDI attacks, PSA decreases as the percentage of deception increases. Usually, PSA is higher for level 1 attacks as there are fewer sensors to randomize in each node. Similarly, Figure 8.2(b) shows how the attacks get exposed, and PEA increases with higher levels and more randomization. However, as there is no active deception at the node level, seed-based remapping cannot detect/expose L-0 FDI attacks. Figure 8.2(c) shows the CSER of the attack vectors for different levels. For attacks at level 2 with 90-100% deception, the figure shows that almost all the compromised sensors are eliminated from the state estimation process. However, for level 1, a maximum of 60% of compromised sensors can be detected and eliminated from the system due to

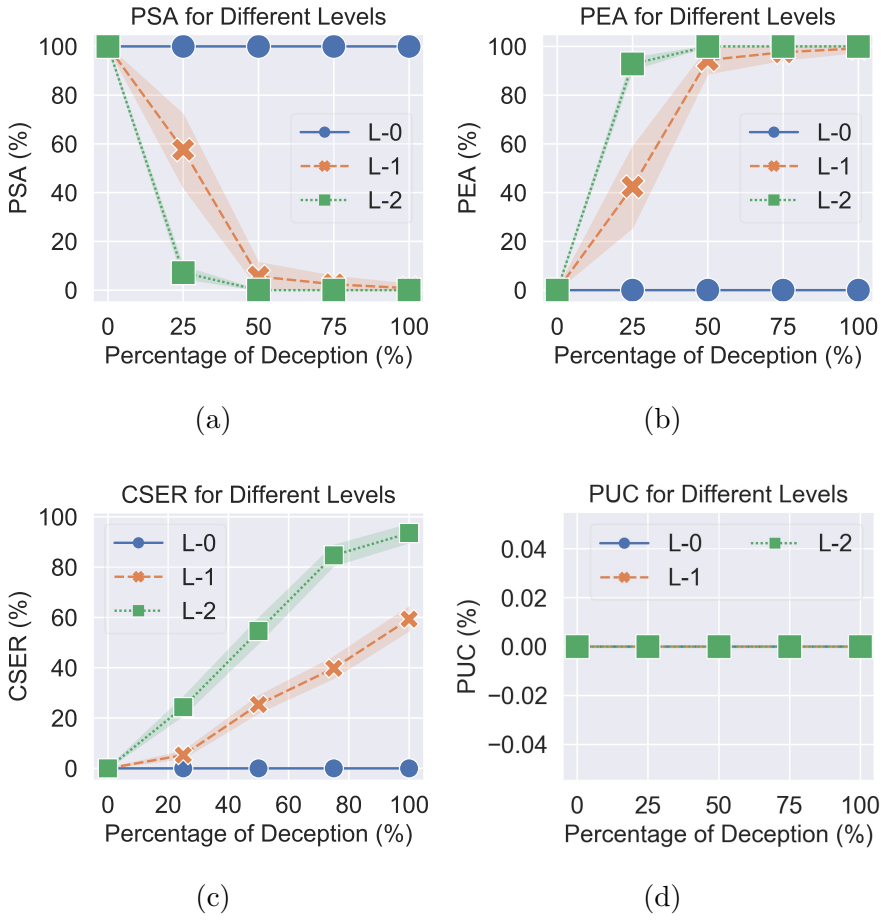


Figure 8.2: Performance evaluation of seed-based DDAF for different levels of FDI attacks, where a) PSA decreases , b) PEA increases, c) CSER increases, and d) PUC remains the same with increasing randomized sensors.

fewer randomization options. Figure 8.2(d) shows that the tree-based randomization ensures that none of the FDI attacks takes the system to unobservability, even with almost 100% CSER. We see from Figure 8.2(a-b) that 50% randomization is sufficient to make almost all the attack vectors exposed to the BDD. Still, Figure 8.2(c) shows that it can only eliminate approximately half of the compromised sensors, allowing the attack to create some deviation in the state estimation. Figure 8.3 shows the estimation deviation of the attacks at different levels. For level 1 attacks, the deviation decreases slowly with the percentage of deception. For levels 2, the deviation

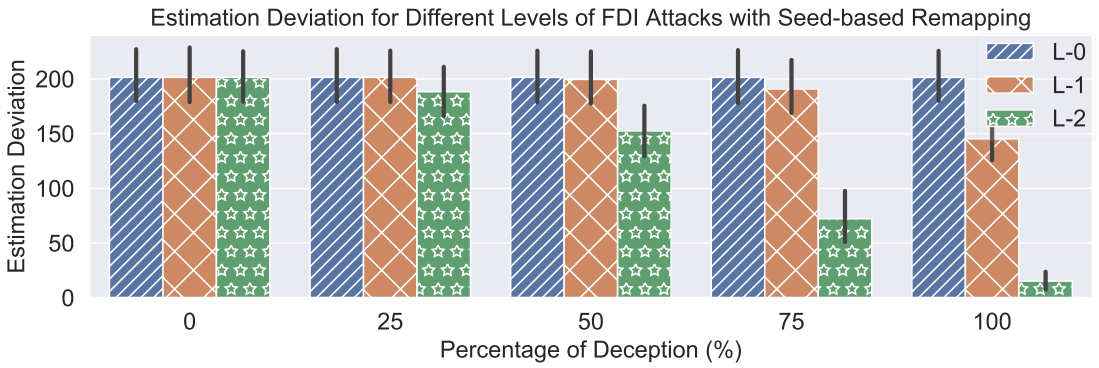


Figure 8.3: Estimation deviation of seed-based DDAF for different levels of FDI attacks, where a higher deception decreases the estimation deviation.

reduces drastically after 50%. Even though there remain a few attacks that create some deviations in the estimation, in most cases, the deviation becomes very low, making the attack less effective with more randomization.

8.4.2 Evaluation of FDI Attack Detection, and Localization

In this step, along with the attack vectors, we inject another 250 random noise vectors and analyze the attack detection and localization performance of DDAF. We skip the L-0 attacks in this case, as they remain stealthy and cannot be detected by the seed-based remapping. Figure 8.4(a) shows the ROC curve for attack detection at different levels. With 100% sensor deception, the AUC for levels 1, 2, and 3 are 0.861, 1.00, and 0.996, respectively. Figure 8.4(b) shows the ROC curve for different randomization and random level attacks. The figure shows that DDAF shows very high performance in attack detection with minimum deception. Even 25% randomization can provide an average AUC of 0.88.

Figure 8.4(c) shows the confusion matrix of detected levels at which the attack happens. Among 250 attacks at level 1, 203 are identified correctly as level 1 attacks. However, for levels 2 and 3, the model predicts with 99% and 100% accuracy, respec-

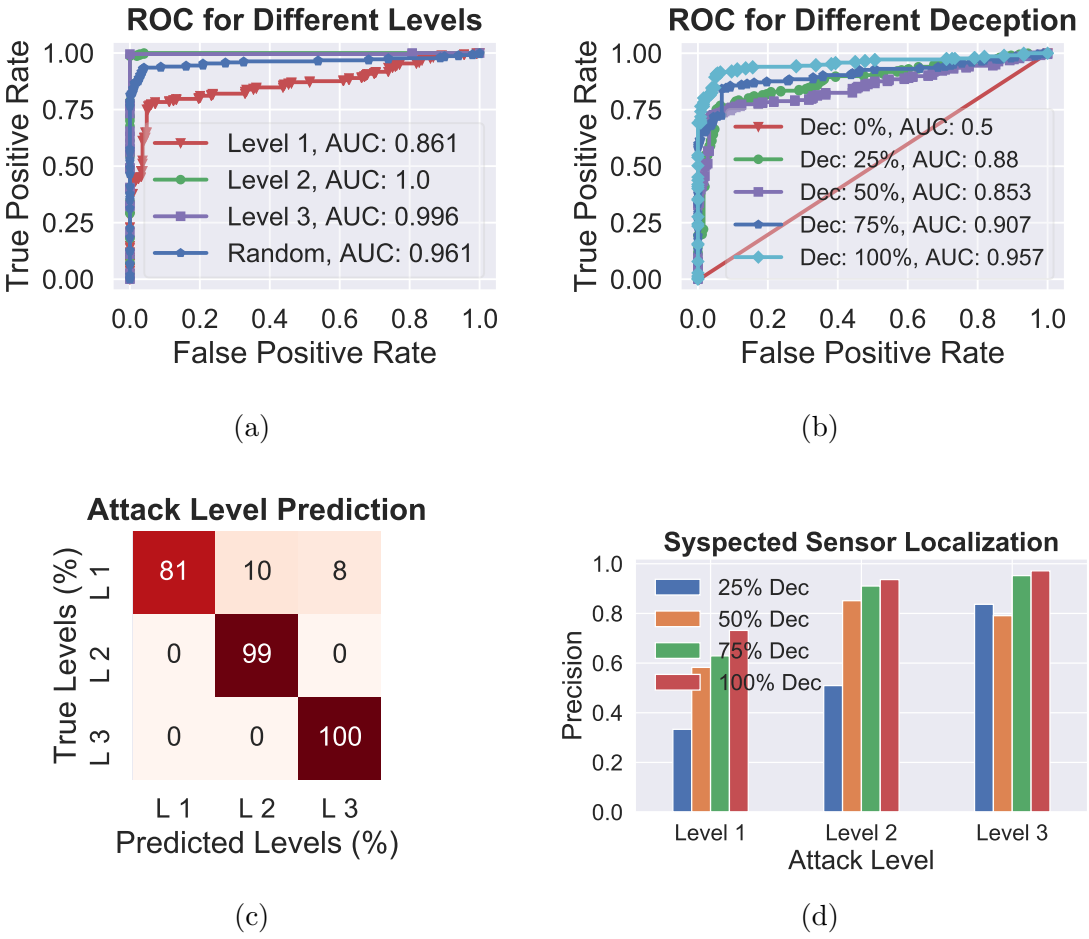


Figure 8.4: Performance evaluation of seed-based DDAF on attack detection, and localization for different levels of FDI attacks. a-b) attack detection, c) attack levels, and d) suspected sensors.

tively. Figure 8.4(d) shows the precision of the prediction on the suspected sensors' IDs for different levels of attacks and deception. The figure shows DDAF has very high precision in predicting the location of the targeted sensors.

8.4.3 Evaluation of DoS Attack Mitigation

Figure 8.5 shows seed-based DDAF's robustness against of DoS attacks. As the figure shows, in the case of the L-0 DoS attack, PUC remains 100% regardless of the

percentage of deception. Thus, seed-based DDAF fails to secure the system against DoS attacks at the sensor level. However, for network-level attacks, PUC decreases with higher deception, and for L-2 attack, it can be even almost zero.

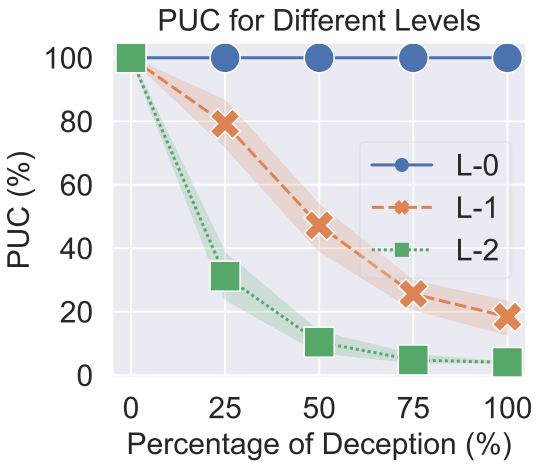
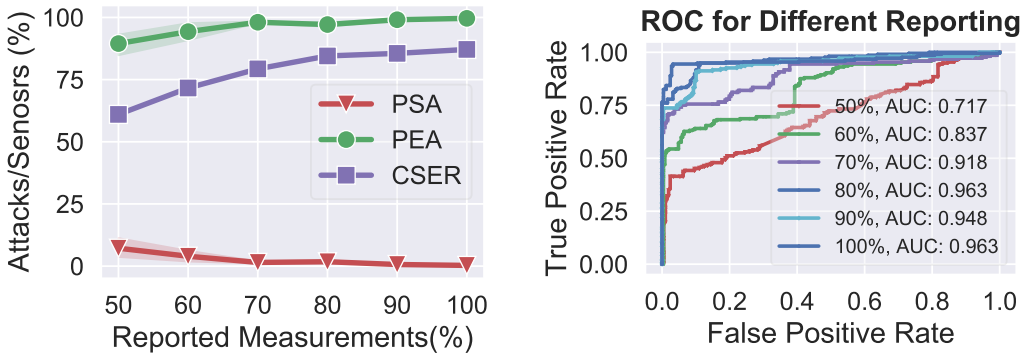


Figure 8.5: Performance evaluation of seed-based DDAF for different levels of DoS attacks, where PUC decreases with higher deception.



(a) (b)

Figure 8.6: Performance of DDAF on percentage of reported data.

8.4.4 Evaluation on Percentage of Reported Data

In the earlier evaluations, we consider that all the sensors are reporting to the EMS. However, in this part, we evaluate the framework's performance for different amounts

of sensors reporting to the EMS. Figure 8.6 shows under DDAF; only 70-80% sensors are sufficient to observe a substantial performance in PSA, PEA, and CSER and attack detection. More sensors participating in the data acquisition make it harder for the attacker to bypass the BDD.

8.4.5 Evaluation of Scalability

In this part, we analyze the impact of the network size on DDAF's performance. To evaluate the framework's scalability, we implement and analyze it on different systems to show that the framework is not system-specific and compatible with any dimensions. We consider the scalability from two perspectives.

Performance of the framework

We analyze the framework's performance on attack detection and mitigation for IEEE 14, 57, and 300 buses. Figure 8.7 (a-c) show the PSA, CSER, and ROC curve for attack detection for the three systems under 100% deception and random level attack. Similarly, it is evident from figures that irrespective of the network size, the framework shows high performance in attack detection/mitigation.

Time Complexity of the Framework

We analyze the time needed for the deception process and the reallocation process for each of the networks. Figure 8.7(d) shows the framework's time complexity for those two tasks. The figure demonstrates a linear relationship between the time complexity and the network size, proving that the framework is highly scalable and generic enough to be implemented for any hierarchical network size.

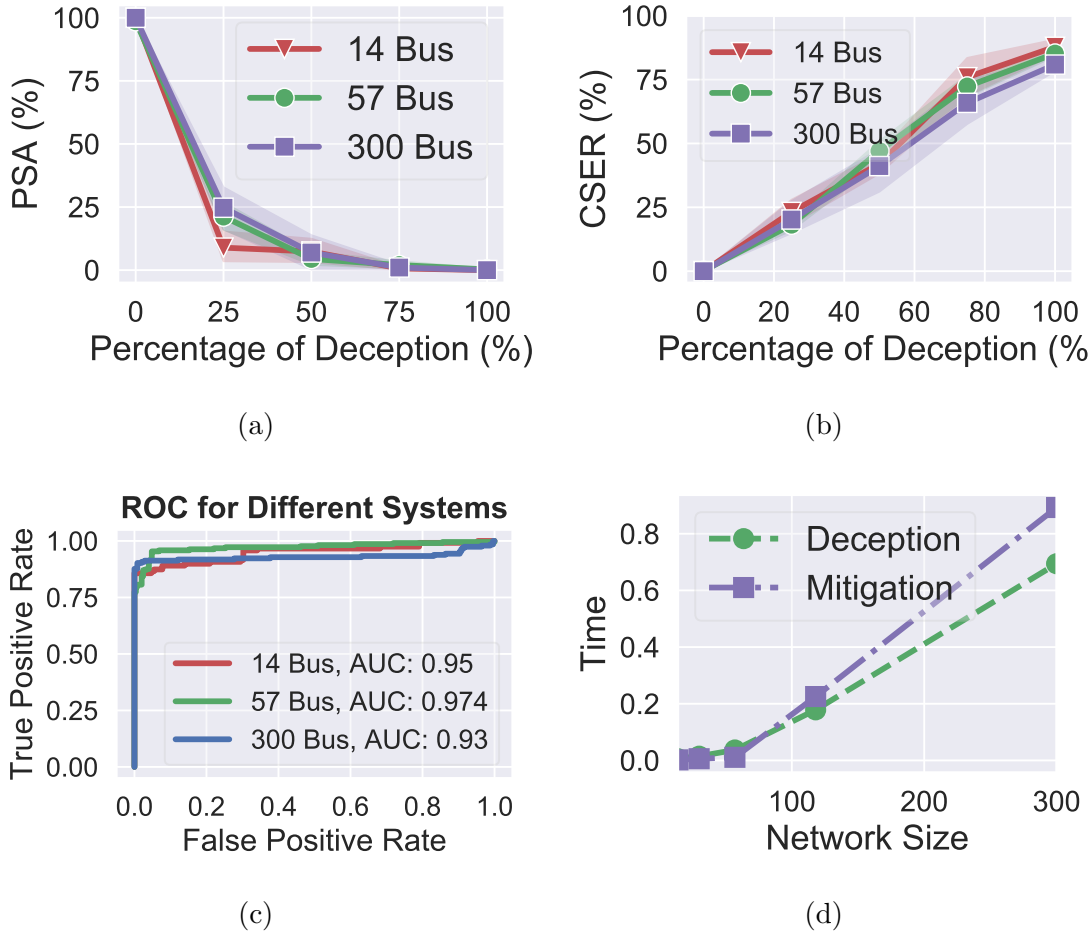


Figure 8.7: Scalability analysis of DDAF for different models. a) PSA, b) CSER c) attack detection, and d) running time (seconds) show DDAF is scalable for different test cases.

8.5 Evaluation of Prediction-based Remapping

This section shows the robustness of prediction-based remapping against FDI and DoS attacks.

8.5.1 Evaluation on FDI Attack Mitigation

We evaluate the performance of the prediction-based DDAF on the FDI attack mitigation capability. Figure 8.8(a) shows that PSA decreases as the percentage of deception

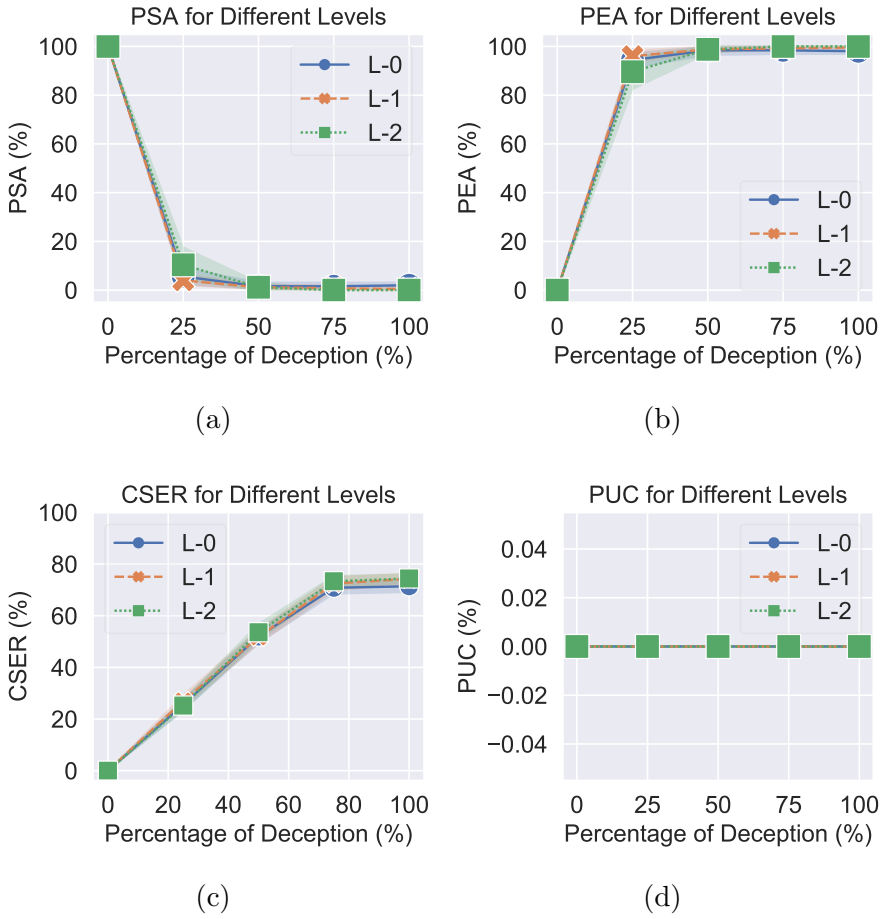


Figure 8.8: Performance evaluation of prediction-based DDAF for different levels of FDI attacks, where a) PSA decreases , b) PEA increases, c) CSER increases, and d) PUC remains the same with increasing randomized sensors.

increases at the same rate for all levels of FDI attacks. Similarly, Figure 8.8(b) shows that PEA increases drastically at the same rate for all the levels. Again, Figure 8.8(c) shows the CSER of the attack vectors for different levels, which can be almost 75%. Here, the remaining 25% compromised sensors contain low injection and are within the threshold α; thus, they are not eliminated during the remapping process. Figure 8.8(d) shows, the tree-based randomization ensures that none of the FDI attacks makes the system unobservable. Figure 8.9 shows the estimation deviation due to the FDI attacks at different levels decreases similarly with the percentage of decep-

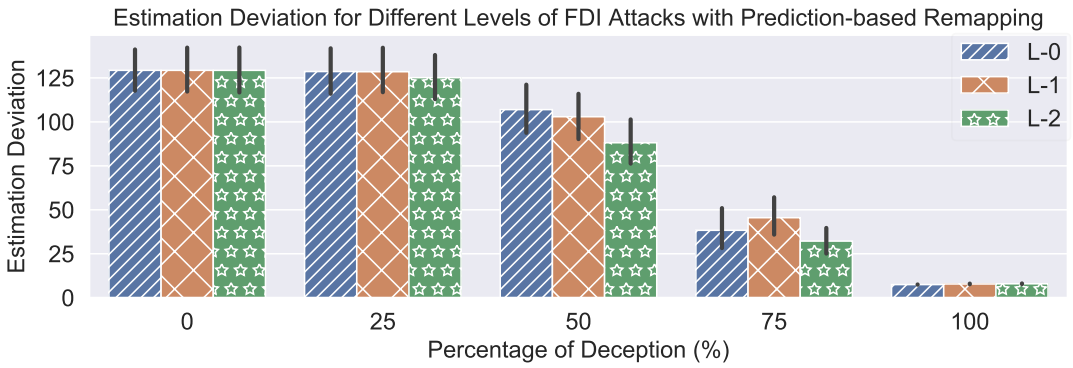


Figure 8.9: Estimation deviation of prediction-based DDAF for different levels of FDI attacks, where a higher deception and levels decrease the estimation deviation.

tion. Thus, the evaluation shows that, whereas the seed-based remapping failed to defend against the L-0 attacks, prediction-based remapping can successfully tackle such attacks with high accuracy.

The optimization algorithm takes 0.025, 0.175, 0.456, and 2.87 seconds on our computer to remap all the sensors of 14, 30, 57, and 118 bus systems. For a powerful server, this time will be much shorter. Moreover, for large-scale power systems, EMS can split the remapping problem into several subproblems for each child node's reported and predicted data. The subproblems can run parallelly, and the combined solution reveals the overall remapping pattern.

8.5.2 Evaluation of DoS Attack Mitigation

Figure 8.10 shows prediction-based DDAF's robustness against of DoS attacks. As the figure shows, in the case of DoS attacks, PUC remains 0% regardless of the percentage of deception and attack levels. Even though the seed-based remapping fails to secure the system against DoS attacks at the sensor level, prediction-based remapping can successfully mitigate such DoS attacks.

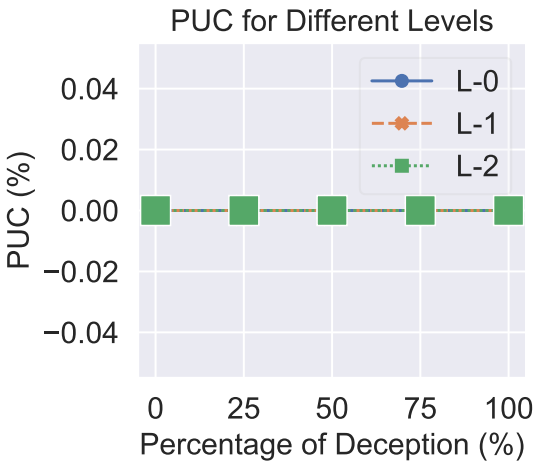


Figure 8.10: Performance evaluation of prediction-based DDAF for different levels of DoS attacks, where PUC remains zero for different deceptions.

Even if many measurements are missing but LSTM is precise enough in predicting them, there will be almost no impact on the state estimation for a short time attack. In the case of a continuous attack on some specific sensors, LSTM's prediction error will have a cascading effect (add up) on the state estimation. However, frequent updates on the sensor sets/groups and the randomization patterns will deviate the attack in another direction. Thus, the effect on a specific state variable will be mitigated. Whereas attacks can have an immediate impact on the existing approaches, our proposed framework will slow down the attack impact, providing the system operator more time to take defensive actions in the case of suspicious events.

8.6 Evaluation of Hybrid Remapping

This section shows the robustness of hybrid remapping against FDI and DoS attacks.

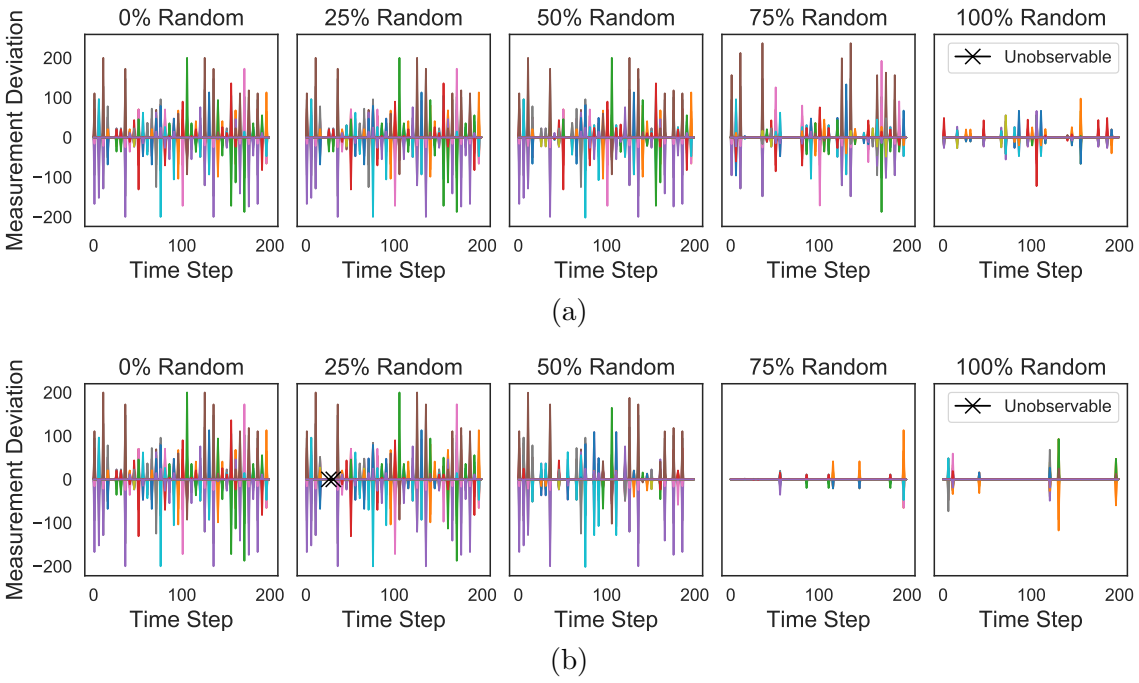


Figure 8.11: Measurement deviations for different sensors only with seed-based remapping under FDI attacks at a) Level 1, and b) Level 2.

8.6.1 Performance Evaluation of Hybrid Remapping

This section provides the evaluation of the hybrid remapping mechanism on the IEEE 57 bus system. Firstly, we analyze the impacts of FDI attacks on the system with only seed-based remapping. Later we use hybrid remapping with data filtering and data imputation to compare how these add-on features make the system more stable and robust. In this case, we implement FDI attacks every 5 seconds and considers 75% of sensors active in the system.

System only with Seed-based Remap

Figure 8.11 shows the sensor-wise estimation deviations of seed-based remapping due to consecutive FDI attacks on different parts of the system and levels. Figure 8.11(a) illustrates that the system is most vulnerable against Level 1 attacks, where the sensors are lightly randomized. Thus, due to limited randomization, the attacks

remain within the system and create significant measurement deviation, even with 100% deception. On the other hand, Figure 8.11(b) shows, the system performs better against level 2 attacks; however, still, there are a lot of spikes in the estimation deviation.

System with Hybrid Remap

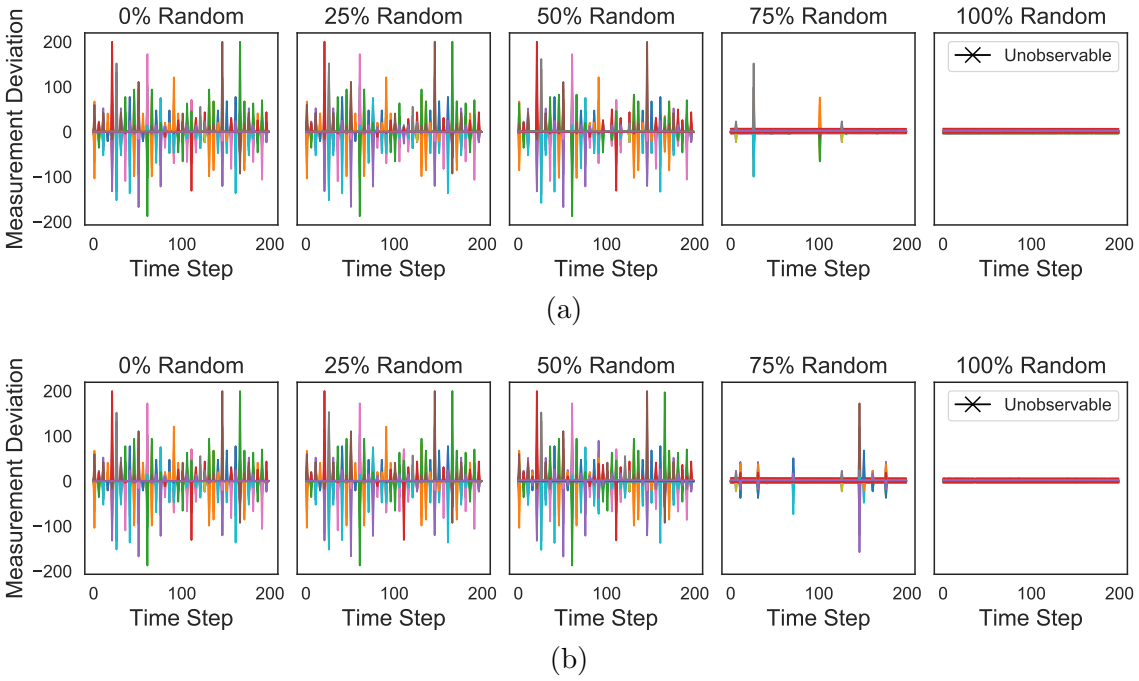


Figure 8.12: Measurement deviations for different sensors only hybrid remapping under FDI attacks at a) Level 1, and b) Level 2

This part analyzes the same scenarios with the hybrid remapping mechanism, equipped with data imputation and predictive filtering. Figure 8.12 shows the measurement deviation due to the FDI attacks, which illustrates that a hybrid remapping mechanism with a predictive filter is highly robust against stealthy FDI attacks, where the seed-based mechanism shows poor performance (Figure 8.11). The predictive filtering eradicates almost all the compromised sensors even with 75% randomization.

8.6.2 Evaluation on FDI Attack Mitigation

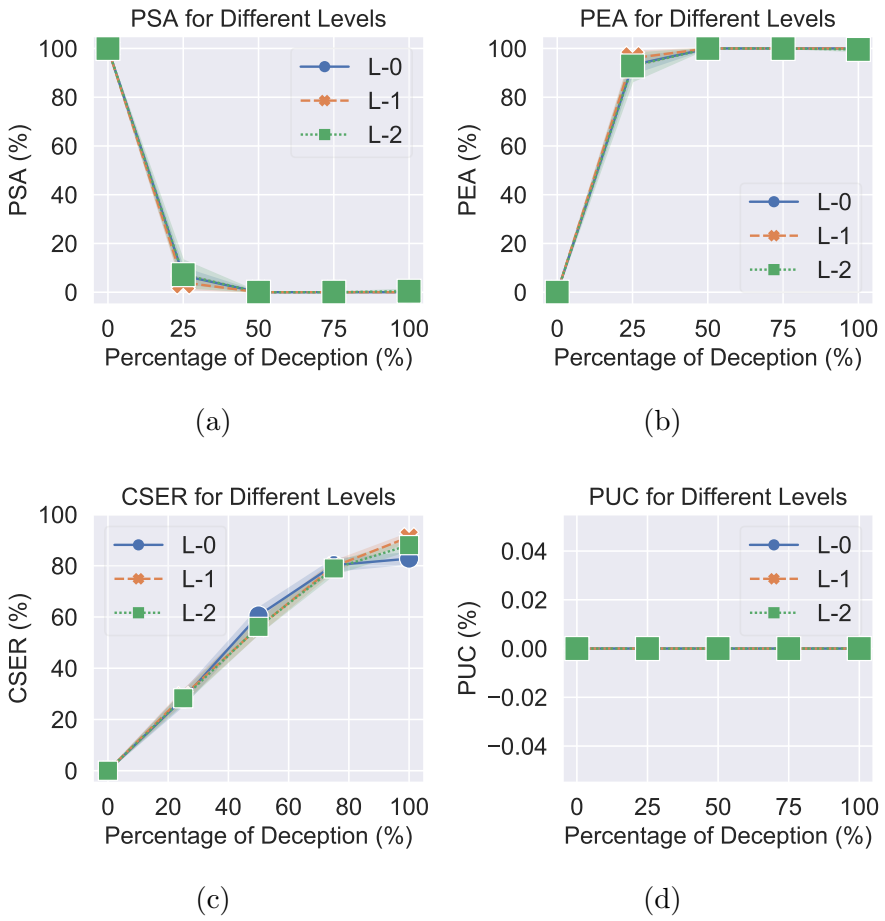


Figure 8.13: Performance evaluation of hybrid DDAF for different levels of FDI attacks, where a) PSA decreases , b) PEA increases, c) CSER increases, and d) PUC remains the same with increasing randomized sensors.

Figure 8.13(a) shows, for all levels of FDI attacks, PSA decreases as the percentage of deception increases at the same rate. Similarly, Figure 8.13(b) shows, for all the levels, PEA increases drastically at the same rate. Similarly, Figure 8.13(c) shows the CSER of the attack vectors for different levels, which can be almost 90%. Figure 8.13(d) shows, the tree-based randomization ensures that none of the FDI attacks makes the system unobservable. Figure 8.14 shows the estimation deviation due to the FDI attacks at different levels decreases similarly with the percentage of

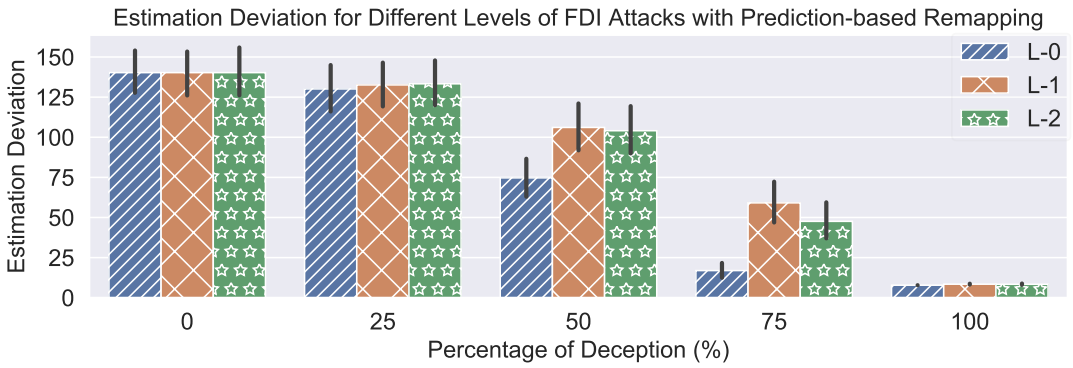


Figure 8.14: Estimation deviation of hybrid DDAF for different levels of FDI attacks, where a higher deception and levels decrease the estimation deviation.

deception. Thus, the evaluation shows, whereas the seed-based remapping failed to provide the defense against the L-0 attacks, hybrid remapping can successfully tackle such attacks with high accuracy.

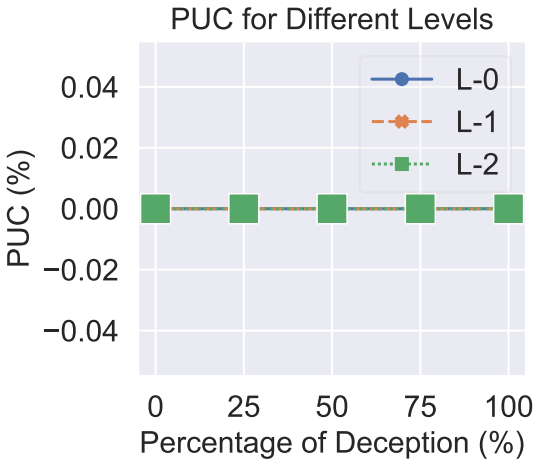


Figure 8.15: Performance evaluation of hybrid DDAF for different levels of DoS attacks, where PUC remains zero for different deceptions.

8.6.3 Evaluation of DoS Attack Mitigation

Figure 8.15 shows prediction-based DDAF's robustness against of DoS attacks. As the figure shows, in the case of DoS attacks, PUC remains 0% regardless of the

percentage of deception and attack levels. Even though the seed-based remapping fails to secure the system against DoS attacks at the sensor level, hybrid remapping can successfully mitigate such DoS attacks.

Attack Type	Defense Type	Seed-based Remapping			Prediction-based Remapping			Hybrid Remapping		
		L-0	L-1	L-2	L-0	L-1	L-2	L-0	L-1	L-2
Reconnaissance	Mitigation	●	●	●	●	●	●	●	●	●
FDI Attacks	Mitigation	●	●	●	●	●	●	●	●	●
	Detection	●	●	●	●	●	●	●	●	●
	Localization	●	●	●	●	●	●	●	●	●
DoS Attacks	Mitigation	●	●	●	●	●	●	●	●	●

● Worst/No Defense ● Moderate ● Best

Figure 8.16: Performance comparison of different remapping mechanisms.

8.7 Performance Comparison of Remapping Mechanisms

This subsection summarizes the comparative study of the resilience of different remapping strategies. Figure 8.16 shows the effectiveness of the remapping mechanisms against three types of attacks. The defense of reconnaissance attacks only depends on the successful deception, not the remapping. During the evaluation process, the deception mechanism is considered the same for all three remappings. Thus, the figure illustrates that the robustness against such passive attack improves with the levels. As there is no active deception at the sensors, DDAF does not have any defense against L-0 passive attacks.

For the same reason, the seed-based remapping performs the worst for L-0 active attacks. However, it shows moderate performance in detection, mitigation, and localization of any types of L-1 FDI attacks, and the best for L-2 FDI attacks. Similarly, it shows a similar performance in the mitigation of DoS attacks.

On the other hand, prediction-based remapping can entirely mitigate any types (FDI/DoS) of active attacks. However, it cannot detect or localize the FDI attacks; it depends on the knowledge of the nodes' randomization pattern, which is not shared with EMS in the case of prediction-based remapping.

Lastly, the hybrid remapping not only mitigates the active attacks, but it can also detect and localize the attack points as it possesses all the features of the seed-based and prediction-based remapping. As the detection and localization performance depends on the mitigation's success, the hybrid remapping utilizes the residuals and the nodes' randomization patterns to detect and localize the FDI attacks accurately.

CHAPTER 9

CONCLUDING REMARKS AND FUTURE WORK

9.1 Summary

CPSs are evolving dramatically, connecting different technologies around the world of different domains. Besides, extensive implementation of 5G will make the system almost real-time. Such dependency creates huge attack space that an attacker can exploit to achieve his/her goals. Moreover, a few SCADA-based control systems' time-constrained operations make it difficult for the nodes to implement a high-level encryption mechanism, as they have limited computational abilities. A few existing works considered deception as defense, but none could provide a complete solution against different cyberattacks.

This thesis proposes DDAF, a generic deceptive defense-based secure data acquisition framework for CPSs hierarchical communication networks, ensuring the CIA (confidentiality, integrity, and availability) triad for the sensor data. Using SDN controllers at the network nodes, DDAF can deceive an attacker by replacing the original sensor IDs with the deceptive IDs and adding intelligently crafted decoy data. Such deception allows EMS to detect, mitigate, and localize the stealthy cyberattacks such as FDI attacks, DoS attacks, and reconnaissance attacks. Initially, we propose two types of unique remapping mechanisms, where each has special features with a few limitations. Finally, we merge them to present a hybrid remapping, highly resilient against both the local sensor-level attacks and the network-level attacks. Experimental results on standard IEEE power systems show that the framework can detect, mitigate, and locate most stealthy cyberattacks with high accuracy. Besides, the framework is highly scalable, which can be implemented for large scale networks.

9.2 Future Work

The future works contain the following subs-tasks:

Evaluation for different CPSs and cyberattacks: We implemented DDAF on the power grid network; however, it can be implemented on any CPSs network that incorporates networked control systems. Thus, in future work, we plan to evaluate DDAF's performance on other CPSs, i.e., water treatment plants, oil and gas systems, manufacturing plant systems, mass transit systems, etc. This thesis considers three types of attacks (e.g., FDI, DoS, and reconnaissance) in the evaluation; however, we will study the robustness of the framework under other emerging, impactful, and zero-day cyberattacks.

Test-bed implementation: The current implementation is limited to the simulation only. Therefore, we will implement the framework on a real test-bed in the future, developed for the power system implementing both the physical network and the communication network. Besides, we will study the overhead and latency due to the deception at the nodes by implementing the framework in a network simulator or on real SDN controllers. We plan to implement the communication network in NS3 and verify the results on real SDN controller devices.

Optimization of SDN controller placement: Moreover, most of the legacy CPSs may not have the capability to support the SDN application at all the network nodes. On the other hand, upgrading the whole system may not be a feasible solution considering the associated cost and benefit. Thus, in the future, we will study the optimal placement of the SDN controller (upgrading the nodes), which will maximize the resiliency with minimum cost. We will also develop a tool to design the optimal network architecture for any CPSs that will provide the best defense running with DDAF.

BIBLIOGRAPHY

[AE04] Ali Abur and Antonio Gomez Exposito. *Power system state estimation: theory and implementation*. CRC press, 2004.

[AFA+20] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 207–218, 2020.

[AFYES18] Abdelrahman Ayad, Hany EZ Farag, Amr Youssef, and Ehab F El-Saadany. Detection of false data injection attacks in smart grids using recurrent neural networks. In *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2018.

[AKK+20] Mamoun Alazab, Suleman Khan, Somayaji Siva Rama Krishnan, Quoc-Viet Pham, M Praveen Kumar Reddy, and Thippa Reddy Gadekallu. A multidirectional lstm model for predicting the stability of a smart grid. *IEEE Access*, 8:85454–85463, 2020.

[ALD13] Ilge Akkaya, Edward A Lee, and Patricia Derler. Model-based evaluation of gps spoofing attacks on power grid sensors. In *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6. IEEE, 2013.

[AQS+15] Salman Ali, Saad Bin Qaisar, Husnain Saeed, Muhammad Farhan Khan, Muhammad Naeem, and Alagan Anpalagan. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. *Sensors*, 15(4):7172–7205, 2015.

[BAF20] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Hall spoofing: A non-invasive dos attack on grid-tied solar inverter. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1273–1290, 2020.

[BBR+02] Jerry R Barker, Michael Bollman, Paul L Ringold, Jennifer Sackinger, and Steven P Cline. Evaluation of metric precision for a riparian forest survey. *Environmental monitoring and assessment*, 75(1):57–78, 2002.

[DBDJH14] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. *Neural network design*. Martin Hagan, 2014.

[DKG+14] Nils Dorsch, Fabian Kurtz, Hanno Georg, Christian Hägerling, and Christian Wietfeld. Software-defined networking for smart grid communications: Applications, challenges and advantages. In *2014 IEEE international conference on smart grid communications (SmartGridComm)*, pages 422–427. IEEE, 2014.

[dMB09] Leonardo de Moura and Nikolaj Bjørner. Satisfiability modulo theories: An appetizer. In *Brazilian Symposium on Formal Methods*, 2009.

[Doh16] Jim Doherty. *SDN and NFV simplified: a visual guide to understanding software defined networks and network function virtualization*. Addison-Wesley Professional, 2016.

[Dri15] D. Drinkwater. Stuxnet-style attack on us smart grid could cost government $1 trillion, 2015.

[GCS19] Jairo Giraldo, Alvaro Cardenas, and Ricardo G Sanfelice. A moving target defense to detect stealthy attacks in cyber-physical systems. In *2019 American Control Conference (ACC)*, pages 391–396. IEEE, 2019.

[GDU+12] Stephen Groat, Matthew Dunlop, William Urbanksi, Randy Marchany, and Joseph Tront. Using an ipv6 moving target defense to protect the smart grid. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–7. IEEE, 2012.

[GES02] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[GPGV14] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet & Information Systems*, 8(12), 2014.

[GWS20] Paul Griffioen, Sean Weerakkody, and Bruno Sinopoli. A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, 2020.

[HBSB10] Paul Hines, Seth Blumsack, E Cotilla Sanchez, and Clayton Barrows. The topological and electrical structure of power grids. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2010.

[HLY+19] Yan Hu, Hong Li, Hong Yang, Yuyan Sun, Limin Sun, and Zhiliang Wang. Detecting stealthy attacks against industrial control systems based on residual skewness analysis. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):74, 2019.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[HSD+20] Nur Imtiazul Haque, Md Hasan Shahriar, Md Golam Dastgir, Anjan Debnath, Imtiaz Parvez, Arif Sarwat, and Mohammad Ashiqur Rahman. Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey. *arXiv preprint arXiv:2010.00661*, 2020.

[HSG+95] Susan L Hartmaier, Philip D Sloane, Harry A Guess, Gary G Koch, C Madeline Mitchell, and Charles D Phillips. Validation of the minimum data set cognitive performance scale: agreement with the mini-mental state examination. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 50(2):M128–M133, 1995.

[idd] iddaf. `https://sites.google.com/view/iddaf/home`.

[IEC] Iec 61850-power utility automation. `https://www.iec.ch/smartgrid/standards/`.

[IEEa] Ieee 14-bus system. `https://www.icseg.iti.illinois.edu/ieee-14-bus-system/`.

[ieeb] Ieee bus systems. `https://icseg.iti.illinois.edu/power-cases/`.

[JH20] Aleksandar Jovicic and Gabriela Hug. Linear state estimation and bad data detection for power systems with rtu and pmu measurements. *arXiv preprint arXiv:2001.10764*, 2020.

[jup] The jupyter notebook. https://jupyter.org/.

[KKC18] Nakyoung Kim, Minkyung Kim, and Jun Kyun Choi. Lstm based short-term electricity consumption forecast with daily load profile sequences. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 136–137. IEEE, 2018.

[Kov16] E Kovacs. Blackenergy malware used in ukraine power grid attacks. securityweek, 2016.

[Lan11] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[LDZ14] Yu Li, Rui Dai, and Junjie Zhang. Morphing communications of cyber-physical systems towards moving-target defense. In *2014 IEEE International Conference on Communications (ICC)*, pages 592–598. IEEE, 2014.

[LHQZ20] Yuancheng Li, Wendan Huo, Rixuan Qiu, and Jing Zeng. Efficient detection of false data injection attack with invertible automatic encoder and long-short-term memory. *IET Cyber-Physical Systems: Theory & Applications*, 5(1):110–118, 2020.

[LKI18] Hui Lin, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. Raincoat: Randomization of network communication in power grid cyber infrastructure to mislead attackers. *IEEE Transactions on Smart Grid*, 10(5):4893–4906, 2018.

[LLWC16] Beibei Li, Rongxing Lu, Wei Wang, and Kim-Kwang Raymond Choo. Ddoa: A dirichlet-based detection scheme for opportunistic attacks in smart grid cyber-physical system. *IEEE Transactions on Information Forensics and Security*, 11(11):2415–2425, 2016.

[LNR09] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. In *ACM Conference on Computer and Communications Security (CCS)*, pages 21–32, 2009.

[LNR11] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):1–33, 2011.

[LW20] Bo Liu and Hongyu Wu. Optimal d-facts placement in moving target defense against false data injection attacks. *IEEE Transactions on Smart Grid*, 2020.

[LY18] Subhash Lakshminarayana and David KY Yau. Cost-benefit analysis of moving-target defense in power grids. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 139–150. IEEE, 2018.

[LZHZ20] Hui Lin, Jianing Zhuang, Yih-Chun Hu, and Huayu Zhou. Defrec: Establishing physical function virtualization to disrupt reconnaissance of power grids' cyber-physical infrastructures. In *The Proceedings of 2020 Network and Distributed System Security Symposium (NDSS)*, 2020.

[MB09] Miles A McQueen and Wayne F Boyer. Deception used for cyber defense of control systems. In *2009 2nd Conference on Human System Interactions*, pages 624–631. IEEE, 2009.

[MMV17] Ramin Moslemi, Afshin Mesbahi, and Javad Mohammadpour Velni. A fast, decentralized covariance selection-based approach to detect cyber attacks in smart grids. *IEEE Transactions on Smart Grid*, 9(5):4930–4941, 2017.

[MVK+16] Hoda Maleki, Saeed Valizadeh, William Koch, Azer Bestavros, and Marten Van Dijk. Markov modeling of moving target defense games. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pages 81–92, 2016.

[MYLR12] Chris YT Ma, David KY Yau, Xin Lou, and Nageswara SV Rao. Markov game analysis for attack-defense of power networks under possible misinformation. *IEEE Transactions on Power Systems*, 28(2):1676–1686, 2012.

[NHS+20] AKM Newaz, Nur Imtiazul Haque, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. Adversarial attacks to machine learning-based smart healthcare systems. *arXiv preprint arXiv:2010.03671*, 2020.

[NSRU19] AKM Iqtidar Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. Healthguard: A machine learning-based security framework for smart healthcare systems. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 389–396. IEEE, 2019.

[NSRU20] AKM Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *arXiv preprint arXiv:2005.07359*, 2020.

[PAG17] Aswin Chidambaram Pappa, Aditya Ashok, and Manimaran Govindarasu. Moving target defense for securing smart grid communications:

Architecture, implementation & evaluation. In *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2017.

[Pow21] Powerworld, 2021.

[RASB14] Mohammad Ashiqur Rahman, Ehab Al-Shaer, and Rakesh B Bobba. Moving target defense for hardening the security of the power system state estimation. In *Proceedings of the First ACM Workshop on Moving Target Defense*, pages 59–68, 2014.

[RASK14] M. A. Rahman, E. Al-Shaer, and R. Kavasseri. Impact analysis of topology poisoning attacks on economic operation of the smart power grid. In *International Conference on Distributed Computing Systems (ICDCS)*, July 2014.

[RSJM19] Mohammad Ashiqur Rahman, Md Hasan Shahriar, Mohamadsaleh Jafari, and Rahat Masum. Novel attacks against contingency analysis in power grids. *arXiv preprint arXiv:1911.00928*, 2019.

[RSM19] M Ashiqur Rahman, Md Hasan Shahriar, and Rahat Masum. False data injection attacks against contingency analysis in power grids: poster. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 343–344, 2019.

[sec] Rsa securid suite. `https://www.rsa.com/en-us/products/rsa-securid-suite`.

[SHRAJ20] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso Jr. G-ids: Generative adversarial networks assisted intrusion detection system. *arXiv preprint arXiv:2006.00676*, 2020.

[SM19] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[Sob19] B. Sobczak. 'denial of service' attack caused grid cyber disruption: Doe, 2019.

[SPA+18] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. A survey on sensor-based threats to internet-of-

things (iot) devices and applications. *arXiv preprint arXiv:1802.02041*, 2018.

[SSU16] Md Hasan Shahriar, Md Jawwad Sadiq, and Md Forkan Uddin. Stability analysis of grid connected pv array under maximum power point tracking. In *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*, pages 499–502. IEEE, 2016.

[SSVE04] Andrew Simmonds, Peter Sandilands, and Louis Van Ekert. An ontology for network security attacks. In *Asian Applied Computing Conference*, pages 317–323. Springer, 2004.

[TTGL18] Jue Tian, Rui Tan, Xiaohong Guan, and Ting Liu. Enhanced hidden moving target defense in smart grids. *IEEE Transactions on Smart Grid*, 10(2):2208–2223, 2018.

[TTS+71] Sun Tzu, Sun Tzu, Wu Sun, Sun Cu Vu, et al. *The art of war*, volume 361. Oxford University Press, USA, 1971.

[VK19] Hemant Verma and Sudhir Kumar. An accurate missing data prediction method using lstm based deep learning for health care. In *Proceedings of the 20th international conference on distributed computing and networking*, pages 371–376, 2019.

[VVP+06] D. Van Hertem, J. Verboomen, K. Purchala, R. Belmans, and W. L. Kling. Usefulness of dc power flow for active power flow analysis with flow controlling devices. In *The 8th IEE International Conference on AC and DC Power Transmission*, pages 58–62, March 2006.

[WXH+17] Y. Wu, Y. Xiao, F. Hohn, Lars N., J. Wang, and W. Zhao. Bad data detection using linear wls and sampled values in digital substations. *IEEE Transactions on Power Delivery*, 33(1):150–157, 2017.

[WZYC16] Shizhong Wang, Yuanrui Zhang, Zhihua Yang, and Yixiang Chen. A graphical hierarchical cps architecture. In *2016 International Symposium on System and Software Reliability (ISSSR)*, pages 97–105. IEEE, 2016.

[XWG+17] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du. Achieving efficient detection against false data injection attacks in smart grid. *IEEE Access*, 5, 2017.

[YHK+12] Mark Yampolskiy, Peter Horvath, Xenofon D Koutsoukos, Yuan Xue, and Janos Sztipanovits. Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach. In *2012 5th International Symposium on Resilient Control Systems*, pages 55–62. IEEE, 2012.

[YZL+20] Liqun Yang, Xiaoming Zhang, Zhi Li, Li Zhoujun, and He Yueying. Detecting bi-level false data injection attack based on time series analysis method in smart grid. *Computers & Security*, page 101899, 2020.

[Z3S] Z3: An efficient smt solver. In *Microsoft Research*. http://research.microsoft.com/en-us/um/redmond/projects/z3/.

[ZDGK10] Ce Zheng, Yimai Dong, Ozgur Gonen, and Mladen Kezunovic. Data integration used in new applications and control center visualization tools. In *IEEE PES General Meeting*, pages 1–7. IEEE, 2010.

[ZDY+19] Zhenyong Zhang, Ruilong Deng, David Yau, Peng Cheng, and Jiming Chen. On effectiveness of detecting fdi attacks on power grid using moving target defense. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2019.