11-5-2018

Secure Control and Operation of Energy Cyber-Physical Systems Through Intelligent Agents

Mohamad El Hariri
melha003@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SECURE CONTROL AND OPERATION OF ENERGY CYBER-PHYSICAL SYSTEMS THROUGH INTELLIGENT AGENTS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Mohamad El Hariri

2018

To: Dean John Volakis
College of Engineering and Computing

This dissertation, written by Mohamad El Hariri, and entitled Secure Control and Operation of Energy Cyber-Physical Systems through Intelligent Agents, having been approved in respect to style and intellectual contents, is referred to you for judgment.

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

Kemal Akkaya

Mohammed Hadi

Sakhrat Khizroev

Mark Roberts

Osama A. Mohammed, Major Professor

Date of Defense: November 05, 2018

The dissertation of Mohamad El Hariri is approved.

Dean John Volakis
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2018

DEDICATION

To the ones whom I cannot untether myself from. Those who stayed within my heart eager for my return. Those who were once a song placed within my ears and have never left.

To my father, Ahmed, my mother, Amal, my siblings, Hiba, Sereen, and Ali, and my aunt Sahar.

ACKNOWLEDGMENTS

I would like to thank my major advisor, Professor Osama Mohammed, for supervising me during my doctoral research and studies at FIU. He has been and endless source ideas which he has always provided and that was the foundation of my success in this research. His guidance, suggestions, and useful discussions have made the difference and made this work of very high quality. I would also like to thank Professor Osama Mohammed for providing me with financial support through appointing me as a Research Assistant at the Energy Systems Research Laboratory at FIU. I have gained a lot of skills and experience at ESRL, which has first-class hardware and software facilities needed to verify my research results experimentally. This helped me complete my dissertation research producing high-level publications and patents. Also, as my doctoral research started to gain traction, Professor Mohammed provided me opportunities to grow within my professional career inside and outside the university. This included; sending me to many international conferences and meetings, being elected in leadership position as the IEEE Power and Energy Society Student Branch Chapter, and allowing me to serve as judge or organizer in several professional conferences and competition activities.

I would also like to thank my committee members for their support and discussion. I would also like to acknowledge the support for the FIU graduate school for providing me with the dissertation year fellowship during my final year in the program.

Finally, thanks to all my colleagues at ESRL, who created a professional and collaborative environment that helped me reach all my research objectives.

ABSTRACT OF THE DISSERTATION

SECURE CONTROL AND OPERATION OF ENERGY CYBER-PHYSICAL SYSTEMS THROUGH INTELLIGENT AGENTS

by

Mohamad El Hariri

Florida International University, 2018

Miami, Florida, USA

Professor Osama A. Mohammed, Major Professor

The operation of the smart grid is expected to be heavily reliant on microprocessor-based control. Thus, there is a strong need for interoperability standards to address the heterogeneous nature of the data in the smart grid. In this research, we analyzed in detail the security threats of the Generic Object Oriented Substation Events (GOOSE) and Sampled Measured Values (SMV) protocol mappings of the IEC 61850 data modeling standard, which is the most widely industry-accepted standard for power system automation and control. We found that there is a strong need for security solutions that are capable of defending the grid against cyber-attacks, minimizing the damage in case a cyber-incident occurs, and restoring services within minimal time.

To address these risks, we focused on correlating cyber security algorithms with physical characteristics of the power system by developing intelligent agents that use this knowledge as an important second line of defense in detecting malicious activity. This will complement the cyber security methods, including encryption and authentication. Firstly, we developed a physical-model-checking algorithm, which uses artificial neural networks to identify switching-related attacks on power systems based on load flow characteristics.

Secondly, the feasibility of using neural network forecasters to detect spoofed sampled values was investigated. We showed that although such forecasters have high spoofed-data-detection accuracy, they are prone to the accumulation of forecasting error. In this research, we proposed an algorithm to detect the accumulation of the forecasting error based on lightweight statistical indicators. The effectiveness of the proposed algorithms was experimentally verified on the Smart Grid testbed at FIU. The test results showed that the proposed techniques have a minimal detection latency, in the range of microseconds.

Also, in this research we developed a network-in-the-loop co-simulation platform that seamlessly integrates the components of the smart grid together, especially since they are governed by different regulations and owned by different entities. Power system simulation software, microcontrollers, and a real communication infrastructure were combined together to provide a cohesive smart grid platform. A data-centric communication scheme was selected to provide an interoperability layer between multi-vendor devices, software packages, and to bridge different protocols together.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF ACRONYMS

ACRONYMS	DETAILS
AC	Alternating Current
AE	Available Energy
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
ASDU	Application Service Data Unit
ASN.1	Abstract Syntax Notation One
APDU	Application Protocol Data Unit
API	Application Programming Interface
APPID	Application ID
BBB	Beagelbone Black
BBH	Battery-based Load Hiding
BER	Basic Encoding Rules
CB	Circuit Breaker
CIM	Common Interface Model
CP	Current Price
CPU	Central Processing Unit
confRev	Configuration Revision
DC	Direct Current

DDS	Data Distribution Service
DER	Distributed Energy Resource
DES	Data Encryption Standard
DoE	Department of Energy
DoS	Denial of Service
DNP	Distributed Network Protocol
ECU	Electronic Control Unit
EFT	Electromagnetic Fast Transient
eMMC	Embedded Multi Media Card
EMS	Energy Management System
EV	Electric Vehicle
FAN	Field Area Network
FDIA	False Data Injection Attack
FIPA	Foundation for Intelligent Physical Agents
FPGA	Field-Programmable Gate Array
G2V	Grid to Vehicle
GDS	Global Data Space
gocbRef	GOOSE control block reference
goID	GOOSE ID
GOOSE	Generic Object Oriented Substation Event
goosePDU	GOOSE Protocol Data Unit
GPIO	General Purpose Input Output

GPS	Global Positioning System
HAN	Home Area Network
HMI	Human Machine Interface
HSR	High-availability Seamless Redundancy
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure
HVDC	High Voltage DC
ICS	Industrial Control System
IDL	Interface Definition Language
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
IP	Internet Protocol
ISO	International Organization of Standardization
LAN	Local Area Network
LB	Lower Bound
LED	Light Emitting Diode
LLD	Local Load Demand
LLH	Load-based Load Hiding
LV	Low Voltage

MAC	Media Access Control
MC	Maximum Capacity
MCC	Microgrid Control Center
MCCA	MCC Agent
MMS	Manufacturing Message Specification
MITM	Man in the Middle
MISRA	Motor Industry Software Reliability Association
MSE	Mean Square Error
MU	Merging Unit
ndsCom	Needs Commissioning
NAN	Neighbor Area Network
NERC	North American Electric Reliability Corporation
NHP	Next Hour Price
NILM	Non-Intrusive Load Monitoring
NIST	National Institute of Standards and Technology
NMRSE	Normalized Mean Root Square Error
NN	Neural Network
NN-F	Neural Network Forecaster
noASDU	Number of ASDU
numDatSetEntries	Number of Dataset Entries
OMG	Object Management Group
OSI	Open System Interconnect

P	Active Power
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
PSO	Particle Swarm Optimization
PV	Photovoltaic
Q	Reactive Power
QoS	Quality of Service
R-GOOSE	Routable GOOSE
RMS	Root Mean Square
RSA	Rivest Shamir Adleman
RTI	Real Time Automation
SAE	Society of Automotive Engineers
SAS	Substation Automation System
SBC	Single Board Computer
SCADA	Supervisory Control and Data Acquisition
seqData	Sequence Data
SFTP	Secure File Transfer Protocol
smpCnt	Sample Counter
smpSynch	Sample Synchronization
SMV	Sampled Measured Value
SoC	State of Charge

sqNum	Sequence Number
SSH	Secure Shell
stNum	Status Number
svID	Sampled Value ID
t	Time
TC	Technical Committee
TCP	Transmission Control Protocol
TL	Transmission Line
TLL	Transmission Line Loading
TLS	Transport Layer Security
TR	Technical Report
UB	Upper Bound
V2G	Vehicle to Grid
VFD	Variable Frequency Drive
VLAN	Virtual LAN
WG	Working Group
WAMPAC	Wide Area Monitoring, Protection, and Control
WAN	Wide Area Network
WLS	Weighted Least Square
XML	eXtensible Markup Language

Chapter 1 Introduction

The smart grid's reliable operation is a function of the configuration and cyber-physical nature of its constituting system components. The smart grid, thus, is a complex system, in which communication and information technologies are the main propellers of its evolution and expansion [1][2]. Lately, the engineering literature showed that researchers are utilizing communication technologies for adding intelligence to the smart grid, allowing system operators to do more with the current physical assets of the grid.

Generically, the communication infrastructure of the smart grid could be divided into several levels depending on its function and location, as seen in Figure 1.1. The lower level of the communication infrastructure consists of the Home Area Network, or the HAN. The HAN extends the capabilities of the smart grid to reach inside the home by enabling two-way information exchange between appliances inside the home and the utility grid for better energy usage. This enables applications, such as building management systems, control of smart appliances or loads, and energy management systems. Control networks of electric vehicle parks and even small microgrids are considered to be on the same level of the HAN. The second layer is the Neighbor Area Network, or the NAN. The NAN is implemented within the distribution network. The NAN allows energy providers to dispatch power to customers based on load demand data collected from smart meters. As can be seen in Figure 1.1, the NAN connects neighboring meters together. On the higher level, the Field Area Network and Wide Area Network, or the FAN and WAN, respectively, consist of groups of NANs, field devices, substation automation networks, and generation control.

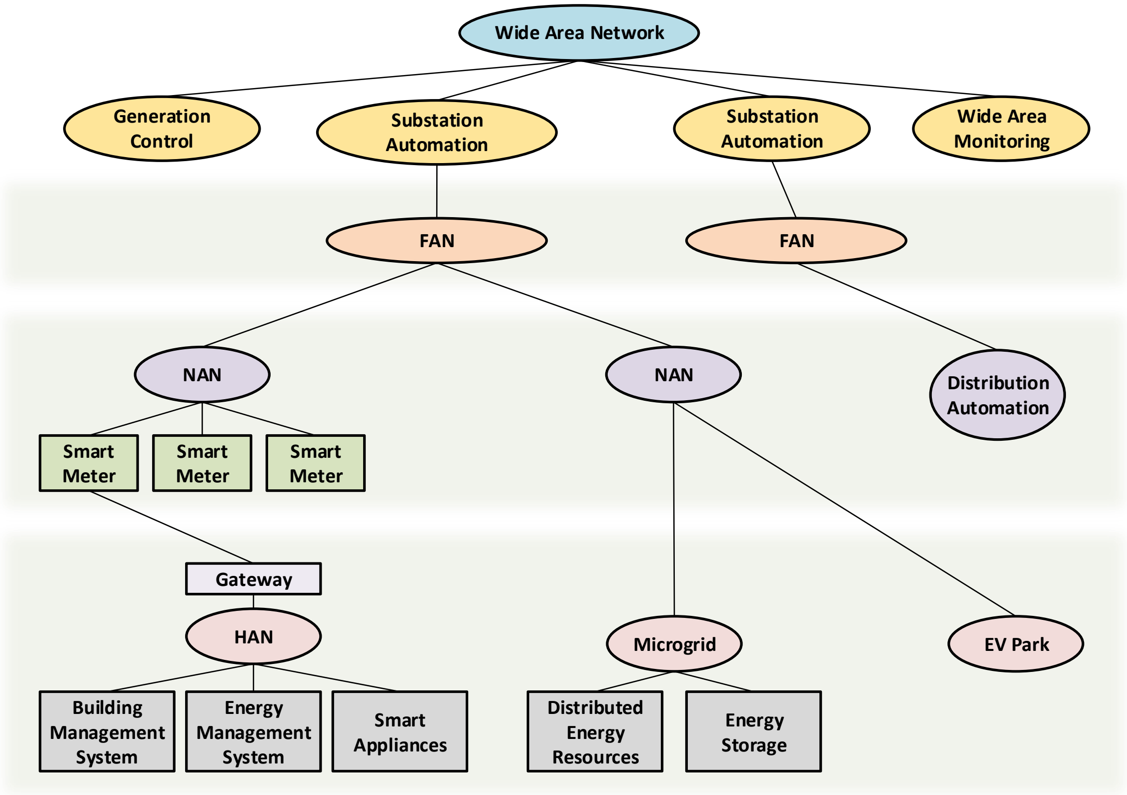


Figure 1.1: Smart Grid Cyber Infrastructure.

This communication infrastructure enables the migration from the conventional centralized control networks to decentralized and eventually distributed control. In centralized control, a single server collects all the measurements, performs the control logic processing, and issues the commands to field devices [3][4]. The server in this case is a single point of failure. Therefore, as can be seen in Figure 1.2, a communication failure or a successful attack against centralized control could lead to serious damage and loss of service. Unlike centralized control, the decentralized control contributes to the security of the power system by avoiding single points of failure. In decentralized control, lower-level components perform tasks based on local information and are supervised by a higher-level agent responsible for their area [5]. Finally, in distributed control, each node exchanges information and cooperates with neighbor nodes in a peer-to-peer fashion to improve the stability of the system and eventually minimize the attack surface, as can be seen in Figure 1.2 [6]. The three control schemes are shown in Figure 1.3.

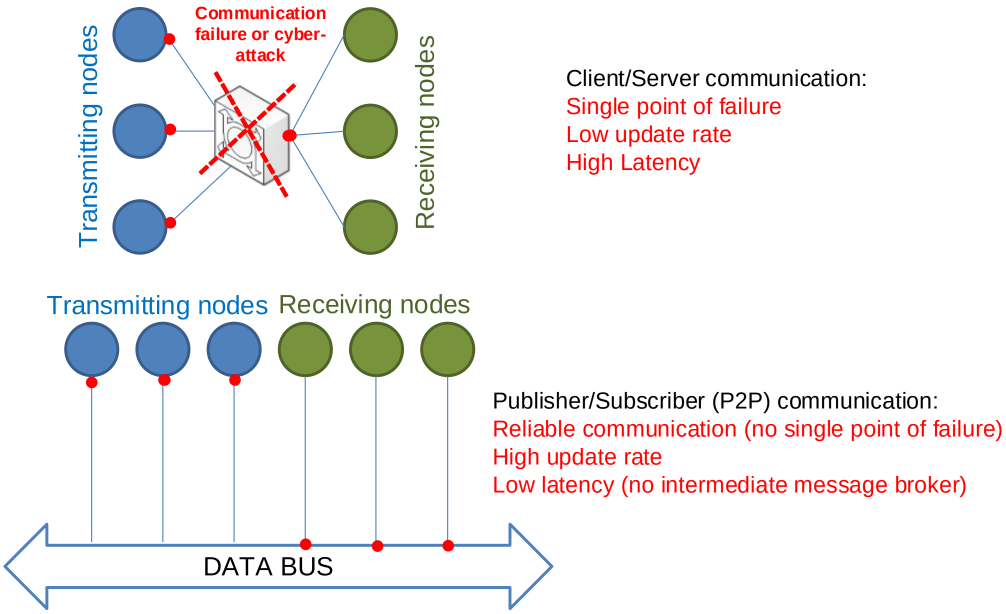

Transmitting nodes
Communication failure or cyber-attack
Receiving nodes
Client/Server communication:
Single point of failure
Low update rate
High Latency
Transmitting nodes
Receiving nodes
Publisher/Subscriber (P2P) communication:
Reliable communication (no single point of failure)
High update rate
Low latency (no intermediate message broker)
DATA BUS

Figure 1.2: Client/Server versus Publisher/Subscriber Communication Schemes.

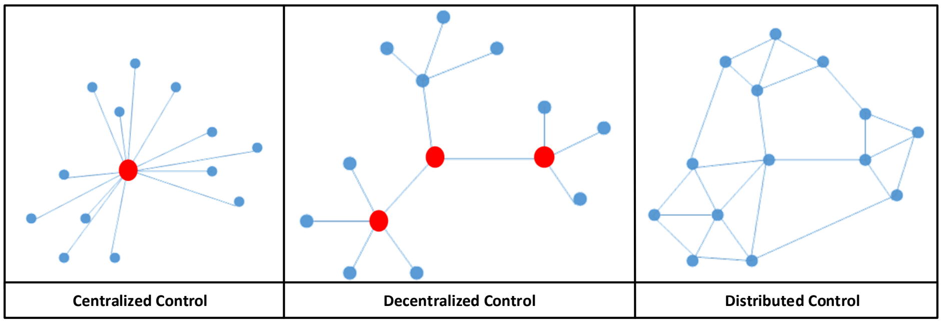


Figure 1.3: Smart Grid Control Schemes.

Although communication technologies brought many advantages to the control and operation of the smart grid, they introduced new sets of threats to the grid that are different in nature from the concerns related to the standard bulk power system operation. The largest threat to the smart grid is that the same technology that is advancing the power grid is being abused by adversaries to exploit vulnerabilities in the grid and maliciously tamper with its operation. Therefore, migrating to a reliable and secure smart grid requires a paradigm shift in the design and implementation of power system applications and control to account for cyber security early on, in the design stages.

In terms of cyber security, regulatory bodies, such as the International Electrotechnical Commission (IEC) and the Institute of Electrical and Electronics Engineers (IEEE), along with researchers, are concerned about changes in system features, attributes, and are worried about system inefficiency, resiliency, and even failure when under cyber-attacks. The ultimate goal is to harden the power grid against cyber-attacks by minimizing the damage and providing quick healing mechanisms. From a broad perspective, research work related to the cyber security of the power grid could be categorized as follows:

- Protection: to develop cyber security solutions that ensure resilient delivery of energy. Work in this area includes authentication, encryption, key management and storage, and others.
- Detection: to develop cyber security solutions that are capable of identifying the occurrence of cyber incidents. Work in this area includes anomaly detection, detection of false data injection, detection of password cracking attempts, and others.
- Response: to develop cyber security solutions that take appropriate measures in response to the detected cyber incidents.
- Recovery: to develop tools and techniques that allow the restoration of energy and other services that were lost due to cyber incidents.

Most of the current security solutions are based on cyber rules and mechanisms, such as digital signature verifications, message encryption, and others. However, different power system applications require different security levels, and at the same time, might impose strict requirements that will hinder the use of conventional security techniques [7]. For instance, based on the IEC 61850 standard, messages within a substation are categorized based on their sensitivity to communication delays. These range from 3 milliseconds up to several 100 milliseconds, depending on the message type and content [8]. For example, for messages with 3 milliseconds time delay, encryption is infeasible [9][10][11].

Recently, there has been a research thrust in complementing cyber detection techniques with physical detection techniques. That is, the security algorithms are now being trained to take into consideration the physical properties of the system that is being protected.

Table 1.1 summarizes the work in the recent literature that provided new technologies related to the cyber security of the power grid. Table 1.1 categorizes the work as being related to one or more of the previously mentioned thrusts: protection, detection, response, and recovery. Also, Table 1.1 identifies the work that uses physical information of the power system in their algorithm.

Table 1.1. Classification of Cyber Security-related Work for Smart Grids.

Ref.	**P**	**D**	**R**	**RY**	**UPP**	**Technique Used**
[12]	✗	✓	✗	✗	✗	Classifies synchrophasor data as anomalous or not using machine learning techniques.
[13]	✗	✓	✗	✗	✗	Surveys machine learning techniques for anomaly detection and analyzes their performance.

[14]	✗	✓	✗	✗	✗	Compares the performance of several data-stream-based intrusion detection techniques to detect anomalous behavior in smart meter data.
[15]	✗	✓	✗	✗	✗	Presents a multiattribute intrusion detection system specific for Supervisory Control and Data Acquisition (SCADA) systems.
[16]	✗	✓	✗	✗	✗	Presents a behavior-based Intrusion Detection System (IDS) for the IEC 61850 protocols using statistical analysis and traditional network features.
[17]	✗	✓	✗	✗	✗	Develops an intrusion detection system that identifies corrupt GOOSE and SMV messages based on predefined security rules using a specification-based algorithm.
[18]	✗	✓	✗	✗	✗	Provides an integrated anomaly detection system on the host and network levels to detect attacks on a single substation or on several substations based on a predefined set of violations indications.
[19]	✓	✗	✗	✗	✗	Proposes a layer 2 security algorithm in HSR Rings in substation networks.
[20]	✓	✗	✗	✗	✗	Proposes a lightweight security algorithm for SASs that provides authentication and authorizations at different levels utilizing public key certificates and zero-knowledge protocol-based server-aided verification and access control mechanisms.
[21]	✓	✗	✗	✗	✗	Proposes a hybrid method of securing IEC 61850 GOOSE messages based on DES and RSA.
[22]	✗	✓	✗	✗	✓	Uses register access indices and power consumption indices to detect attacked smart meter readings. Did not actually implement the physical indices.
[23]	✗	✓	✗	✗	✓	Detects network traffic that deviates from the expected communication pattern or physical limitations that the system must obey. Applied to overcurrent protection application of a transformer.
[24]	✗	✓	✗	✗	✓	Utilizes three types of agents with different physical rules, based on current and voltage deviations, to enhance the security of protection schemes against attacks that maliciously control circuit breaker.
[25]	✗	✓	✗	✗	✓	Uses common paths mining to detect intrusions on an electric transmission distance protection system

						using system data from synchrophasors and network logs.
[26]	✓	✓	✗	✗	✗	Monitors the traffic behavior of GOOSE messages to forecast or detect any deviation from the performance of GOOSE messages.
[27]	✓	✓	✓	✗	✗	Uses data mining and forensics tools to identify users' login behaviors, thus, preventing, detecting, and blocking insiders or attackers from performing malicious activities in closed environments.
[28]	✗	✓	✓	✓	✓	Compares physical system candidate invariants based on dynamic analysis of the microgrid with the actual invariants to detect false data injection attacks and take appropriate mitigation action to restore services.
[29]	✓	✓	✓	✗	✓	Proposes a multi-agent scheme for protection and emergency control when a power system is under attack.
[30]	✓	✓	✓	✗	✓	Utilizes a modified AC power flow algorithm to detect control-related attacks in power grids.
[31]	✓	✓	✓	✗	✓	Utilizes a model-checking approach for detection and mitigation of attacks on automatic generation control in power systems.
[32]	✗	✓	✓	✗	✗	Proposes a framework for detecting and responding to intrusions on the media access control level of wireless networks in smart grids based on the notion of defense-in-depth.
[33]	✗	✓	✓	✗	✗	Develops a GOOSE attack detector, which identifies Ethernet storms and other fraudulent GOOSE frames using standard stipulations.

*P: Protection; D: Detection; R: Response; RY: Recovery: UPP: Use Physical Properties.

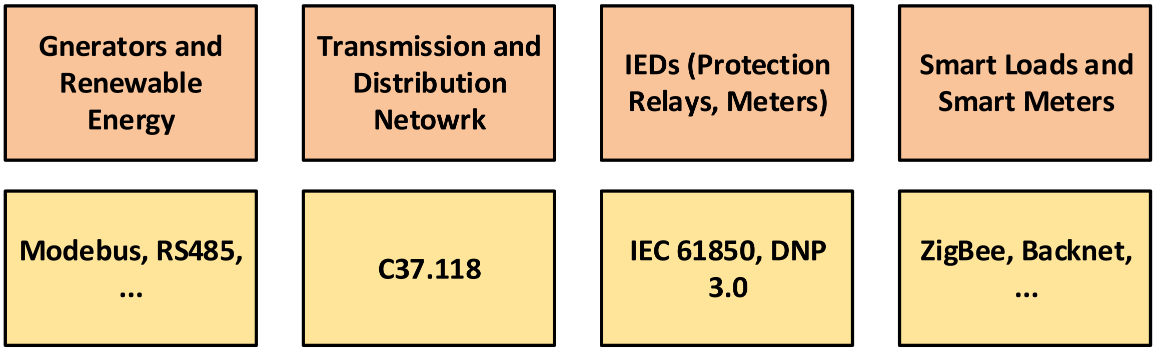
Following that trend, in this dissertation, we focused on developing cyber-physical techniques for securing smart energy cyber-physical systems. That is, in this dissertation, we focus on environment awareness by providing intelligent agents that are capable of learning the characteristics of the physical environment within which they operate. The agents, then, use this knowledge as an important second line of defense in detecting malicious activity, thus complementing the cyber security methods, such as encryption and authentication. Therefore, the decisions made by these agents are based on properly correlating the requirements of the cyber and the physical domains to detect and mitigate malicious activity that is trying to tamper with the grid operation. Unlike the work presented in the literature, the agents developed in this dissertation utilize artificial intelligence to characterize the power system. As will be shown later, artificial intelligence techniques can learn the characteristics of the power system with high accuracy and have fast response time, which will not add significant latency to the operation and control of the smart grid.

1.1 Interoperability and Security Challenges of the Smart Grid as a Cyber-Physical System

1.1.1 Interoperability Challenges

Communication and data exchange between various system components in the smart grid are a prerequisite for the realization of modern monitoring and control services in the future smart grid. However, the different systems of the smart grid are usually owned by different entities and are governed by different rules. This poses an interoperability problem. Also, as seen in Figure 1.4, there are a variety of standards and protocols that have been developed for automating the operation of the smart grid. In order to establish a link between multi-protocol and multi-vendor devices, mapping the data protocols from one to another is required.

As such, there is a strong need for a common data bus that allows application developers to easily develop smart grid applications that link different sectors of the grid, such as energy management applications and demand side management applications [34].



Gnerators and Renewable Energy	Transmission and Distribution Netowrk	IEDs (Protection Relays, Meters)	Smart Loads and Smart Meters
Modebus, RS485, …	C37.118	IEC 61850, DNP 3.0	ZigBee, Backnet, …

Figure 1.4: Protocols associated with different physical layers of the grid.

1.1.2 Security Challenges

Establishing a bullet-proof digital communication infrastructure is not an easy task to perform and spans a wide area of research topics, such as threat identification and detection, protection against known threats, and responding to/recovering from unknown malicious cyber incidents.

With the advent of the electric grid, security was taken for granted by the utility industry. Cyber security was not a design requirement since utility networks were private and not connected to the public internet. However, recent threats to the grid and other critical industrial systems raised the level of awareness across people in the industry and in research and development. In fact, in 2010, Stuxnet was the first malware to be discovered, targeting SCADA along with Programmable Logic Controllers (PLCs) control networks [35]. In fact, it has been said that the “power grid infrastructure is teetering on the brink of a hacker-induced meltdown.” As seen in Figure 1.5, recent statistics show that “more than 85% of all ICS vulnerabilities have been disclosed since 2011 - the year following the discovery of Stuxnet. The Open-Source Vulnerability Database tracked more

ICS vulnerabilities than other sites, including the National Vulnerability Database ...” [36].

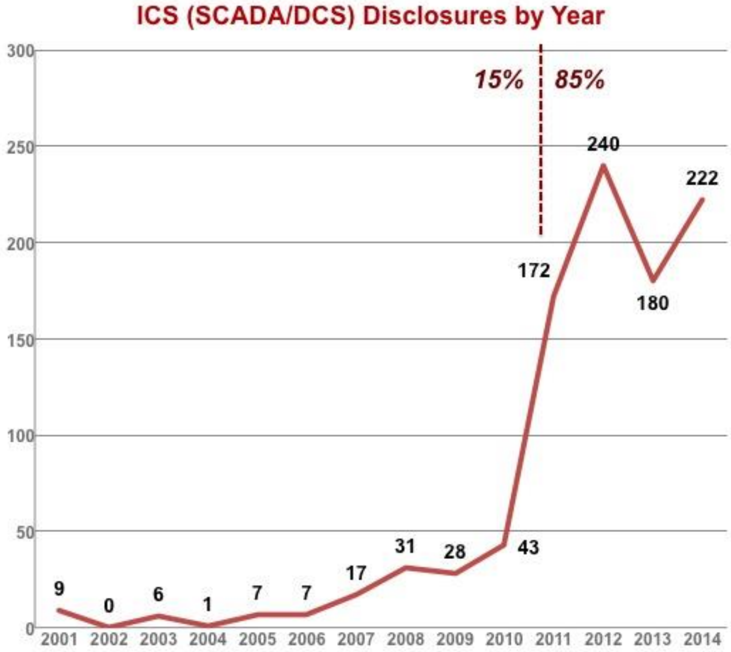


Figure 1.5: Data obtained from the former Open-Source Vulnerability Database [36].

Several researchers were able to prove the ability of a determined adversary to tamper with the grid’s operation. In [37], a stealth error vector was devised that was unrecognizable by state estimators. This stealth vector was also able to fool estimators into using erroneous data. The importance of state estimation led researchers to assess state estimation uncertainties and impacts of the communication system delays [38]. Global Positioning System (GPS) spoofing techniques to disrupt the operation of Phasor Measurement Units (PMUs), which directly affect the stability of the power system, were discussed in [39] and [40]. The work in [41] and [42] discussed an important security shortcoming in IEC 61850 GOOSE messaging. By breaching a substation’s network, the authors were able to successfully launch a data manipulation attack to maliciously open and close circuit breakers. As it is well known, a chain is as strong as its weakest link.

Therefore, with a widespread and complex system, such as the smart grid, there is always a possibility of a zero day attack. That is, the grid is always under the risk of exploiting previously unknown vulnerabilities.

Today, cyber security is at the core of the smart grid. It is critical to everything we do to try to modernize the grid, such as integrating renewables, introducing energy independence, and reducing the reliance on foreign oil. It is true that the smart grid itself is a complex interconnected system; however, the operation of commercial, industrial, medical, military, and many critical infrastructures is critically dependent on the cyber-physical smart grid. Looking at the critical industries today, the smart grid is said to be the first among equals [43].

1.2 Problem Statement

Utilizing microgrids as building blocks of the smart grid with increased penetration of distributed energy resources - whether renewable or not - controllable smart loads, and energy storage, will provide a lot of new services that will radically enhance the operation and control of the current power system. However, to manage the distributed components of the microgrids and integrate the microgrids together require a shift from centralized control to decentralized and distributed control schemes. This, in its turn, will necessitate reshaping the current communication infrastructure of the grid to interface with different types of distributed software/hardware controllers and sensing devices.

However, the enhanced operation of the grid came at the cost of new interoperability challenges and security risks. To address the interoperability challenges, there is a need for a platform that seamlessly integrates the components of the smart grid together, especially since they are governed by different regulations and owned by different entities. To achieve that, one first needs to develop a strong understanding between the cyber information flow and the physical information flow in the smart grid, and the interrelationship between them. On the other hand, to address the security risks, there is a strong need to come up with solutions that are capable of defending the grid against cyber-attacks, minimizing the damage in case a cyber-incident occurs, and restoring services within minimal time.

1.3 Research Objective

In this dissertation, we present the concept of enhancing the robustness of cyber security solutions against cyber-attacks by complementing their decision with information about the physical characteristics of the power system through the use of intelligent agents. The idea behind this concept stems from the fact that the cyber information flow and the physical information flow in the smart grid are correlated. That is, physical processes are influenced by the decisions of the control logic, and control algorithms are influenced by the states of the physical system.

Therefore, information pertinent to the physical behavior of the system should be treated as an indispensable element, alongside a detailed analysis of communication protocols used in power system automation, when designing algorithms that defend the power grid against malicious attacks.

In order to reach that objective, we analyzed the cyber and physical information flow in a safe environment and in a high-fidelity communication network-in-the-loop co-simulation setup. We also used the setup to emulate industrial communication protocols, such as the IEC 61850 GOOSE messaging and the IEC 61850 SMV protocols. Based on the analysis of these protocols, we applied the concept of physical-model-checking to detect switching-related attacks on power systems. Also, we investigated the use of neural network forecasters to detect spoofed sensor measurements.

1.4 Original Contribution of this Dissertation

The main goal of the research work in this dissertation is to provide security solutions that harden the smart grid against cyber-attacks, taking into consideration not only the cyber domain, but also the physical characteristics of the grid. To achieve that, we developed a communication network-in-the-loop co-simulation platform, and performed an analysis of the effects of the cyber information flow on the physical dynamics of the power system, and vice-versa, to develop an understanding for the strong correlation between the cyber and physical domains of the smart grid. In this co-simulation framework, the dynamics of the power system were simulated on software packages, the control logic was programmed on hardware embedded microcontrollers, and the data exchange was performed over a real communication network. We utilized the Data Distribution Service (DDS) middleware as a common data bus that linked the components of the system together. Also, we performed protocol emulation to link commercial devices into the co-simulation setup. We performed three studies, with three different objectives, to verify the effectiveness of the developed co-simulation framework. : 1) Hierarchical control of Electric Vehicles (EV) charging in microgrids; 2) IEC 61850 protocol emulation for protection of active distribution networks; and 3) Online lost packet forecasting for IEC 61850 Sampled Measured Values. The results showed that the co-simulation platform provides a high fidelity design, analysis, and testing environment for cyber information flow and their effect on the physical operation of the smart grid, as they are experimentally verified, down to the packet, over a real communication network.

Since it is the most industry-accepted power automation standard, we conducted a detailed analysis of the IEC 61850 protocol mapping. The GOOSE messaging protocol, which is the protocol used by modern digital relays for event-driven communication, such as the opening and closing of circuit breakers, was analyzed. Based on this vulnerability analysis, we developed a GOOSE poisoning malware. The malware is able to sniff GOOSE messages, filter them based on their unique identifiers, change any field of the message, and then, re-transmit it in a legitimate GOOSE packet format. We utilized this malware to analyze the different implementations of the GOOSE messaging protocol on commercial devices and to recommend best practices.

To defend against GOOSE poisoning, especially manipulating the data fields and spoofing the source MAC address to maliciously open and close circuit breakers, we developed an artificially intelligent physical-model-checking security algorithm. The algorithm utilizes artificially intelligent neural networks to learn the load flow characteristics of the power system and benefits from the fast responses of the AI to decode and understand contents of network packets. Then, the output of the AI is processed through an expert system to verify that incoming control commands do not violate the physical operational constraints of the power system and do not put the power system in an insecure state.

Similarly, we analyzed in detail the IEC 61850 SMV protocol, which is used to digitally transmit current and voltage measurements over the process bus. We found that the structure of the SMV protocol is robust against several types of known attacks, but it is still vulnerable to spoofing attacks. In order to mitigate that, we investigated the feasibility of using neural network forecasters to detect spoofed SMV packets. Although forecasters showed a high detection accuracy, they were still vulnerable to the accumulation of forecasting errors. Accordingly, we developed another malware, which is targeting SMV packets. This malware allows the user to eavesdrop on SMV packets and inject malicious measurements at his/her discretion. We utilized the malware as a vaccine to enhance the resiliency of neural network forecasters against the accumulation of forecasting errors. To that end, we added a second layer to the developed neural network forecasters to detect the accumulation of the forecasting errors based on lightweight statistical indicators.

On the application layer, we developed an Energy Management System (EMS) not only to help customers reduce their energy bills, but also to hide their load pattern profiles through proper utilization of energy storage devices. This is extremely important for military bases that are operating on foreign lands and connected to foreign utility grids.

The work in this dissertation sheds the light on the importance of communication technologies and the criticality of hardening the smart grid against cyber-attacks. Therefore, it is important to educate the current and future users of the cyber infrastructure with the tools and techniques that are necessary to keep advancing and protecting the smart grid. Accordingly, a demonstration of utilizing some of the tools required to develop the algorithms in this dissertation is presented.

Finally, the developed security algorithms in this dissertation were verified in hardware experimentation on the Smart Grid Testbed at Florida International University.

1.5 Dissertation Organization

Chapter 2 emphasizes the importance of understanding the cyber information flow and the physical information flow in the smart grid, and the interdependency between them. This chapter also explains the concept of co-simulation as it relates to energy cyber-physical systems, and presents a literature review on the available co-simulation setups. Finally, a network-in-the-loop framework to analyze cyber and physical information flow in the smart grid utilizing the DDS as the system's orchestrator is proposed.

Chapter 2 also presents three case studies for the purpose of verifying the effectiveness of the network-in-the-loop framework. The first case study is about hierarchical charging control of electric vehicles and is focused on showing how the DDS is used as a common data bus to seamlessly interface the components of the system together. The second case study is about protocol emulation. It gives an example about emulating the IEC 61850 GOOSE and SMV protocols for use in protection of active distribution networks. Finally, the third case study focuses on the ability of the proposed framework to provide a credible platform for testing cyber algorithms, such as dropped packet forecasting, study their effects on the physical system, and verify them, down to the packet level, on a real network.

Chapter 3 presents an analysis of the IEC 61850 GOOSE messaging protocol and its recommended implementation procedures by the IEC 62351 standard. Then, a study is conducted on two different IEC 61850 GOOSE compliant devices. Miss-implementation issues are found and the chapter is concluded by a set of standard practices for proper implementation of the GOOSE protocol.

Chapter 4 analyzes the standards that are currently used for switching-related activities in power systems (i.e. GOOSE, R-GOOSE, and DNP 3.0) and the cyber threats associated with those protocols. Next, a physical-model-checking technique is proposed to detect switching related attacks. The proposed technique is experimentally verified on a laboratory-scale power system.

Chapter 5 analyzes in detail the IEC 61850 SMV protocol, which is used to transmit current and voltage sensor measurements over Ethernet networks. The benefits of the SMV process bus are identified. The current security countermeasures is also outlined. Chapter 5 also investigated the use of neural network forecasters to detect spoofed sampled measured values.

Chapter 6 presents the steps of developing a malware script that performs SMV spoofing attacks. The results show that this malware is capable of reducing the accuracy of neural network forecasters. It is also proposed in this chapter that this malware is to be used as a vaccine to harden the robustness of the neural network forecasters against sophisticated attacks.

Chapter 7 discusses the importance of complementing cyber intrusion detection techniques with physical rules. In this chapter, a bi-layer algorithm to detect spoofed SMV messages is proposed. The proposed algorithm is verified in hardware on a laboratory-scale hybrid AC/DC microgrid.

Chapter 8 addresses issues related to the privacy of the customers’ data in the artificial metering infrastructure. An energy management system, which utilizes energy storage, to camouflage the customers’ load demand profiles and reduce their electricity bills is presented in this chapter. The energy management system is verified on the network of the smart grid testbed at FIU.

Chapter 9 discusses security issues related to the charging of electric vehicles. Comments on the importance of including security in the design stage of smart infrastructure, such as electric vehicle charging, were made.

Chapter 10 educates the reader on necessary tools to develop and prototype applications for energy cyber-physical systems.

Chapter 11 concludes this dissertation and gives insight on future work.

Chapter 2 Design, Analysis, and Testing Framework for Cyber Physical Interactions in the Smart Grid

Being a complex and highly interdependent cyber-physical system makes capturing the relation between the cyber and physical domains of the smart gird a necessity to understand the effect that each one has on the other. This chapter presents a network-in-the-loop co-simulation platform that formalizes the understanding of cyber information flow, the dynamic behavior of physical systems, and captures the interactions between them in applications related to the smart grid. Here, power system simulation software packages, embedded microcontrollers, monitoring and logging services, and a real communication infrastructure are combined to provide a cohesive smart grid cyber-physical platform. A data-centric communication scheme, with automatic network discovery features, is selected to provide an interoperability layer between multi-vendor devices, software packages, and to bridge different protocols. This chapter also provides a review and an analysis about the available co-simulation frameworks in the literature. The effectiveness of the framework proposed in Chapter 2 is verified in three case studies: 1) Hierarchical control of electric vehicles charging in microgrids; 2) IEC 61850 protocol emulation for protection of active distribution networks; and 3) Online lost packet forecasting for IEC 61850 Sampled Measured Values. The results showed that the co-simulation platform provides a high fidelity design, analysis, and testing environment for cyber information flow and their effect of the physical operation of the smart grid, as they are experimentally verified, down to the packet, over a real communication network.

2.1 Introduction

Bulk centralized generation has been the foundation of the power system for decades. In fact, in the early stages of the grid, power system was divided into two parts: bulk generation and transmission on one hand, and the distribution system on the other hand. In this model, the main concern was the balance of supply and demand. The utilities have been studying their customers' demand for power and were managing and allocating their resources to meet that demand. A typical daily load demand profile shows low demand for power in the early morning when most of the people are sleeping, a peak in demand during midday when industrial activities are ongoing, and the demand remains high when people leave work and come back home to cook dinner. The demand for power then decreases at night when people go to sleep. Given that, utilities have divided their customers' demand for power as base load, which is served by nuclear power plants or large coal units, intermediate load, which is served by medium size gas and coal units, and peak load, which is served by gas units. As a precaution measure, utilities also have an operating reserve for reliability purposes [43].

Therefore, the distribution grid was regarded as a passive power consumer. That is, power was flowing in a unidirectional manner from generation to transmission to distribution. However, with the advancement of technology and changes in lifestyles, the demand for power increased drastically. A comparative study on the demand for power between the industrial, commercial, and residential sectors showed that from 1956 to 2007, the residential and commercial demand for power increased from 48% to 73%, whereas the demand for power in the industrial section dropped from 52% to 27% [43].

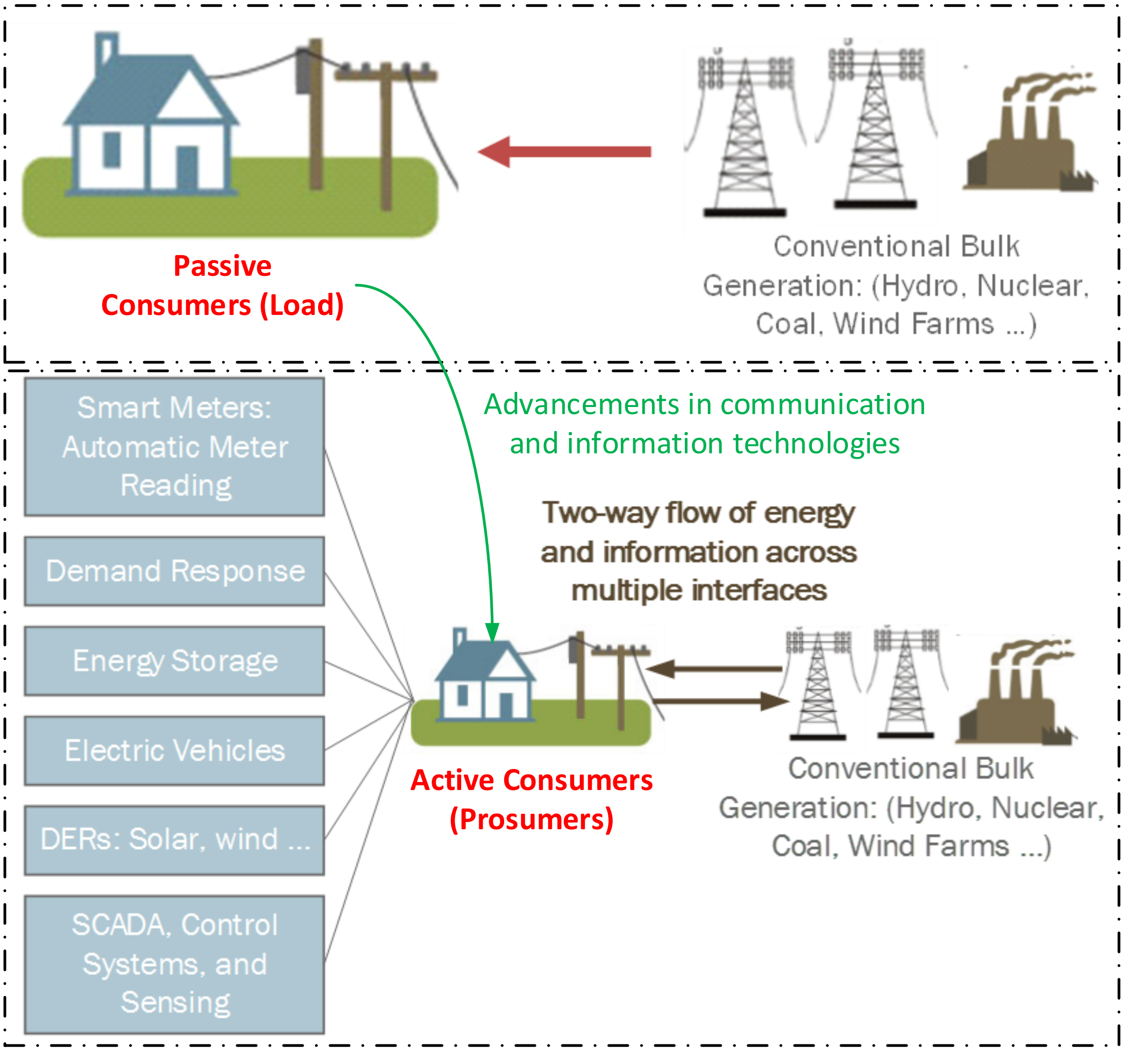


Figure 2.1: The transformation of the power industry.

It would be very expensive and time consuming if utilities were to build new bulk centralized plants to meet the new increasing demand. However, coming up with novel technologies to manage/control the demand in the distribution side to reshape and peak demands would be faster and more economical. Advancements in communication and information technologies where the main reason behind the transformation of the power industry to the recently known Smart Grid. In the new power grid, the distribution system transformed from passive consumers, or load, to active consumers, or prosumers. As can be seen in Figure 2.1, a lot of new technologies have been integrated into the distribution side, such as smart meters, distributed and renewable energy resources, energy storage, SCADA systems, etc …

In order to manage the two-way flow of power in the new smart grid, the concept of microgrids emerged. A microgrid is a small geographical area with a group of distributed energy resources, mainly composed of renewable resources, that provide power to local customers and has a connection point to the main utility grid. The smart grid is, thus, composed of a group of interconnected smaller microgrids. Each microgrid is controlled locally, and is coordinating with neighboring microgrids and the utility grid to maintain reliable power delivery.

Therefore, the reliable operation of the smart grid is a function of the configuration and cyber-physical nature of its constituting subcomponents. However, the power system and the communication network differ in terms of their dynamic behavior. Therefore, understanding the dynamics of the smart grid’s cyber and physical domains, and the ability to model, analyze, and test the interactions between them, are important in order to ensure reliable power delivery to the critical infrastructures of current and future cities [44]. In pursuit of this, co-simulation has emerged as an effective method for the assessment and validation of cyber-physical energy systems [45]. Several works in the literature have introduced the concept of co-simulation and, as seen in Figure 2.2, they can be categorized into two clusters:

1. Integrating two software packages and providing synchronization among them.
2. Hardware-in-the-Loop testbeds.

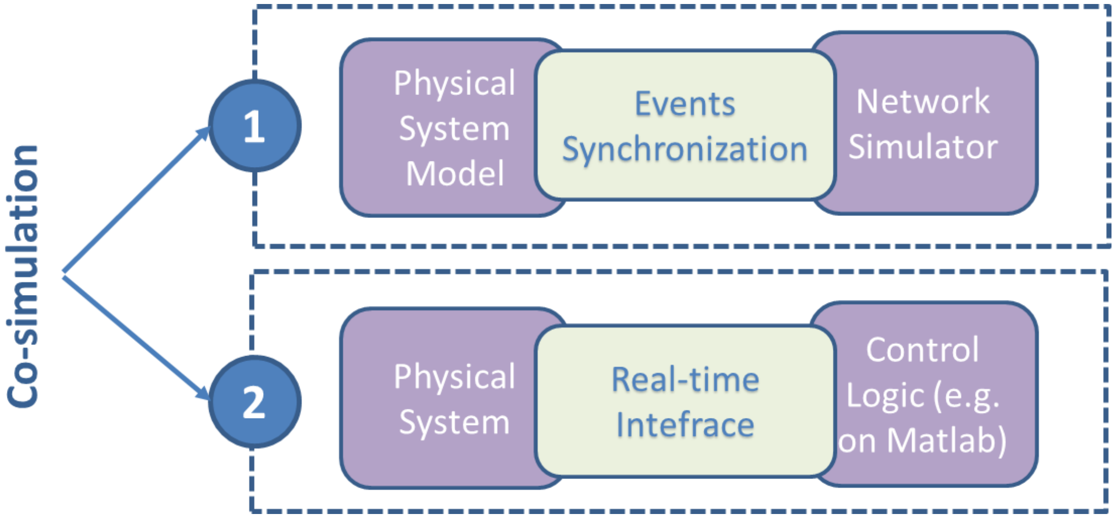


Figure 2.2: Categories of co-simulation platforms present in the literature.

2.2 Integrating software packages and providing synchronization among them.

As the section title indicates, in this category of co-simulation, the efforts to provide a cohesive cyber-physical energy platform comprises the use of a power system simulation tool, a network simulation tool, and an event synchronization tool to harmonize the flow of events between the former and the latter.

In [46], co-simulation is described as the process of integrating two software packages together and providing synchronization among them. In [47] a co-simulation framework was developed by combining OpenDSS for simulation of the power system and OMNeT++ for simulation of the communication network to investigate wide area monitoring systems. Similarly, [48] presented a framework for simulating a power routing algorithm for clusters of microgrids by integrating a real-time digital simulator and OMNeT++. Authors in [49] argued that there is a large gap in the area of simulating cyber-physical systems and more efforts are needed to be put in the co-simulation area. For that, they introduced an event driven co-simulation module based on OpenDSS and Network Simulator NS2. A co-simulation platform of a low voltage grid based on IEC 61850 was presented in [50], where MATLAB's SimPowerSystems and SimEvents toolboxes were used to model the system's physical and cyber information flow, respectively. The following table, which is adopted from [45], provides more examples about co-simulation frameworks and is presented here for the convenience of the reader.

Table 2.1. Application-Specific Simulation-Based Co-simulation Frameworks.

Co-Simulation Framework	Power System Simulator	Network Simulator	Synchronization Strategy	Application
[51][52]	PSLF	NS-2	Global Event-Driven	PMU-based WAMPAC
[53]	ADVES	NS-2	DEVS	WAMPAC
[54]	VTB	OPNET	Master-slave	WAMPAC
[55]	PowerTech TSAT	GridStat	Time stepped	WAMPAC
[56]	Modelica	NS-2	Master-slave	Control
[57]	NETOMAC	NS-2	Time stepped	Evaluation of DERs
[58][59]	RTlab	OPNET SITL	Asynchronous	WAMC, HVDC
[60]	OpenDSS	NS-2	N/A	DER Integration
[61]	PowerWorld	OPNET	N/A	Cyber-security of SCADA
[62]	PSCAD	OMNet++	Global Event-Driven	Cyber-security of LV Grid

* PMU: Phasor Measurement Unit; WAMPAC: Wide Area Monitoring, Protection, and Control; DER: Distributed Energy Resource; WAMC: Wide Area Monitoring and Control; HVDC: High Voltage DC; SCADA: Supervisory Control and Data Acquisition; LV: Low Voltage.

The mentioned works represent an important step towards properly modeling the cyber and physical domains of a cyber-physical system. Nonetheless, these simulators are not implemented over a real communication network. Therefore, they will not be able to account for practical issues with high fidelity, as they are limited with the functionalities provided in the network simulation software. For instance, network simulators do not work on the packet level. They usually model networks on the large scale and use statistical and/or probabilistic models to predict delays. Also, practical issues due to different firmware implementations cannot be realized in network simulators. Another drawback of the previously discussed frameworks is that they are application specific and hard to expand. As shown in Table 2.1, each framework was tailored to a specific application in the smart grid.

2.3 Hardware-in-the-Loop Testbeds

From another perspective, several works included hardware in the loop with simulation platforms. There are two methodologies for this approach:

1. Integrating power equipment such as generators, actuators, and converters into the simulation environment to test control algorithms on real hardware.
2. Integrating IEDs and other embedded devices with traffic generation software packages for designing and testing communication networks.

A hardware in the loop simulation test-bed for distributed microgrid management was presented in [63]. The presented system is based on the Zigbee protocol and the simulation and hardware were integrated through an I/O conditioning board. Also, in [64] actual PMUs were integrated with a real-time digital simulator via an IEC 61850 bus to model passive islanding schemes. These two implementations are application specific and are hard to expand and manage the complex communication requirements for other smart grid applications. From a pure networking perspective, [65] proposed a method for testing IEDs in a platform that incorporates several protocols. However, the integration of these protocols is based on a proprietary Distributed Test Manager, and therefore, is not easily expandable and requires special libraries to interface with.

Both modeling approaches presented are single sided and do not provide a comprehensive framework to properly model cyber-physical systems, and most importantly the interactions between their cyber and physical domains. Also, the second approach (2) is usually concerned with a single or a few protocol combinations and will require a lot of engineering and programming efforts to integrate different applications and devices together.

Additionally, and as explained in [66], comprehensive security mechanisms are required to deal with cyber security and vulnerabilities brought about by communication-assisted control and operation of the smart grid. Therefore, co-simulation, which provides a cohesive platform for the physical and communication infrastructures of the smart grid are important in the design and testing stages of cyber security algorithms.

Accordingly, the work in this chapter presents a communication network-in-the-loop framework to analyze the cyber and physical information flow in the smart grid. Further to what is presented in the literature, which are well-detailed but application oriented co-simulation platforms, the contributions of this framework are as follows:

1. Providing a holistic cyber-physical energy systems framework, with ease of integration of simulation packages, software, and hardware by utilizing a data-centric communication backbone to manage information exchange and seamlessly orchestrate the components of the system together.

2. Enabling protocol emulation and translation that allows the integration of a wide range of multi-protocol/multi-vendor devices into the developed framework.

3. Supporting scalability to a wide range of smart grid applications, as demonstrated in the case studies of Chapter 3.

4. Providing remote connectivity, data monitoring, and logging services.

2.4 Interoperability and Capturing the Relations between the Cyber and Physical Components of Smart Grid Applications

Before moving forward in discussing the details of the developed co-simulation framework, an overview about the importance of interoperability in the envisioned smart grid will be provided. This is important because the future grid is seen as a complex system with different components communicating together and lots of efforts are being placed to find a common ground to achieve harmony among all the devices and applications.

2.4.1 The Human Interactions Analogy

The IEEE Standard Computer Dictionary defined the term Interoperability as the ability of two or more systems or components to exchange information and to properly use the information that has been exchanged [67]. The Utility Standards Board (USB) explained this definition by introducing the human interoperability paradigm as an analogy to cyber interoperability [68]. USB argues that people speak their native language within their own local environment, similar to applications within their own computing platform. However, if geographically dispersed groups need to communicate with each other, then English would be used as a common language. This is because the English language has been accepted and is known by most societies worldwide. The concept of English as a common language is illustrated in Figure 2.3. Similarly, there is a strong need for the adoption of a common cyber language, or backbone, especially for large and complex systems, such as the smart grid.

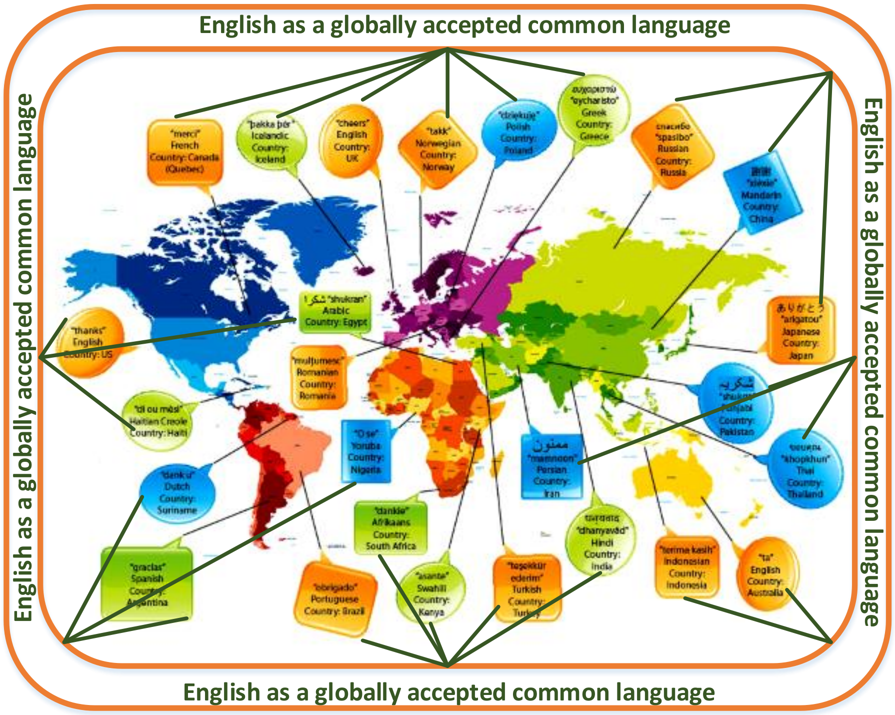

English as a globally accepted common language
English as a globally accepted common language
English as a globally accepted common language
English as a globally accepted common language
"merci" French Country: Canada (Quebec)
"þakka þér" Icelandic Country: Iceland
"cheers" English Country: UK
"takk" Norwegian Country: Norway
"dziękuję" Polish Country: Poland
ευχαριστώ "eycharisto" Greek Country: Greece
спасибо "spasibo" Russian Country: Russia
谢谢 "xiexie" Mandarin Country: China
ありがとう "arigatou" Japanese Country: Japan
شكرا "shukran" Arabic Country: Egypt
"thanks" English Country: US
"mulţumesc" Romanian Country: Romania
"di ou mèsi" Haitian Creole Country: Haiti
"O ṣe" Yoruba Country: Nigeria
ممنون "mamnoon" Persian Country: Iran
شکریہ "shukriya" Punjabi Country: Pakistan
ขอบคุณ "khopkhun" Thai Country: Thailand
"dank u" Dutch Country: Suriname
"dankie" Afrikaans Country: South Africa
धन्यवाद "dhanyavād" Hindi Country: India
"gracias" Spanish Country: Argentina
"obrigado" Portuguese Country: Brazil
"asante" Swahili Country: Kenya
"teşekkür ederim" Turkish Country: Turkey
"terima kasih" Indonesian Country: Indonesia
"ta" English Country: Australia

Figure 2.3: Interoperability in Human Interactions. People from difference countries use English language as a common ground to establish communications as indicated by the green connection lines (map adopted from [69]).

2.4.2 Language, Platform, and Vendor Independence

The envisioned smart grid is a complex system, which is in continuous evolving and expanding. Communication and information technologies are the main propellers of the evolution and expansion of the smart grid. These technologies leveraged the intelligence level of the smart grid. In fact, the term smart was not present before the integration of the communication technologies into the power grid. With the realization of the importance of communication technologies in making the former possible, more effort is going towards this need, with particular interest in the concept of interoperability. Several national and international entities are working with electricity utilities, vendors, regulatory bodies, and in some cases end users on developing standards that will make interoperability in the smart grid a reality. These include IEEE, IEC, National Institute of Standards and Technology (NIST), American National Standards Institute (ANSI), North American Electric Reliability Corporation (NERC), and many others.

Like the English language, a common cyber language should have a profound word diction and a well-defined set of communication rules. Therefore, standards development bodies have a lot of functionalities that need to be well orchestrated together to achieve sound interoperability. Middleware technology is a practical solution which ensures language, platform, and vendor independence in smart grid applications. Recently, authors in [70] proposed the use of a data-centric middleware to manage the communication in a laboratory scale smart grid system. As shown in Figure 2.4, using this network architecture, i.e. the common middleware data bus, applications exchange messages via the middleware without worrying about each other's computing devices and configurations. The middleware, thus, introduces an abstraction level to the smart grid from the heterogeneous nature of the communication networks, operating systems, and programming languages [71].

In this work, the DDS machine-to-machine middleware from the Object Management Group (OMG), [72], was utilized to enable real-time integration of various applications within the developed co-simulation framework. DDS is a data-centric middleware that follows a publisher-subscriber model for distributed control applications. It possesses a relational data modelling method in a decentralized data space, which decouples applications in time and space. DDS supports automatic network discovery and has a rich set of quality-of-service (QoS) profiles, which are configured depending on the applications' needs [72][73]. More details about the DDS will be explained later in this chapter.

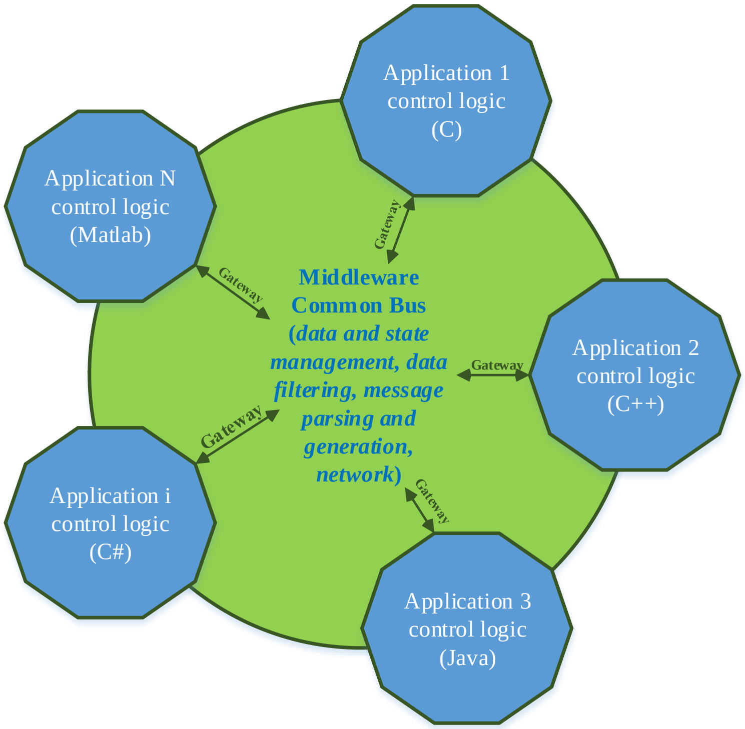


Figure 2.4: Utilizing a common data bus (middleware service) as an interoperability layer for different smart grid applications.

2.4.3 Cyber Security, Vulnerability, and Impact Analyses

As explained earlier, standardization in smart grid communications is necessary to facilitate complex operations of modern power system functions. However, the strong coupling between the cyber and physical domains of the contemporary grid exposes the system to more vulnerabilities. As such, standards need to be continuously assessed for reliability and are expected to be implemented properly on field devices [41][42]. The developed platform becomes more useful when we add cyber security, vulnerability analysis, and impact analysis into the equation of the smart grid. For instance, cyber security for industrial control systems in general, and power systems in particular, is different from enterprise cyber security because the former usually operates in real-time and has stringent time delay requirements. As an example, the IEC 62351 part 6, which covers cyber security for Layer 2 messages in substation automation systems, recommends not to use encryption on these messages because their required transfer time is limited to 4 ms end-to-end [42]. Therefore, developing and testing new security and authentication mechanisms is a necessity. However, it is infeasible, dangerous, and highly expensive to test these algorithms directly on physical systems. The developed framework provides a safe and high-fidelity platform to test cyber security algorithms over a real network with physical embedded computing platforms (e.g. could be commercial IEDs) integrated with a scalable set of simulation software packages, operating in real-time, to capture the physical information flow as well.

Also, vulnerability analysis on industrial communication systems and standards could be performed on the developed platform. The impact of these vulnerabilities could be investigated safely on a simulated system. Maximum reverse current fed back into a distributed generator exciter due to malicious data sent to the synchronization controller could be investigated. Questions about how a certain device would operate under an attack or a network outage could be answered. The ability of the system to reconfigure and self-heal could also be investigated as well as many other scenarios involving cyber-physical interactions.

2.5 Framework Development and Description

Figure 2.5 shows a conceptual representation of the proposed framework. The developed framework consists of 4 main components:

1. Power system simulation packages.
2. Embedded microcontrollers.
3. Information exchange interface and protocol translation.
4. Monitoring, logging, and remote connectivity.

2.5.1 Power System Simulation Packages

In the smart grid, cyber information flow affects physical processes and the dynamics of physical systems influence the operation and decision of cyber processes. However, in the analysis, design, and testing phases of smart grid applications, which are the scope of this chapter, it is impractical and highly expensive to test newly developed algorithms directly on field devices. Also, in the design stage, physical system dynamics and their tolerance to software and communication constraints cannot be ignored. Given the fact that the laws of physics that govern the operation of the physical processes are well formulated in mathematical models with high accuracy, the interrelation between the power systems on one hand, and the software and communication realms on the other, could be adequately and safely captured through the incorporation of power system simulation software. In this work, SimPowerSystems from Matlab/Simulink was utilized as the power system simulation package due its popularity and extensive libraries.

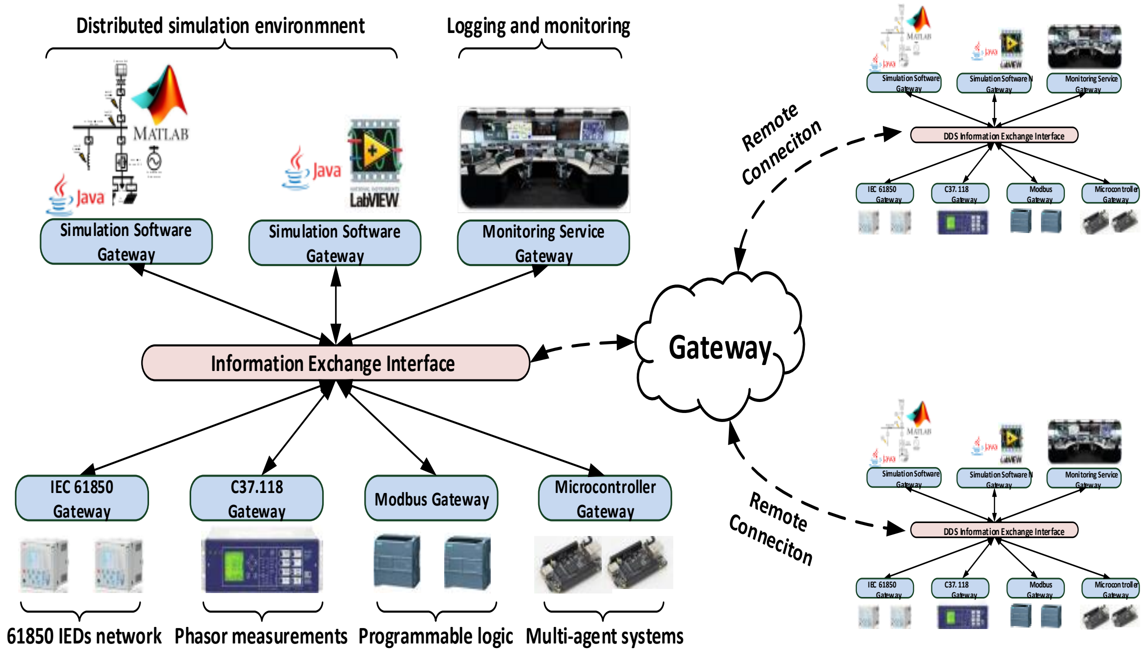

Distributed simulation environmnent
Logging and monitoring
MATLAB
Java
LabVIEW
Simulation Software Gateway
Simulation Software Gateway
Monitoring Service Gateway
Information Exchange Interface
Gateway
Remote Conneciton
Remote Conneciton
IEC 61850 Gateway
C37.118 Gateway
Modbus Gateway
Microcontroller Gateway
61850 IEDs network
Phasor measurements
Programmable logic
Multi-agent systems
Simulation Software Gateway
Simulation Software Gateway
Monitoring Service Gateway
DDS Information Exchange Interface
IEC 61850 Gateway
C37.118 Gateway
Modbus Gateway
Microcontroller Gateway
Simulation Software Gateway
Simulation Software Gateway
Monitoring Service Gateway
DDS Information Exchange Interface
IEC 61850 Gateway
C37.118 Gateway
Modbus Gateway
Microcontroller Gateway

Figure 2.5: Conceptual representation of the overall framework.

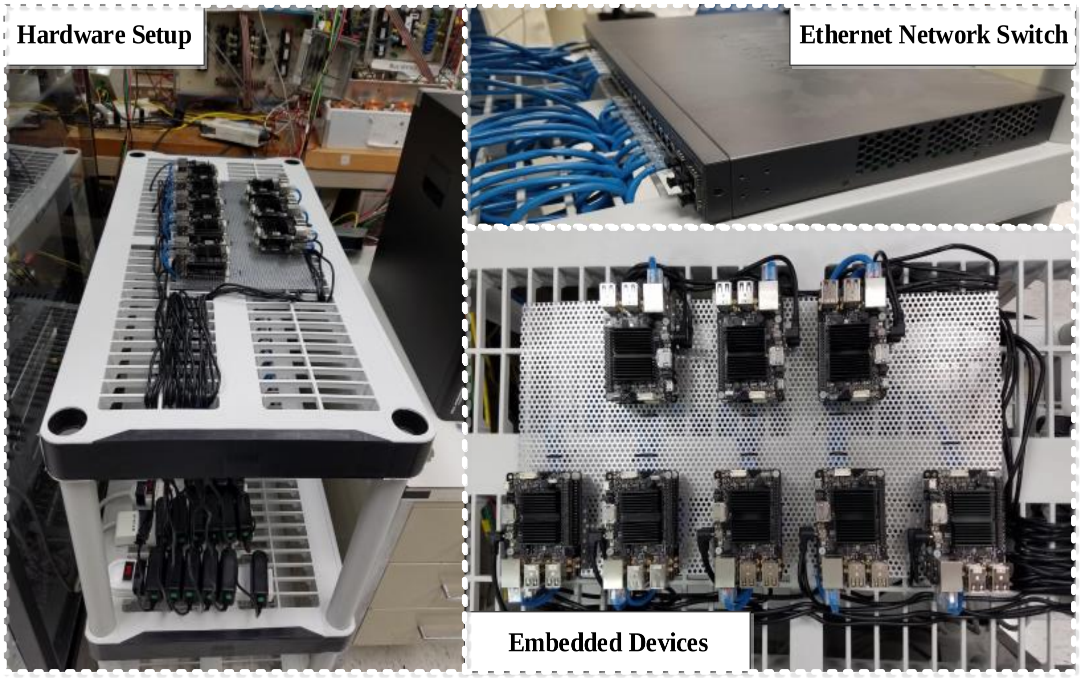


Figure 2.6: Embedded microcontrollers utilized in the co-simulation framework.

2.5.2 Embedded Microcontrollers

Software developers face many challenges when implementing their algorithms on microcontrollers. These challenges include limitations in speed of the processors, memory allocation, interfacing with the network, and other practical issues. Since software algorithms will be controlling critical processes, it is important that the mentioned challenges are exhausted in a safe and high fidelity environment. Therefore, in this work, all the software and control logic will be implemented on hardware-embedded devices.

2.5.3 Information Exchange Interface and Protocol Translation

Middleware technology is a practical solution, which ensures language, platform, and vendor independence in smart grid applications. Using the common middleware data bus, applications exchange messages via the middleware without worrying about each other's computing devices and configurations. The middleware, thus, introduces an abstraction level to the smart grid from the heterogeneous nature of the communication networks, operating systems, and programming languages [71].

As mentioned earlier, in this work, the Data Distribution Service machine-to-machine middleware from the Object Management Group was utilized to enable real-time integration of various applications within the developed co-simulation framework. The reason for choosing the DDS as a common data bus for the developed platform is that it has an Application Programing Interface (API) that facilitates mapping of other industrial protocols, such as Common Interface Model (CIM), IEC 61850 GOOSE messages and SMV messages into DDS, as will be explained later.

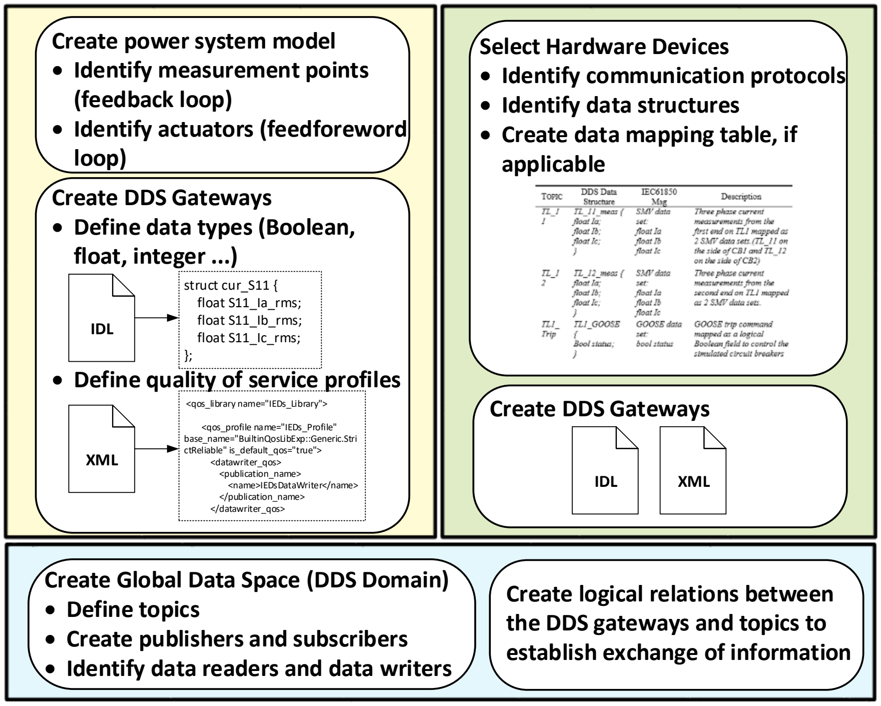
In this work, protocol translators were developed to enable seamless interfacing of multi-vendor/multi-protocol devices with the developed framework.

The focus of this work is providing a credible platform that harmonizes information flow between different smart grid applications and an understanding to their cyber-physical nature. By utilizing the DDS middleware layer here, the proposed framework has the ability to support multiple power system communication protocols and standards, thus, providing interoperability between devices from different vendors. Since DDS could be interfaced with many simulation tools, the developed platform supports fully distributed applications over an expandable number of different simulation stations. The operator does not need to worry about integrating all these devices together as the DDS provides automatic network discovery features, which also learn the various data structures and types of the newly joined devices and/or applications. Therefore, conceptually, a holistic smart grid framework could be modelled over a distributed network architecture, managed by the DDS, and is allowed to interact with real hardware devices over a real network.

2.5.4 Monitoring, Logging, and Remote Connectivity

A monitoring and logging service was developed for the operator to keep track of the operation of the tested systems. The monitoring service also provides a visualization tool in the form on a relational graph that shows the logical relation between the components of the system and their data structures in real-time. The data logging service stores the exchanged information and the network packets for post-processing.

There are a lot of situations in the smart grid where controllers interact with each other and collect sensor data over the wide area network, such as smart metering and demand side management. Therefore, in this work, remote connectivity to the developed framework was established through routing and web integration services.


Create power system model
• Identify measurement points (feedback loop)
• Identify actuators (feedforeword loop)
Create DDS Gateways
• Define data types (Boolean, float, integer ...)
IDL
• Define quality of service profiles
XML
Select Hardware Devices
• Identify communication protocols
• Identify data structures
• Create data mapping table, if applicable
Create DDS Gateways
IDL
XML
Create Global Data Space (DDS Domain)
• Define topics
• Create publishers and subscribers
• Identify data readers and data writers
Create logical relations between the DDS gateways and topics to establish exchange of information

```
struct cur_S11 {
    float S11_Ia_rms;
    float S11_Ib_rms;
    float S11_Ic_rms;
};
```

```
<qos_library name="IEDs_Library">

    <qos_profile name="IEDs_Profile"
base_name="BuiltinQosLibExp::Generic.Stri
ctReliable" is_default_qos="true">
        <datawriter_qos>
            <publication_name>
                <name>IEDsDataWriter</name>
            </publication_name>
        </datawriter_qos>
```

TOPIC	DDS Data Structure	IEC61850 Msg	Description

TL_11	TL_11_meas {
	float Ia;
	float Ib;
	float Ic;
	}
		SMV data set:
		float Ia
		float Ib
		float Ic
			Three phase current measurements from the first end on TL1 mapped as 2 SMV data sets.(TL_11 on the side of CB1 and TL_12 on the side of CB2)

TL_12	TL_12_meas {
	float Ia;
	float Ib;
	float Ic;
	}
		SMV data set:
		float Ia
		float Ib
		float Ic
			Three phase current measurements from the second end on TL1 mapped as 2 SMV data sets.

TL1_Trip	TL1_GOOSE
	{
	Bool status;
	}
		GOOSE data set:
		bool status
			GOOSE trip command mapped as a logical Boolean field to control the simulated circuit breakers

Figure 2.7: Steps to combine the framework components.

Figure 2.7 shows the general procedure to link the components of the developed co-simulation framework together. In terms of the power system, the simulation model needs to be created first. Next the developers need to identify the measurement points (or the feedback loops) and the actuators (or the feed foreword loops). The next step is to create the DDS Gateways. In this step, it is important to identify the nature of the measured data. If the data is circuit breaker status, then it is defined as Boolean, if the data is current measurements, then it is defined as Float, and so on. After that, the data structures are organized in an Interface Definition Language (IDL) file, one file for each gateway.

Finally, an eXtensible Markup Language (XML) file was created to define the quality of service profiles for each application.

In terms of the cyber control, the first step is to select the hardware devices and identify their communication protocol and programming language. Next, a data mapping table, if needed, is to be developed to organize the translation procedure between the protocol of the hardware devices and the data structures defined in the DDS gateways. Finally, the quality of service profiles are defined in XML files.

To link the communication software with the hardware devices, a Global Data Space (GDS or a DDS domain) needs to be created. In that domain, different topics need to be created. The creation of the data topics allows the developers to draw logical relational graphs between all of the components of the system. That is, a relation graph will be created between all the publishers (and the corresponding data writers) and the subscribers (and their corresponding data readers).

Figure 2.6 shows the hardware microcontrollers that were selected to implement the control logic of all applications in the case studies that will be detailed later. The devices are Odroid C2 embedded microcontrollers that have ARM® Cortex®-A53 1.5Ghz processors and are running on a real-time Linux Kernel. Also, a dedicated Ethernet switch was utilized in this framework.

2.6 Verification Case Studies

This section presents three case studies that were implemented on the developed co-simulation framework. The first case study will be about hierarchical control of electric vehicle charging. The emphasis of this case study will be on the utilization of the DDS common information exchange bus to link the components of the co-simulation framework together. The second case study will be a protection application for active distribution networks. Its focus will be on showing the ability of the developed framework to incorporate multi-vendor/multi-protocol devices via protocol translation and emulation. Finally, the last case study will be lost packet forecasting and will focus on the importance of the developed framework in cyber-resiliency studies.

2.6.1 Hierarchical Control of Electric Vehicle Charging

A block diagram of the microgrid in this case study is shown in Figure 2.8. It consists of four buses with a local conventional generator connected to bus B1 and an inverter-based photovoltaic (PV) source connected to bus B4. The microgrid has four constant loads, each connected to a different bus, and two 3-phase dynamic loads connected to buses B3 and B4. An EV is connected to every bus. This is done to take into consideration the anticipated high penetration of electric vehicles. The PV source is controlled through an inverter, which controls the amount of injected power to the microgrid. The charging of the EVs is done through controlling the pulse width modulation of the DC-DC converter.

As shown in Figure 2.8, a cyber-layer is present on top of the physical layer. It is composed of a two-level hierarchy of agents. The lower layer of the hierarchy is composed of 4 EV agents, which are responsible for calculating the requested charging current for the EV chargers. The higher layer of the hierarchy is composed of a microgrid control agent (MCCA), which is responsible to set the final reference charging current for every EV to charge the batteries while maintaining the voltages at the different buses within the ANSI standards [74].

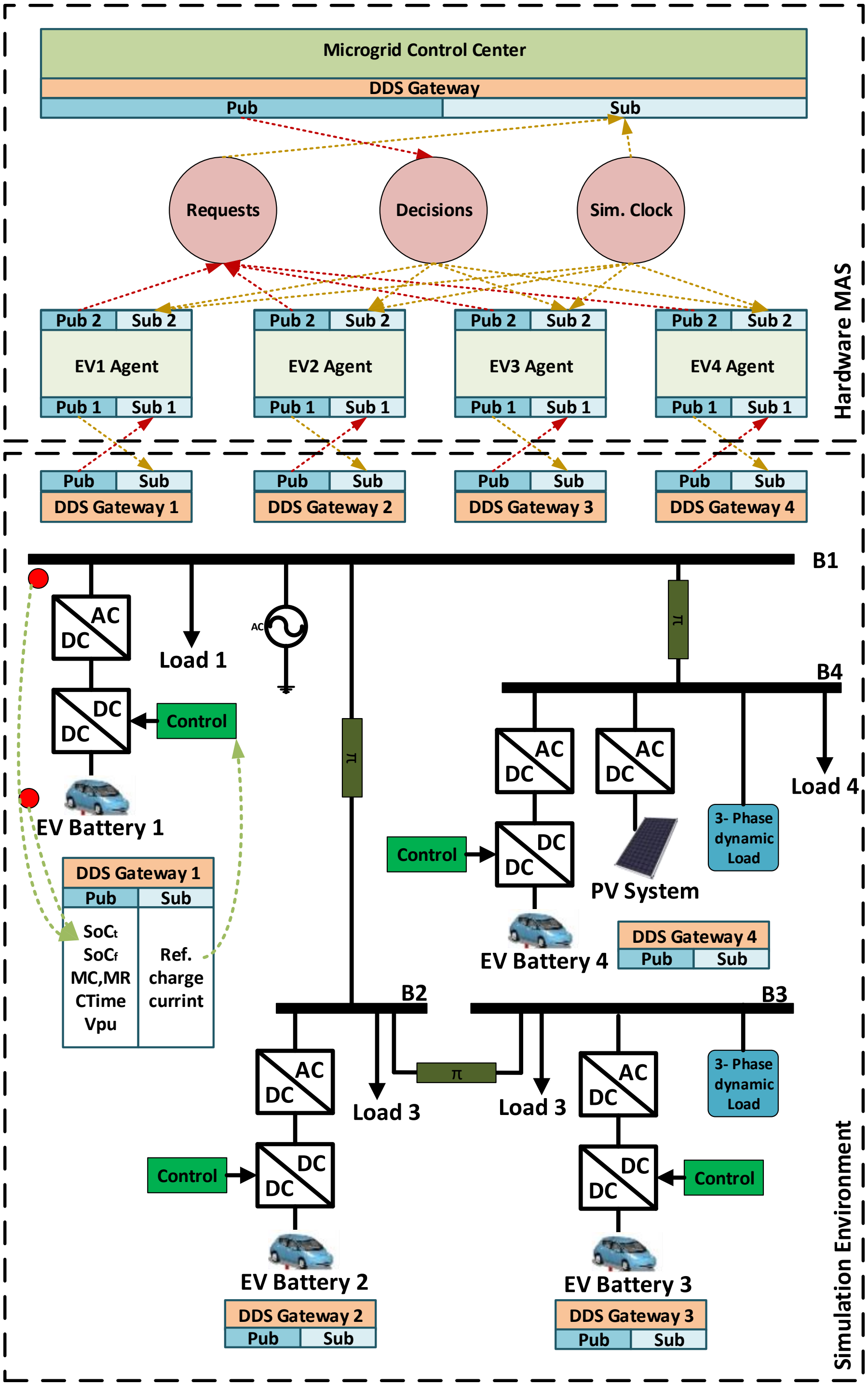


Figure 2.8: Hierarchical control of EV charging.

A communication layer links the physical layer to the cyber layer through a Global Data Space, which contains three topics: Requests, Decisions, and Simulation Clock. Due to the complexity of the simulated power system and the limitation on the available hardware resources, the simulation clock is much slower than the real-time clock. Therefore, the *Simulation Clock* topic was developed to synchronize the hardware agents with the clock of the simulation. A DDS gateway was created for every bus of the simulated microgrid. On one hand, these gateways will collect and publish the necessary input for the hardware agents. On the other hand, they will subscribe to the commands issued by the hardware agents and execute them on the simulated microgrid.

Every 6 minutes, each EV Agent reads the current state of charge (SoC_c) of the EV battery and calculates the requested charging current (I_{req}) based on equations (2.1) and (2.2).

$$I_{req} = \frac{\left(SoC_f - \ SoC_c\right) \times MC}{t} \tag{2.1}$$

$$I_{req\_f} = \min(I_{req}, I_{rated}) \tag{2.2}$$

where SoC_f is the final state of charge, which is set to 80%, $MC = 24\ kWh$ is the battery capacity, $I_{rated} = 15\ A$ is the maximum charging rate, and t is the requested time to reach SoC_f. In this study $t = 4, 6, 7, and\ 5\ hrs$ for EV1, EV2, EV3, and EV4 respectively. After calculating the requested charging current, each EV agent writes a charging request vector to the *Requests* topic, as shown in equation (2.3).

$$R = \left\{EV_{ID}, I_{req_f}, V_{pu}\right\} \tag{2.3}$$

where EV_{ID} is the unique identifier of the EV and V_{pu} is the p.u. voltage of the bus at which the EV is connected.

Next, the Microgrid Control Center (MCC) Agent collects all the charging requests and readjusts the charging currents for each EV according to the heuristic rules in (2.4) and (2.5). The MCCA utilizes (2.4) to guide its decision in the cases where the bus voltages are healthy, i.e. no under voltage. However, the MCCA utilizes (2.5) to guide its decision when there is an under voltage on at least one bus.

$$I_{ref_i} = \begin{cases} I_{req_i} & if \ \ (V_i > 0.97) \ \forall \ i \in [1,4] \\ 0.75 \ I_{req_i} & if \ (0.955 < V_i \leq 0.975) if \exists i \epsilon [1,4] \end{cases} \qquad (2.4)$$

$$I_{ref_i} = \left\{ k \ I_{req_i} \quad \begin{matrix} k = 0 \ if \ V_i \leq 0.955 \ \ i\epsilon[1,4] \\ k = 0.5 \ otherwise \end{matrix} \right. \qquad (2.5)$$

The MCCA publishes its decisions vectors $D = \{EV_{ID}, I_{ref}\}$ to the *Decisions* topic.

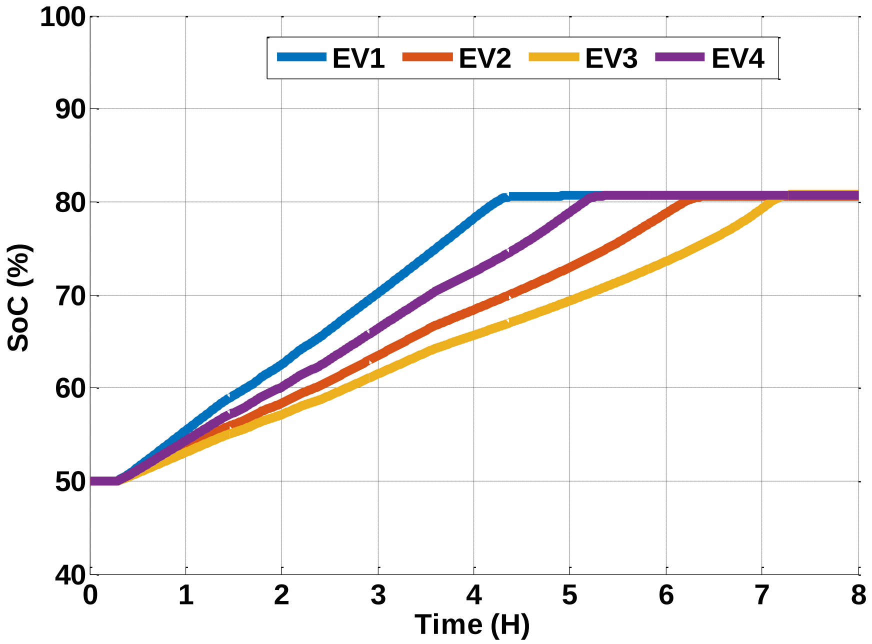


Figure 2.9: Battery state of charge.

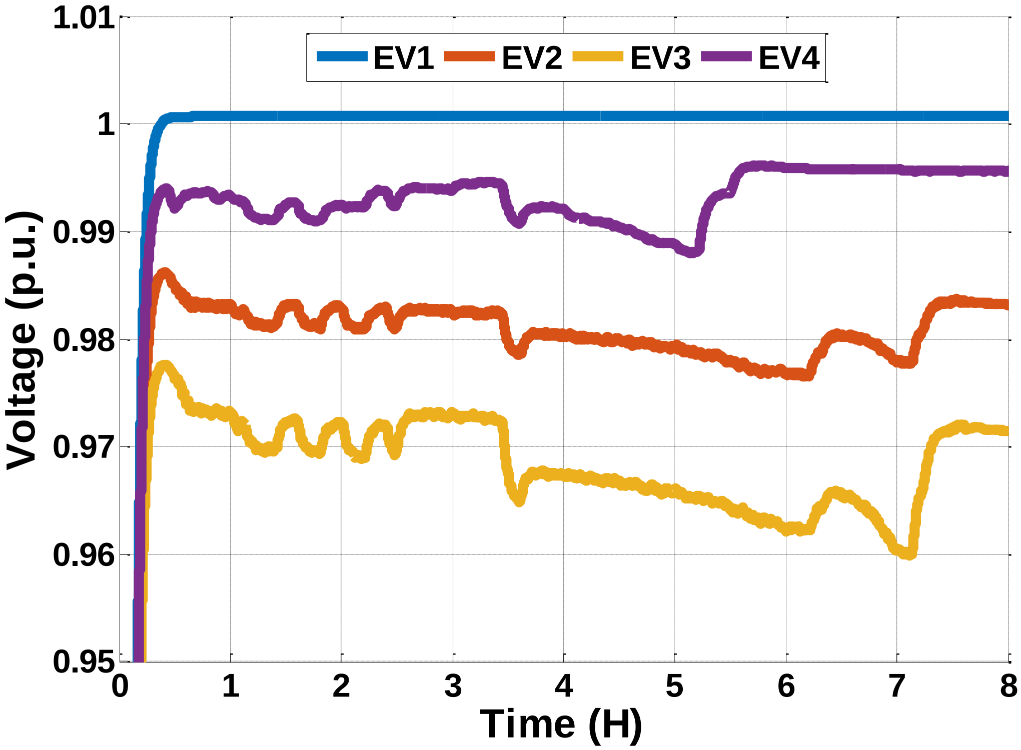


Figure 2.10: Bus voltages in p.u.

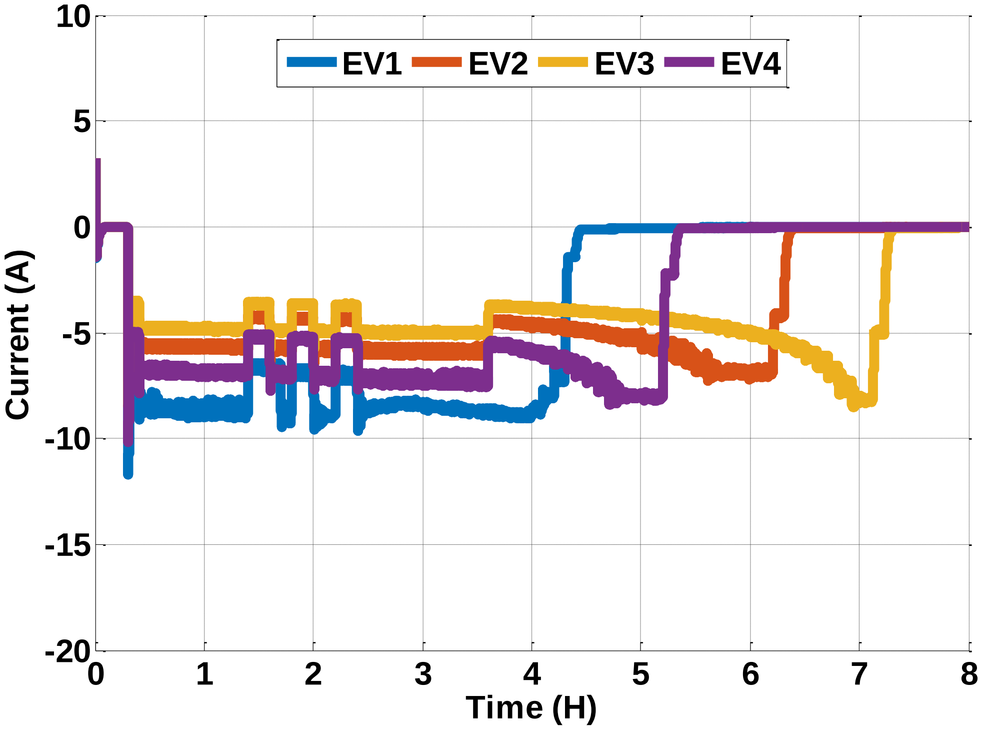


Figure 2.11: Battery currents.

Figure 2.9 shows the SoC of the EV batteries over time. It can be seen that all EVs reached 80% SoC within the requested periods (t=4.2, 6.2, 7, and 5.1 hrs for EV1, EV2, EV3, and EV4, respectively). The voltages in per unit at the four different buses are shown in Figure 2.10. During the experiment, the voltages at all buses were maintained above 0.95 p.u, which is desired. One can see that the voltage at B1 is the highest since it is connected directly to the generator. At bus 4, the voltage drop across the feeder is compensated for by the power injected from the inverter. Therefore, B4 maintained high voltage values. However, buses B2 and B3 have lower voltages since they are far from the generation buses, especially B3, since it is connected at the far side of the feeder. The fluctuations in the charging currents of the EV batteries can be seen in Figure 2.11. It is evident that the charging currents are following the reference values set by the MCCA.

As can be appreciated from the results of this case study, the developed framework was successful in providing a smooth link between a multi-agent hardware/software infrastructure and a simulated power system. Through this link, the effect of the control logic, which was implemented in C++, was tested and the response of the power system to the control logic was analyzed.

2.6.2 Protection of an Active Distribution Network

In this case study, we focus on how the developed framework supports the incorporation of multi-protocol devices through protocol translation. To achieve that, an IEC 61850-based differential protection algorithm is designed and interfaced with a 4-bus active distribution network.

The developed microgrid model is shown in Figure 2.12. The microgrid under study consists of 3 transmission lines, named TL1, TL2, and TL3. Various types of AC and DC loads are connected to Buses 3 and 4. Transmission line one links the microgrid to the utility grid through Circuit Breakers (CBs) 1 and 2. Power is transferred from either the main utility grid or the two distributed generators (G1 and G2) to the system busses and their corresponding loads over transmission lines TL1 and TL2. Three DDS gateways were incorporated into the model in order to provide an interface link between the simulated microgrid on one hand, and the physical Merging Units (MUs) and IEDs on the other. The DDS gateways receive the three phase current measurements from Simulink and publish them into the DDS GDS, each to its corresponding topic.

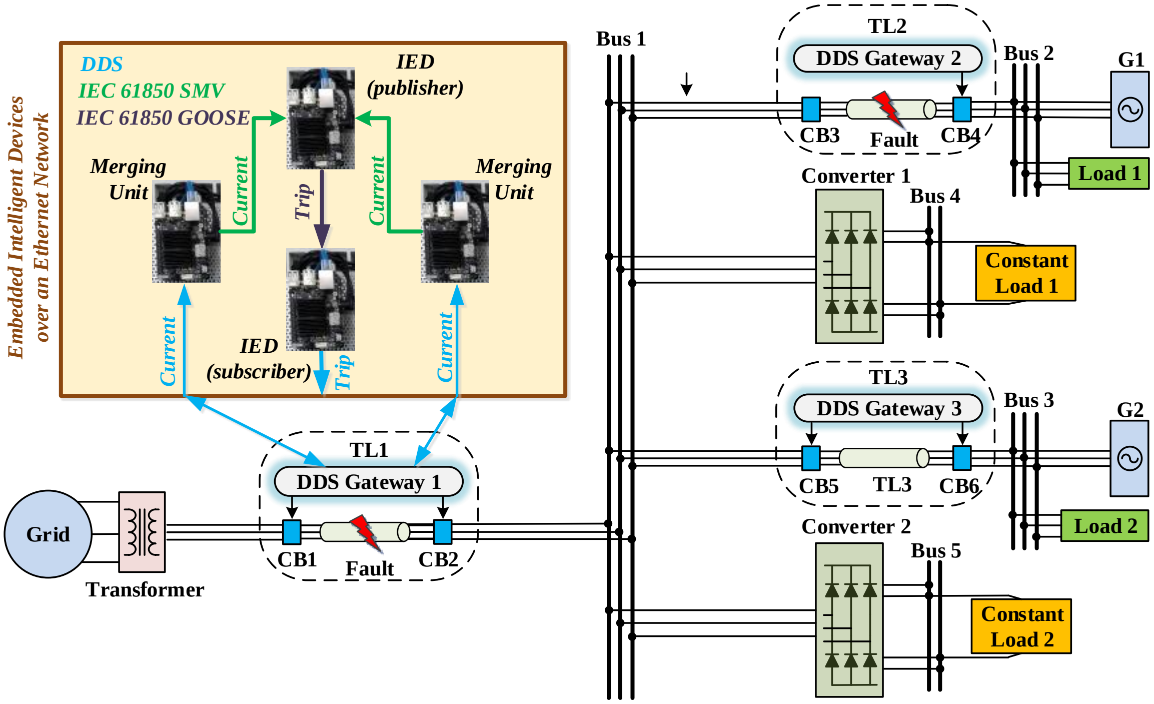


Figure 2.12: Simulated distribution network linked to the IEC 61850 agents.

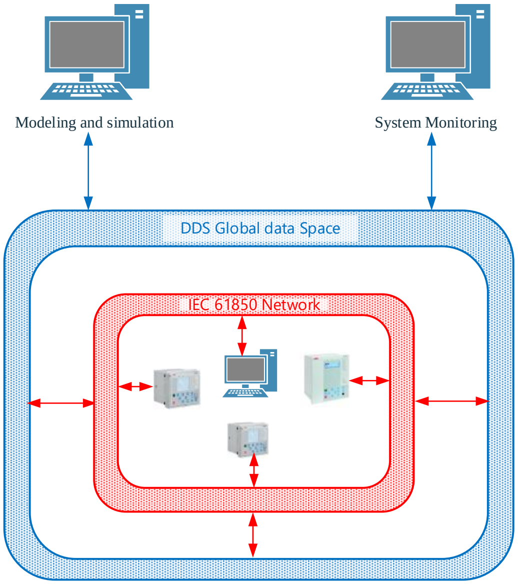


Figure 2.13: Architecture of the communication network.

Table 2.2. Example DDS Data Structure & Corresponding Iec 61850 Messages

Topic	**DDS Data Structure**	**IEC61850 Msg**	**Description**

TL_11	*TL_11_meas {*
	float Ia;
	float Ib;
	float Ic; }
		SMV data set:
		float Ia
		float Ib
		float Ic
			Three phase current measurements from the first end on TL1 mapped as 2 SMV data sets.(TL_11 on the side of CB1 and TL_12 on the side of CB2)

TL_12	*TL_12_meas {*
	float Ia;
	float Ib;
	float Ic; }
		SMV data set:
		float Ia
		float Ib
		float Ic
			Three phase current measurements from the second end on TL1 mapped as 2 SMV data sets.

TL1_Trip	*TL1_GOOSE {*
	Bool status; }
		GOOSE data set:
		bool status
			GOOSE trip command mapped as a logical Boolean field to control the simulated circuit breakers

The merging units then receive the measurements, translate them into IEC 61850 SMV packets, and publish them over the Ethernet network. For each transmission line, a publisher IED reads the published measurements and issues a trip GOOSE message according to the protection scheme coded into its firmware. The subscribing IED reads the GOOSE commands and translates them into DDS messages to control the status of the circuit breaker in the simulated microgrid. The mapping between the DDS and the IEC61850 GOOSE and SMV messages was coded in C with the help of libraries from the DDS API from Real Time Innovation (RTI) OMG and the open source library libIEC61850 [72][75]. An IDL file, which contains the IEC 61850 data modules, is passed to the automatic publisher/subscriber code generator from RTI. The generated code is then interfaced with the routines available from libIEC61850. The developed codes were downloaded on Odroid C2 single board computers (SBCs) running a Linux kernel. The communication took place over a dedicated Ethernet network switch, as shown in Figure 2.13. Table 2.2 provides an example correspondence between DDS and IEC 61850 SMV and GOOSE messages for the section covering TL1. Measurements are assigned float data types, whereas circuit breaker statuses are assigned Boolean data types.

As mentioned earlier, the DDS provides a flexible and standard API to integrate with different systems and programming languages. Since DDS is a data-centric communication middleware, the message structure is driven directly from the system data model without the need for a predefined set of structures for messages. In addition, the automatic code generation for publishers/subscribers based on the data models defined by XML and/or IDL files simplifies the integration with different data types for other protocols utilized by different IED and remote units' vendors. Figure 2.13 shows how the IEC 61850 is linked to the simulation environment through the DDS global data space.

A three-phase-to-ground (ABCG) fault was applied on TL2 of

Figure 2.12 at t = 5.50 seconds. DDS Gateway 2 reads the current measurements at both ends of the transmission line and publishes them as DDS messages. Two physical merging units are subscribing to the values of the current published from each end of the transmission line. From this point, the actual IEC 61850 process bus is implemented.

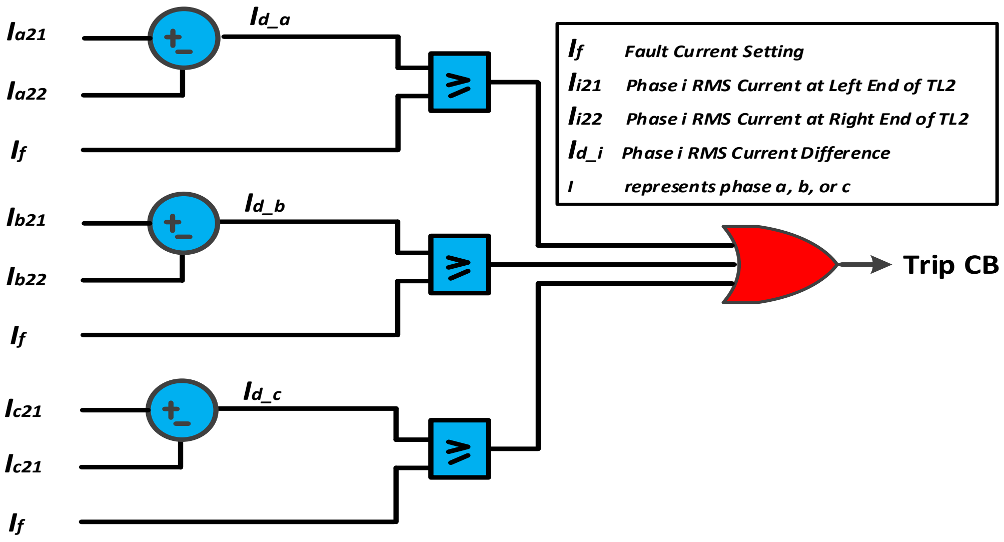

Ia21
Ia22
If
Id_a
+
−
≥
Ib21
Ib22
If
Id_b
+
−
≥
Ic21
Ic21
If
Id_c
+
−
≥
If Fault Current Setting
Ii21 Phase i RMS Current at Left End of TL2
Ii22 Phase i RMS Current at Right End of TL2
Id_i Phase i RMS Current Difference
I represents phase a, b, or c
Trip CB

Figure 2.14: Protection logic.

These merging units publish the measured current values as SMV packets at a 4,800 Hz publishing rate, as set by IEC 61850 for 60 Hz systems. An IED, which has the protection logic implemented on it, will then subscribe to these SMV messages. First, the IED will calculate the Root Mean Square (RMS) value of all received measurements. Next, the IED will calculate the difference in the RMS current values for each phase according to equation (2.6)

$$I_d^{abc} = I_{21}^{abc} - I_{22}^{abc} \tag{2.6}$$

where I_d^{abc} is the per phase difference in current, I_{21}^{abc} is the phase current at the left end of transmission line TL2, and I_{22}^{abc} is the phase current at the right end of TL2. If the difference in phase A current, phase B, or phase C currents at both ends of TL2 is greater than or equal to the fault current setting, the IED will issue a trip command in the form of a GOOSE message. This is represented in the logical diagram of Figure 2.14

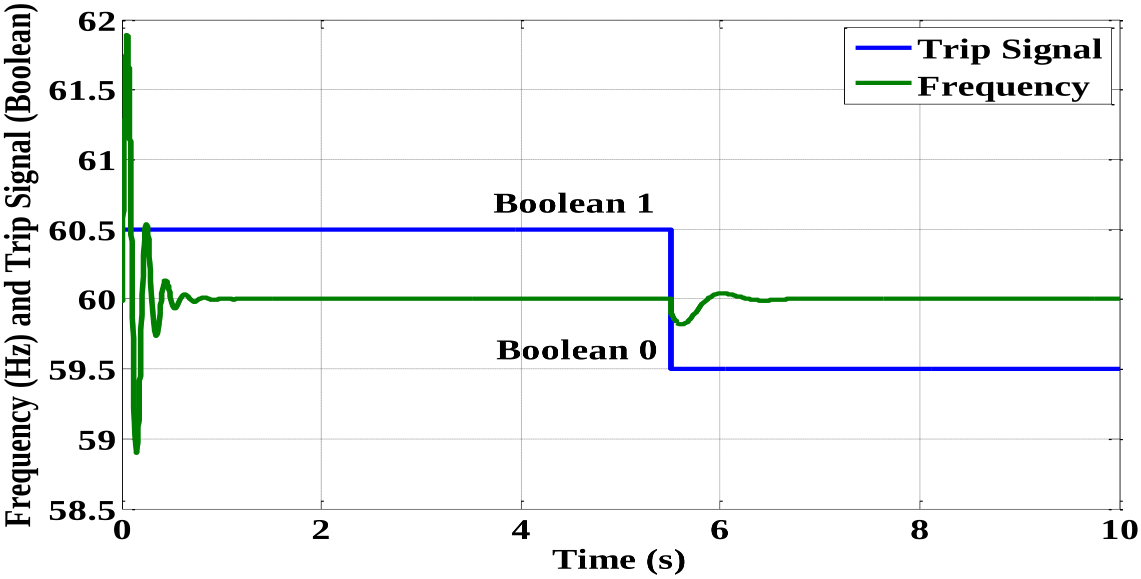


Figure 2.15: System frequency and status of CB3.

$$CB\ Status = \begin{cases} 1 & (I_{d\_a} \geq \ I_f)\ ||\ (I_{d\_b} \geq \ I_f)\ ||(I_{d\_c} \geq \ I_f) \\ 0 & otherwise \end{cases} \tag{2.7}$$

Finally, an IED subscribing to this GOOSE message will map it into a DDS Boolean command to control the status of CB3 and CB4 in the simulated microgrid model. The frequency of the system was recorded, and is shown along with the status of the circuit breakers in Figure 2.15.

Figure 2.15 shows a drop in the system frequency from 60 Hz to a minimum of 59.82 Hz after the fault occurs at t = 5.50 seconds. The system frequency was restored to 60 Hz after the circuit breakers (CB3 and CB4) opened in response to a GOOSE trip message

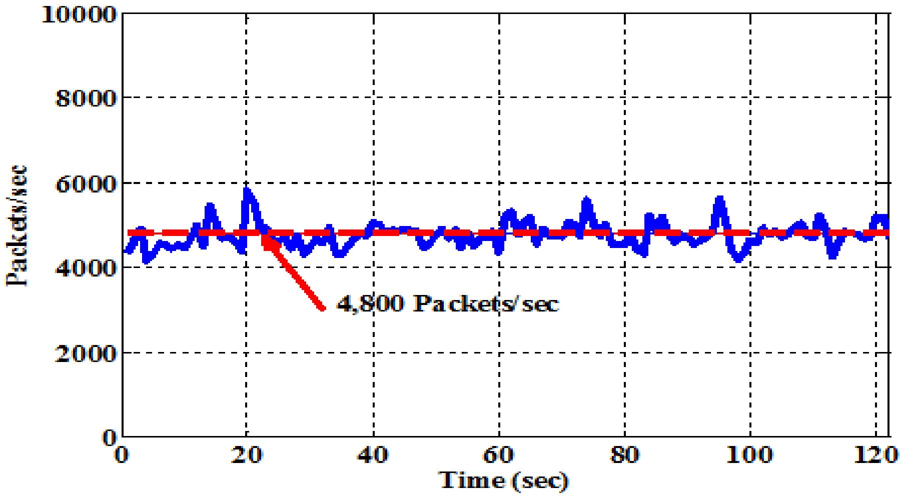


Figure 2.16: Performance of SMV messages.

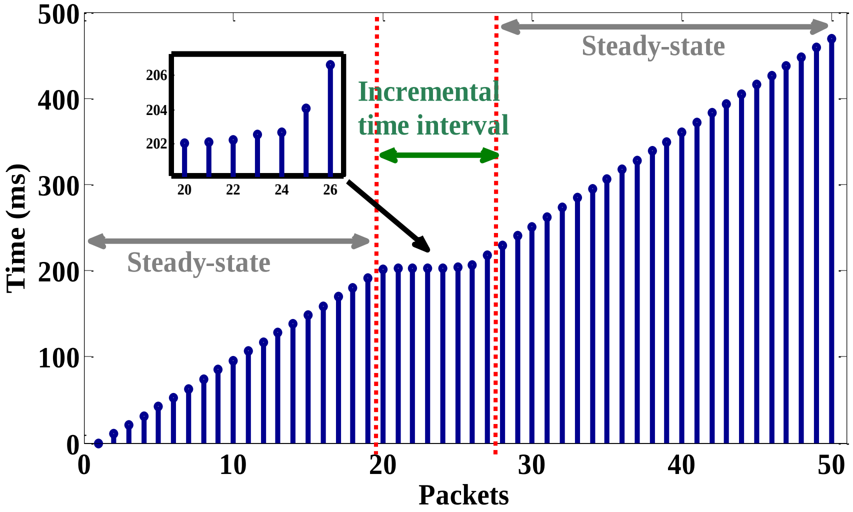


Figure 2.17: Performance of GOOSE messages.

Table 2.3. Recorded Time Delay

MESSAGE TYPE	Average Delay Time
SMV	*120 μs*
GOOSE	*11.2 μs*

sent by the intelligent electronic device. As can be appreciated from the results, the microcontrollers were able to sense, locate, and send trip commands to isolate the fault in the simulated model in a timely manner based on feedback measurements sent to the controllers through the DDS Gateways. Similarly, the simulation model received the correct trip commands from the controllers through the DDS gateway and opened the appropriate circuit breakers to clear the fault. The cyber part of the system, which was exchanging SMV and GOOSE messages over a real Ethernet network, was successfully integrated with the simulation software, which was recording the dynamics of the power system in response to the control actions. Again, this verifies the ability of the proposed framework to accurately capture the relation between the cyber information flow and the physical information flow in power systems.

Additionally, the co-simulation framework was used to analyze the performance of the IEC 61850 process bus in terms of sampling rate and transmission delays. SMV and GOOSE messages were recorded using Wireshark network analyses tools. The transmission rate of SMV messages was recorded in Figure 2.16. The average transmission rate of SMV messages was found to be around 4.76 KHz, which is close to the 4.8 KHz set recommended by IEC 61850 [8][41][42]. On the other hand, Figure 2.17 shows the re-transmission rate of GOOSE messages controlling the status of CB3. Figure 2.17 shows that during steady state operation of the microgrid, the GOOSE messages were being retransmitted with a fixed rate of about 10 ms. Packet number 20 reflects the event of a fault. During such events, the IED sends GOOSE messages with an incrementing transmission rate until the system reaches steady state again. Table 2.3 shows that the average delays recorded for GOOSE and SMV messages fall within the 4 ms constraint.

2.6.3 Online Lost Packet Forecasting for IEC 61850 Sampled Measured Values

Today, MUs play a significant role in the IEC 61850 process bus. In modern substations, MUs are installed near primary equipment in switchyards and are meant to provide synchronized phase voltage and current measurements, digitize them into an SMV packet format, and publish them to the substation process bus [8][76][77]. The deployment of the Process Bus in SAS has introduced several advantages, including reduced copper wiring, modular substation architectures, easier commissioning and maintenance processes, and enhancing protection and control applications by leveraging data availability.

From a network implementation perspective, several points need to be considered in the deployment of the IEC 61850 process bus. IEC 61850 recommends that for a power system operating at a 50-60 Hz frequency, SMV messages should have a transmission rate of 80 samples per AC cycle. Therefore, 4,800 samples are transmitted, for a given set of measurements, per second. Assuming an SMV frame size of 124 bytes, as more merging units are publishing packets, more network bandwidth is consumed, and thus, more sampled packet delays are incurred, leading to the loss of some samples [78]. Also given the fact that all MUs need to be synchronized [77], SMV traffic will be simultaneously generated, increasing the latency of these messages. The work in [78] further emphasizes the consequences of lost SMV packets in SAS.

Complex forecasting techniques, such as non-linear interpolating polynomials and time series, have been widely used in the literature in applications requiring sensor measurements [79][80][81]. Although very accurate, these highly nonlinear models require a lot of computational time, which is not feasible in the case of SMV. Therefore, the problem of compensating for lost sampled values within the limited time delays and limited computational power of field devices, is poorly addressed in the engineering literature.

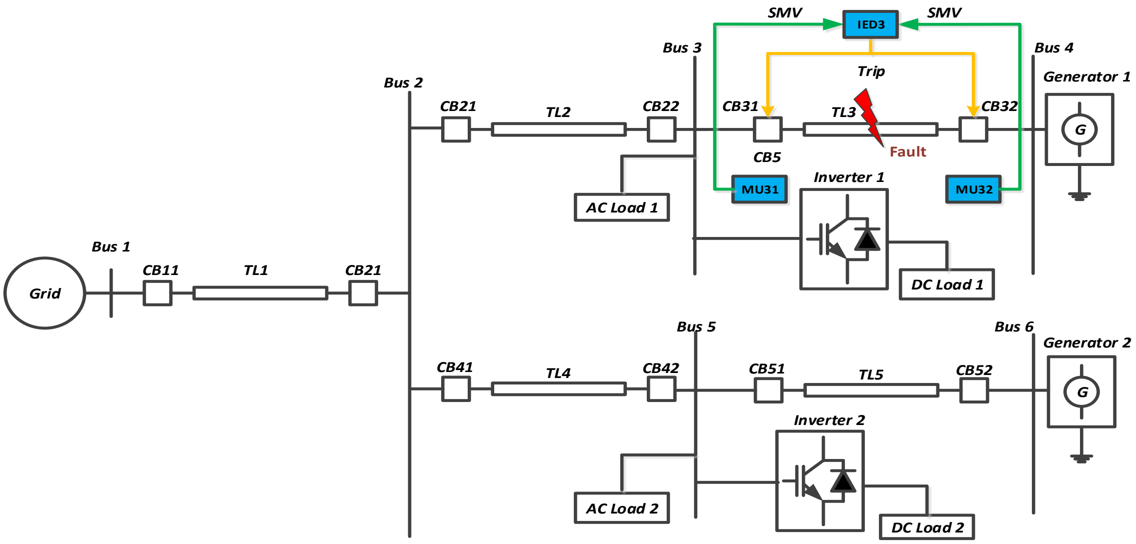


Figure 2.18: Block Diagram of the Physical System.

As such, in this case study, an online lost packet forecasting algorithm using time series Neural Networks (NNs) for IEC 61850 SMV messages is presented. The system takes advantage of the fast responses of the NN in order to meet the time requirements of SMV messages. IEC 61850-based MUs and IEDs were programmed and connected to the co-simulation platform’s network. This to assess the performance of the system in a real-time environment and over a real network. The results showed excellent performance in accurately forecasting lost samples.

Figure 2.18 shows the system used to generate the proper training data for the lost sample forecasting NN. The power system, which was simulated on MATLAB Simulink, consists of 5 transmission lines, to which various types of linear and non-linear loads are connected. Non-linear loads were present to add harmonic contents to current and voltage measurements. Transmission line TL1 connects the system under study to the main utility grid. Transmission lines TL2 to TL5 transfer power from either the utility or the generators, G1 and G2, to the system busses and their corresponding loads. All transmission lines are modeled to be10 Km in length.

In order to guarantee accurate responses of the neural network, care must be put in generating and selecting appropriate training data. Two major cases were studied: the first one is grid-connected mode of operation, whereas the second is islanded mode of operation. For each of the two mentioned cases, and on each transmission line, 5 types of faults were applied. These fault types are single-line-to-ground (A-G), line-to-line (B-C), double-line-to-ground (B-C-G), three-phase (A-B-C), and 3-phase-to-ground (A-B-C-G) faults. Additionally, each type of fault was applied on the beginning (10%), middle (50%), and end (90%) of each transmission line.

For all the cases mentioned above, current and voltage measurements from MUs on TL3 were recorded, since this is where the proposed algorithm was implemented. To make sure not to over train the network, only samples that represent the shift from normal operation to the unstable fault operation were utilized for training. Also, one AC cycle after the fault clearance was used for training.

As mentioned earlier, MUs digitize voltage and current measurements and send them over the processes bus as broadcast messages for various SAS applications to use. As shown in Figure 2.18, IED3 subscribes to SMV messages published by MU31 and MU32. In a simple differential protection scheme, IED3 calculates the difference between the received current values and compares the results to a predefined threshold. If the difference is greater than the threshold, then IED3 sends a trip signal in the form of a GOOSE message in order to isolate the fault. Therefore, the proposed lost samples forecasting system is implemented on IED3. Since the forecasting problem in this case depends only on a single variable, which is time, then a time series approach was used in training the developed neural network. In order to capture all the features in the measured signals and decrease the forecasting error, a window of 20 samples covering a quarter AC cycle for a 60 Hz system was used to train the NN. The developed NN has an input layer with twenty neurons to account for the 20 history samples, a hidden layer with 10 neurons, and an output layer with 1 neuron for the forecasted sample.

As seen in the flowchart in Figure 2.19, after 20 samples are received, the system forecasts a new SMV only if the 4ms time constraint is exceeded. Next, if the new SMV is received before 4ms, therefore, it is not lost and it will be used.

Figure 2.20 shows the results of forecasting lost SMV packets for various fault types and locations according to the proposed algorithm. Randomly selected packets were dropped (shown with zero values on the blue waveform), covering fault and normal operating conditions of the studied microgrid. Here, it is noteworthy to emphasize that the developed forecasting system was able to forecast accurately lost samples during fault peaks. This is of particular importance, since the IED settings are commonly programmed at these higher current values. If a fault occurred and those samples were dropped, the IED will not sense the fault and thus renders the protection scheme failed. However, the proposed scheme ensures smooth operation of protection the application, even under congested or poor network conditions.

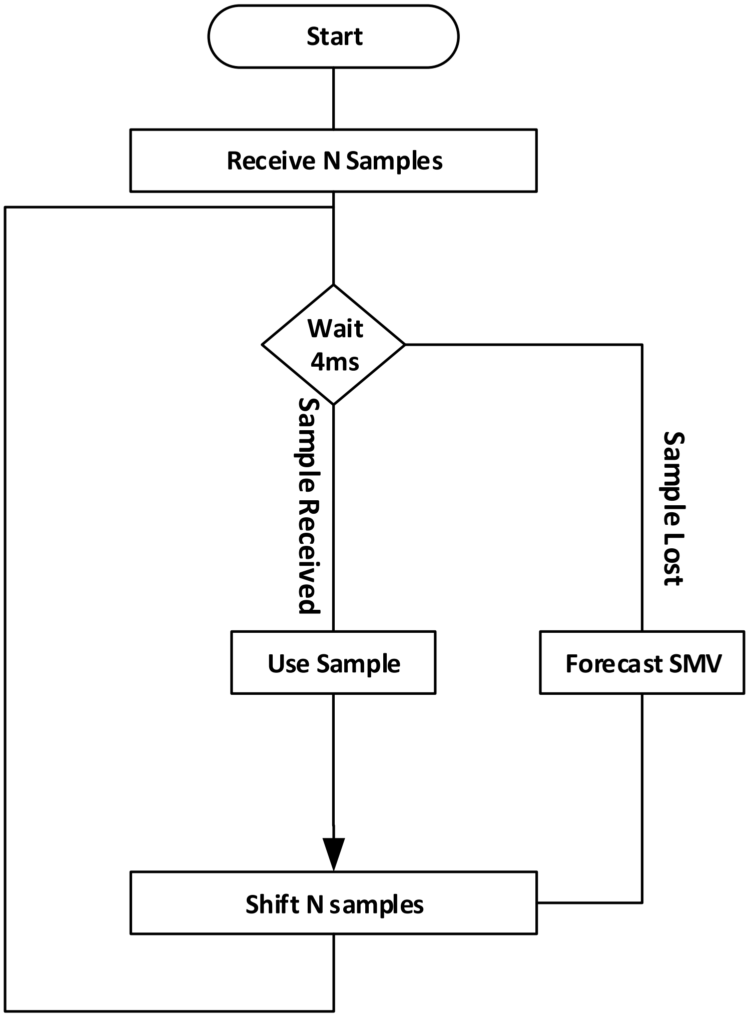


Figure 2.19: Online IDS and Forecasting Algorithm

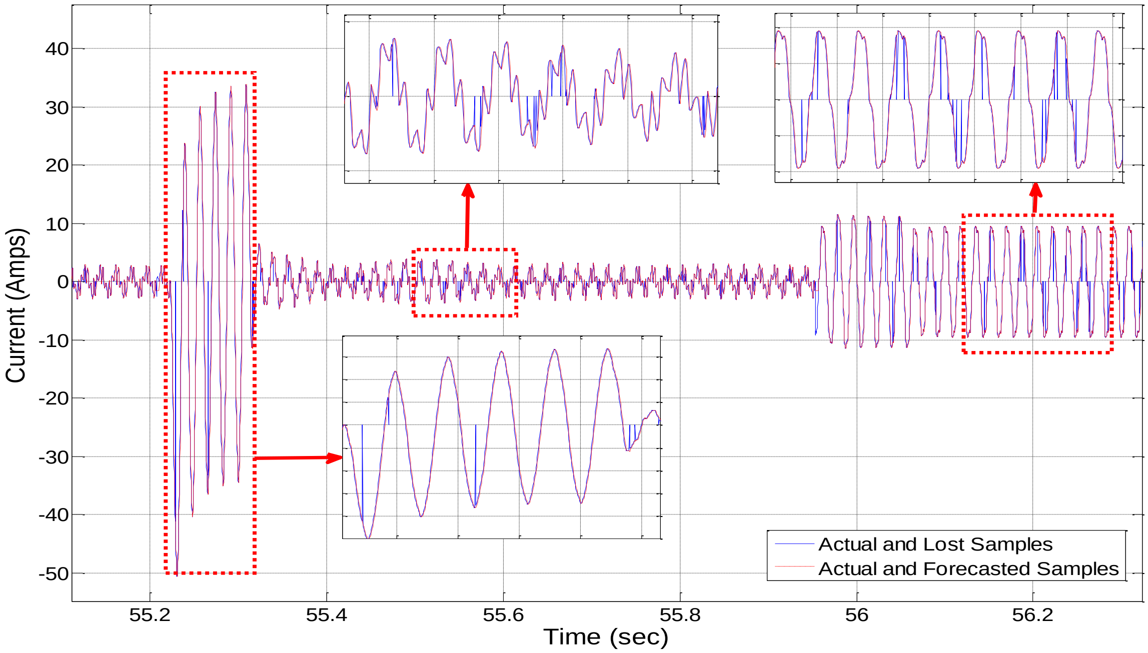


Figure 2.20: Response of the Lost Sample Forecasting Algorithm

2.7 Conclusion

In this chapter, a survey analyzing the recent co-simulation platforms for smart grid applications was developed and provided. We have shown that there is a need to provide a comprehensive co-simulation platform capable of modeling the relation and interaction between the cyber and physical parts of the smart grid. Accordingly, this chapter presented a network-in-the-loop co-simulation platform that formalizes the understanding of cyber information flow, physical power system dynamics, and the interrelation between them. A discussion about the importance of interoperability in large and highly complex systems is given. The data-centric DDS communication middleware was selected to orchestrate the components of the developed co-simulation platform together and to bridge the different communication protocols and software applications.

Three case studies were presented to verify the effectiveness of the developed network-in-the-loop co-simulation platform in analyzing the interrelation between the cyber and physical information flow in smart grid applications. First, an electric vehicle charging algorithm was implemented. Second, an IEC 61850-based microgrid protection algorithm was developed, in which two protocols, GOOSE and SMV, and several applications implemented in C and MATLAB were interleaved together through the DDS common data bus. Finally, to satisfy the purpose of the co-simulation platform in testing new algorithms that enhance the resiliency of the power system to cyber vulnerabilities, a lost sample forecasting algorithm for SMV messages was implemented and tested on the developed framework. The results showed the plausibility of such an expandable hybrid framework and emphasized its importance in the analyses of physical and cyber information flow and testing of new control algorithms in a safe and credible environment.

Chapter 3 On the Implementation of the IEC 61850 Standard

Standardization in smart grid communications is necessary to facilitate complex operations of modern power system functions. However, the strong coupling between the cyber and physical domains of the contemporary grid exposes the system to vulnerabilities and thus places more burden on standards' developers. Therefore, standards need to be assessed continuously for reliability and are expected to be implemented properly on field devices. However, the actual implementation of common standards varies between vendors, which may lead to different behaviors of the devices even if present under similar conditions. The work in this chapter tested the implementation of the IEC 61850 GOOSE messaging protocol on commercial Intelligent Electronic Devices (IEDs) and the open source libiec61850 library - which is also used in commercial devices - which showed different behaviors in identical situations. Based on the test results and analysis of some features of the IEC 61850 GOOSE protocol itself, this chapter proposes guidelines and recommendations for proper implementation of the standard functionalities.

3.1 Introduction

Communication protocols are the basis for determining how a cyber-physical system gathers information and sends it as control signals. Therefore, an accurate definition of communication protocols is of paramount importance in defining the architecture of control systems [9]. However, intricate interdependencies between the cyber and physical components of a cyber-physical system increase the difficulty of devising communication protocols that ensure proper information flow in such systems, and thus complicates the design process of control algorithms. The challenge lies in the fact that in a highly interconnected cyber-physical system, a slight exploit in the cyber domain can have a significant impact in the physical domain and vice-versa [2]. In current days, the operation of commercial, industrial, medical, military, and many critical infrastructures relies on the cyber-physical smart grid. The reliance of such critical infrastructure on the smart grid means reliance on the grid’s cyber domain, physical domain, and most importantly, the interactions between them [2]. Therefore, understanding and modelling data exchange in the smart grid is a noticeably challenging process with considerable effort placed on accurately capturing the interactions between both the cyber component and the physical component of the grid.

In order to solve the information flow modelling problem and facilitate the design of cyber-physical smart grid applications, various data communication standards were developed for different parts of the smart grid. One of the vital standards in the electrical automation systems around which many automation projects have been built is IEC 61850 [82][83][84]. Communication between substation devices, namely IEDs, is integral for substations to keep up with their real-time operations. IEDs are embedded microcontroller systems that support Ethernet-based communication and perform several protective and control functions in an SAS, such as data and file transfer [82][85]. In order to ensure interoperability between IEDs, the IEC 61850 standard was developed by the IEC Technical Committee Number 57 Working Group 10 (TC57 WG10) and IEEE for Ethernet (IEEE 802.3)-based communication in electrical substations [86]. The IEC 61850 provides a comprehensive data modelling and organization method that unifies data structure definitions across all IED brands. The standard abstracts the definition of the service and data items to be independent from the underlying protocols. The abstracted data items and services can thus be mapped into any other communication protocol. IEC 61850 maps the data to three different protocols based on the application: the Manufacturing Message Specification (MMS) protocol is used for control and automation functions, whereas the GOOSE and SMV protocols are used for real-time operations [42]. Recently, the IEC 61850 has been extended (IEC 61850-90-1) to cover applications that require inter-substation communication, such as tele-protection, which requires the use of GOOSE messaging protocol over WAN for fast propagation of control signals [87][89]. A close look at the current engineering literature also shows a trend of utilizing IEC 61850 GOOSE messages in some microgrid control applications [90][91].

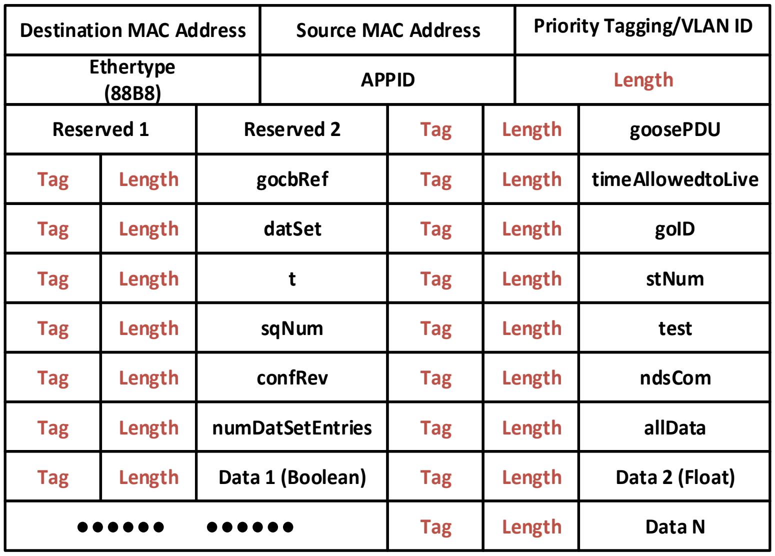
Standards are developed to be implemented. Nonetheless, due to different device topologies and hardware used, not all vendors follow the same implementation process. Here, concerns arise about the degree of compliance of devices from different vendors with the standard being implemented. Due to its criticality, failing to implement the IEC 61850 standard properly on field devices may expose the overall system they operate in to unwanted vulnerabilities. In fact, the criticality of IEC 61850-based communications in terms of data transfer, reliability, availability and efficiency has been the concern of several research works [92][93]. In this work, a case study of the implementation of IEC 61850 GOOSE messaging on commercial IEDs present at the Smart Grid test bed at Florida International University and the open source libiec61850 [75] library, which is also implemented on commercial devices, was performed. GOOSE messaging protocol in particular is of paramount importance due to its application in transporting time-critical power system protection commands. Several experiments were conducted to test critical features of the standard, which are detailed later in this chapter. The results showed that different devices produce different responses under similar conditions. This paved the way to launch a successful data manipulation cyber-attack on the devices under study.

3.2 IEC 61850 GOOSE Messaging Protocol

GOOSE messaging is a fast, non-routable, and reliable data exchange protocol between IEDs defined in IEC 61850-8-1, which is the basis of critical power system functions, such as power line protection. GOOSE messages are Ethernet messages sent over layer 2 of the Open System Interconnect (OSI) model (IEEE 802.3) following a publish/subscribe model, unlike MMS messages, which are routable and sent over layer 3 of the OSI model. That is, the publishing IED creates a multicast message to which a number of destination IEDs subscribe concurrently. In order to ensure delivery of the message, at every substation event, the publishing IED repeatedly transmits the same GOOSE message with an increasing transmission period until a maximum predefined period is reached [8].

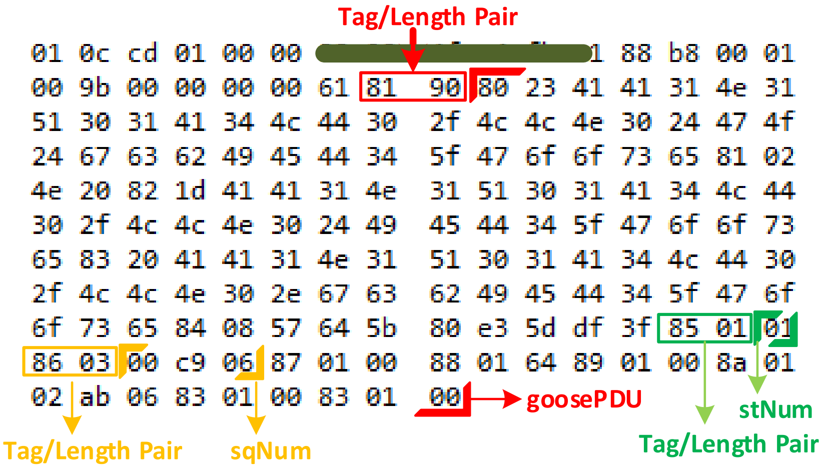
3.2.1 The Anatomy of a GOOSE Message

As shown in Figure 3.1, the GOOSE datagram follows a modified Abstract Syntax Notation One (ASN.1) Basic Encoding Rules (BER) Tag/Length pair encoding scheme [8]. The Tag field represents the type of information, which is represented in the following GOOSE frame. Each of the fields has a unique tag value specified by the standard. As shown in Figure 3.2, the tag for the GOOSE Protocol Data Unit (goosePDU) field is 81, whereas the tags for the stNum and sqNum fields are 85 and 86, respectively. The Length field represents the number of bytes in the following GOOSE frame. For example, the sqNum in Figure 3.3 has a Length field of 03, which means that the following three hex pairs (00-c9-06) are the sqNum itself. The sqNum here is in hexadecimal.



<table>
<tr><td colspan="2">Destination MAC Address</td><td colspan="2">Source MAC Address</td><td colspan="2">Priority Tagging/VLAN ID</td></tr>
<tr><td colspan="2">Ethertype
(88B8)</td><td colspan="2">APPID</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved 1</td><td>Reserved 2</td><td>Tag</td><td>Length</td><td>goosePDU</td></tr>
<tr><td>Tag</td><td>Length</td><td>gocbRef</td><td>Tag</td><td>Length</td><td>timeAllowedtoLive</td></tr>
<tr><td>Tag</td><td>Length</td><td>datSet</td><td>Tag</td><td>Length</td><td>goID</td></tr>
<tr><td>Tag</td><td>Length</td><td>t</td><td>Tag</td><td>Length</td><td>stNum</td></tr>
<tr><td>Tag</td><td>Length</td><td>sqNum</td><td>Tag</td><td>Length</td><td>test</td></tr>
<tr><td>Tag</td><td>Length</td><td>confRev</td><td>Tag</td><td>Length</td><td>ndsCom</td></tr>
<tr><td>Tag</td><td>Length</td><td>numDatSetEntries</td><td>Tag</td><td>Length</td><td>allData</td></tr>
<tr><td>Tag</td><td>Length</td><td>Data 1 (Boolean)</td><td>Tag</td><td>Length</td><td>Data 2 (Float)</td></tr>
<tr><td colspan="3">●●●●●● ●●●●●●</td><td>Tag</td><td>Length</td><td>Data N</td></tr>
</table>

Figure 3.1: Structure of a Generic Object Oriented Substation Event (GOOSE) Datagram.



```
01 0c cd 01 00 00 [redacted]1 88 b8 00 01
00 9b 00 00 00 00 61 81 90 80 23 41 41 31 4e 31
51 30 31 41 34 4c 44 30 2f 4c 4c 4e 30 24 47 4f
24 67 63 62 49 45 44 34 5f 47 6f 6f 73 65 81 02
4e 20 82 1d 41 41 31 4e 31 51 30 31 41 34 4c 44
30 2f 4c 4c 4e 30 24 49 45 44 34 5f 47 6f 6f 73
65 83 20 41 41 31 4e 31 51 30 31 41 34 4c 44 30
2f 4c 4c 4e 30 2e 67 63 62 49 45 44 34 5f 47 6f
6f 73 65 84 08 57 64 5b 80 e3 5d df 3f 85 01 01
86 03 00 c9 06 87 01 00 88 01 64 89 01 00 8a 01
02 ab 06 83 01 00 83 01 00
```

Figure 3.2: Hexadecimal Representation of a GOOSE Datagram.

The GOOSE datagram starts with the Destination MAC Address, which is a multicast address reserved for IEC 61850 applications always starting with 01-0c-cd, and is followed by a six-octet source MAC address. This is the MAC address of the publishing IED. A

GOOSE message has an IEEE 802.1Q Virtual Local Area Network ID (VLAN ID) and a unique Ethernet type (88-b8). The Application ID (APPID) field is a four-octet field, which the subscribing IEDs use to identify messages to which they are subscribing. The Length field represents the length of the overall GOOSE datagram minus eight bytes, and is followed by two reserved fields left out by the standard for future use. The goosePDU field itself is composed of twelve subfields that follow the modified ASN.1 BER encoding scheme as well [8]. The goosePDU consists of the following:

- gocbRef: GOOSE control block reference
- timeAllowedtoLive: the time a receiver waits before receiving a re-transmitted message
- datSet: name of the Data Set
- goID: ID of publishing IED
- t: time stamp indicating a new GOOSE event
- stNum: counter that increment with every GOOSE event
- sqNum: counter that increment with every repeated GOOSE message
- test: specifies if a message is/is not intended for testing
- confRev: number of times Data Set has changed
- ndsCom: needs commissioning field
- numDataEntries: number of data elements in allData
- allData: actual data being sent (bool, integer, float, …)

3.2.2 IEC Standard Guidelines for Processing GOOSE Messages

IEC 61850-8-1 defines the structure of a GOOSE message and the means by which it is communicated over a network. Despite its criticality, IEC 61850 advanced in an era where substations operated in isolated proprietary networks and thus did not include any cyber security measures for data communication. However, this will no longer be the case, as operators are moving towards open networks and remote access of substation control systems through the aid of contemporary communication technologies, such as cloud services. For instance, authors in [94] investigated the use of IEC 61850 for tele-protection outside the boundaries of a single substation over WAN. Also, in an effort called Cloud IEC 61850, authors in [95] investigated the idea of having virtualization and cloud technologies as the underlying infrastructure of electrical automation systems with a specific example of a substation automatic voltage control. Recent literature also shows the application of IEC 61850 in hierarchical microgrid control, where the authors in [96] propose a comprehensive hybrid agent framework combining the Foundation for Intelligent Physical Agents (FIPA), IEC 61850, and DDS standards. With the realization of this modern communication infrastructure, IEC 62351 emerged in order to tackle the shortcomings of IEC 61850 in terms of communication security. IEC 62351 was developed by IEC TC57 WG15 and consists of eleven parts to cover end-to-end security issues in power system communications [9]. IEC 62351-6 covers communication security within the boundaries of a substation covering MMS, GOOSE and SMV protocols.

IEC 62351-6 devises an algorithm for proper processing of GOOSE messages in order

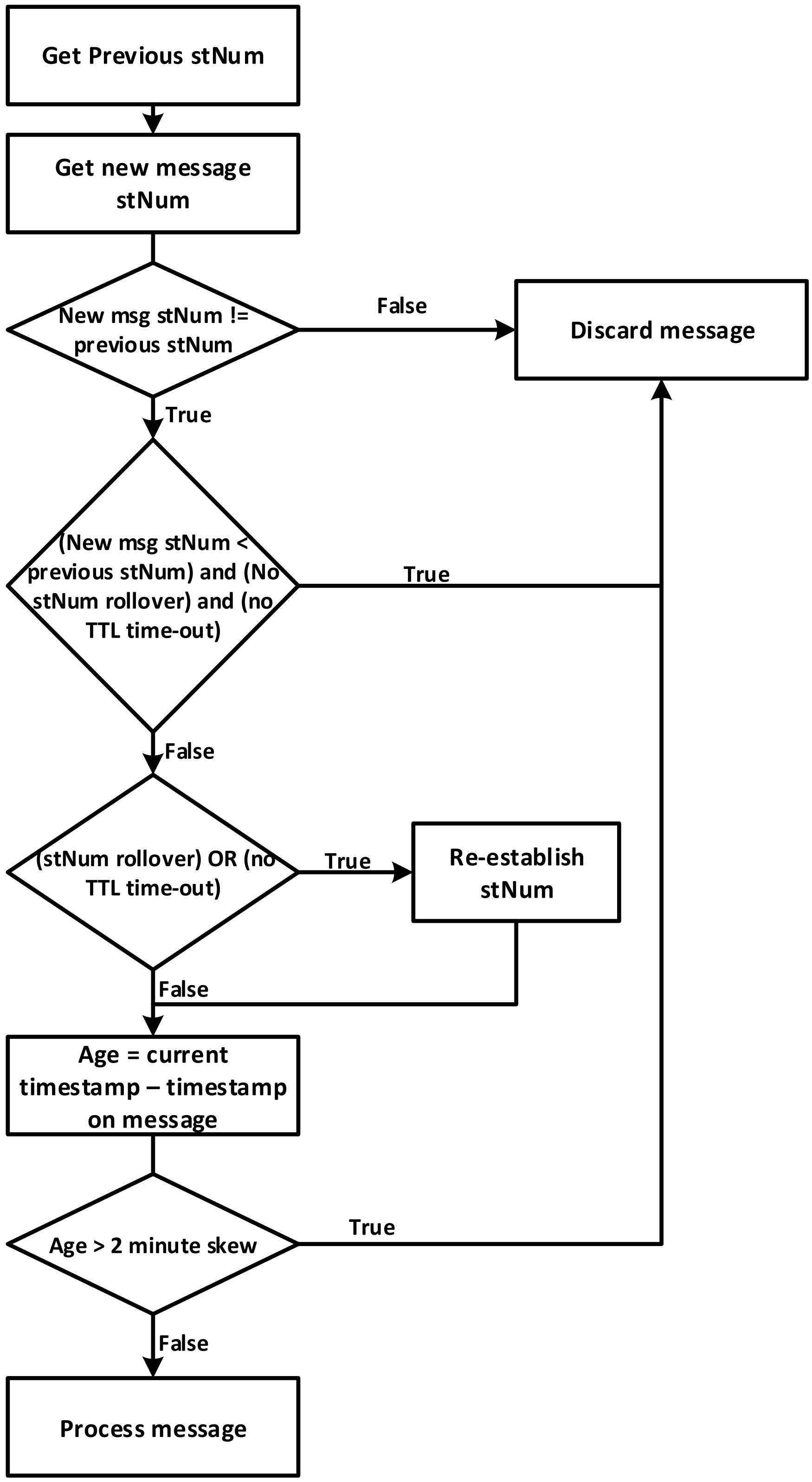

Get Previous stNum
Get new message stNum
New msg stNum != previous stNum
False
Discard message
True
(New msg stNum < previous stNum) and (No stNum rollover) and (no TTL time-out)
True
False
(stNum rollover) OR (no TTL time-out)
True
Re-establish stNum
False
Age = current timestamp – timestamp on message
Age > 2 minute skew
True
False
Process message

Figure 3.3: GOOSE Messages Processing Algorithm.

to mitigate some cyber-attacks, such as replay and man-in-the-middle attacks. From the publishing IED side, each GOOSE message has a status and sequence number fields (stNum and sqNum, respectively). When a substation event occurs, for example, an overcurrent is sensed, the publishing IED instantly transmits a message with an incremented stNum field. The message is then repeated with variable increasing time delay until the maximum defined period is reached. The sqNum counter increments with every repeated message until the maximum number count ($2^{32}-1$) is reached, the point at which the sqNum counter rolls over. IEC 62351-6 states that a subscriber IED that detects a new message with a new stNum must discard any message having an stNum less than or equal to the previous message and which time allowed to live has not timed-out yet, unless a rollover of the stNum counter occurs. If none of the above conditions are true, the subscribing IEDs process the messages. A flowchart describing the algorithm for processing GOOSE messages set by IEC 62351-6 is presented in Figure 3.3.

3.3 Testing of Commercial IEDs Communicating with IEC 61850 GOOSE Messaging Protocol

A case study of the implementation of IEC 61850 GOOSE messaging on commercial IEDs present at the Smart Grid test bed at Florida International University and the open source libiec61850 [75] library, which is also implemented on commercial devices, was performed.

Figure 3.4 shows the experimental setup with the commercial IEDs having the vendor’s proprietary implementation of IEC 61850. Manufacturer details of the commercial IEDs under test are intentionally oitted. Under normal conditions, the publishing IED is programmed to broadcast a GOOSE message with two Boolean data fields set to False (00-00). The subscribing IEDs read the Boolean data and control the status of the circuit breaker accordingly. In this case, the data read (False) maintains the relay's un-tripped status and the circuit breaker's closed status.

Similarly, Figure 3.5 shows the experimental setup in which the libiec61850 GOOSE open source library has been implemented on two embedded boards. The publishing IED has the *goose_publisher* routine implemented on it, whereas the receiving IED has the *goose_subscriber* routine. More details about the implemented routines can be found on the open source library's website [75]. It is worth mentioning that this library has also been implemented on other commercial devices. The device on the left is the publishing IED, whereas the device on the right is the subscribing IED. The publishing IED also transmits Boolean data (either True or False), which the subscribing IED reads and triggers a digital output accordingly, as marked in red in Figure 3.5. The subscribing boards were designed with connection capabilities to the solid-state circuit breakers shown in Figure 3.4.

In order to perform the tests, a Python script was written in conjunction with network traffic capturing and packet crafting libraries from Scapy [97][98]. The developed script takes advantage of the simplicity of the unencrypted GOOSE message structure defined in Section II in order to monitor the Local Area Network (LAN) and capture Ethertype (88-b8) GOOSE messages. Each field in the captured GOOSE messages was properly decoded based on the IEC 61850-8-1 modified ASN.1 BER mechanism. The script then modifies the content of the messages, encodes all the fields, crafts the new fake packet, and broadcasts it over the LAN. For each test, a certain field in the GOOSE message was modif-

-ied and the results of the various tests are discussed below. A general overview of

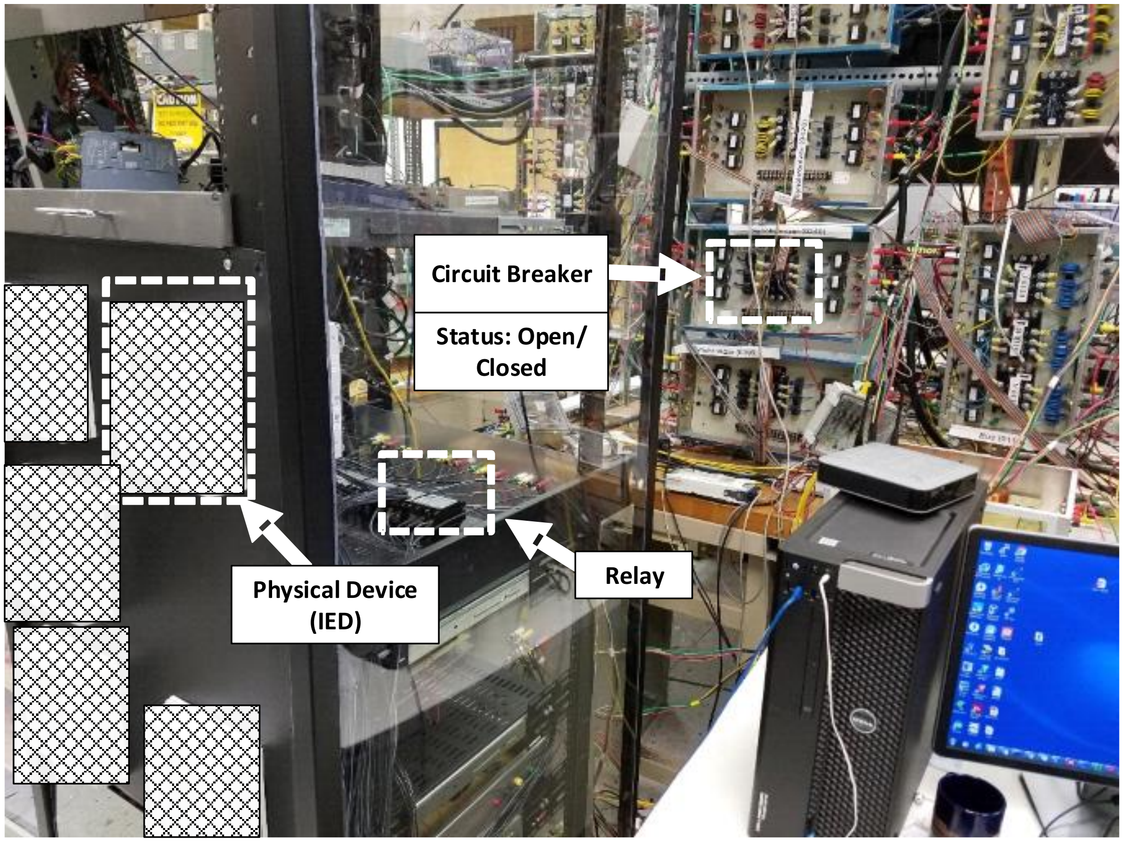


Figure 3.4: Experimental Setup with Commercial Intelligent Electronic Devices (IEDs).

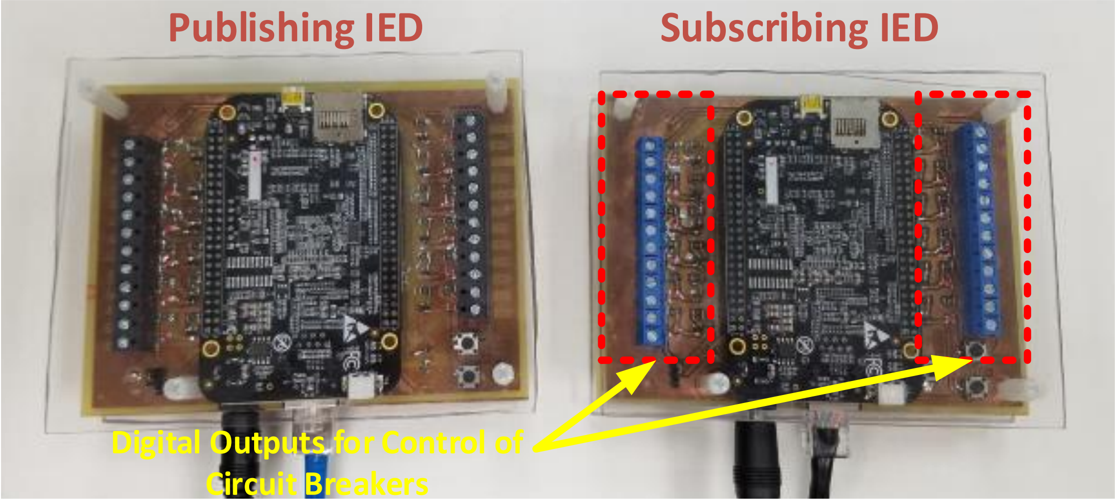


Figure 3.5: Experimental Setup with libiec61850 Implemented on Embedded Boards.

the data manipulation procedure performed is shown in Figure 3.6. Figure 3.7 shows a screen shot of the captured messages by the developed script and the corresponding decoded fields. The script was run on a Virtual Machine operating on Ubuntu Version 16.04.

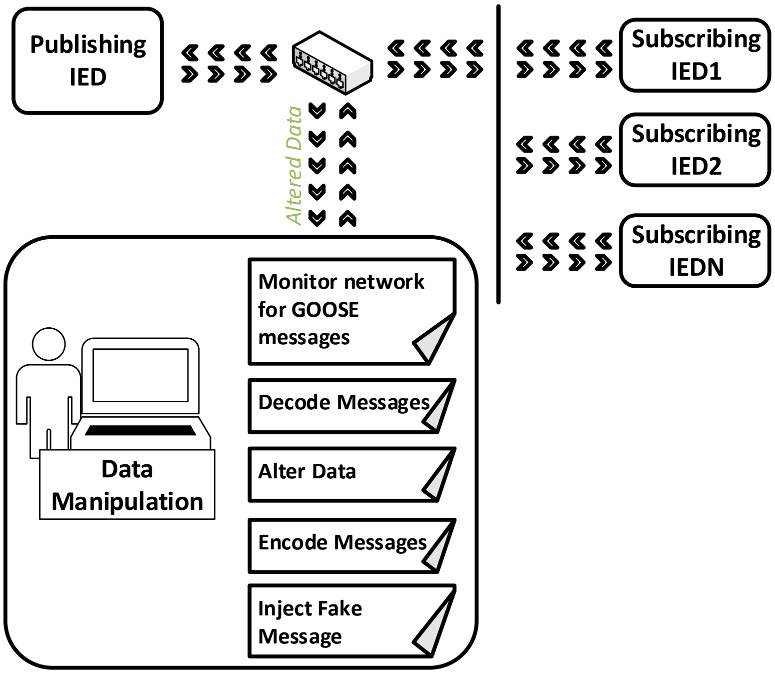
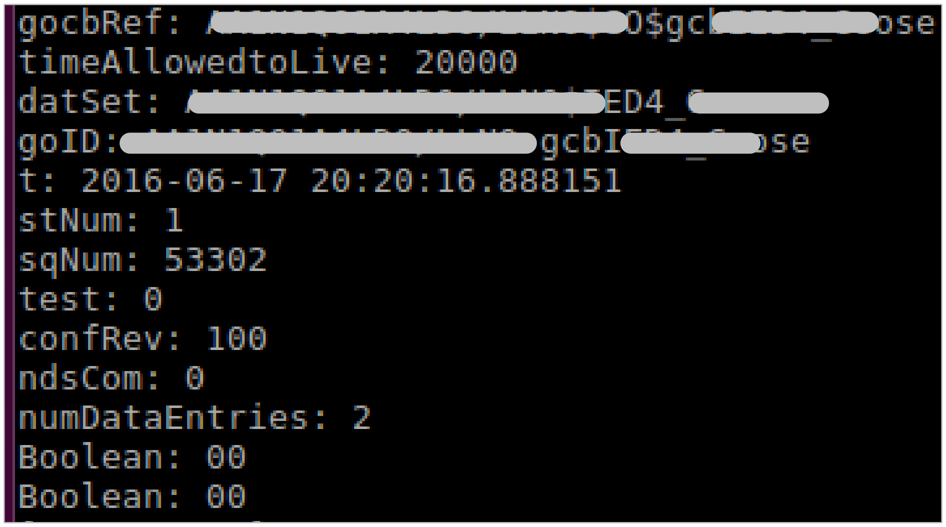


Figure 3.6: GOOSE Messages Data Manipulation Overview.



```
gocbRef: [redacted]O$gc[redacted]ose
timeAllowedtoLive: 20000
datSet: [redacted]ED4_[redacted]
goID:[redacted]gcbI[redacted]ose
t: 2016-06-17 20:20:16.888151
stNum: 1
sqNum: 53302
test: 0
confRev: 100
ndsCom: 0
numDataEntries: 2
Boolean: 00
Boolean: 00
```

Figure 3.7: Captured Packet (manufacturer’s details are intentionally omitted).

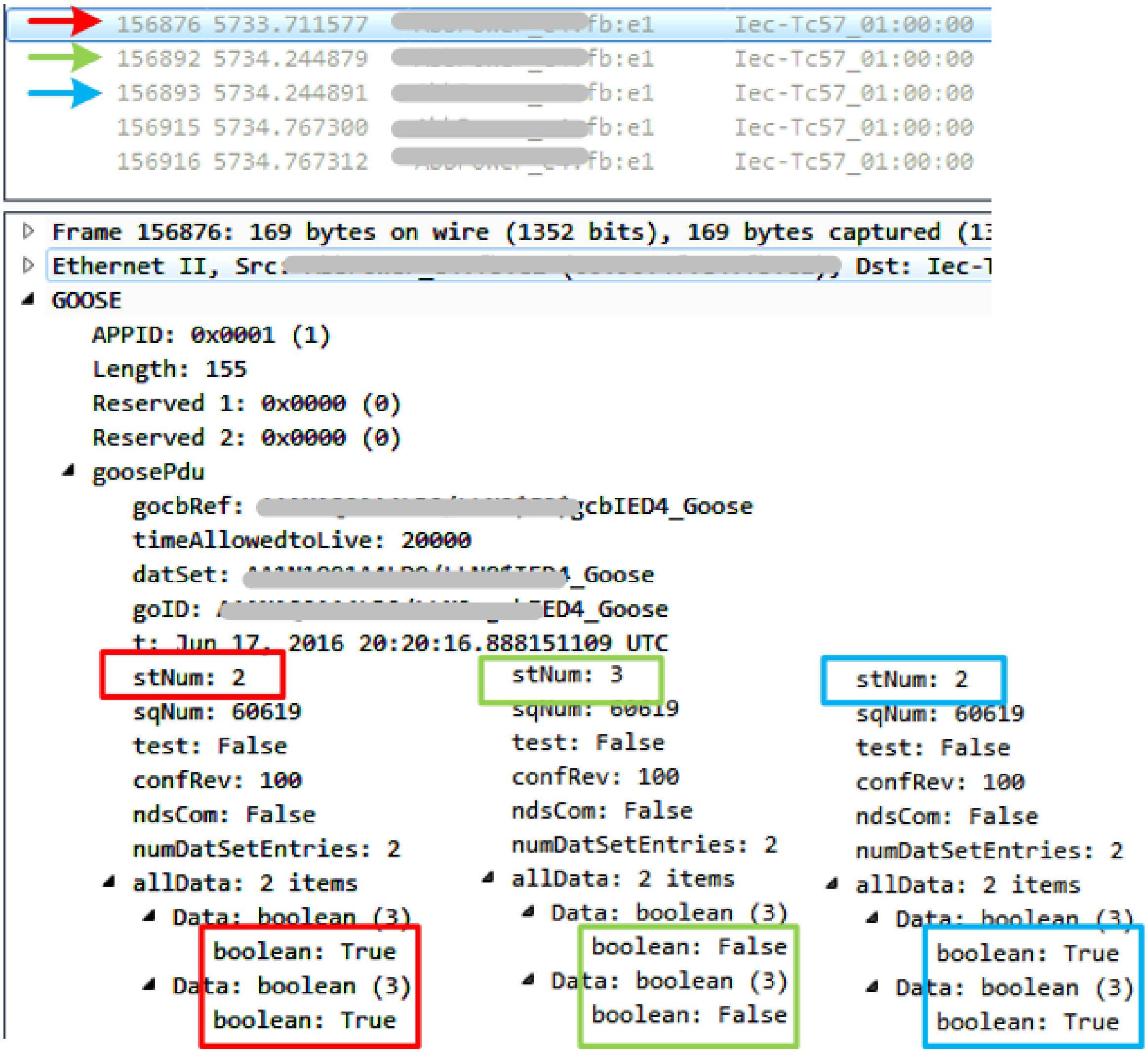
In the experimental setup for both the commercial IEDs and the developed embedded devices running the open source libiec61850 library, the publishing IED transmits messages with a status number stNum = 1, False Boolean Data fields (00-00), and incrementing sequence numbers.

3.3.1 Processing of Status Number

As explained in the flowchart of Figure 3.3, any GOOSE message with a status number different than that of its predecessor shall be discarded if it has an stNum equals to or less than that of the previous message and is still within its valid time allowed to live.

3.3.1.1 Commercial IEDs

In this test, first a message with stNum = 2 (>1) and True (01-01) Boolean data fields was sent. As anticipated from the standard, this message was processed and the circuit breakers status changed from closed to open. Next, a message with stNum = 3 (>2) and False Boolean fields (00-00) was transmitted and was also processed. Finally, another fake message with stNum = 2 (<3) with True Boolean fields (01-01) was broadcasted. Although this final message had a lower stNum than its predecessor, it was processed and the status of the circuit breaker changed from closed to open. The GOOSE datagrams of the broadcasted messages are shown in Figure 3.8. It should be noted here that all messages had the same time stamp, which was three days old. When compared with the subscribing IED time stamp, it was noticed that the 2-min time skew mentioned in Figure 3.3 was exceeded; however, the messages were still processed.



```
156876 5733.711577 [redacted]fb:e1   Iec-Tc57_01:00:00
156892 5734.244879 [redacted]fb:e1   Iec-Tc57_01:00:00
156893 5734.244891 [redacted]fb:e1   Iec-Tc57_01:00:00
156915 5734.767300 [redacted]fb:e1   Iec-Tc57_01:00:00
156916 5734.767312 [redacted]fb:e1   Iec-Tc57_01:00:00

▷ Frame 156876: 169 bytes on wire (1352 bits), 169 bytes captured (1
▷ Ethernet II, Src: [redacted], Dst: Iec-T
◢ GOOSE
    APPID: 0x0001 (1)
    Length: 155
    Reserved 1: 0x0000 (0)
    Reserved 2: 0x0000 (0)
  ◢ goosePdu
      gocbRef: [redacted]gcbIED4_Goose
      timeAllowedtoLive: 20000
      datSet: [redacted]_Goose
      goID: [redacted]ED4_Goose
      t: Jun 17, 2016 20:20:16.888151109 UTC
      stNum: 2                       stNum: 3                       stNum: 2
      sqNum: 60619                   sqNum: 60619                   sqNum: 60619
      test: False                    test: False                    test: False
      confRev: 100                   confRev: 100                   confRev: 100
      ndsCom: False                  ndsCom: False                  ndsCom: False
      numDatSetEntries: 2            numDatSetEntries: 2            numDatSetEntries: 2
    ◢ allData: 2 items             ◢ allData: 2 items             ◢ allData: 2 items
      ◢ Data: boolean (3)            ◢ Data: boolean (3)            ◢ Data: boolean (3)
          boolean: True                  boolean: False                 boolean: True
      ◢ Data: boolean (3)            ◢ Data: boolean (3)            ◢ Data: boolean (3)
          boolean: True                  boolean: False                 boolean: True
```

Figure 3.8: Wireshark Capture of Transmitted Messages (manufacturer's details are intentionally omitted), red: original message with low status number, green: fake message with high status number, blue: original message retransmission with low status number.

3.3.1.2 Libiec61850

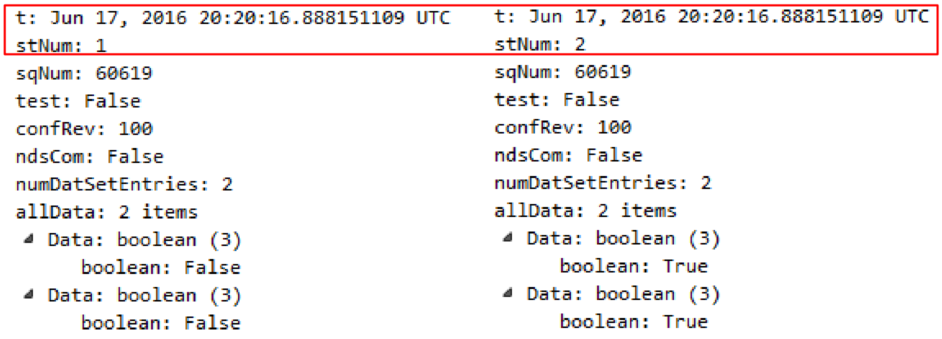
The same test was repeated on the open source libiec61850 library implemented on the two developed embedded devices. Here, the final message with a low stNum (2 < 3) was not processed. This is because libiec61850 has an IsValid() function which checks if the TimeAllowedToLive timeout is not elapsed and if GOOSE messages were received with correct state and sequence IDs [75].

3.3.2 Message Time Stamp

Each GOOSE message has a time stamp field, which is updated with each increment of the status number (i.e. with each substation event). Therefore, subscribing IEDs receiving a new message with changed data fields and an incremented stNum field must expect to have a message with an updated time stamp.

3.3.2.1 Commercial IEDs

In this test, we sent a fake GOOSE message with an incremented status number (stNum = 2) and altered data True (01-01), but with an old time stamp (three days old). As shown in Figure 3.9, the device processed the message and the status of the circuit breaker changed from closed to open.



```
t: Jun 17, 2016 20:20:16.888151109 UTC    t: Jun 17, 2016 20:20:16.888151109 UTC
stNum: 1                                  stNum: 2
sqNum: 60619                              sqNum: 60619
test: False                               test: False
confRev: 100                              confRev: 100
ndsCom: False                             ndsCom: False
numDatSetEntries: 2                       numDatSetEntries: 2
allData: 2 items                          allData: 2 items
 ◢ Data: boolean (3)                       ◢ Data: boolean (3)
      boolean: False                            boolean: True
 ◢ Data: boolean (3)                       ◢ Data: boolean (3)
      boolean: False                            boolean: True
```

Figure 3.9: Wireshark Capture of Transmitted Messages with Same Time Stamp (t) and Incremented status number (stNum).

3.3.2.2 LibIEC61850

The test was repeated on the open source libiec61850 library implemented on the two developed embedded devices. The subscribing IEDs processed the messages even though they have the same time stamps and incremented status numbers. The red LED in Figure 3.10 indicates that the message was processed and a digital output (HIGH) was produced, signaling a circuit breaker trip.

It is noteworthy to point out that, according to the flowchart of Figure 3.3, IEC 61850 recommends checking for a message's time stamp only if it recognizes an stNum different than that of the previous message. The experiments revealed that when sending new messages with three-day-old time stamps exceeding the 2-min skew, they were processed as long as they had status numbers equal to or higher than the previous message.

3.3.3 Processing of Source MAC Address

All fake messages in the three tests performed above were sent from the virtual machine with a spoofed MAC address mimicking that of the publisher IED. That is, all IEDs subscribing to this message process the fake messages as if they were originating from the publisher IED. One common network defense procedure to counter MAC address spoofing incidents is to apply a MAC address filter to network switches. This will deny any machine connected to the network from sending a message with a source MAC address other than its own. After applying this filter, the messages with the fake MAC address were blocked from being sent over the network. However, in this test, we sent fake GOOSE messages with the MAC address of the virtual machine and altered data, and noticed that the subscriber IEDs processed these messages. The circuit breaker's status changed from closed to open.

This test actually exploits a vulnerability in the GOOSE messaging protocol itself rather than its implementation in commercial devices. In the GOOSE protocol, subscribing IEDs use the APPID field to subscribe to desired GOOSE messages. Since the subscribing IEDs in this case do not check for the source MAC address, they will process any message with their defined APPID, regardless of its origin.

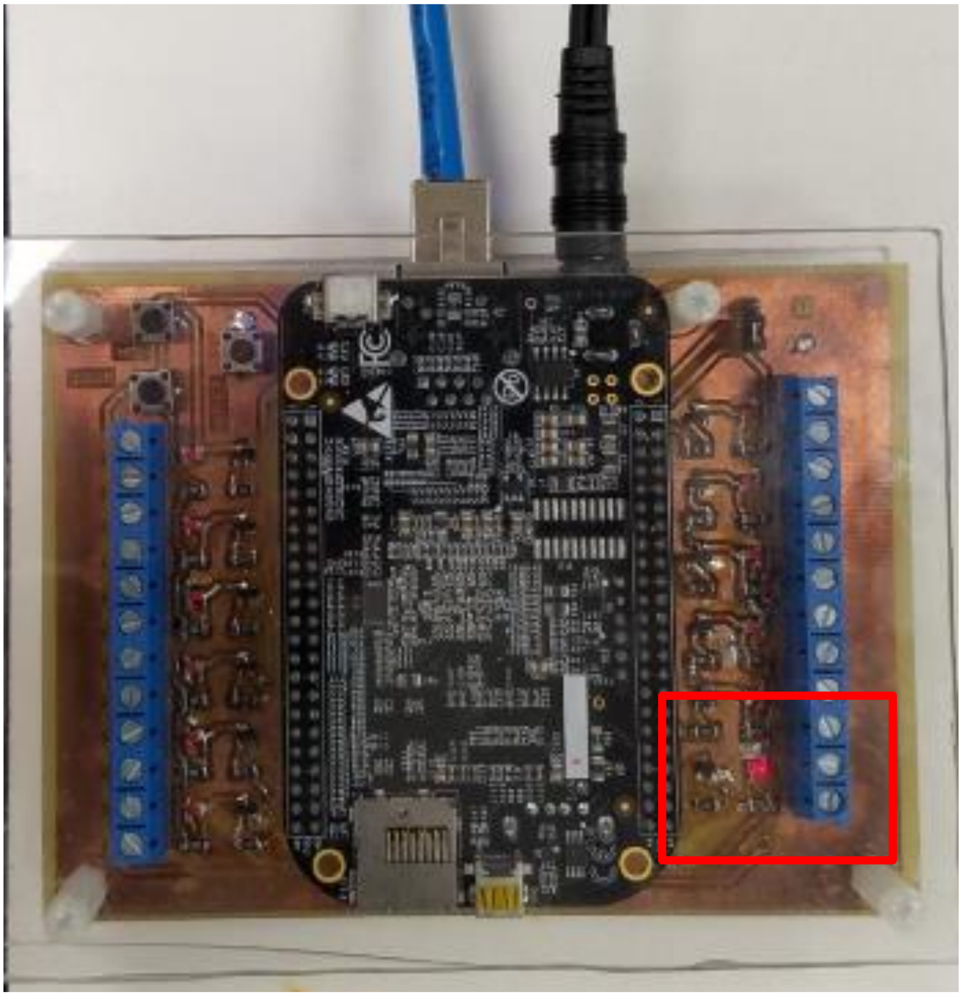


Figure 3.10: Subscribing IED Processing Fake Message (output port triggered as indicated in the red box).

3.4 Guidelines for Proper Implementation of IEC 61850 GOOSE Protocol

Table 3.1 summarizes the results of the performed tests on both the commercial IEDs and libiec61850.

It can be concluded from the results of the performed experiments that the actual implementation of IEC 61850 and its associated IEC 62351 cyber security standard on field devices depends on the vendors themselves. While vendors try to fully abide by the standard, differences in the implementation process might still be found as shown in this paper. The presence of such differences in the implementation process might expose the system to unwanted vulnerabilities, which might be exploited by prying eyes to launch cyber-attacks on the power grid [10][86]. As GOOSE messaging is the base protocol for critical applications, such as power system protection, any vulnerability in the system might lead to devastating consequences, ranging from system disturbances to complete blackouts.

Recent literature shows several security concerns about the IEC 61850 standard itself [42]. Therefore, in order to avoid additional exploits, extreme care must be placed on implementing IEC 61850 functionalities on commercial devices, as well as abiding by the cyber security requirements set by IEC 62351. The analysis of the outcome of this work distinguishes between two levels of vulnerabilities: one on the device level and the other on the network level. On the device level, when devices are configured to communicate via GOOSE messages, the firmware on the subscribing IEDs must be tested for proper processing of messages as stated by the IEC 61850-8-1 and IEC 62351-6 standards. Since

Table 3.1. Compliance Test Results.

Test	**Commercial IED’s**	**Libiec61850**	**Standard Practice**
Processing of messages with lower stNum	Y	N	Discard message
Processing of messages with outdated time stamp	Y	Y	Discard message
Processing of messages with unspoofed MAC address	Y	Y	Not specified by standard

Y: Message processed; N: Message not processed.

IEC 61850 does not provide any cyber security measure by itself, manufacturers should also make sure that their devices comply with IEC 62351 requirements. First, as stated by IEC 62351-6, messages with repeated or old status numbers must not be processed by subscribing devices. In fact, the open source libIEC61850 has an *IsValid()* function to ensure this, whereas the tested commercial IEDs lack this important check and thus processed fake messages. In addition, the association of a new time stamp with every increment of the status number must be checked for before publishing and/or processing messages. In the case of a new GOOSE event (i.e. an incremented stNum), it is important to compare a newly received message's time stamp with the subscribing machines time to check whether or not the 2-min skew set by IEC 62351 was exceeded. Also, every change in the Data fields of a GOOSE message must be checked for association with an increment in the status number field. Finally, repeated messages with a change in their control signal (i.e. data fields) must be rejected. Message retransmissions should be identical, with no alterations in any field except for an incrementing sequence number.

On the network level, in order to avoid compromised machines from publishing fake GOOSE messages using a spoofed MAC address, MAC filters must be applied to all switches in the substation's local area network to prevent MAC address spoofing. As concluded from the presented case studies, MAC filtering did not prevent subscribing IEDs from processing fake messages with unspoofed MAC addresses. Therefore, the source MAC address field in GOOSE messages must be checked to be belonging to an authenticated machine authorized to communicate via the GOOSE protocol within a substation's local area network. In fact, this vulnerability has not been accounted for, neither in IEC 61850 nor in IEC 62351. Until the standards cover this issue, it is up to the substation's network administrators to make sure that only authenticated devices can communicate via GOOSE messages.

3.5 Conclusions

Testing of two different available implementations of the IEC 61850 GOOSE messaging protocol was performed on commercial IEC 61850-based devices and on the open source libiec61850 library. The results demonstrated that different implementations of the same standard might lead to different behaviors even if the devices were present under similar conditions. Deviation from the actual procedures set forth by the IEC 61850 standard and its complementary cyber security IEC 62351 standard were found in the responses of the devices. From the experiments in this chapter, it was found that the processing of the GOOSE messages status number was not properly implemented on the commercial devices as recommended by IEC 62351. This vulnerability provides a strong attack surface for prying eyes to inject malicious activities in power systems, such as the data manipulation attack demonstrated in this work. Additionally, all the tested devices were processing messages with old time stamps, which is another attack surface for launching replay attacks. This point is of importance since GOOSE messages are broadcast in nature and, therefore, sniffing and replaying them is possible when an attacker is in the same LAN. Moving to the network level, it was shown that as long as it has a valid APPID field, a GOOSE message is processed whether originating from an authentic device or a malicious one. Since both IEC 61850 and 62351 do not clearly outline how to present clear rules for authenticating source MAC addresses, it is up to the substation network designers to take this issue into consideration and apply the appropriate defense mechanisms. Thus, this work raises a serious issue as such devices are out in the field and are controlling critical and potentially dangerous power system operations. The work in this chapter also proposes guidelines to better enhance utilization of IEC 61850. Proper processing of a message's source MAC address, better utilization of the time stamp field to check for messages' validity, and the association of new message content with a status number increment are advised.

Finally, the vulnerabilities presented in this chapter present a strong attack surface for attackers to perform GOOSE poisoning that will be successful in controlling the switching activity of circuit breakers in power systems. Realizing the criticality of that, the next chapter will present a physical-model-checking security algorithm to detect such types of switching-related attacks on power systems.

Chapter 4 Physical-Model-Checking to Detect Switching Related Attacks on Power Systems

Recent public disclosures on attacks targeting the power industry showed that savvy attackers are now capable of obfuscating themselves from conventional rule-based network IDS, bringing about serious threats. In order to leverage the work of rule-based IDS, this chapter presents an artificially intelligent physical-model-checking intrusion detection framework capable of detecting tampered-with control commands, from control centers of power grids. The work in this chapter utilizes AI to learn the load flow characteristics of the power system and benefits from the fast responses of the AI to decode and understand contents of network packets. The output of the AI is processed through an expert system to verify that incoming control commands do not violate the physical system operational constraints and do not put the power system in an insecure state. We tested proposed content-aware IDS in simulation on a 14-bus IEEE benchmark system. Also, we carried out experimental verification on a small power system, with an IEC 61850 network architecture. The results showed the accuracy of the proposed framework in successfully detecting malicious and/or erroneous control commands.

4.1 Introduction

Resilient and secure operation of the power grid relies on judicious cooperation between several cyber and physical entities. Cyber processes, for instance, read the physical states of the grid and interact with it by actuating physical devices. Therefore, communication signals could be feedback from sensory devices or control commands to actuating devices.

On several occasions, the literature showed the ability of attackers to exploit vulnerabilities in the communication networks of electricity grids and tamper with control fields in network packets. For instance, [30] presented how a switching control command could be manipulated by an attacker to maliciously open circuit breakers causing blackouts. The work in [99] showed how the power could be interrupted by tampering with sensor measurements. Reference [100] showed that blackouts could be caused due to the sequential removal of substations or transmission lines by malicious acts. Similarly, [101] discussed attack scenarios capable of causing cascaded failures in power systems.

Furthermore, recent public disclosures emphasized the brutality of control-related attacks on critical processes, such as the Stuxnet and the Crash Override malwares targeting industrial control systems and power plants [102][103]. In the Stuxnet malware, attackers targeted PLCs by changing the control signals going to Variable Frequency Drives (VFDs) of motors. In a similar approach, the Crash Override malware targeted intelligent electronic devices by altering the switching commands sent to open and close circuit breakers.

Notwithstanding the fact that the aforementioned attacks targeted critical infrastructure, the gravity of these attacks is also accentuated by their ability to obscure themselves from conventional rule-based IDSs. In such attacks, the modified control fields are re-encoded in the proper packet format before being transmitted on the network [30][41]. Rule based IDS rely on information in the header of network packets and compare them against standard stipulations or perform statistical analysis on network traffic to identify anomalous ones based on cyber rules. By that, conventional IDS disregard the actual data fields. Accordingly, there is a need for new innovative solutions that detect attacks that might disrupt of the operation of the power grid.

Attacks on power control loops can be categorized as:

1. False Data Injection Attacks (FDIAs) that target sensor or meter measurements. In these types of attacks, attackers attempt to feed back to the controller fake sensor measurements to alter its operation. For instance, a malware in [104] showed that injecting high current values into a substation's network could cause controllers to issue unwanted trip signals, jeopardizing the reliability of the power system. There are extensive efforts on detecting and mitigating FDIAs. For instance, the authors in [105] presented an algorithm based on the linear Weighted Least Square Error (WLS) to detect bad or corrupted sensor measurements in digital substations. However, there are practical limitations on the WLS method, such as the latency. In [28], a false data injection attack detection mechanism, which is based on identifying a set of candidate invariant microgrid parameters, was introduced. This method was designed specifically for DC microgrids. In [99], the authors focused on detecting fake sensor measurements in power systems and enhancing the reliability of power grid by forecasting the values of lost measurements, due to network congestion, based on historical trends. Similarly, there are plenty of other works that are focused on FDIAs, such as [106][107][108].

2. Control-related attacks that target control commands going to actuators and field devices. Several recent works that have been placed to detect control-related attacks in the energy sector that incorporates physical rules along with cyber rules. In [30], a semantic analysis framework, which integrates network IDS with power flow analysis was proposed to detect malicious control commands. To achieve acceptable detection latency, this technique requires adapting the power flow analysis algorithm, leading to a tradeoff between accuracy and latency, as the system expands. In [31], an anomaly detection algorithm, which is specific for detecting attacks on automatic generation control, is proposed. In the former, the control signal is executed on the physical system only if it is regarded as legitimate by the anomaly detection engine, otherwise, a signal from a model-based automatic generation control is utilized. This work relies on the assumption that the feed-back frequency and tie-line measurements are trusted and do not discuss their security requirements. In [24], faults are distinguished from cyber-attacks by following a mathematical formulation that incorporates PMU data, event status, and monitoring logs. Similarly, the work in [109] utilizes lookup tables for current measurements and circuit breakers statuses to compare current and previous states for attack detection. Both [24][109] require that data collection for the detection algorithms to be performed by a trusted entity, which is not always the case [110].

Therefore, there is a need for security systems that not only are capable of detecting anomalous network activity based on cyber rules, but also are aware of the content of network packets to be able to understand and assess their consequences on the physical grid. Since the goal of most attackers is to disrupt the operation of the power system, attackers have more incentive to directly alter control commands, rather than tamper with sensor measurements to affect the controllers' actions. Accordingly, the focus of this chapter will be on the detection of switching attacks on circuit breakers, which falls under the category of control related attacks.

This chapter proposes a multi-agent security framework to detect and prevent cyber-attacks targeting circuit breakers in a power system. Unlike the work presented in the literature, the work in this paper utilizes AI to learn the load flow characteristics of the power system and benefits from the fast responses of the AI to decode and understand the contents of network packets. The output of the AI is processed through an expert system to decide on whether an incoming control command contains malicious content or not.

The contributions of the chapter are as follows:

While the work in the literature assesses network packets against mathematical models of the power system, to the best of the authors' knowledge, this is the first effort to discuss the use of machine intelligence to develop a content-aware intrusion detection and prevention system that decodes and understands the physical meaning of the content of network packets.

The use of AI reduces the online computational burden as compared to complex mathematical models and therefore, accelerates the attack detection and decision-making processes (in the range of microseconds).

Taking appropriate preventive action upon detecting malicious control commands and not only detecting intrusions.

Finally, implementing the developed security multi-agent system on a hardware laboratory scale power system with an IEC 61850 communication architecture, taking into consideration the practical aspects that arise from the hardware implementation of the power system, agents as embedded microcontrollers, and communication network. The obtained results from the experimental setup proved the feasibility of the proposed security algorithm in real-time physical power systems.

The performance of the proposed framework was tested in simulation on a 14-bus IEEE benchmark system for 36 test cases covering all N-1 contingency scenarios. The results showed that all the malicious commands that will place the system in an insecure state, have been prevented from actuating the circuit breakers. The framework was also verified experimentally, where the security algorithm was compiled on embedded microntrollers that are interfaced with a 5-bus hardware power system testbed setup having an IEC 61850 communication infrastructure.

4.2 The Artificially Intelligent Physical-Model Checking Approach

The proposed multi-agent security framework is shown in Figure 4.1. In this work, the power system is sectionalized into several zones. In each zone, an agent is responsible for: (1) local control and (2) security actions.

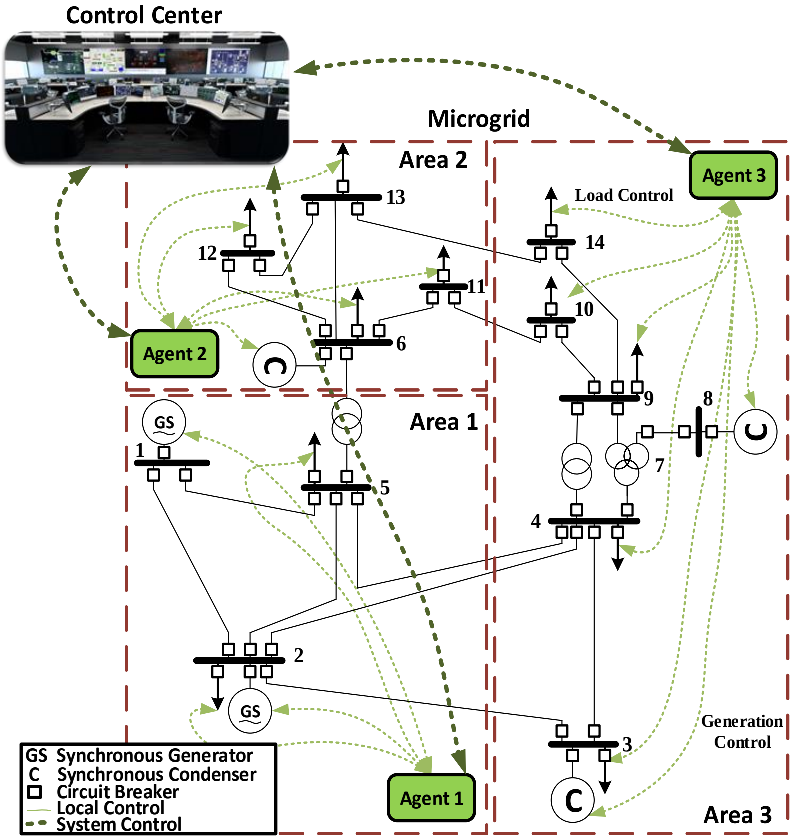


Figure 4.1: 14-Bus IEEE benchmark system with decentralized and hierarchical control.

In terms of local control, each agent is interfaced with sensors, actuators, and field devices within its zone. Based on the feedback that it gets from the sensors, agents control the actuators and the field devices. The agents communicate among each other to achieve the global objective of stable and reliable operation of the power system. The agents also act as mediators between control centers and field devices/processes. This will allow control centers to poll information/data from sensors to monitor the entire system and to send control actions to actuators and field devices. Such a control architecture is referred to as decentralized control. The standard industrial communication protocols that allow the adoption of the decentralized control are discussed in Section 4.2.1.

Unlike centralized control, the decentralized multi-agent framework in this chapter contributes to the security of the power system by avoiding single points of failure. In centralized control, all the sensors and field devices communicate with a single server, which from a security view-point is a bottle-neck and single point of failure. However, in decentralized control, even if one of the agents failed, the system will not entirely collapse. In addition to that, by processing data locally and performing local control actions, the amount of data to be transferred to the control center and the communication bandwidth will be reduced. The required processing power on control centers will be less and the system reliability will be improved.

In terms of the security actions, the goal of the proposed algorithm is to add an additional security layer to the operation of the power system by detecting malicious actions, which might occur on switching commands traveling the communication network. Switching commands that come from the system operator can be tampered with. The agents in this work are capable of assessing the physical consequences of these switching commands before they are executed, to ensure reliable operation of the power system and that the voltage levels and line loadings do not violate the safe operational limits set by standards [74][111]. The agents' ability to assess the physical consequences is coming from intensive training of its neural network, by performing all the possible N-1 contingency analysis and all possible switching attack combinations on the loads and generators, and feeding all this data to the neural network. The proposed security algorithm is discussed in detail in Section 5.2.2.

4.2.1 Current Standards and Associated Threats

The two most used protocols for microgrid operation and control in the power industry are the DNP3.0 for SCADA systems and IEC 61850 MMS, GOOSE, and SMV messages in more recent systems [30][109]. Although these protocols enabled decentralized, robust, and more accurate microgrid control, they also introduced some vulnerabilities in terms of cyber security. Each of the aforementioned protocol suites has its own vulnerabilities that were previously exploited to launch successful attacks on power grids. For instance, [30] presented a successful data manipulation attack on a DNP3 packet, which has 4 control relay objects to operate 4 circuit breakers in a substation. A man-in-the-middle attack was also presented in [41] to generate malicious circuit breaker control commands as GOOSE messages. As mentioned previously, these attacks remained obscure form the network IDS, since the attackers established fake data, as legitimate network packets, but with malicious content.

A major facilitator of such attacks is that power system communication networks need to accommodate the real-time operation of the grid. Therefore, strict time-delay requirements are imposed on the exchange of communication signals. Since current microcontrollers and Intelligent Electronic Devices (IEDs) have low processing power, such industrial control networks are left unencrypted, and sometimes, without authentication. In fact, a study conducted in [9] shows that even the latest processor technologies cannot meet the 4 ms end-to-end time delay requirement set forth by the IEC 61850 standard stipulations on GOOSE messages.

Since this work targets the detection of control-related attacks, the publisher/subscriber GOOSE messaging protocol is selected for controlling the statuses of circuit breakers in the studied system. As will be shown later in the chapter, the latency of the proposed detection algorithm falls within the 4 ms time delay set for GOOSE messaging.

Attack Model:

In order to understand the threat models assumed in this chapter for DNP3.0 and GOOSE switching commands, we differentiate between two types of switching commands, as in [30]:

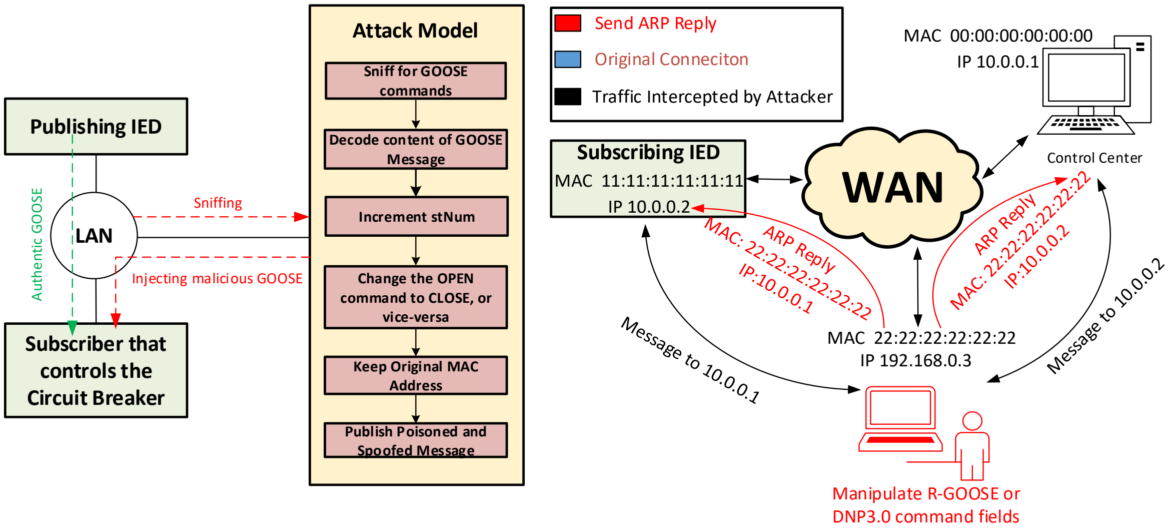
1. Automatic Switching Commands: These are commands exchanged between IEDs/Agents to clear short-circuit faults, and they are usually exchanged over a LAN. Typically, these messages are either DNP3.0 switching commands or GOOSE commands.

2. Manual Switching Commands: These are commands sent by the system operator in the control center over a WAN. These messages could be either DNP3.0 or R-GOOSE messages, as defined in IEC Technical Report TR 61850-90-5 for Routable GOOSE over WAN.

As reported in Table 4.1, GOOSE messages are Layer 2 messages of the OSI, which are exchanged over a LAN. The OSI model divides a network into seven abstraction layers with the goal of providing interoperability to communication systems. In this work, we assume that an attacker is able to perform a GOOSE Spoofing and Poisoning attack. First, the attacker sniffs the network for GOOSE messages. Since these messages are unencrypted, the attacker could decode the content of the GOOSE message and modify the data fields (i.e., change the OPEN command to CLOSE, or vice-versa). Here, it is important to understand that GOOSE messages are event-driven, and each message is associated with an incremental counter, called stNum. For example, the IED starts by sending a GOOSE message with stNum = 1. If a fault happens, the IED senses this fault and issues a new GOOSE commands with stNum = 2, to open the circuit breaker and clear the fault. Knowing that, the attacker then publishes the poisoned GOOSE message with a new incremented stNum and a spoofed MAC address. That is, the attacker uses the MAC address of the original sender. This process is depicted in Figure 4.2.

Table 4.1. Classification of Switching Commands and the Assumed Attacks.

Message Type	**Open System Interconnect Layer**	**Network**	**Assumed Attacks**
GOOSE	Layer 2 Data Link (MAC)	LAN	GOOSE Poisoning and Spoofing
R-GOOSE	Layer 3 Network (IP)	LAN/WAN	ARP Poisoning Man-in-the-Middle
DNP3.0	Layer 4 Transport (TCP/IP)	LAN/WAN	ARP Poisoning Man-in-the-Middle



(a) GOOSE Poisoning and Spoofing

(b) ARP Spoofing and Man-in-the-Middle

Figure 4.2. (a) Goose Poisoning and Spoofing Procedure; (b) ARP Poisoning and R-GOOSE and DNP3.0 Man-in-the-Middle Attack.

Table 4.1 also shows that R-GOOSE and DNP3.0 are Layer 3 and Layer 4 messages, respectively. This means that they are exchanged over an IP network. Accordingly, they are susceptible to Address Resolution Protocol (ARP) Poisoning and Man-in-the-Middle (MITM) attacks. ARP is a communication protocol used to convert IP addresses into MAC addresses [42]. As shown in Figure 4.2b, the attacker sends an ARP Reply to install a fake IP address and MAC address mapping to other hosts on the network. Therefore, the IP address of the attacker is, now, associated with an incorrect MAC address. This allows the attacker to intercept the messages exchanged between the control center and the subscriber IED. The attacker can, then, perform a Man-in-the-Middle attack and manipulate the data fields in the R-GOOSE or DNP3.0 packets.

Finally, although most industrial communication networks are not open to the public internet, we assume that they can still be penetrated through corporate networks or personal devices of the employees with techniques such as password cracking, backdoors, and malwares among others [30].

4.2.2 The Proposed Security Algorithm

As seen in Figure 4.1, the proposed multi-agent framework requires sectionalizing of the microgrid into several zones and assigning an agent to each zone, which will be responsible for local control and security actions. A block diagram of the security module of each agent and its network interfaces is shown in Figure 4.3. As can be seen in Figure 4.3, the security functionality in each agent is divided into two layers: an AI module and an Expert System module.

4.2.2.1 The AI Module

In the first layer, the agent is continuously listening to incoming control commands from the control center through its Ethernet network interface. Once a command is received, the agent decodes the content of the network packets and checks if the requested change is within its area or not. The agent gets activated only if the change is within its area. Once the agent is activated, the AI module will check if the command is to disconnect a generation unit or a critical load. This command will be processed only if the agent sees an override signal from the system operator over its isolated network interface. For all other commands, the AI module will pass the commands through a trained neural network that will solve the power flow problem for the system. The AI module will output the minimum voltage in per unit, the bus number on which this minimum voltage is anticipated, and the maximum transmission line overloading. The NN in this work is a feed forward NN trained using the back-propagation algorithm.

The mathematical processes guarding the operation of the AI module are explained below.

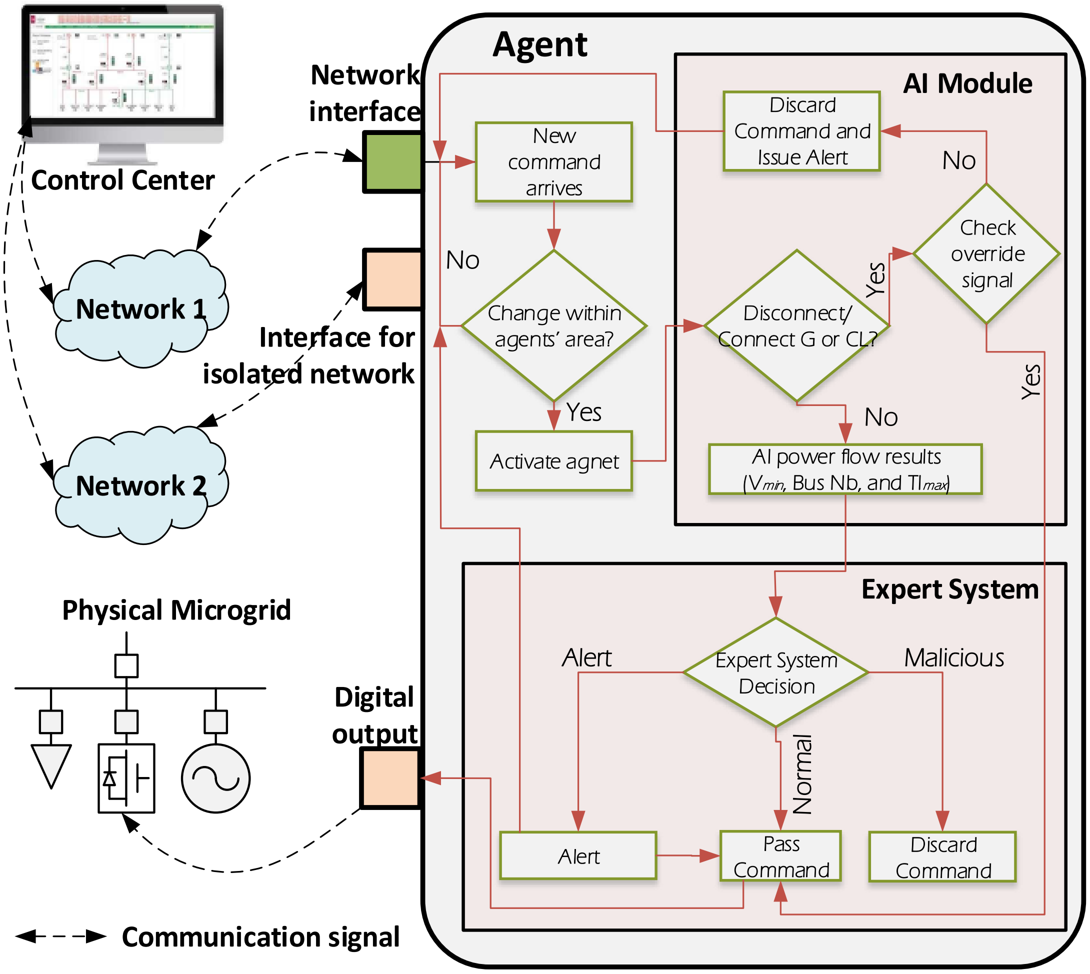


Figure 4.3: Agent operation and the physical-model-checking approach.

The NN utilized in this model is a three layer one composed of an input layer, a hidden layer, and an output layer. Let $X1[I+1]=\{x1_1, x1_2,..., x1_i,1\}$ $\mathrm{X1}[\mathrm{I}+1] = \{\mathrm{x1}_1, \mathrm{x1}_2, \dots, \mathrm{x1}_\mathrm{i}, 1\}$be the input coming to the NN, where $i \in \{1, I\}$, I Iis the dimension of the input, and $x1_{I+1} =1$ $\mathrm{x1}_{\mathrm{I+1}} = 1$ is the bias for the input layer. In the input layer, the inputs are multiplied by the weights ($w1_{h,i}$ $\mathrm{w1}_{\mathrm{h,i}}$) to get the vector $N1[H]=\{n1_1, n1_2,..., x1_h\}$

$N1[H] = \{n1_1, n1_2, \ldots, n1_h\}$. H H is the dimension of the hidden layer and $h \in \{1, H\}$ $h \in \{1, H\}$. The elements of $N1$ N1are calculated in accordance to (1).

$$n1_h = \sum_{i=1}^{I+1} x1_i \times w1_{h,i} \tag{4.1}$$

Next, each element in $N1$ N1 will be processed through a neuron in the hidden layer, which will result in $X2[H+1] = \{x2_1, x2_2, ..., x1_h, 1\}$ $X2[H + 1] = \{x2_1, x2_2, \ldots, x2_h, 1\}$. The elements in $X2$ X2are calculated according to (2) and (3).

$$x2_h = \frac{2}{1+e^{-n2_h}} - 1 \tag{4.2}$$

$$x2_{H+1} = 1 \tag{4.3}$$

$x2_{H+1}$ $x2_{H+1}$is the bias of the hidden layer. The equation in (4.2) is considered as the activation function for the hidden layer neurons, which represents a sigmoid function.

Finally, $X2$ X2will be processed by the output layer to get the output vector $O[K] = \{o_1, o_2, ..., o_k\}$ $O[K] = \{o_1, o_2, \ldots o_k\}$, where K Kis the dimension of the output and $k \in \{1, K\}$ $k \in \{1, K\}$. The elements in O Oare calculated according to equation (4.4).

$$o_k = \frac{2}{1+e^{-\sum_{h=1}^{H+!} x2_h \times w2_{h,k}}} - 1 \tag{4.4}$$

$w2_{h,k}$ $w2_{h,k}$ are weights between the hidden and output layers. The sigmoid function is also used as the activation function for the neurons in the output layer, as in (4.4).

The output vector is then interpreted to get the minimum voltage, the bus number at which this voltage occurred, and the maximum transmission line loading.

4.2.2.2 The Expert System Module

This module consists of a fuzzy inference system. The outputs of the AI module, are passed to the fuzzy inference system for further processing. These are the minimum bus voltage recorded in the power system, the number of the bus at which that minimum voltage occurred, and the value of the maximum transmission line overloading. It is to be noted here that the expert system module does not account for over-voltage cases. This is due to the fact that such cases will be the result of disconnecting multiple loads from the power system; however, the proposed security algorithm will automatically counteract such incidents.

The value of the bus voltage, which is passed to the expert system module, is fuzzified using four membership functions. These membership functions are designed to reflect the behavior of the power system according to the recommendations of the ANSI C84.1-2006 standard, from the voltage point of view [74]. The membership functions that correspond to the voltage are:

- Very Low (VL), repressing severe under voltage and Very High (VH), representing severe over voltage.
- High (H), representing normal condition and Low (L), representing mild undervoltage. For the latter case, a corrective action is necessary, such as reactive voltage support.

In this work, it is assumed that different priorities are assigned to the power system buses by the system operator. That is, high priorities will be assigned to the buses where main generator units or critical loads are connected, whereas low priorities will be assigned to busses with normal loads that can be shed. Due to the limited capacity in the generators adopted in this model, all the buses with generators were considered as critical buses (buses 1, 2, 3, 6, and 8), because the disconnection of any of the generators can result in cascaded failures in the whole system, which will result in a partial or full blackout. Also, generators are expensive equipment and are hard to replace within a reasonable time during emergencies. The study by Assante, which was conducted in 2007, showed how a massive diesel generator could be physically and permanently broken with only digital commands [101]. In fact, manipulating digital commands actually occurred in real life in the Crash Override malware. The behavior of the power system in response to the Crash Override malware that targeted the Ukrainian power grid was considered as an example in assigning the critical busses in the test bench system presented in this paper [105]. In the aforementioned malware, the attackers targeted switching commands from the SCADA system, which controlled the status of circuit breakers. They were able to de-energize some of the substations resulting in blacking out a portion of the Ukrainian capital equivalent to a fifth of its total power capacity. As reported in [103]: "The command sequence polls the target device for the appropriate addresses. Once it is at the subset of known addresses, it can then toggle the value. The command then begins an infinite loop and continues to set addresses to this value effectively opening closed breakers. If a system operator tries to issue a close command on their Human Machine Interface (HMI) the sequence loop will continue to re-open the breaker. This loop maintaining open breakers will effectively de-energize the substation line(s) preventing system operators from managing the breakers and re-energize the line(s)."

We assigned the loads at buses 2, 3, 4, and 9 to be critical loads. The bus number is therefore considered as another second input to the fuzzy system. Finally, the fuzzy system takes the maximum transmission loading (TLLmax) as its third input. The membership functions for this input are divided according to the rating procedures, that can be found in [111], to normal loading (represented by N), allowable overloading (represented by LTE), and unallowable overloading (represented by STE). The acronyms are in accordance with those is [111].In this work, the trapezoidal membership functions, shown in relation (4.5), were considered.

$$f(x) = \begin{cases} 0, (x < a) \cup (x > d) \\ \dfrac{x-a}{b-a}, a \le x \le b \\ 1, b \le x \le c \\ \dfrac{d-x}{c-d}, c \le x \le d \end{cases} \tag{4.5}$$

The boundaries $\{a,b,c,d\}$ {a, b, c, d} for each input level are defined in Table 4.2.

Table 4.2**.** Ranges of Membership Functions.

		a	**b**	**c**	**d**
Voltage (V p.u.)	VL	0	0	0.872	0.92
	L	0.9	0.924	0.924	0.95
	H	0.94	1	1	1.05
	VH	1.03	1.04	2.082	2.36
TLL (%)	N	0	0	73	105
	LTE	100	115	115	130
	STE	128	148	200	200

As for the busses' number input, impulse membership function are defined, as in equation (4.6).

$$f(x)=\begin{cases}1, x\in\{1,N\}\\ 0, otherwise\end{cases} \tag{4.6}$$

Where N Nis the number of busses in the system and $x\in Z^{+*}$ x ∈ Z^{*+}.

The output of the expert system is defuzzified based on the weighted sum for the Sugeno technique following equation (4.7):

$$output=\frac{\sum w_i x_i}{\sum w_i} \tag{4.7}$$

Since this is a Sugeno-like fuzzy system, the output is a number, which value is interpreted to be one of the following three cases: normal, alert, or malicious. In the normal case, the control commands are executed as they come, whereas in the malicious case, the commands are blocked, since they are suspected to put the power system in an insecure state. Finally, in the alert case, the control commands are executed, but an alert is passed to the system operator over the out-of-band channel to take corrective actions, if necessary. The controller makes its decision according to the following four fuzzy rules:

1. For the cases in which the control commands will lead the power system to have extreme over-voltage (represented by VH) or extreme under-voltage (represented by VL) on any of its busses, or extreme overloading of any of its transmission lines (represented by STE), the control command will be considered malicious by the fuzzy inference system. This is because, under no reasonable circumstance, will the system operator perform circuit breakers switching actions that will put the power system in an insecure state, which might be the cause of cascaded blackouts.

2. For the cases in which the control commands will lead the power system to have low voltages (represented by L) on one of the busses that has a main generator and/or a critical load connected to it, the fuzzy inference system will consider command will also consider the command as malicious. This is because such critical buses must maintain good voltage conditions at all times.
3. For the cases in which the voltage ends up to be low (represented by L) on the buses, which are not accounted for in the second rule (Rule 2), or when the transmission line loading condition is expected to be LTE, the control commands will be passed. However, the system operator will receive an alert signal over the out-of-band channel, in order to see if further actions are necessary.
4. For the remaining cases, in which the incoming control commands do not place the power system in an insecure state, the fuzzy inference system will pass the control commands without any issue.

Based on the previous rules, and following the standards for voltage rating and line loading [74][111], respectively, the voltage at the different busses should be maintained within acceptable limits and the line loading should be maintained within certain values to avoid overheating. Accordingly, the expert system was designed to ensure that any malicious action, which is anticipated to put any part of the system outside the acceptable standard limits, whether it is voltage or line loading or loss of generation or critical loads, will not be processed.

It is important to note here that the system operator will need to perform some tasks, such as maintenance tasks, which require the temporary disconnection of transmission lines or shutdown of generators. For that purpose, the system operator has the ability to communicate with all the agents over an encrypted and out-of-band communication channel and send a signal to override the decision of the agents.

4.3 Simulation Results

First, the proposed algorithm was verified in simulation on the 14-bus IEEE benchmark system, which is shown in Figure 4.1. The system was divided into three zones, in such a way that each zone has at least one generation unit and one load. Specifications of the system are found in detail in [112] and in Table 4.3.

Table 4.3**.** Ratings of the Simulated System

Generator Number	**Real Power (MW)**		**Reactive Power (MVAR)**	
	Min	**Max**	**Min**	**Max**
1	10	160	0	10
2	20	80	−42	50
3	20	50	23.4	40
4	--	--	-6	24
5	--	--	-6	24

Bus Number	**Load Real Power (MW)**	**Load Reactive Power (MVAR)**
1	0	0
2	21.7	12.7
3	94.2	19.1
4	47.8	-3.9
5	7.6	1.6
6	11.2	7.5
7	0	0
8	0	0
9	29.5	16.6

10	9.0	5.8
11	3.5	1.8
12	6.1	1.6
13	13.8	5.8
14	14.9	5.0

Line Number	From Bus	To Bus	MVA Rating
1	1	2	120
2	1	5	65
3	2	3	36
4	2	4	65
5	2	5	50
6	3	4	65
7	4	5	45
8	4	7	55
9	4	9	32
10	5	6	45
11	6	11	15
12	6	12	32
13	6	13	32
14	7	8	32
15	7	9	32
16	9	10	32
17	9	14	32
18	10	11	12
19	12	13	12
20	13	14	12

As mentioned earlier, the AI module of the first layer of the agent is trained to learn the characteristic of the system. This is done according to the following procedure:

- *Generating the Training and Test Target Data Sets*: First, in this work, we assume *N*-1 contingency cases (i.e., disconnecting the transmission lines, one at a time) and the disconnection of the generators and the loads. For the simulated 14-bus system, this gives us 36 cases plus the normal case, where all the circuit breakers are closed. For each case, the power flow problem was solved and the real power (*P*), reactive power (*Q*), and bus voltage (*V*) at the different busses, and transmission line loading (*TLL*) results were recorded as follows:

$$\begin{bmatrix} P1Bus1 & ... & P1BusN & Q1Bus1 & ... & Q1BusN & V1Bus1 & ... & V1BusN & TLL11 & ... & TLL1K \\ P2Bus1 & ... & P2BusN & Q2Bus1 & ... & Q2BusN & V2Bus1 & ... & V2BusN & TLL21 & ... & TLL2K \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ P37Bus1 & ... & P37BusN & Q37Bus1 & ... & Q37BusN & V37Bus1 & ... & V37BusN & TLL371 & ... & TLL37K \end{bmatrix}$$

where, $[P_iBus1 \dots P_iBusN]$, $[Q_iBus1 \dots Q_iBusN]$, and $[V_iBus1 \dots V_iBusN]$ are the real power, reactive power, and voltage of the system busses, respectively, and $i \in \{1,37\}; N = 14$. $[TLL_i1 \dots TLL_iK]$ are the transmission lines loading, and $K = 20$.

- *Generating the Training and Test Input Data Sets*: Second, in this case study, the switching commands corresponding to each contingency were mapped to a 6-bit binary code, as shown in the matrix below. For instance, the code 000010 will be utilized to represent a control signal to actuate circuit breakers connecting bus 1 to bus 2.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} ----> \begin{bmatrix} Case1 \\ Case2 \\ Case3 \\ \mathrm{M} \\ Case37 \end{bmatrix}$$

- *Training the Neural Network*: The neural network of the agent was trained according to the Back Propagation algorithm to accurately predict the system response in terms of the real power, reactive power, and voltage of the buses and the transmission line loading.

The accuracy of the neural network is shown in Figure 4.4. The maximum errors recorded were 2.6% for the active and reactive power, 10^{-4} for bus voltages, and 0.614% for transmission line loading.

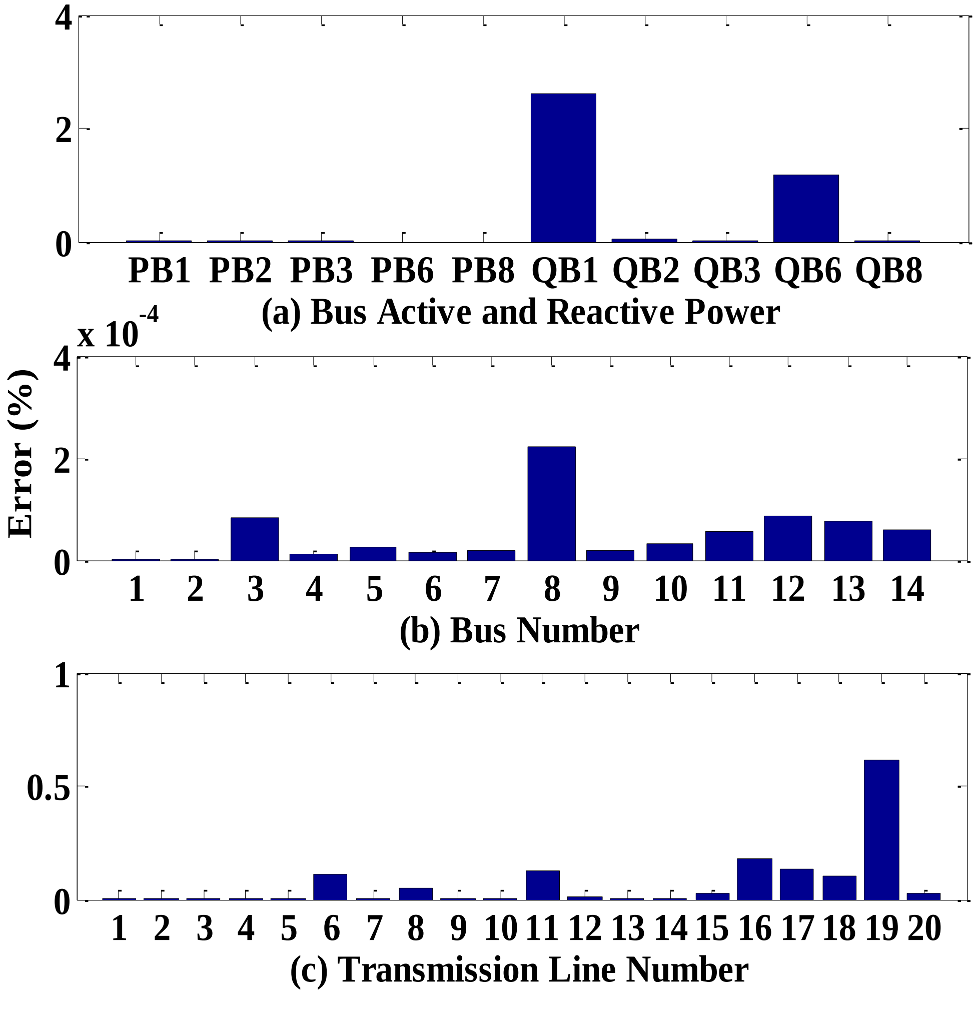


Figure 4.4: Goodness of Fit of AI Module in Terms of Mean Squared Error: (a) Generators Active and Reactive Power; (b) Bus Voltages; (c) Transmission Line Loading.

Table 4.4: Simulation Results

					Compare w. NC			Sequential								Compare w. NC			Sequential		
	Case	V_{min} (pu)	Bus Nb.	TLL_{max} (%)	A1	A2	A3	A1	A2	A3		Case	V_{min} (pu)	Bus Nb.	TLL_{max} (%)	A1	A2	A3	A1	A2	A3
	NC	1.02	4	103	🟨	🟨	🟨	🟨	🟨	🟨		**NC**	1.02	4	103	🟨	🟨	🟨	🟨	🟨	🟨
1	**B1-B2**	1.02	5	252	🟥			🟥			**19**	**12-13**	1.02	4	103		🟨			🟨	🟨
2	**B1-B5**	1.01	5	134	🟥			🟥			**20**	**13-14**	1.02	4	106		🟨	🟨		🟨	🟨
3	**B2-B3**	1.01	3	156	🟥		🟥	🟥		🟥	**21**	**GB1**	n/a	n/a	n/a	🟥			🟥		
4	**B2-B4**	1.01	4	180	🟥		🟥	🟥		🟥	**22**	**G B2**	n/a	n/a	n/a	🟥			🟥		
5	**B2-B5**	1.01	5	113	🟩			🟩		🟩	**23**	**G B3**	n/a	n/a	n/a			🟥	🟥		🟥
6	**B3-B4**	1.02	5	102			🟩	🟩		🟩	**24**	**G B6**	n/a	n/a	n/a		🟥			🟥	🟥
7	**B4-B5**	1.02	4	131	🟥		🟥	🟥		🟥	**25**	**G B8**	n/a	n/a	n/a			🟥		🟥	🟥
8	**B4-B7**	1.02	4	130			🟥	🟥		🟥	**26**	**CL B9**	n/a	n/a	n/a			🟥			🟥
9	**B4-B9**	1.02	4	123			🟨			🟨	**27**	**L B6**	1.02	4	103		🟨			🟨	🟨
10	**B5-B6**	0.99	12	198	🟥	🟥		🟥	🟥	🟥	**28**	**L B5**	1.02	4	102	🟨				🟨	
11	**B6-11**	1.02	4	108		🟨		🟨	🟨		**29**	**CL B4**	n/a	n/a	n/a			🟥			🟥
12	**B6-12**	1.02	4	103		🟨			🟨		**30**	**CL B3**	n/a	n/a	n/a			🟥			🟥
13	**B6-13**	0.99	13	108		🟩			🟩		**31**	**CL B2**	n/a	n/a	n/a	🟥			🟥		🟥
14	**B7-B8**	0.02	8	100		🟥	🟥		🟥	🟥	**32**	**L B14**	1.03	4	98			🟩	🟩		
15	**B7-B9**	1.02	14	129			🟨			🟨	**33**	**L B13**	1.02	4	103		🟨			🟨	🟨
16	**B9-10**	1.02	5	108			🟩			🟩	**34**	**L B12**	1.02	4	102		🟨			🟨	
17	**B9-14**	0.99	14	117			🟩			🟩	**35**	**L B11**	1.02	4	102		🟨			🟨	
18	**10-11**	1.02	4	107		🟨	🟨		🟨	🟨	**36**	**L B10**	1.02	4	102			🟨		🟨	🟨

* NC: Normal Case; B: Bus; G: Generator; L: Load; CL: Critical Load; A1: Agent 1; A2: Agent 2; A3: Agent 3; V: Voltage; TLL: Transmission Line Loading; Yellow: Alert; Red: Malicious; Green: Normal.

Simulation of the proposed framework were carried out in two different scenarios and the results are tabulated in Table 4.4. In the first scenario, the response of the agents is assessed for each contingency case separately. That is, after each command, the system is reverted to normal case before executing the next command. The results for the first scenario are as follows:

- Cases 21-25 and 29-31 correspond to the disconnection of either a generation unit or a critical load from the system. Since the override signal was set to zero throughout the experiment, these commands were regarded as malicious and, therefore, were not passed to the circuit breakers.
- Cases 9, 11, 12, 15, 18-20, 27, 28, and 33-36 resulted in alert situation, where the commands were passed but an alert was issued to the system operator. It was noticed that there was no severe bus voltage deviations in these cases from the allowable limits. In fact, in all the cases, the minimum bus voltage was 1.02 p.u.. Also, the maximum recorded transmission line loading was 129%, which falls into the LTE state of Table 4.2.
- Cases 1-4, 7, 8 10, and 14 violated the physical operation constraints of the microgrid, and thus, were regarded as malicious by the agents.
- The rest of the cases were regarded as normal. It is worth noting that in these cases, commands to disconnect non-critical loads, such as cases 27 and 28, were passed. Although these commands might be malicious or erroneous, they were passed by the multi-agent system since they did not put the microgrid in a contingency state and the microgrid maintained its stability. Therefore, the multi-agent system was

successful in satisfying its purpose by ensuring that only signals that do not violate the stable operational limits of the microgrids will be passed.

Therefore, out of all the simulated cases, the multi-agent security framework allowed the passage of a command that led to the disconnection of a non-critical load 2 times. This is equivalent to 5.56% of the simulated cases. To address this issue, each agent generates periodic log reports and sends them to the system operator over its isolated and encrypted network interface. This allows the system operator to get feedback about the statuses of the circuit breakers and utilize this feedback to take corrective actions, when deemed necessary. Also, it can be seen form Table 4.4 that the proper agent was activated for each case. That is, in case 1, the control command is intended to actuate the circuit breakers on the transmission line connecting bus 1 to bus 2. In this case, the change is in Area 1 only. Thus, only Agent 1 was activated. On the other hand, in case 7, the control command is intended to actuate the circuit breakers on the transmission line connecting bus 4 to bus 5. This line connects Areas 1 and 3. Therefore, Agent 1 and Agent 3 were activated. In the second scenario, the control commands were sent in a sequential manner, one after the other, without getting back to the normal case. The results in this scenario produced similar outputs to the previous one, with the major difference being the number of activated agents in each case. For instance, after executing case 4 in Table 4.4, returning to the normal case, then executing case 5, only Agent 1 was activated and gave normal condition. However, when executing case 4 then directly executing case 5, Agent 1 and Agent 3 were activated and gave normal condition. This is because Agent 1 sensed a change in the status of the

CB connecting bus 2 to bus 4 (from 0 back 1) and Agent 2 sensed a status change in the CB connecting bus 2 to bus 5 (from 1 to 0).

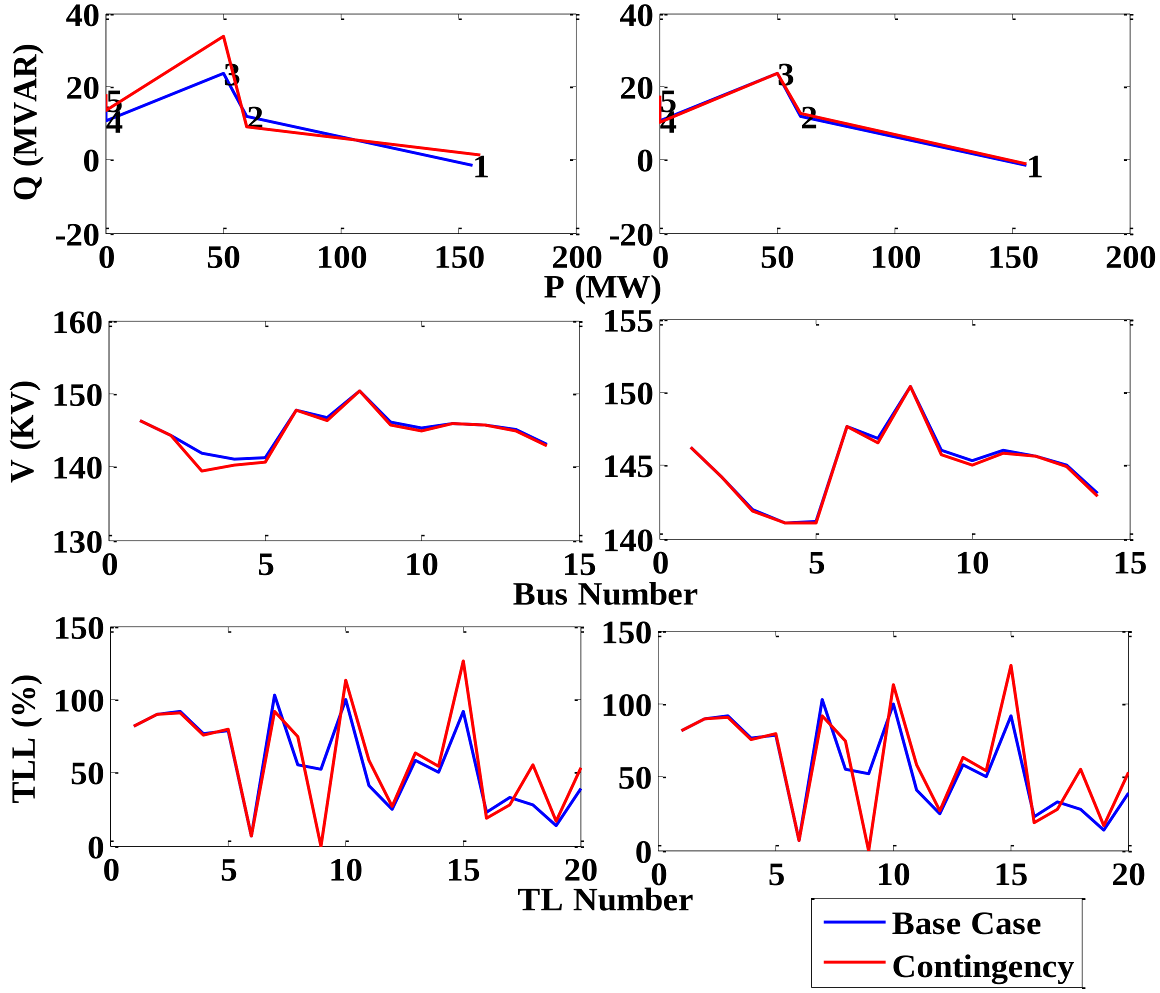


Figure 4.5: (Left Column) Comparison of Real and Reactive Power, V, and TLL of Malicious Case to Base Case; (Right Column) Comparison of Real and Reactive Power, V, and TLL of Alert Case to Base Case.

A sample report of the post processing done by the system operator based on the log reported by the agents is shown in Figure 4.5. A comparison between the real and reactive power of generation units, bus voltages, and transmission line loading of the normal case and case 3, which is a malicious situation, is shown in Figure 4.5 (Left Column). Figure

4.5 (Right Column) shows the same for an alert situation, which corresponds to case 9. The reported data for case 3 shows that if the agent were to process that control command, the system would significantly deviate from its normal case. On the other hand, the graphs corresponding to the alert state show that the processing of the control command would not result in a significant deviation from the normal case in terms of generators' power and bus voltage. However, the transmission line loading would change but will not exceed the allowable limits. These graphs are useful visualization tools for system operators and designers that will assist them in future plans and lessons learned.

4.4 Hardware Setup and Experimental Results

4.4.1 Description of the Hardware Setup

The performance of the proposed security framework was tested on the 5-bus microgrid shown in

Figure 4.6 (a). The microgrid has the following components:

- Two generation units, Generator 1 and Generator 2, with 13.8 KVA 230 V and 5 KW and 10.3 KVA 230 V and 3 KW, respectively.
- Seven distribution lines with a typical π-model.

  Three loads, each having 10 levels of parallel resistive loads, ranging from 300-W to 3-kW. In this experiment, L1, L2, and L3 are set at 600 W each. L3 is considered to be a critical load, and therefore, has a redundant path to the generation units.

- Each of the five buses has three sets of three-phase inputs and outputs with 530V/25A solid-state relays, whose switching can be controlled by digital inputs.

Each phase has its own potential and current transformer for measurement data collection.

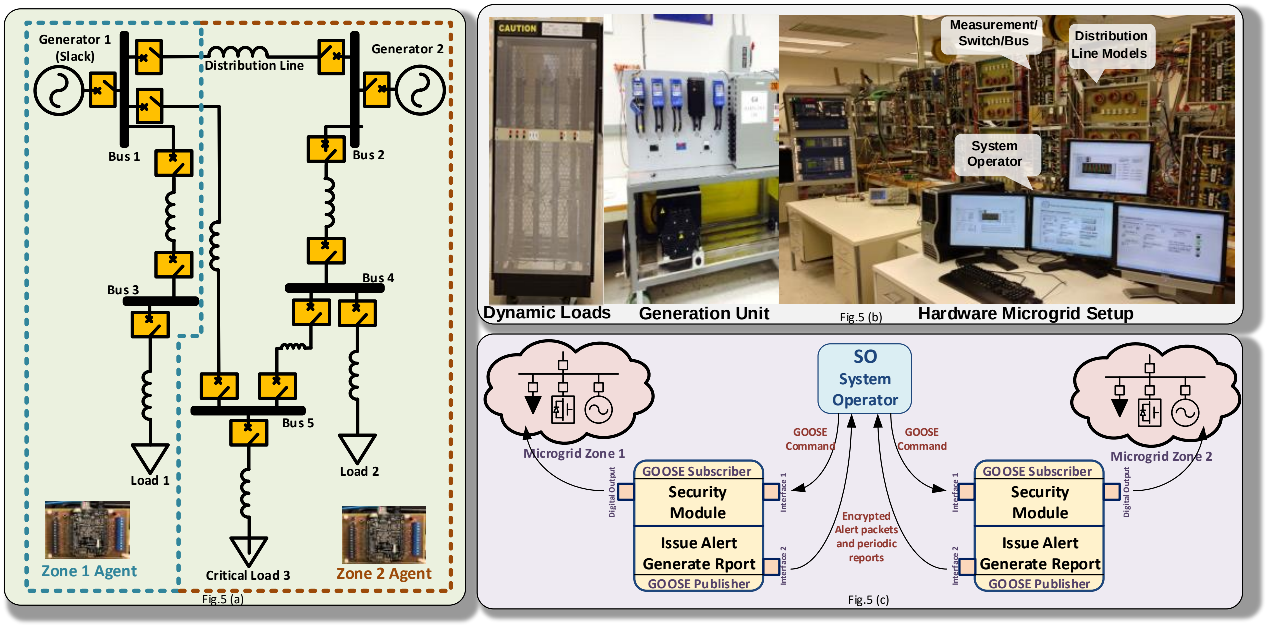


Figure 4.6: (a) One-line diagram of the 5-bus hardware microgrid; (b) Actual laboratory setup; (c) Network topology of agents.

Complete specifications of the system components are found in [1][113]. It is be noted that the experimental setup is not a portion of the simulation network. It is a totally new physical system, with its own parameters and components. Contingency analysis for the experimental system were also done independently and were fed to the neural network for training. Therefore, the two neural networks, the one used for simulation and the one used for the experimental work, are different networks and are not meant to mitigate attacks for the same power network. We emphasize that the experimental setup is not a portion of the simulation network. It is rather a different small-scale testbed benchmark that was used to validate our algorithm experimentally.

Table 4.5. Parameters of the Hardware Setup

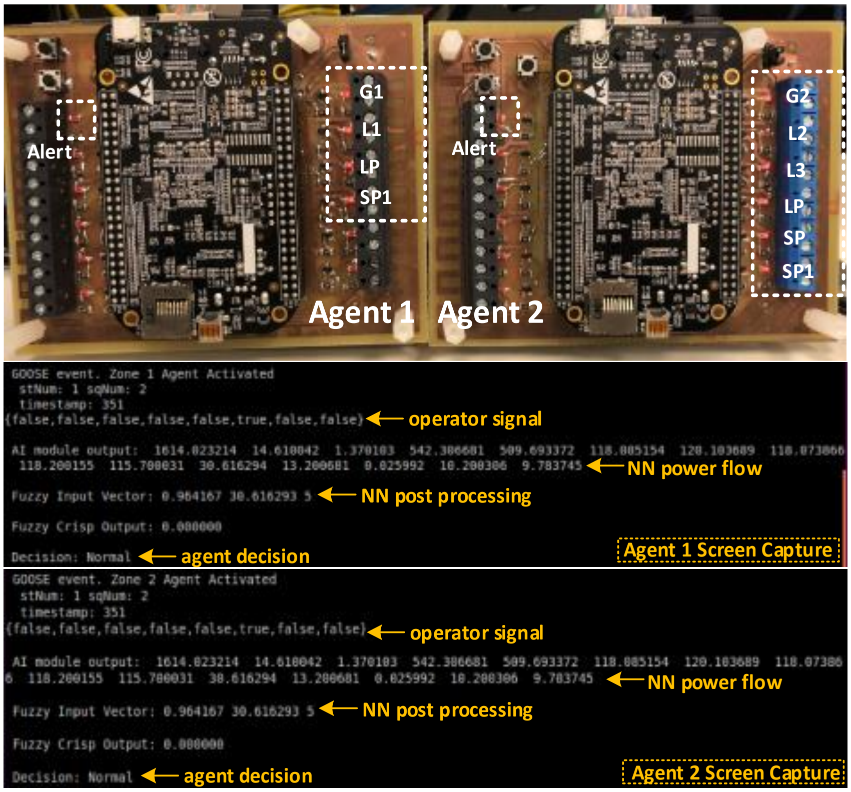
Generator Number	Real Power (KW)		Reactive Power (KVAR)	
	Min	Max	Min	Max
1	0	5	0	4.8
2	0	3	0	7.5

Bus Number	Load Real Power (W)	Load Reactive Power (VAR)
1	0	0
2	0	0
3	600	0
4	600	0
5	600	0

Line Number	From Bus	To Bus	KVA Rating (3-Phase)
1	1	2	5.4
2	1	3	5.4
3	1	5	5.4
4	2	4	5.4
5	4	5	5.4

4.4.2 Information Exchange and Agent Development

Figure 4.6 (c) depicts the information exchange between the developed agents and the system operator at the control center, which follows the IEC 61850 GOOSE publisher/subscriber model for high speed communication. The software embedded into the agents has two threads running in parallel. The first thread is a GOOSE Subscriber. This thread listens to incoming GOOSE commands over network interface 1 and processes them through its security module before interacting with the physical microgrids through its digital outputs. The second thread is a GOOSE publisher. This thread waits for an internal flag from the security module to issue an alert to the system operator over the isolated and encrypted network interface.



```
GOOSE event. Zone 1 Agent Activated
 stNum: 1 sqNum: 2
 timestamp: 351
{false,false,false,false,false,true,false,false}

AI module output:  1614.023214  14.610042  1.370103  542.306681  509.693372  118.005154  120.103689  118.073866
 118.200155  115.700031  30.616294  13.200681  0.025992  10.200306  9.783745

Fuzzy Input Vector: 0.964167 30.616293 5

Fuzzy Crisp Output: 0.000000

Decision: Normal
```

```
GOOSE event. Zone 2 Agent Activated
 stNum: 1 sqNum: 2
 timestamp: 351
{false,false,false,false,false,true,false,false}

AI module output:  1614.023214  14.610042  1.370103  542.306681  509.693372  118.005154  120.103689  118.07386
6  118.200155  115.700031  30.616294  13.200681  0.025992  10.200306  9.783745

Fuzzy Input Vector: 0.964167 30.616293 5

Fuzzy Crisp Output: 0.000000

Decision: Normal
```

Figure 4.7: Developed agents and their corresponding human machine interface.

Figure 4.7 shows the actual hardware agents with their digital output extension board. The embedded microcontroller on each agent has an AM335x 1GHz ARM® Cortex-A8. 512MB DDR3 RAM processor running a Linux kernel. Agent 1 has 4 digital outputs interfaced with the circuit breakers connected to Generator 1 (G1), Load 1 (L1), the Long Path (LP) connecting busses 1 and 5, and the Short Path (SP1) connecting busses 4 and 2.

4.4.3 Results and Discussion

A data set including bus measurements of previously recorded events and measurement data collected from a simulated model of the microgrid for various contingency cases was used to train the AI module of the two developed agents. The simulated microgrid was developed in Matlab/Simulink and was verified by comparing bus voltages, currents, and power measurements with experimental data for three different cases shown in Figure 4.8 (a), (b), and (c). The results show that the model accurately represents the actual system.

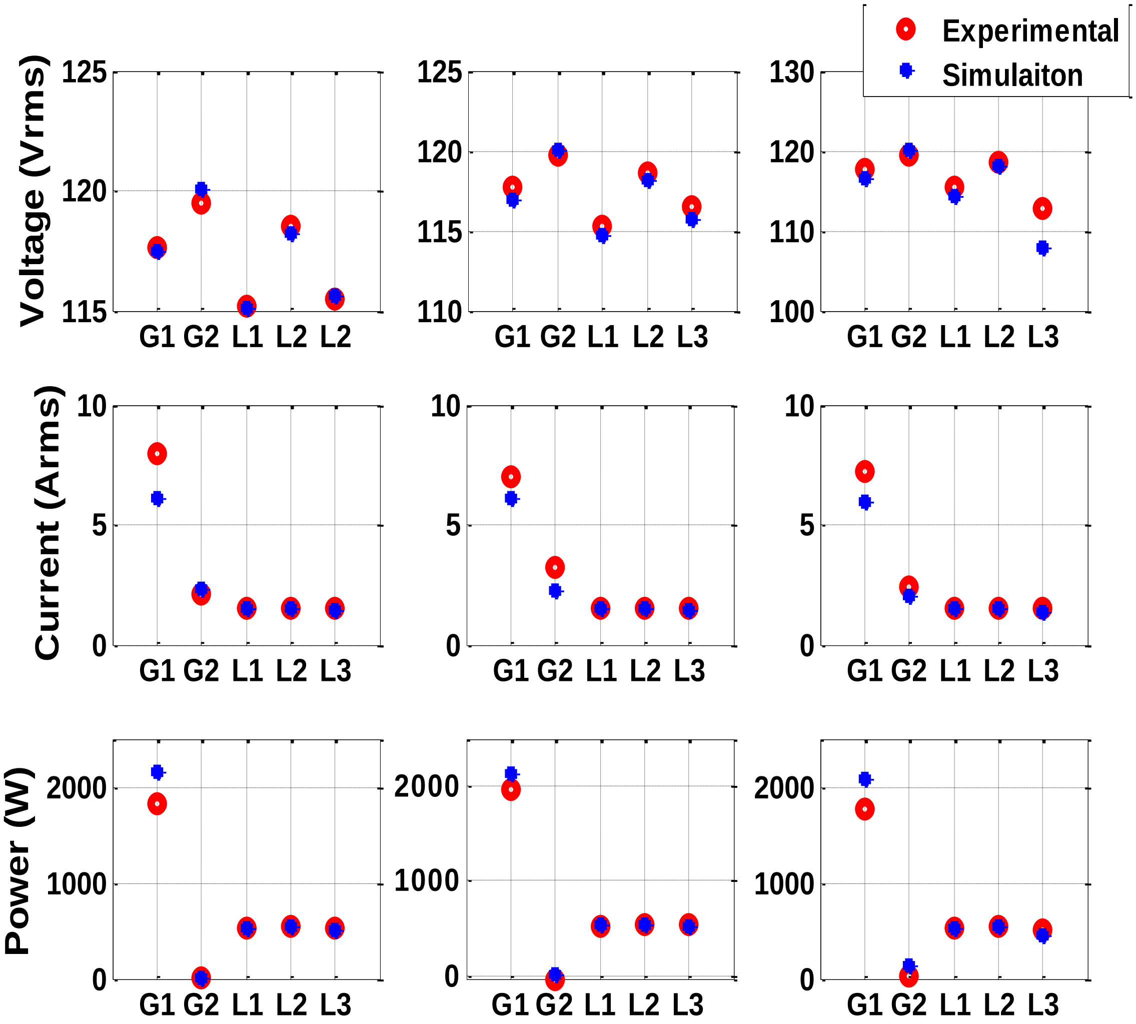


Figure 4.8: Comparison of the voltage, current, and power on all 5 buses between the simulation model and experimental setup in three cases: (a) normal case; (b) supplying CL3 from redundant path only; (c) under voltage case on CL3.

The performed experiment comprises of 9 control command signal combinations, which are:

- R1: Normal condition. All circuit breakers are closed.
- R2: Disconnection of slack generator (G1).

- R3: Disconnection of generator 2 (G2).
- R4: Disconnection of load 1 (L1).
- R5: Disconnection of load 2 (L2).
- R6: Disconnection of critical load 3 (CL3).
- R7: Disconnection of the circuit breakers between bus 4 and bus 5, which represents the main path to CL3.
- R8: Disconnection of the circuit breakers between bus 1 and bus 5, representing the redundant path to CL3.
- R9: Disconnection of the path between G1 and G2.

The power measurements on each of the buses were plotted throughout the duration of the experiment in Figure 4.9 to visualize which control commands actually passed and which were considered as malicious, and therefore, prevented. Note that in Figure 4.9, there is no relation between the regions (R1–R9) and the time scale. Since this is a hardware setup, which requires careful synchronization among the generators, the power system was ran and the attacks were performed. The power was plotted throughout the duration of the experiment to show which commands were passed by the agents and which ones were blocked. Also, Table 4.6 shows the output of the AI module and the decision of each agent in response to each command signal. As can be seen from the results of the normal case (R1), each of the three loads was at around 600 W summing to around 1800 W, which was provided by G1. G2 was acting as a synchronous condenser. As can be seen from Table 4.6 and Figure 4.9, the commands that attempted to disconnect G1 (R2), G2 (R3), and critical load L3 (R5) were directly considered as malicious commands. This is because the override signal sent over the isolated network interface to the agents was set to zero. The disconnection of the main path to CL3 (R7) was considered a malicious command, since the AI module anticipated an under voltage of 0.90 p.u. on the critical load bus.

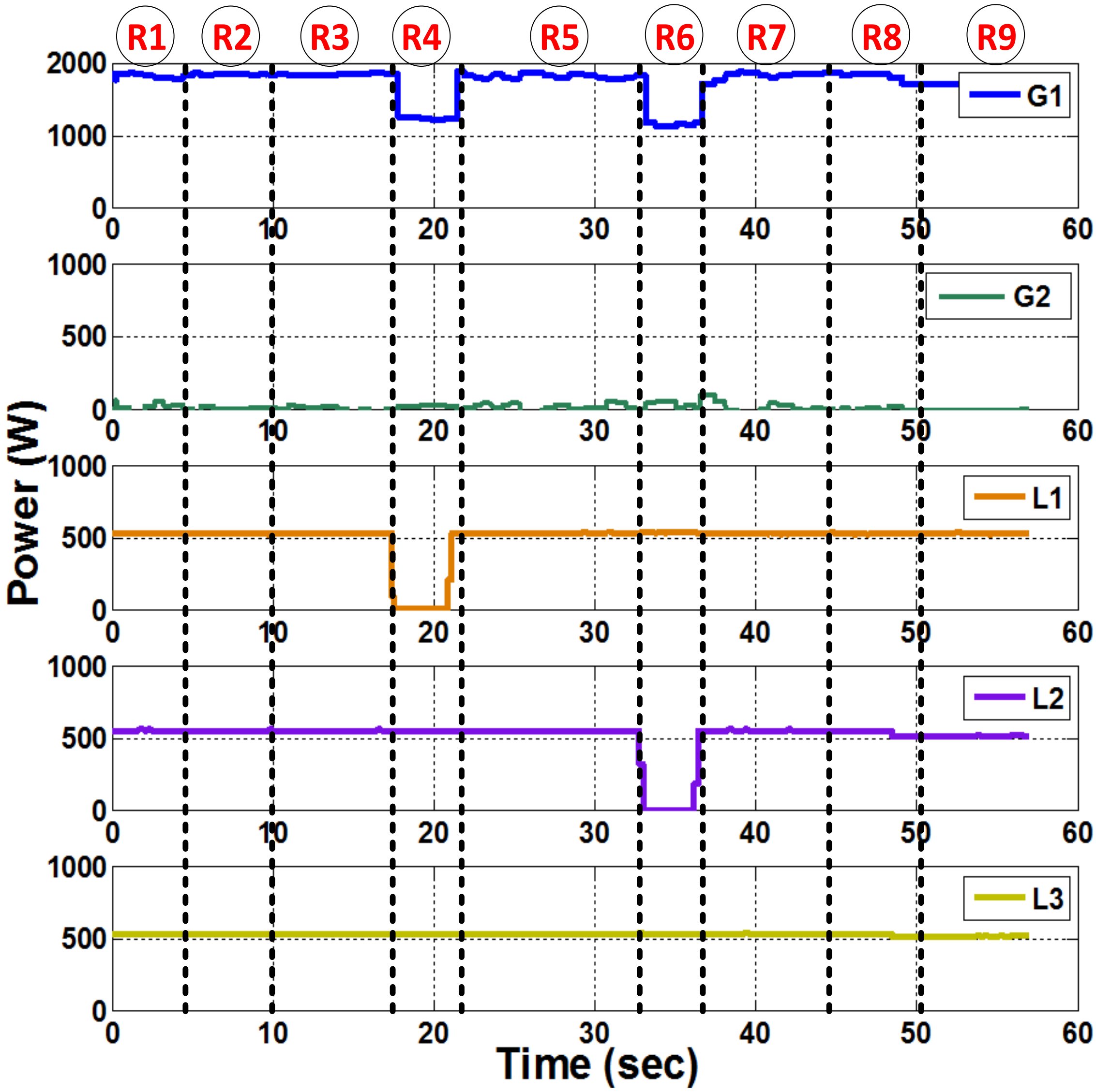


Figure 4.9: Power on each of the 5 buses of the studied microgrid. The figure is divided into 9 regions (R1 – R9) covering all 9 test cases.

It is noted that the disconnection of L1 and L2 was considered as a normal command by the agents. This is shown by the drop of power on L1 and L2 from around 600 W to 0

W in Figure 4.9. Although this command was actually not issued by the system operator, the agents still passed these commands, since they did not put the microgrid in an insecure state. As mentioned earlier, this is compensated for by the periodic reports that each agent sends to the system operator to take corrective action. In this experiment, after receiving periodic reports, the system operator restored power to L1 and L2, as shown in Figure 4.9. Similarly, the command to open the redundant path to CL3 (8) was passed. Finally, in the case where the circuit breakers connecting bus 1 to bus 2 (R9) were open, the minimum recorded voltage was 0.92 p.u., which was regarded as an alert situation and an alert signal was sent to the system operator. It is also noted in Table 4.6 that each agent was activated only when a change was detected within its zone.

Table 4.6. Experimental Evaluation of the proposed security system.

Cases (Fig.8)	Description	V_{min} (pu)	Bus Nb.	TLL_{max} (%)	A1	A2
R1	**NC**	0.95	3	40.8	N	N
R2	**Open CB G1**	---	---	---	M	
R3	**Open CB G2**	---	---	---		M
R4	**Open CB L1**	0.96	5	30.6	N	
R5	**Open CB L2**	0.96	3	30.5		N
R6	**Open CB CL3**	---	---	---		M
R7	**Open CB B4-B5**	0.90	5	40.0		M
R8	**Open CB B1-B5**	0.95	3	40.9	N	N
R9	**Open CB B1-B2**	0.92	3	52.7	A	A

Finally, a comparison of the online detection latency of the proposed method, which utilizes artificial intelligence to characterize the power system, with the work in [30], which uses a modified power flow analysis technique, is presented. [30] is chosen for comparison because it targets the same type of control-related attack detection utilizing power flow models. The results reported in [30] show that the online detection latency increases with the expansion of the system and can reach up to 200 ms. This is because their detection algorithm requires solving the adapted power flow problem online, every time a control command is issued. Thus, the time to solve the power flow problem is directly proportional to the size of the system. However, in the proposed framework the detection latency is in the order of (297 microseconds) and remains marginally constant as the number of the buses in the power system increases. This is because in this work, the computation time required for the AI module to produce an output is relatively constant regardless of the number of buses. It is worth noting that the performed comparison is for the online detection latency. That is, detecting attacks while the system is up and running. It is true that more than 200 ms would be required to collect training data and train the AI module; however, this will be performed offline and will not significantly affect the detection latency.

4.5 Conclusion

The work in this chapter proposed an artificially intelligent physical model-checking approach to detect malicious and erroneous control commands controlling the state of circuit breakers in power systems. The security algorithm was established as a multi-agent system, in which a given power system is sectionalized into separate areas and an agent is assigned to each area. The purpose of the work is to push enough intelligence into the agents controlling the microgrid to enable them to assess the consequences of control commands before taking actions on the physical system. The proposed security framework was tested on a 14-bus IEEE benchmark system. The results showed the accuracy of the AI module in characterizing the system under study and its effectiveness in not allowing the system to go into an insecure state. Next, the proposed multi-agent security framework was verified experimentally on a 5-bus power system with an IEC 61850 communication architecture.

Chapter 5 Online False Measurement Data Detection System Using Time Series Neural Networks

Migrating to a smart grid requires a paradigm shift in the implementation of power system applications. With the advent of IEC 61850, contemporary SASs are utilizing electronic instrument transformers and merging units to transmit current and voltage measurements over Ethernet as SMV. Also, the strict 4ms time constraint imposed on SMVs makes encrypting these messages nearly impossible. As such, this chapter this presents a detailed analysis of the Sampled Measured Values protocol and its benefits, then, it identifies its vulnerabilities and derives the associated cyber threats. Secondly, current security measures are outlined and the feasibility of using neural network forecasters to detect spoofed sampled values is investigated. The proposed forecasting algorithm was implemented in a system composed of merging units and intelligent electronic devices developed for this purpose. Real-time experimental results of the proposed algorithm over a real IEC 61850 network showed that neural network forecasters have good potential in terms of detecting fake messages and increasing the robustness of control and protection algorithms.

5.1 Introduction

IEC 61850 tackles data exchange in modern substation automation systems on three levels, being the process, field, and station levels, as in Figure 5.1. To achieve interoperability, IEC 61850 standardizes self-describing data, services, networks, and configuration for a complete description of a field device [8]. In order to achieve flexibility in SAS communications, the concept of a process bus has emerged as an Ethernet-based communication network between bay level Protection and Control IEDs and switch yard equipment [76]. Today, MUs play a significant role in an IEC 61850 process bus. In modern substations, MUs are installed near primary equipment in switchyards and are meant to provide synchronized phase voltage and current measurements, digitize them into a SMV packet format, and transmit them over a substation process bus network [76][79]. The deployment of the Process Bus in SAS has introduced several advantages including reduced copper wiring, modular substation architectures, easier commissioning and maintenance processes, as well as enhancing protection and control applications by leveraging data availability.

Given all the advantages of IEC 61850 process bus, extreme care must be placed on

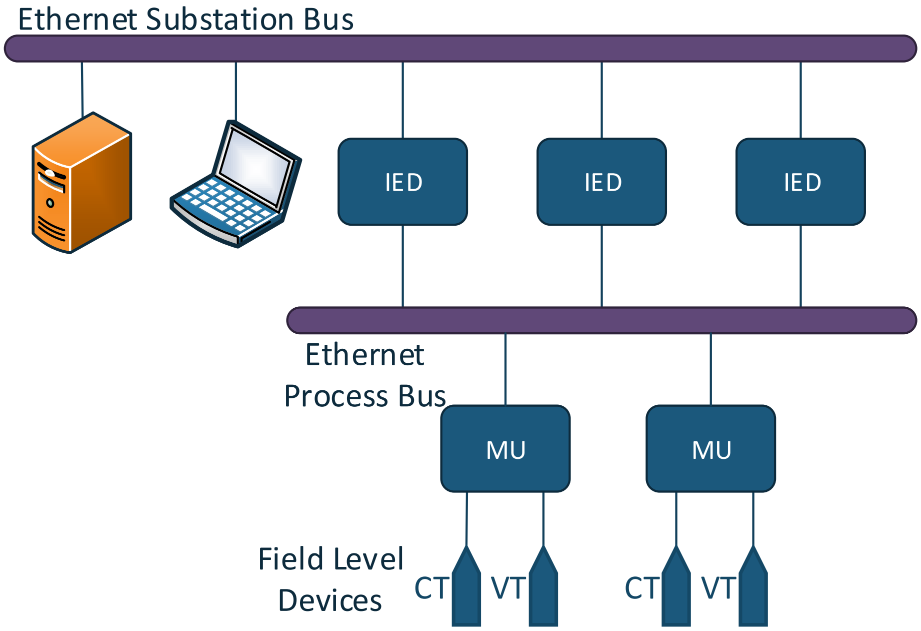


Figure 5.1: Communication between Bay and Process Level Devices.

its deployment from a cyber-security and network implementation perspective, as it carries extremely time-critical messages for sensitive substation protection and control applications. As per the IEC 61850-9-2, the maximum end-to-end time delay allowed for SMV messages is 4ms, which includes the time for processing, queuing, and transmission. This tight limitation on message transmission time makes it nearly impossible to encrypt SMV packets, especially with the low processing power of publishing MUs and receiving IEDs. IEC 62351-6, which covers the cyber security of SMV messages and relieves time-critical SMV messages from the burden of being encrypted, asserted this fact. The lack of encryption provides a great and easy attack surface for prying eyes, if a substation network is breached, to manipulate and send fake measurements, leading to catastrophic consequences in the power system [42].

As such, this chapter presents an analysis of the SMV protocol and discusses its advantages and disadvantages. Then, it outlines the cyber vulnerabilities and current security countermeasures. Finally, this chapter investigates the feasibility of using neural network forecasters to detect spoofed SMV measurements.

We conducted the study in this chapter on the cyber-physical smart grid test bed at Florida International University. IEC 61850-based MUs and IEDs were programmed and connected to the test bed's network in order to assess the performance of the system in a real-time environment and over a real network. The results showed excellent performance of the false data detection system in detecting fake measurements. It is worth noting that the proposed system did not violate the 4ms time constraint imposed on SMV protocol.

5.2 Overview of the SMV Process Bus Benefits

Generically, merging units are used to digitize the voltage and current measurements from potential and current transformers, respectively, and publish them as SMV messages to the process bus. In some situations, non-conventional instrument transformers directly publish measurements as SMV messages.

As described in IEC 61850–9–2, and later in the 9-2 Light Edition, the SMV protocol broadcasts measurement data across a LAN using a switched Ethernet-network to communicate. SMV messages are directly mapped into the Data Link Layer, which is Layer 2 of the OSI model. The OSI model divides the network into seven abstraction layers with the goal of providing interoperability to computer networks.

SMV messages that are broadcasted to the process bus are classified based on their Sampled Value ID (svID) field. Since SMV messaging follows the publisher/subscriber model, the controllers, that are commonly IEDs, subscribe to SMV messages based on the svID. This means that the IED does not have to receive all the measurements on the process bus, rather it filters the required measurements based on their svID.

This architecture introduces a lot of advantages to the control and operation of individual microgrids, and eventually the smart grid, from different perspectives.

- Firstly, in terms of system monitoring, the networked process bus simplifies the connections required for monitoring systems, thus, improving the visibility of the power network [79].

- Secondly, in terms of cost, the use of the process bus eliminates the need for point-to-point analogue connections. Therefore, this will drastically reduce the use of expensive copper wiring and will also reduce the cost of installation and maintenance.
- Thirdly, in terms of control, the SMV process bus facilitates the migration from centralized control network architectures to decentralized and distributed architectures that enhance the reliability of the power system. In centralized control, a single server communicates with all the sensors and issues most of the control commands. From a reliability view-point, the server is a bottleneck and a single point of failure. This vulnerability is avoided in decentralized and distributed control.
- Finally, in terms of data handling, the SMV process bus provides data interoperability between multi-vendor devices. By standard practice, all IEC 61850 SMV-compliant merging units and IEDs should abide by the message structure shown in Section III A. Therefore, the process bus is vendor-independent.

However, the benefits of the process bus do not come without risks. The following section will outline the vulnerabilities of the SMV protocol and current countermeasures efforts.

5.3 Threat Identification and Current Security Countermeasures

In this section, we analyze the vulnerabilities of the SMV process bus in terms of cyber-attacks. We start by discussing the sources of attacks on the process bus. Then, for each attack type, we explain how the attack works, if the SMV process bus is resilient to it, and the possible countermeasures.

Despite its benefits, the process bus has vulnerabilities that need to be addressed to ensure its security. Since SMV are directly mapped to the data link layer of the OSI model, they are non-routable and non-blocking. Therefore, in this discussion, we consider the cyber threats that could potentially occur on the SMV protocol within a local area network.

The sources of attacks on an IEC 61850 network could be an inside person, who has access to the IEC 61850 network, and has the capability to infect the system with malware. This action could be intentional, from a disgruntled employee, or unintentional by improper use of infected devices. Another source of attack could start from the supply chain, where a device, IED, is infected with malware during production [86]. A savvy attacker could also penetrate a LAN through corporate networks or personal devices of the employees with techniques, such as password cracking, spam emails, and backdoors [30]. Two major public disclosures on successful attacks on industrial systems and power system control networks are the Stuxnet incident, where a nuclear plant control network was attacked, and the Crash Override incident, where several substation networks in the Ukrainian power grid were attacked [102][103].

Once the LAN is compromised, the attacker has the potential to launch several types of attacks on the SMV process bus.

1. <u>Denial of Service (DoS)</u>: as the term indicates, a DoS attack is when a user or a machine is maliciously prevented from accessing a service. One possible way

to perform a DoS attack is by flooding the network to delay message delivery past the critical flooding rate through congesting the communication channel and exhausting the computation resources of the communicating nodes. Flooding attacks can happen on the network or the application layer of the OSI model. In this work, we are interested in DoS attacks on the process bus. A DoS by flooding the network could occur on the data link layer by broadcasting SMV messages with a high publishing rate, imposing delays on legitimate SMV messages beyond 3 ms. This form of DoS could be detected by classifying the SMV packets based on the APPID and monitoring the rate of publishing of each one on the process bus. According to IEC 61850-9-2LE, there are two allowed publishing rates for SMV messages, either 80 samples/sec or 256 samples/sec. If these rates are violated for a certain SMV, a DoS attack could be detected.

If an attacker, however, uses more than one physical or virtual device to publish SMV packet streams within the acceptable publishing rates, s/he could still congest the network. Here, the operator could create a list that defines the allowed svIDs and the corresponding data in the PDU. Utilizing an undefined svID is, thus, easily detected. Also, since each SMV is associated with a sample counter, if an attacker uses one or more the registered svIDs, the operator can detect repeated or out-of-sequence sample counters. This is because Layer 2 messages cannot be blocked. Finally, small and separated message bursts, which do not trigger the preset publishing thresholds, could also be detected by monitoring the sample counter.

2. Eavesdropping: this is a passive attack, where an attacker sniffs network traffic. Nonetheless, this attack allows the attacker to gain knowledge about the power system. For applications that are on the IP Layer and above, defense strategies against eavesdropping include encryption, such as utilizing Secure HTTP (HTTPS), Secure File Transfer Protocol (SFTP), and Secure Shell (SSH). Defense mechanisms, such as monitoring network traffic and scanning network cards in promiscuous mode, could be used [115].

   However, in practice, and due to the timing limitations on SMV messages, they remain unencrypted. In fact, IEC 62351-6, which covers security for SMV, states that for applications using SMV and requiring 3 ms response times, multicast configurations, and low Central Processing Unit (CPU) overhead, encryption is not recommended [117]. Therefore, eavesdropping on the process bus is a cyber-threat, which leverages the capabilities of attackers.

3. Replay: in a replay attack, an attacker sniffs network packets and replays them at a later time. Since each SMV message is associated with a sample counter, and given the fact that Layer 2 messages cannot be blocked, a replay of SMV messages could be detected by monitoring the process bus for the messages with repeated sample counters or out-of-sequence sample counters.

4. Man-in-the-Middle (MITM): a MITM attack occurs when an attacker forces traffic between two communicating nodes to go through the attacker's machine before reaching the intended recipient. This is possible through ARP Poisoning. ARP is a communication protocol used to convert IP addresses into MAC addresses [42]. In ARP Poisoning, the attacker sends an ARP Reply to fake the

IP and MAC addresses mapping on the network. As such, the IP address of the attacker is, now, associated with an incorrect MAC address. This allows the attacker to intercept the messages exchanged between the communicating nodes. The attacker can, then, perform a MITM attack by manipulating the intercepted data.

Being broadcast Layer 2 messages, a MITM is not possible on the SMV protocol, except when a control network in physically compromised by a person from the inside.

5. SMV Spoofing: in this attack, an attacker publishes an SMV message, with manipulated data, to the process bus from his/her machine using the source MAC address of the original sender. This is possible because the process bus network is unencrypted. An attacker can sniff an SMV message, decode its fields, perform changes on certain fields, and publish the message back with a spoofed MAC address.

   To defend against this attack, IEC 62351 recommends the use of RSA-based signatures to authenticate SMV packets and ensure their integrity. However, RSA-based authentication is unsuitable for the SMV protocol, which requires a 3 ms response time, since it is computationally expensive [9][11].

   The use of a Hash-based Message Authentication Code is also recommended by IEC 62351, and it was found to have an average latency in the range of few hundred microseconds. As reported in [117], this latency significantly increases with the message size. The latency studies in the literature were performed on processors with very high computational powers, such as Intel core i7 processors and Field-Programmable Gate Arrays (FPGAs) [117][118]. This kind of processing power comes at the expense of high monetary costs and is not commonly available on IEDs and MUs, especially those devices currently available in the field.

Also, in multicast applications, such as SMV, MAC authentication does not provide non-repudiation [117]. That is, any subscriber can create a message and a MAC and send it as if they were the publisher. This of course, requires the attacker to know the secret key. Several attacks on the subject matter were reported in the literature, including length extension attacks, internal state attacks, key recovery attacks, and forgery attacks [119][120][121].

If an attacker successfully sends a spoofed SMV message, which is verified by the subscriber, the attacker cannot stop the original publisher from sending the original SMV message. Therefore, the subscriber will have two SMV messages with either a repeated sample counter, or an out-of-sequence one. The problem that arises here is that the subscriber does not know which packet is valid and which packet is spoofed. The same logic applies to the replay attack. The IED will not know which message stream to use for the control operation to continue.

It can be concluded from the previous discussion that the structure of the SMV protocol and the associated recommendations of the IEC 62351-6 make the SMV process bus resilient to several types cyber-attacks. However, devoting the necessary time and effort, savvy attackers can still perform SMV spoofing attacks to disrupt the operation of the power system.

As mentioned earlier, issuing a spoofed SMV will result in a repeated or out-of-sequence sample counter. However, the subscribing device will not be able to distinguish which message holds the true measurement value. Accordingly, the next sections study the feasibility of using neural network forecasters to identify spoofed and legitimate messages.

5.4 System Description and Generation of Training Data

Figure 5.2 shows the simulation microgrid model used to generate the proper training data for the fake data detection system. In fact, the same microgrid model was used in Chapter 3, Section 3.4 for the online lost packet forecasting algorithm.

Similarly, the same training data set, as in Section 3.4, was utilized to train the developed neural network. A small recap on the selection of the training data is presented here for the convenience of the reader.

Two major cases were studied:

1. The grid-connected mode of operation.
2. The islanded mode of operation.

For each of the two cases mentioned above, and on each transmission line, five types of faults were applied, namely:

1. Single-line-to-ground (A-G).
2. Line-to-line (B-C).
3. Double-line-to-ground (B-C-G).
4. Three-phase (A-B-C).
5. Three-phase-to-ground (A-B-C-G) faults.

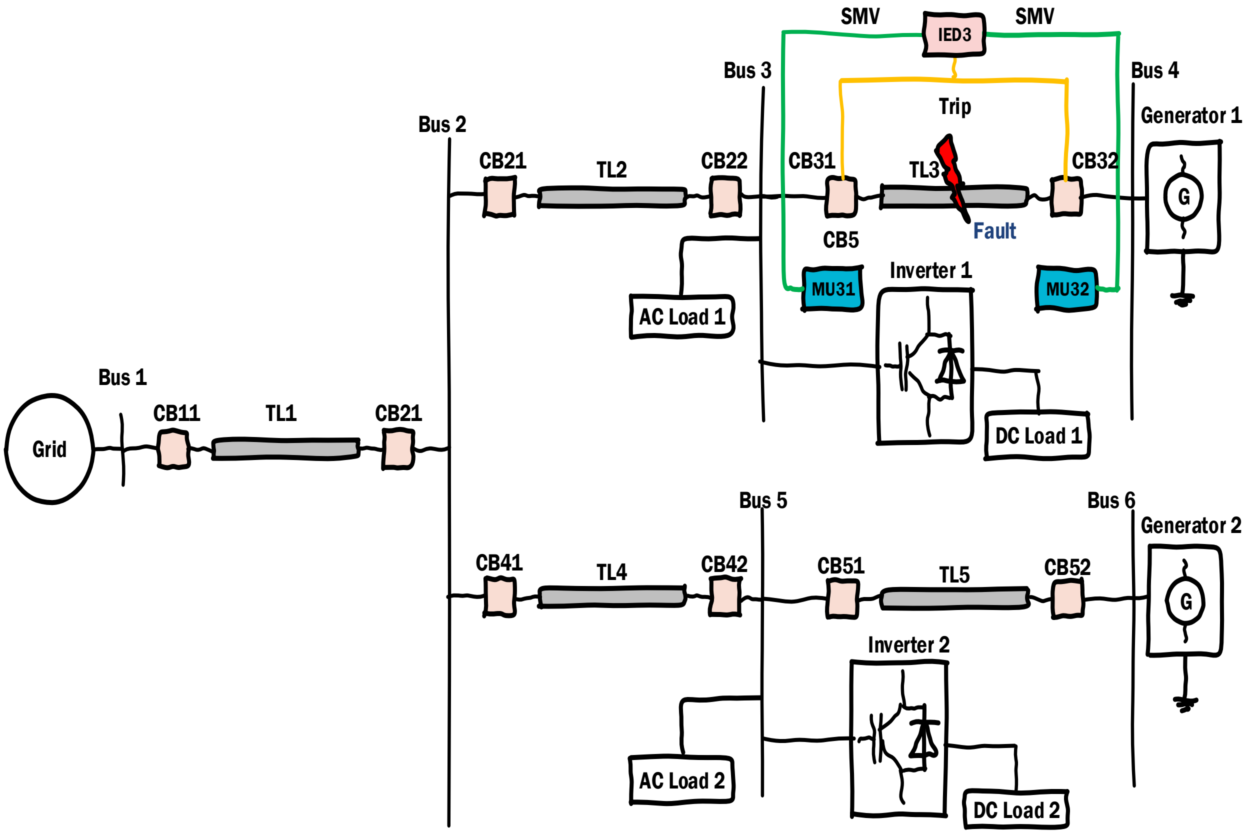


Figure 5.2: Block Diagram of the Physical System.

Additionally, each type of fault was applied on the beginning (10%), middle (50%), and end (90%) of each transmission line.

For all the cases mentioned above, current and voltage measurements from the merging units on transmission line TL3 were recorded, as this is where the proposed algorithm was implemented.

In order to make sure not to over train the network, only samples that represent the shift from normal operation to the unstable fault operation are utilized for training. Also, one AC cycle after the fault clearance was used for training.

5.5 The False Data Detection Algorithm

As shown in Figure 5.2, IED3 subscribes to SMV messages published by MU31 and MU32. In a simple differential protection scheme, IED3 calculates the difference between the received current values and compares the results to a predefined threshold. If the difference is greater than the threshold, then IED3 sends a trip signal in the form of a GOOSE message in order to isolate the fault. Therefore, the proposed IDS is implemented on IED3. The developed MUs and IED are detailed Section 6.4.

Here, the forecasting problem depends on a single variable, which is time. Therefore, a time series approach was used in training the developed neural network. To capture all the features in the current waveforms and decrease the forecasting error, a window of 20 samples was selected. These 20 samples cover a quarter AC cycle for a 60 Hz system.

The developed NN has an input layer with twenty neurons to accommodate the 20 previous samples, a hidden layer with 10 neurons, and an output layer with 1 neuron that produces the forecasted sample.

If an anomaly in the sequence number field is detected, the IED passes the two samples through its first thread, which is the Neural Network Forecaster (NN-F). The NN-F utilizes 20 previous samples to forecast the value of the incoming measurement. The NN-F will compare the error between the received samples and the forecasted value. Only the received SMV, which has an error less than a specified threshold, will be marked as benign and is passed to the control logic. This process is depicted in

Figure 5.3. It is worth noting that the Normalized Mean Root Square Error (NMSRE) is used as opposed to the Mean Square Error Error (MSE) because of the presence of small measured values.

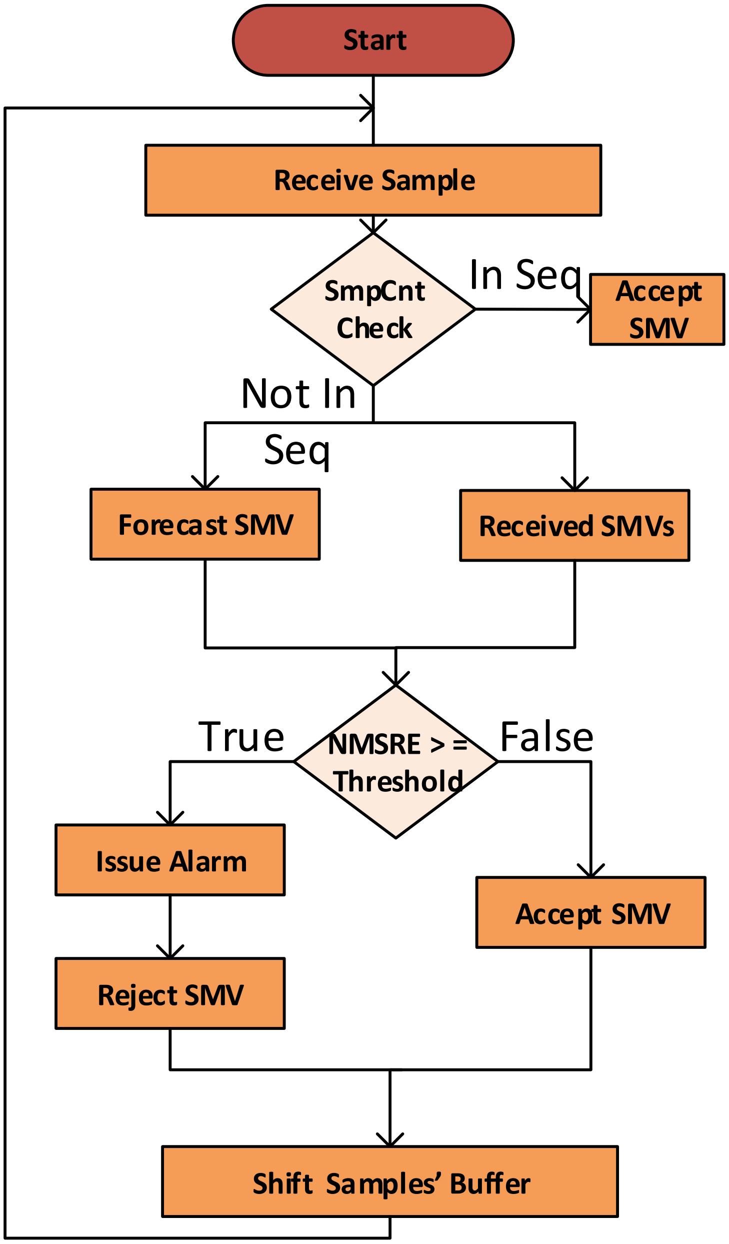


Figure 5.3: Online IDS and Forecasting Algorithm.

In such situations, the MSE would yield very high error percentages, and thus, the NMSRE is a better statistical error indicator. The proposed system is intended for data manipulation attacks, since it relies on measuring the distance between forecasted and received samples.

5.6 The Developed MUs and IEDs

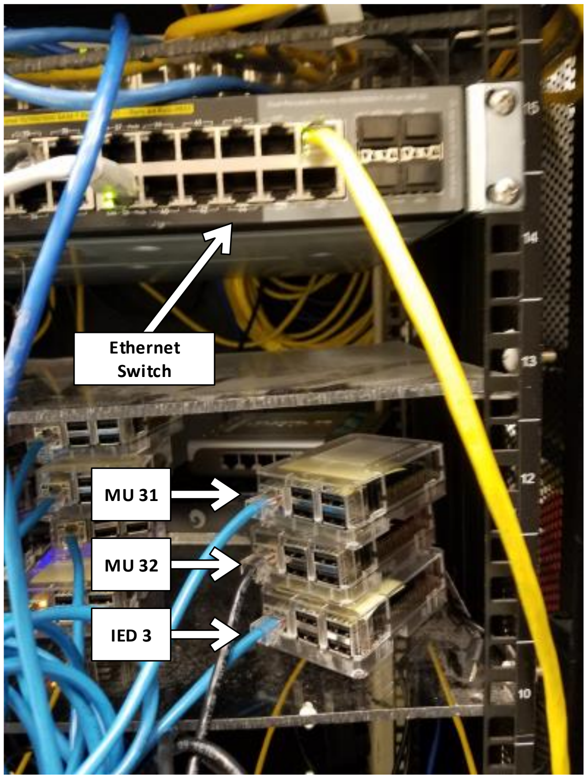


Figure 5.4: Process Bus Ethernet Network.

In order to assess the performance of the developed spoofed sample detection system, two merging units that publish the 3-phase current measurements at both ends of TL3, and one IED, which subscribes to those messages and issues a trip signal, were developed. The SMV publisher and SMV subscriber were coded in C language, utilizing the open source IEC 61850 library, libIEC61850 [75]. As per the IEC 61850 standard, the collected measurements were published with a 4,800 Hz frequency. The code of the SMV subscriber, which was implemented on IED3, was modified to include the proposed forecasting and IDS algorithms. The developed codes were downloaded on 3 Odroid C2 single board computers (SBCs) running a Linux kernel. The communication took place over a dedicated network switch, as shown in Figure 5.4.

5.7 Results and Discussion

Several experiments were performed on the proposed forecasting system and are reported in this section. First, the forecasting accuracy of the NN under different conditions was tested. Second, fake measurements were randomly injected in the collected. A log file with all the fake data detection alarms was collected and analyzed from the subscribing IED performing the differential protection application previously described.

5.7.1 Performance of the Neural Network

Test cases with 43 AC cycles each, covering fault and normal operation periods, were performed. The NN was subjected to data for all types and locations of faults and the forecasting errors were recorded in Figure 5.5. Analyzing all the forecasted samples revealed that 99% of the data have a forecasting error less than 1.5%. Two random test

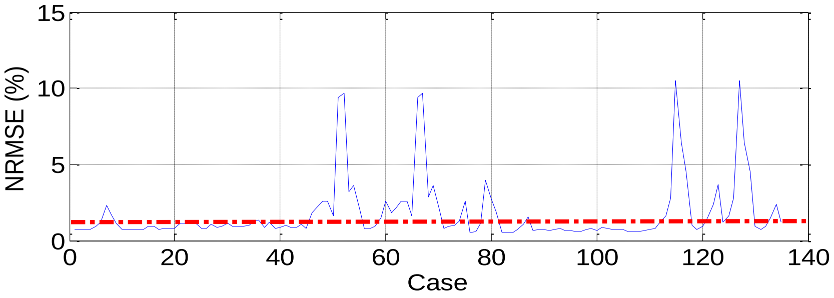


Figure 5.5: Maximum Normalized Root Mean Square Error for all Fault Types and Locations.

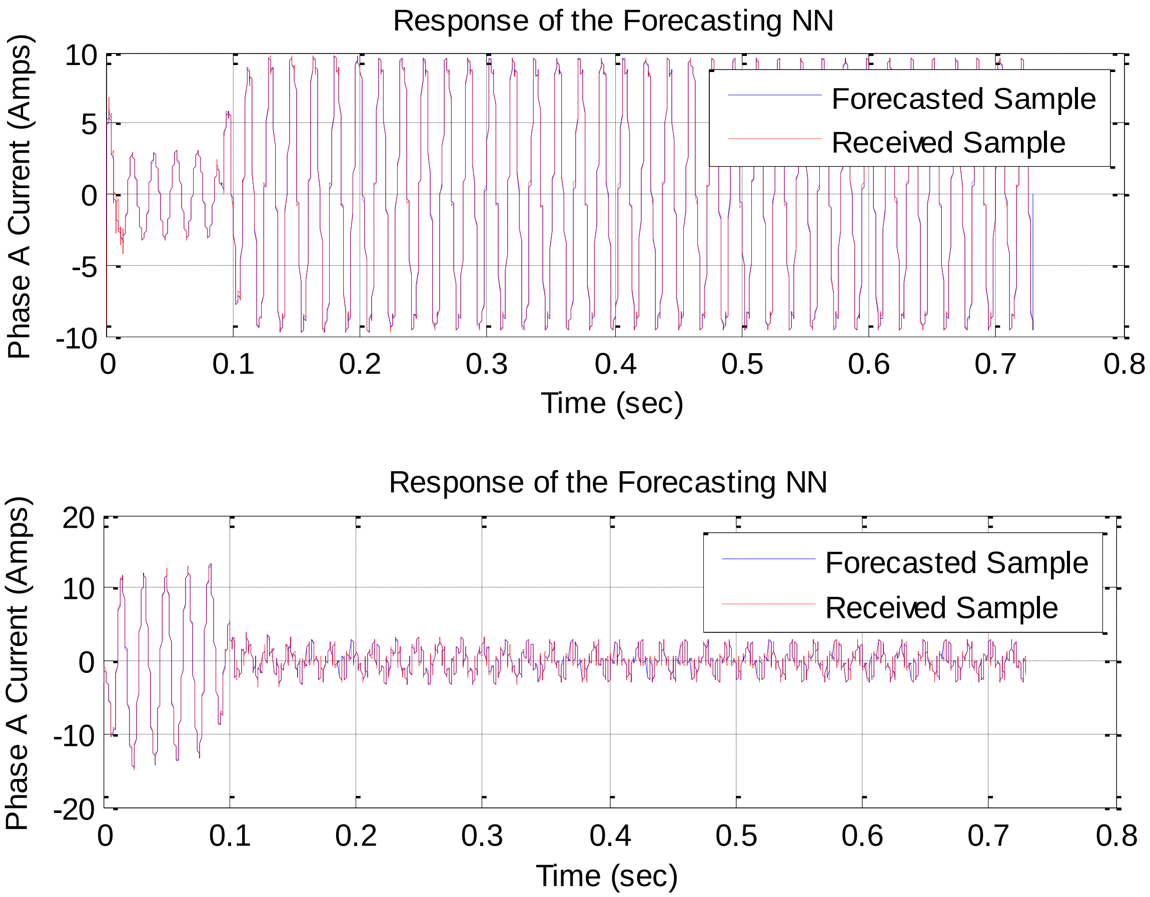


Figure 5.6: Comparison of Forecasted and Actual Current Measurements.

cases were selected, representing a line-to-line fault at the end of TL4 and another 3-phase-to-ground fault at the middle of TL3. The forecasted and actual results of these cases were plotted in Figure 5.6. The results showed very low error rates, which indicates a robust NN with high forecasting accuracy. It is worth noting from this graph that the forecasting accuracy of the samples corresponding to the fault period, which is a critical period in protection applications, is high.

In fact, the proposed cyber security solution depends on the ability of the artificially intelligent module (the neural network in our case) to learn the physical properties, i.e.

features, of the system. There is extensive work in the literature that relies on artificial intelligence techniques in intrusion detection [81][122][123]. However, such techniques do not incorporate the physical characteristics of the system in intrusion detection. They are purely reliant on cyber rules, such as detecting high bandwidth consumption. There are few recent efforts to incorporate the physical properties in securing power systems [29][124]. However, these are based on a set of predefined mathematical equations against which an incoming command/packet is compared. The complexity and the number of these equations will add significant processing time, which needs to be studied given the low processing power of current IEDs. The work in this chapter introduces the incorporation of the physical properties' dimensionality in intrusion detection, utilizing artificial intelligent modules. The use of artificial intelligence for characterizing complex systems has been justified in the scientific and engineering literature. In addition to learning the features of the power network, the fast response of properly trained neural networks makes them excellent candidates in time-critical applications, such as protection. The overall transmission and processing time of an SMV packet was measured to be less than the 4 ms constraint set by the IEC 61850 standard.

5.7.2 Performance of the Spoofed Data Detection Algorithm.

Fake packets with high current values were intensively injected in the system in order to trick the IED into sending trip messages. During the normal operating conditions, the system showed that it can successfully detect fake measurements with high current values. On the other hand, during fault conditions, the proposed system was able to detect low current measurements successfully. The performance of the system is shown for several normal and fault operating conditions in Figure 5.7. As can be seen from Figure 5.7, the blue curve represents the fake data, while the red curve shows the real and the forecasted data. Therefore, the proposed system did enhance the resiliency of the protection scheme against fake current measurements. Here, it is important to note that the choice of the decision threshold between real and fake measurements is of paramount importance and requires fine tuning based on system observations. In the performed case studies, a 2% threshold level was selected and resulted in 273 false positive measurements from a pool of 48,343 measurements, representing 0.56% of all the published data, as shown in Table 5.1.

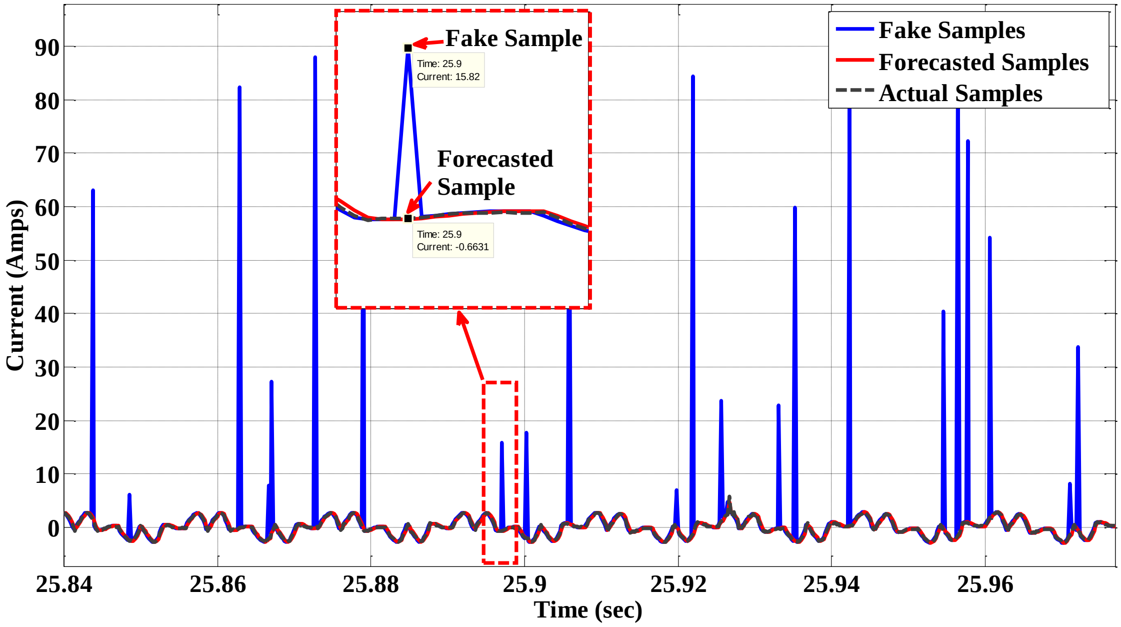


Figure 5.7: Performance of the Fake Data Detection System.

Table 5.1: Performance of Neural Network IDS

Total Number of Samples	False Positives	False Positive (%)
48,434	273	0.56

Here, one would ponder the response of the forecaster against fast electric transients (EFT). In fact, fast transients in power systems can be divided in general into fast transients generated by natural causes (e.g. lightning) or fast transients generated through the use of equipment (e.g. de-energizing heavy duty motors, switching in and out power factor correction equipment, and opening and closing circuit breakers) [125][126]. As per the IEC 61000-4-4 specifications, which define the requirements for immunity to repetitive fast transients and the necessary test methods, an Electromagnetic Fast Transient (EFT) pulse has a minimum period of 200 microseconds at a 5 Khz burst. The waveform, as adopted from [126], is shown in Figure 5.8.

The developed neural network is intended to forecast IEC 61850 SMV. IEC 61850 – 90 – 2 specifies a publishing rate of 80 samples per AC cycle for 50-60 Hz systems, which is equal to 4.8 KHz sampling frequency. Therefore, even if the communication network is ideal, the sampling frequency set by the standard won't be able to properly capture fast transients, which typically have frequencies of 5 KHz and above (4.8 KHz < 5*2 KHz, thus failing to satisfy Nyquist's theorem). Also, since the sampling frequency is 4.8 KHz, the merging unit is usually accompanied by a low pass filter with sampling frequency fsampling/2 = 2.4 KHz. On the other hand, IEC 61000-4-4 defines various test levels for EFT immunity. Such transients have their own protection schemes and devices (surge arrestors, proper system grounding and insulation …), which are different than the protection functions played by the IEDs and MUs used for dealing with regular system faults.

Finally, one does not need to forget here that the number of consecutively lost or manipulated samples plays a critical role in the success of such algorithms. This is due to the fact that every time a packet is dropped or manipulated, its estimated value is used for forecasting the next sample. Having several consecutive lost or manipulated packets will lead to the accumulation of the forecasting error.

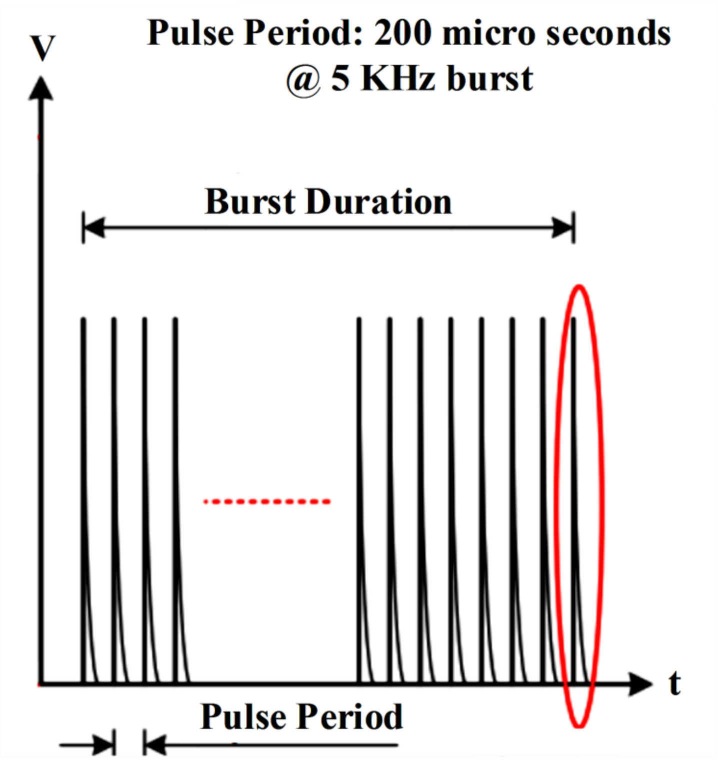


Figure 5.8: Fast Electrical Transient Pulse per IEC 61000-4-4 [126].

A large window size could thus be a possible solution to allow the system to compensate for a good number of lost or manipulated samples. However, this will lead to a tradeoff between the window size used for forecasting and the computation time. Given the strict 4ms time constraint on SMV messages, a window of 20 previous samples covering a quarter AC cycle has been used in this study.

5.8 Conclusion

This chapter presented an online fake data detection and dropped packet forecasting system for IEC 61850 SMV messages. The system utilizes a combined NN – time series forecasting module to detect fake measurement data injection attacks. The proposed system proved its effectiveness in enhancing the resiliency of modern protection schemes against false data injection attacks. The system was implemented on a real IEC 61850 Ethernet network and was able to meet the 4ms time constraint set on SMV messages. Finally, it is important to emphasize the ability of the system in detecting fake messages during critical events, such as peak fault currents.

Chapter 6 Developing a Targeted Attack for Enhancing Resiliency of Intelligent Intrusion Detection Modules in Energy Cyber Physical Systems

Secure high-speed communication is required to ensure proper operation of complex power grid systems and prevent malicious tampering activities. In the previous chapter, artificial neural networks with temporal dependency were introduced for spoofed data identification and mitigation for broadcasted IEC 61850 SMV messages. The fast responses of such intelligent modules in intrusion detection make them suitable for time-critical applications, such as protection. However, care must be taken in selecting the appropriate intelligence model and decision criteria. As such, this chapter presents a customizable malware script to sniff and manipulate SMV messages and demonstrates the ability of the malware to trigger false positives in the response of the neural network. The developed malware is intended to be a vaccine to harden the intrusion detection system against data manipulation attacks by enhancing the neural network's ability to learn and adapt to these attacks, as will be seen in Chapter 7.

6.1 Introduction

The power industry is increasingly relying on robust communication infrastructures to transmit and analyze transmission and distribution measurements in adaptive protection and control schemes. As the reliance on these technologies increases, so does the threat posed by attackers. If not properly secured, these communication-enabled technologies will be vulnerable and pose a potential to cripple the reliability and economy of the grid. The main enabler of automated adaptive protection schemes is the IEC 61850 data modeling standard. One of the main security challenges faced by modern IEC 61850-based protection techniques is data manipulation attacks within the process bus. According to the IEC 61850 model, the process bus is the medium where current and voltage measurements and event triggered commands are communicated as SMV and GOOSE messages, respectively, within a local area network. The core vulnerability is in the fact that these time-critical messages are broadcasted over the LAN unencrypted. Therefore, in the event of a network breach or the presence of a malicious insider within the network, data manipulation of such messages is an easy task, and thus, the opening and closing of circuit breakers is possible via injecting fake current and voltage values [41][42].

In fact, the catastrophic impacts of data manipulation and false data injection attacks on the reliable operation of the power system have been widely researched in recent literature. Authors in [31] showed how false measurements feedback to automatic generation control could impact the physical system stability by causing sudden declines in the system frequency, which in its turn causes unwanted load shedding schemes. The work in [37] demonstrates two realistic false data attack scenarios, in which attackers introduce arbitrary errors to state variables to achieve a false state estimation of the power system. In the study conducted in [127], the authors quantitatively analyze the damage caused by false data injection with regards to the power system operation and security.

On the other hand, there are several works in the literature that focus on defense strategies to minimize service loss through several defense mechanisms. In [128], the authors recognize the potential impacts of data injection on the process bus and proposed an agreement algorithm to detect, locate, and prevent malicious data from being accepted by the IEDs and protection devices. In [129], the authors presented an overview of vulnerabilities in the IEC 61850 protocol suite and discussed a method of GOOSE message modification using a malware script to sniff, manipulate, and inject control messages into the process bus in detail. In [130], the authors presented an intrusion detection system that is capable of filtering malicious messages based on predefined rules and known malicious signatures. In [18], an intrusion detection system based on GOOSE and SMV rule violation indicators was presented. This system, similar to other rule based systems, will not be able to detect unknown attacks that are not defined in their rule base.

Even though the IDS will filter out uncoordinated attacks, it is still not robust enough to secure IEC 61850 automation processes. These solutions are also network-based and are themselves vulnerable to data manipulation from a savvy attacker. Attackers with sufficient information can spoof different data fields to obfuscate themselves from this IDS.

To accomplish the goal of robust security, machine learning techniques have been introduced to leverage the intelligence of rule-based IDS. Machine-learning systems use the approach of anomaly detection, in which a model is defined and positioned as normal and if an outlier is detected, it is considered to be an anomaly [131]. This need is recognized by the authors in [99], and to that end, they incorporated a protection algorithm using a trained coupled time-series NN that predicts incoming current measurement based on the microgrid's recent operating history. Then, an intrusion is detected and announced if a real measured value deviates from the predicted one. The potential of this solution was tested on data collected from a simulated microgrid and the results in [99] are promising for rapid verification of data integrity. However, the accuracy demonstrated was cultivated in a controlled testing environment using high current values for the malicious data injection. In a case where the attack is designed for the target system, the accuracy of the IDS deviates from the results in [99]. This issue must be addressed in the NN to ensure reliability in real-world attack scenarios.

Accordingly, the work in this chapter is an extension of the author's work in the previous chapter, where a configurable malware is developed to be used as a tool to examine and quantify the reduction in accuracy experienced by the NN in targeted attack scenarios. The purpose of this tool is to be used as a vaccine to harden the IDS against smart attacks by fine tuning its decision criteria and model parameters.

6.2 System Description

The targeted attack was performed on the co-simulation setup described in Chapter 5, Ssection 5.2 and is shown in Figure 6.1 here.

Figure 6.1 also shows the hardware setup built to test the IDS and implement the developed malware script over a real IEC 61850 process bus architecture. The recorded current values of both transmission lines from the simulated microgrid model were recorded in a database of coefficient files and fed as inputs to the merging units. The measuring units then publish these measurements as IEC 61850 SMV packets. IED3 is programmed to subscribe to these messages and processes them through the IDS NN, which it hosts. The firmware for MU31, MU32, and IED3, along with the IDS, were coded in C and downloaded on three different Odroid C2 devices running a Linux kernel.

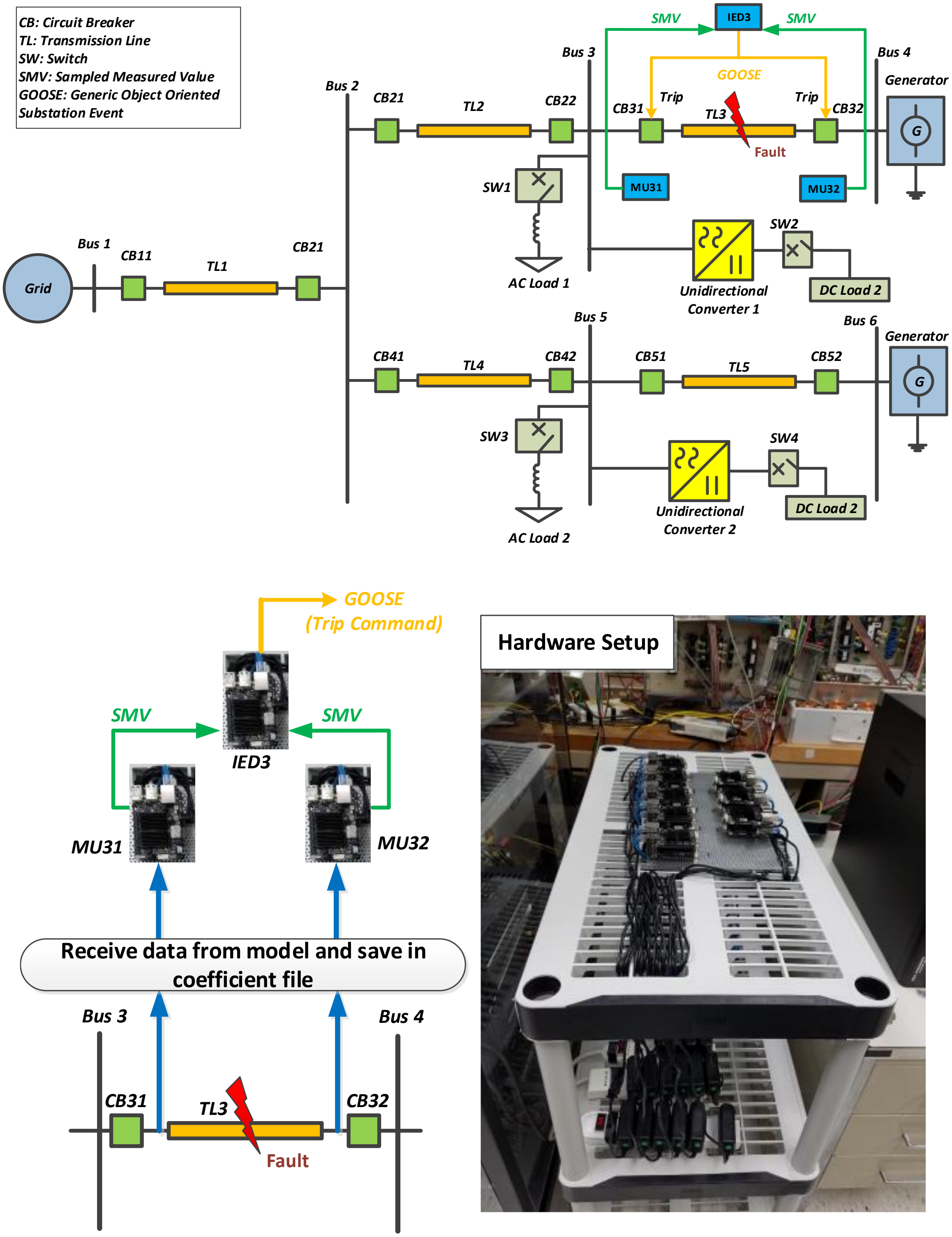


Figure 6.1: Software and Hardware Setup.

6.3 Malware Development

6.3.1 SMV Message Structure

<table>
<tr><td colspan="2">Destination Address</td><td colspan="2">Source MAC Address</td><td colspan="2">Priority Tagging/ VLAN ID</td></tr>
<tr><td colspan="2">Ethertype (88BA)</td><td colspan="2">APPID</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved 1</td><td colspan="2">Reserved 2</td><td colspan="2">APDU</td></tr>
<tr><td>Tag</td><td>Length</td><td>noASDU</td><td>Tag</td><td>Length</td><td>svID</td></tr>
<tr><td>Tag</td><td>Length</td><td>SmpCnt</td><td>Tag</td><td>Length</td><td>ConfRev</td></tr>
<tr><td>Tag</td><td>Length</td><td>SmpSynch</td><td>Tag</td><td>Length</td><td>Sample 1</td></tr>
<tr><td>Tag</td><td>Length</td><td>Sample 2</td><td>Tag</td><td>Length</td><td>Sample 2</td></tr>
<tr><td colspan="3">••••• •••••</td><td>Tag</td><td>Length</td><td>Sample N</td></tr>
</table>

Figure 6.2: SMV Datagram Structure.

In order to better understand the malware development procedure shown in Figure 6.3, the structure of the SMV message will be explained first. An SMV datagram follows a modified ASN.1 BER Tag/Length pair encoding scheme [88]. The Tag field represents the type of information, which is represented in the following SMV frame.

As shown in Figure 6.2, the SMV datagram starts with the Destination MAC Address, which is a multicast address reserved for IEC 61850 applications always starting with 01-0C-CD and is followed by the source MAC address. An SMV message has an IEEE 802.1Q VLAN ID and a unique Ethernet type (88-BA). The APPID field is a 4 octet field, which the subscribing IEDs use to identify messages they are subscribing to. The

Length field represents the length of the overall SMV datagram and is followed by two reserved fields left out by the standard for future use.

The second layer of an SMV message is the Application Protocol Data Unit (APDU), which consists of one or more Application Service Data Units (ASDU). The number of ASDUs is in the noASDU field [132]. Each ASDU then contains the following subfields [8]:

- svID: unique identifier for each SMV message.
- SmpCnt: incrementing counter with each published SMV.
- ConfRev: counter for configuration changes.
- SmpSynch: A boolean value indicating synchronization with a clock signal.
- SeqData: List of data values related to the data set definition.

6.3.2 Malware Development

In order to properly inoculate the NN against smart attacks, a malware script for targeted attacks against the process bus was developed. The malware was written in Python in conjunction with network sniffing and packet crafting libraries from Scapy. As mentioned earlier, SMV messages are broadcasted over the LAN and are unencrypted. Once the malware is run, it starts to passively sniff packets from the network. Next, it filters those messages looking for the destination MAC address assigned for SMV messages, the Ethertype (88xBA), and the designated APPID identifier.

Once the SMV packet has been identified, it is converted into a list of hexadecimal pair strings that will make searching the packet more convenient. After the

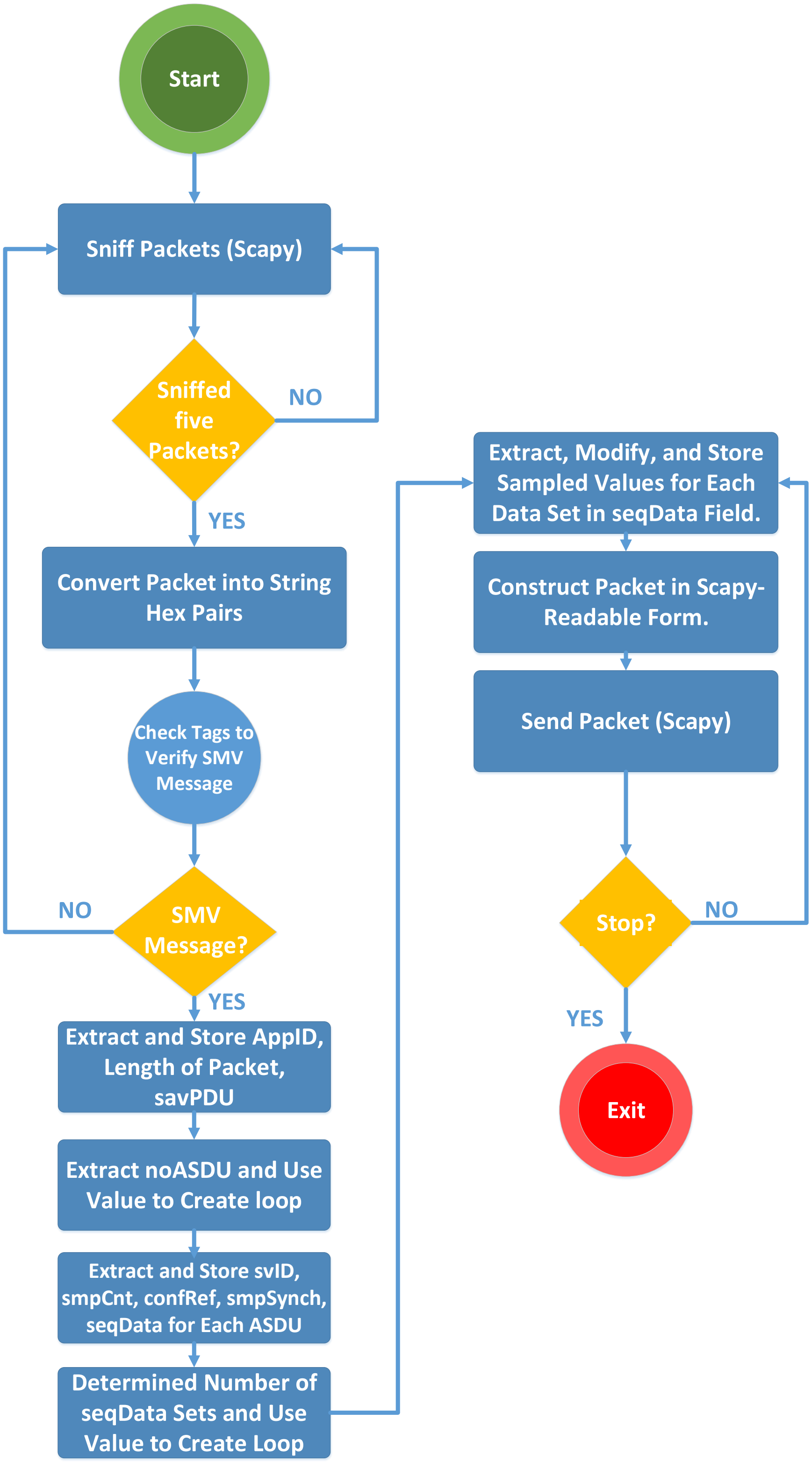

Start
Sniff Packets (Scapy)
Sniffed five Packets?
NO
YES
Convert Packet into String Hex Pairs
Check Tags to Verify SMV Message
NO
SMV Message?
YES
Extract and Store AppID, Length of Packet, savPDU
Extract noASDU and Use Value to Create loop
Extract and Store svID, smpCnt, confRef, smpSynch, seqData for Each ASDU
Determined Number of seqData Sets and Use Value to Create Loop
Extract, Modify, and Store Sampled Values for Each Data Set in seqData Field.
Construct Packet in Scapy-Readable Form.
Send Packet (Scapy)
Stop?
NO
YES
Exit

Figure 6.3: Malware Development Process.

packet has been properly decoded and stored as hex pairs, the malware can be utilized to spoof any desired field of the captured SMV packet. However, the developed malware will spoof all the fields (source MAC address, destination MAC address, APPID …) and change only the seqData field, which holds the value of the current measurement. This is intended to trick the IED into recognizing this packet as being sent from its original merging unit.

In order to manipulate the current measurements, a class called "ASDU" was created and used to store the data collected from the packet, and to conveniently build the new packet before it is injected into the process bus. A major obstacle in the design of this script is to write it in a way that will be able to process any number of ASDUs and seqData for the SMV packet. These values are not static and can be changed by the network administrators based on how often they want the collected measurements to be sent. In order to address this issue, we implement our datafield search function, which is called in an incremental manner, to make sure the packet will be of the appropriate length and that it modifies all ASDUs and seqData fields detected by the script.

The first field searched for is the noASDU field, as this will describe the number of ASDUs present in the packet. Once this value is determined, a loop is created that iterates through each ASDU and records the information into the ASDU class. An array of pointers to ASDU objects is used to easily navigate through the collected data, as this can become quite tedious when an extremely large amount of ASDUs are introduced. For each ASDU, the datafield type, length, and value of each field is stored into its own array of hex pairs for reasons that become apparent when the script re-crafts the packet with the modified data. As each field is decoded, it is stored into the corresponding ASDU class member. Once the entire packet has been decoded, the final step is to modify the existing data and send the new packet.

It is important to note that the attacker must configure which fields they wish to modify and either manually change the value of the data, create an arithmetic function to manipulate the data, or read in predetermined values from a file. Depending on the method chosen, the script will begin to modify the data based on the attacker’s configuration, and overwrite the original ASDU class members with these new values. Once the modification is complete, the packet is rebuilt in the correct order and the spoofed packet with false data is broadcasted into the LAN. The flowchart in Figure 6.3 presents the malware algorithm for visualization of the process.

6.4 Results and Discussion

Once this tool had been developed, spoofed messages were injected into the process bus to test the response of the IDS. The malware is assumed to be running on a computer device connected to the LAN of the microgrid under study. In order to test the efficacy of the NN for intrusion detection, two data modification methods were used and the results for each were observed.

First, the values of the currents were recorded and analyzed for the first few AC cycles. It was noticed that the amplitude of the recorded current waveform was around 3 A. As appreciated from [99], injecting values far above 3 A triggered an intrusion alarm. However, as explained earlier, the NN was trained to recognize the current values for normal and fault conditions. Therefore, in the first attack, fake packets were injected with alternating 3.5 A and -3.5 A at a fixed rate to signal a fake beginning of a fault situation. When configuring the malware script, the two values will be written to an input file and then the malware changes the data fields of the sniffed packets by alternating between these values. The following algorithm was used to inject the fake packets and is written generically to be applicable for many systems:

Table 6.1. Algorithm to Inject Fake Packets.

```
f = fopen(“FakeSV.txt”, r+)
Generate Fake Values (inject +/- 3.5)
f.write(Fake Values)
SetSampleRate(pps)
while(!f.EndofFile){
f.readline(n)
sendpacket(rate, mod_pkt)
n += 1
}
```

It is to be noted that the user of the malware tool is free to experiment with other algorithms or arithmetic functions for generating the fake values. The user can also experiment with the packet sending rate for a completely configurable data injection tool. The received and the forecasted samples from the IED were recorded in a log file and are plotted in Figure 6.4. The blue waveform represents the SMV packets received by the IED, whereas the red waveform represents the forecasted values by the NN. The actual measurements (blue) were broadcasted at a rate of 4,800 Hz in accordance to the recommendations of the IEC 61850 standard, whereas the malware tool was transmitting

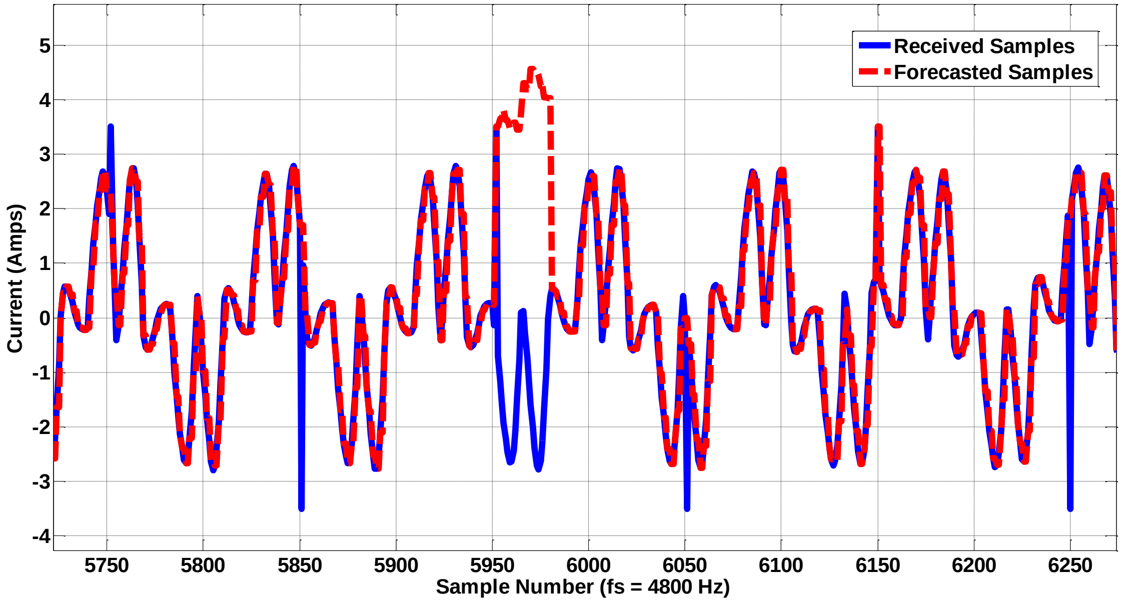


Figure 6.4: Incorrect Forecasting Triggered by Malicious Alternating Sampled Values.

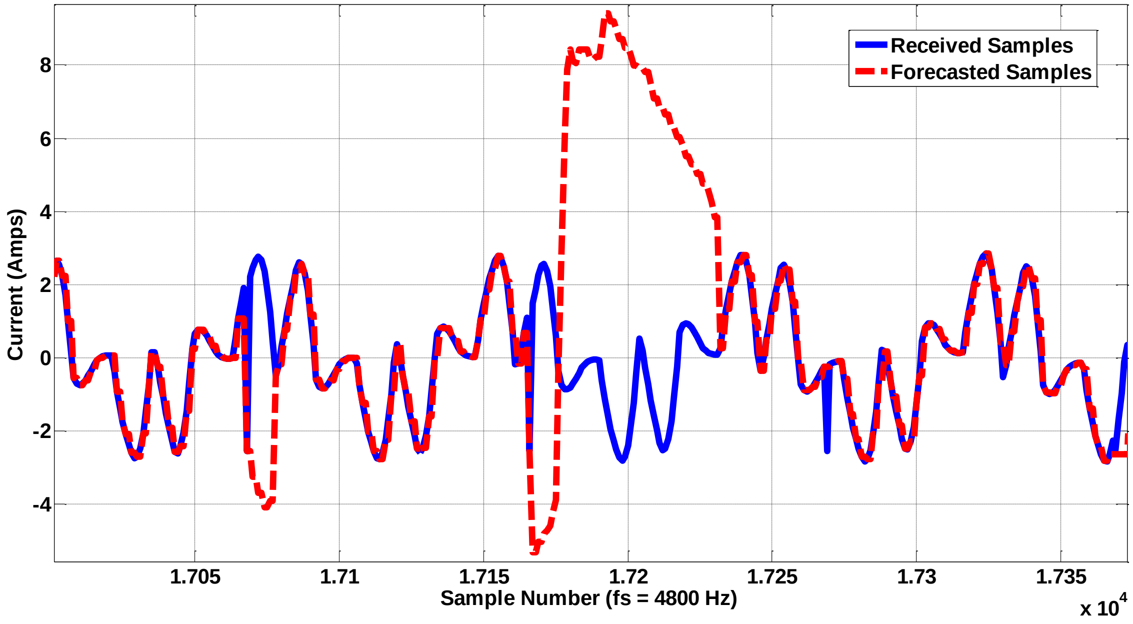
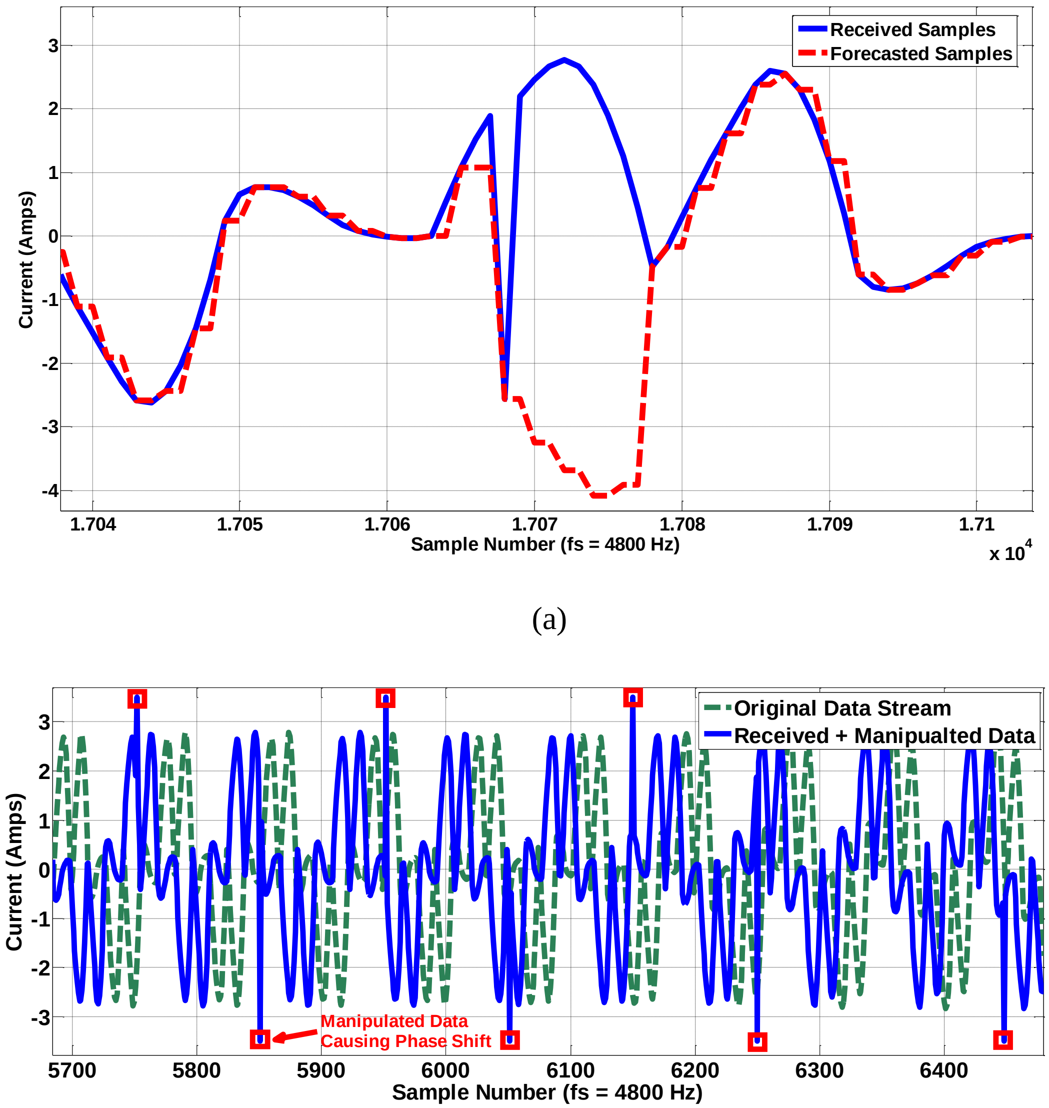


Figure 6.5: Results for Injected Measurements Simulating a Real AC Waveform Measurements.



(a)

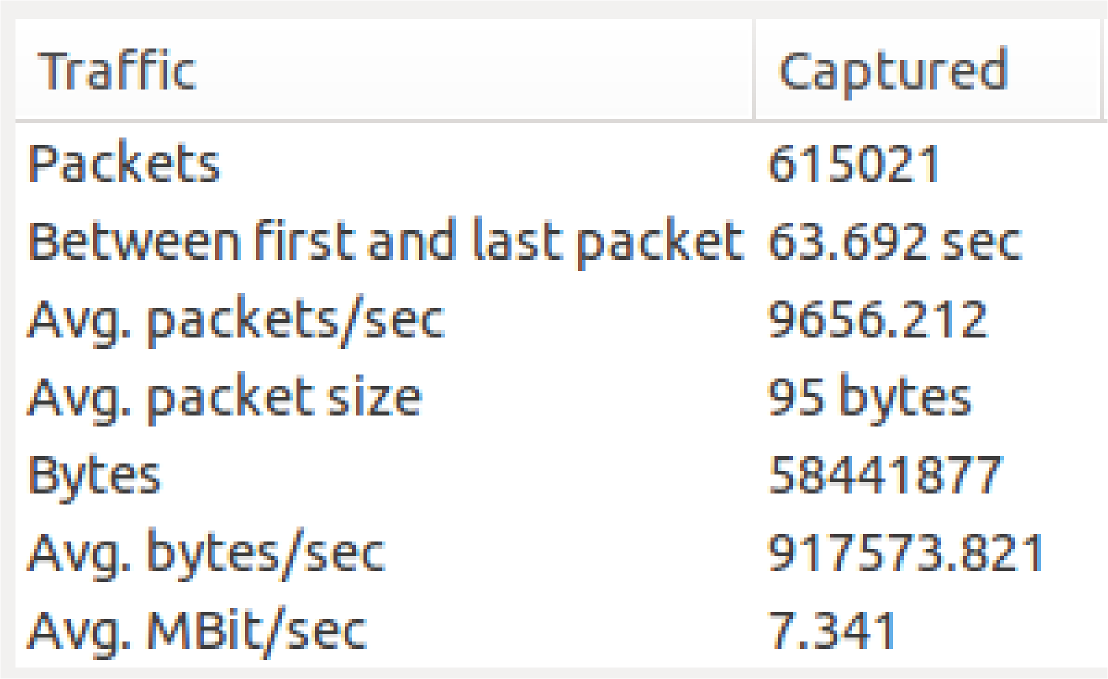
(b)

Figure 6.6: Sample Shifting.

its fake data at a rate of 48 Hz (1 fake packet for every 100 true packets). A stream of 9,400 real current samples was published and in the meantime, the malware published 80 manipulated packets. Out of the 80 attempts, 11 attempts (13.75%) were successful in attacking the NN and thus passing into the IED protection logic false values reaching a maximum of around 4.5 A. This value is 1.5 times the rated value and was enough to issue a false positive and trigger a trip command.

The next data injection method is one that showed more promising results in successfully corrupting a network. This attack continuously replays values from a recoded log file that simulates the same waveform that the NN had been trained on for forecasting a fault. This method will increase the period of the injected values; however, the results show that the NN was not prepared to handle this kind of targeted attack. In Figure 6.5, it is observed that the NN incorrectly forecasted the sampled values in two different locations within a window of 200 samples between sample 17,050 and sample 17,250. Any one of these incorrectly forecasted paths would trigger a false positive and cause service interruption. If the injected data results in a false negative for more than three AC cycles, permanent damage can be inflicted upon the power grid's equipment.

It is known that Layer 2 broadcast messages, such as SMV messages, cannot be blocked. An important observation from the results of these attacks is that when the IED received the spoofed packet, it processed it along with true message, thus introducing a misleading entry in the buffer used for forecasting. This is shown in packet number 17,166 in Figure 6.6 (a). As can be seen from Figure 6.6 (a), when the IED received the fake packet, the original data stream of legitimate SMV messages was shifted by 1 sample. This shift was accumulated as more fake SMV packets were injected. As can be seen in Figure 6.6 (b), the shift accumulation led to the IED receiving SMV messages that are totally out of phase from the original data stream. Here again, the utilization of the developed malware exploited another vulnerability in the studied process bus, which needs to be addressed.



Traffic	Captured
Packets	615021
Between first and last packet	63.692 sec
Avg. packets/sec	9656.212
Avg. packet size	95 bytes
Bytes	58441877
Avg. bytes/sec	917573.821
Avg. MBit/sec	7.341

Figure 6.7: Malware Data Injection Statistics.

It is also important to mention that the malware provides the user a means to control the rate of sending fake packets using the Scapy library function “pps” or packets per second. A test has been conducted to test the maximum speed at which the malware could broadcast fake SMV messages. Figure 6.7 shows a summary of the statistics of the conducted experiment. For around one minute, the malware was sending at an average speed of 9,656 Hz, which is almost double that set by IEC 61850 (4,800 Hz). These statistics are affected by the message length and the specifications of the machine hosting the malware. In this test, the malware was run on a Linux machine with Intel i7 processor rated at 3.50 GHz with an average packet size of 95 bytes.

The previous results showed the effectiveness of the developed malware in testing the efficacy of the proposed predictive IDS. Using this developed malware as a training tool, not only can this NN be trained to detect several targeted attacks, but it can also be used to fine tune event thresholds to prevent service interruption during targeted attacks. Moreover, the developed malware tool can be used to benchmark other NN-related intrusion detection algorithms present in the literature. As mentioned earlier, the developed tool is configurable. Therefore, the user can control the type of attack by manually changing the value of the measurements, providing the tool with an arithmetic function to simulate certain scenarios or replay given values from a log file. Also, the user can adjust the rate of false data injection as desired. By these experimentations, the user can quantitatively analyze the performance of his or her IDS in terms of the ability of the tool to produce false positives and false negatives.

6.5 Conclusion

This chapter developed the design and implementation of a targeted data injection attack that will simulate real AC waveforms in an attempt to interrupt power flow in a compromised power network. The targeted malware was developed in a configurable manner to allow the attacker to choose different methods of data injection. A predictive NN-IDS was then tested against the targeted malware to observe how it would handle the smart attack. The results of the experiment demonstrated that the NN is yet to be properly trained or fine-tuned enough to detect malicious measurements. Fake measurements could lead IEDs to issue trip signals that would result in service interruptions in a real system. The NN showed resistance to less sophisticated data injection methods; however, it is still vulnerable to malicious data injection methods. Also, the NN demonstrated very little resistance to the simulated AC waveform. In order to better prepare the NN, the developed malware will also be used as a training tool to identify attack signatures, and allow the user to tune the event thresholds that would result in controlling messages being sent to the IEDs. The malware is configurable, easy to understand, and is simple to use to train predictive intrusion detection systems.

Chapter 7 Content-Aware Intrusion Detection for Operational Security in Microgrids

Reliable microgrid operation is becoming heavily reliant on microprocessor-based controllers and communication networks, making it prone to cyber-attacks. The work in the previous two chapters analyzed the IEC 61850 Sampled Measured Values protocol and investigated the feasibility of using neural networks to detect spoofed sampled measured values. It was shown that although such forecasters have high spoofed-data-detection accuracy, they are prone to the accumulation of forecasting error. Accordingly, in this chapter, an algorithm to detect the accumulation of the forecasting error based on lightweight statistical indicators is proposed.

We illustrated experimentally the practical relevance of the proposed security framework against fake data injection attacks on a hybrid AC/DC laboratory-scale cyber-physical microgrid. The results showed that the artificially intelligent forecaster has a 95.6% attack detection accuracy. It was also shown that Layer 2 is also capable of detecting the normal accumulation in forecasting error, which grows naturally over time, and allows operators to adjust the parameters of the forecasting module. Finally, experimental demonstrations showed that the overall detection latency of the proposed system is near real-time, in the range of 1-2 ms.

7.1 Introduction

Microgrids are becoming the building blocks of modern power systems. Lately, resilient and secure operation of microgrids has become heavily reliant on judicious cooperation between cyber and physical actors.

Recent public disclosures and research efforts showed the ability of attackers to exploit vulnerabilities in power system communication protocols and to use them in launching successful attacks that lead to catastrophic consequences [102][103]. Although most industrial communication networks are not open to the public internet, they can still be penetrated through corporate networks or personal devices of the employees with techniques, such as password cracking, backdoors, malwares, among others [30].

Here, it is important to mention that the power grid requires real-time control, which needs strict time-delay requirements on information exchange ranging from milliseconds to few seconds depending on the application. Therefore, most industrial control networks are left unencrypted and sometimes without authentication [41] This gives attackers more freedom to launch sophisticated data manipulation attacks and imposes challenging constraints on intrusion detection developers.

While other works, which were discussed in the introduction of Chapter 6, provide a tremendous amount of insight into a multi-agent based model checking intrusion detection system, none of the literature addresses the major limitations involved in intrusion detection and prevention for power systems. Thus, there is a need for a new generation of IDS with enhanced awareness to be able to understand messages' contents and have a low detection latency, which does not hinder the real-time operation requirements of grid processes. Accordingly, this chapter presents a low-latency, content-aware, and bi-layer intrusion detection system for secure operation and control of microgrids. Unlike the work presented in the literature, the first layer of the proposed IDS utilizes AI to learn the characteristics of the microgrid in different operating states, and benefits from the fast responses of the AI to decode network packets and verify the legitimacy of received measurements. If the AI module flags a certain measurement sample as suspicious, the second layer of the proposed IDS is invoked, where a collaboration between control agents and their associated merging unit is instantiated over a secure out-of-band private network. A final decision, with the corresponding preventive measure, is made based on statistical formulations.

The contributions of this chapter are summarized in what follows. First, whereas the work in the literature assesses network packets against mathematical models of the power system, this work utilizes machine intelligence to develop a low-latency content-aware intrusion detection system that decodes the content of network packets and decides on their physical relevance. Second, the use of AI accelerates the attack detection and decision-making processes to the millisconds range, compared to what is presented in the literature. This is because AI reduces the online computational burden compared to complex system models. Most of the time, however, is spent on offline training. Third, the proposed Finally, experimentally verifying the developed IDS on a hardware laboratory scale AC/DC cyber-physical microgrid setup.

7.2 Analysis of Hierarchical Microgrid Control

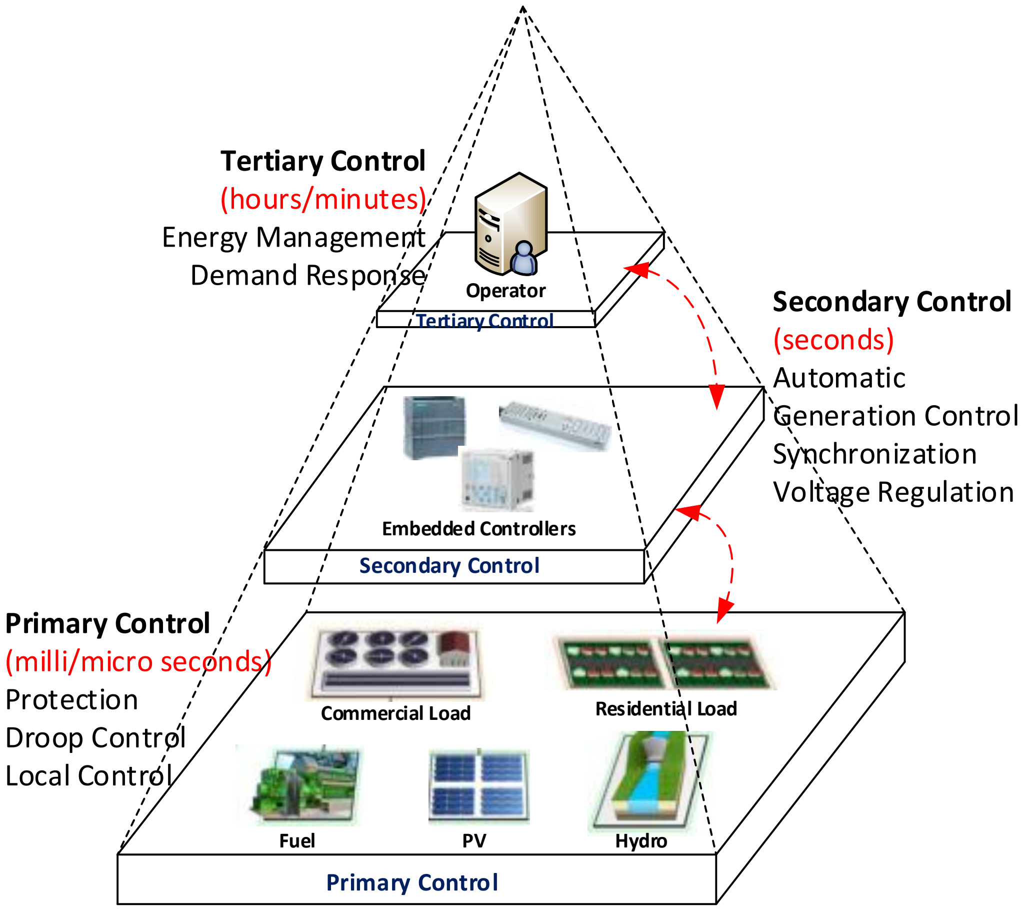


Figure 7.1: Hierarchical microgrid control (red dashed curves represent information exchange).

The decentralized nature of microgrids, which constitutes both energy production (distributed energy resources) and energy consumption (loads) entities, makes information exchange a challenge for proper control operations. To address that, the hierarchical control architecture of microgrids has been proposed in the literature [96][133][134]. As seen in Figure 7.1, the hierarchical control architecture divides the control operations of a microgrid into three levels, depending on the latency and information update time requirements. In the lower level, primary control applications, such as droop and local control, require fast responses in the ranges of milli- to microseconds. In the middle level, secondary control applications, such as automatic generation control, require a response time in seconds. The upper layer has a much more relaxed response speed, in the minutes to hours range, for applications such as energy management and demand response.

To properly implement this control architecture and to manage the exchange of information within and between all three layers in an orderly fashion, a link to industrial standards with interoperable data and protocols is necessary [96]. The IEC 61850 standard has been the most widely industry-accepted standard that provides a comprehensive data modelling and organization method, which unifies data structure definitions across equipment from different manufacturers. Recently, IEC 61850 has been used for microgrid control applications. The standard is well fit for decentralized and distributed control architectures because it abstracts the definition of the service and data items to be independent from the underlying protocols. The abstracted data items and services can thus be mapped into any other communication protocol [41]. Of particular interest to us in this work is the SMV protocol and the concept of the process bus. In decentralized industrial control networks, controllers acquire data about the surrounding environment through sensor readings and then issue control commands to actuators accordingly. The IEC 61850 standard stipulations have introduced an intuitive method to make sensor measurements simultaneously available to all controllers in a microgrid network by introducing the concept of the process bus. As shown in Figure 7.2, this is done by splitting the input/output of control agents and their control logic, and placing a communication bus between them.

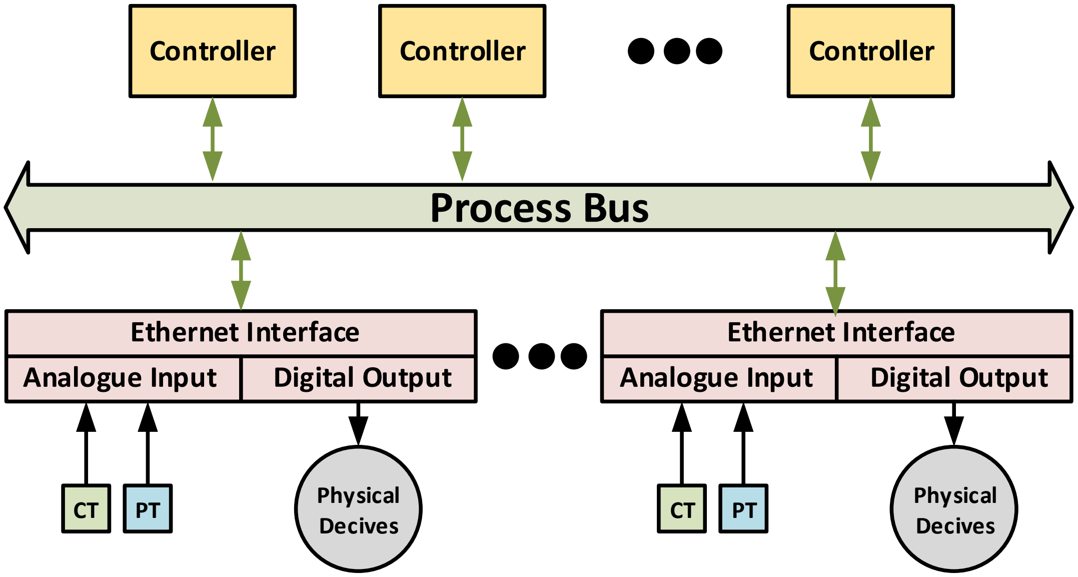


Figure 7.2: IEC 61850 process bus.

According to IEC 61850-9-2, the maximum end-to-end time delay allowed for SMV messages is 3 ms. This tight limitation on message transmission time makes it nearly impossible to encrypt SMV, packets especially with the low processing power of publishing MUs and receiving IEDs. This fact is further asserted by IEC 62351-6 security standard, which covers the cyber security of SMV messages, and relieves time-critical SMV messages from the burden of being encrypted. Therefore, in the event of a network breach, as mentioned in the introduction, manipulating digital measurement data is an easy task. Similarly, the literature showed several methodologies on fake data injection attacks on other industrial protocols [30][102].

7.3 Proposed Content-Aware Intrusion Detection and Prevention Mechanism

Since SMV is Layer 2 and non-blocking, issuing a spoofed SMV will result in a repeated or out-of-sequence sample counter. However, the subscribing device will not be able to distinguish which message holds the true measurement value. Accordingly, this section studies the feasibility of using neural network forecasters to identify spoofed and legitimate messages.

As can be seen in Figure 7.3, the cyber layer detection engine is complemented by a physical layer forecaster. The cyber layer looks at standard stipulations, such as repeated sample counters, whereas, the physical layer refers to the ability of the forecaster to predict the value of the incoming SMV measurement, thus, being aware of the physical characteristics of the system.

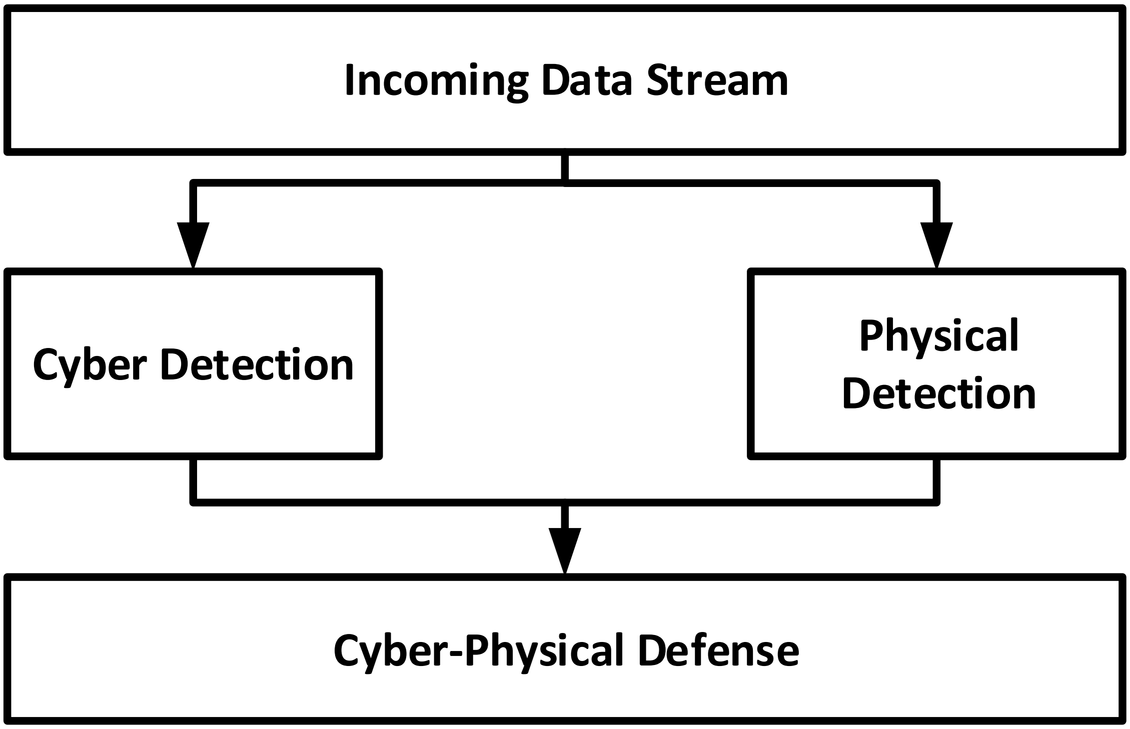


Figure 7.3: Cyber-physical detection of attacks.

With a fundamental understanding of the SMV datagram structure, it is possible to develop malware that specifically targets the measurement data sent along the process bus.

Layer 1: The Neural Network Forecaster

In the proposed framework, each of the control agents (IED)s and the SMV publishing agents (MU)s will have two threads running in parallel, as seen in Figure 7.4. Thread 1 of the publishing agent receives analogue measurements (via current or potential transformers), digitizes them through its analogue to digital converter, and publishes them over the LAN as SMV packets. On the other hand, the control agent subscribes to these measurements. If an anomaly in the sequence number field is detected, the control agents passes the two samples through its first thread, which is the NN-F. As will be detailed later, the NN-F utilizes N previous samples to forecast the value of the incoming measurement. The NN-F will compare the error between the received samples and the forecasted value. Only the received SMV, which has an error less than a specified threshold, will be marked as benign and is passed to the control logic.

The NN-F has three layers: one input, one hidden, and one output layer. The input layer has 20 neurons corresponding to 20 previous samples, whereas the output layer has 1 neuron corresponding to the forecasted sample. The number of neurons in the hidden layer is 10. The forecasting accuracy of the neural network against the computation time was studied. Based on this empirical study, it was found that 20 previous samples and 10 neurons in the hidden layer produce the highest accuracy in the least amount of time.

- *Generating the Training, Target, and Test Data Sets*: the effectiveness of using neural network forecasters to distinguish spoofed from legitimate messages was studied on the hardware microgrid setup shown in Figure 7.9. Data corresponding to different events of the hardware microgrid considered in this

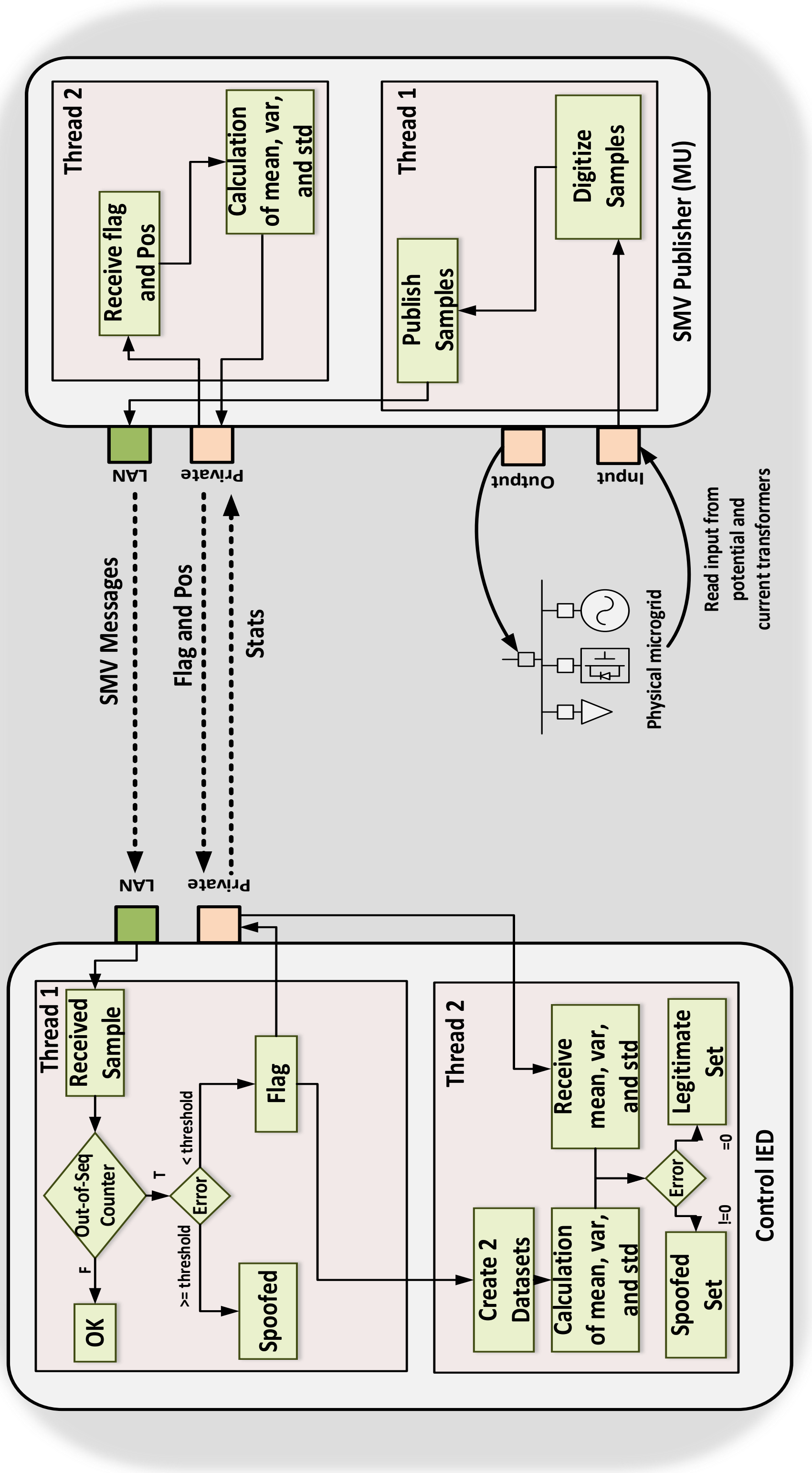


Figure 7.4: Proposed algorithm for detection of spoofed measurements.

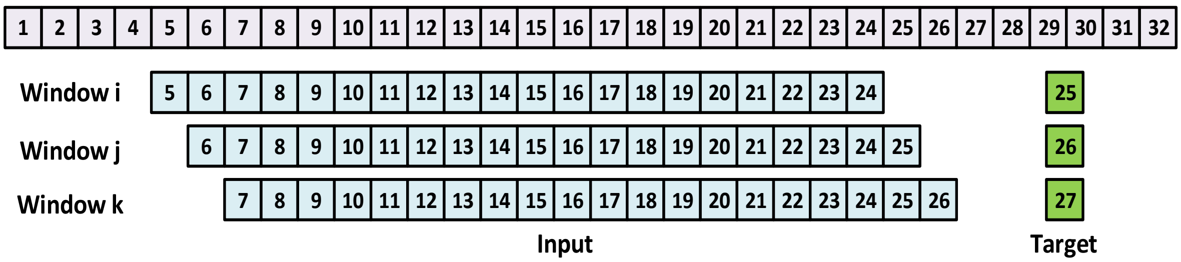


Figure 7.5: Sliding window training/target data generation.

work were collected from history logs. These events include different fault events that has previously happened and other contingencies, such as loss of transmission lines or generation units. Next, an exploration simulation approach was adopted to generate a rich set of training, target, and test data. That is, an accurate model of the microgrid was developed on Matlab/Simulink to generate data for the remaining contingency cases. Details on the accuracy of the developed model are given in Section VI.

- *Training the Neural Network*: the neural network was trained with the back-propagation algorithm with a sliding window approach. Starting from the first sample, 20 samples were counted as input and the 21st sample was set as the target output. Next, the window will move one sample, where the input will become samples 2 to 21, inclusive, and the target output is sample 22, and the process continues. The general process is depicted in Figure 7.5.
- *Setting the NN-F Decision Threshold*: to set the decision threshold of the NN-F, Monte-Carlo simulations were performed for more than 2,500 test cases, each with 2001 measurement samples. For each of the test cases, random fake data was injected at different instants according to equation (7.1). The fake data

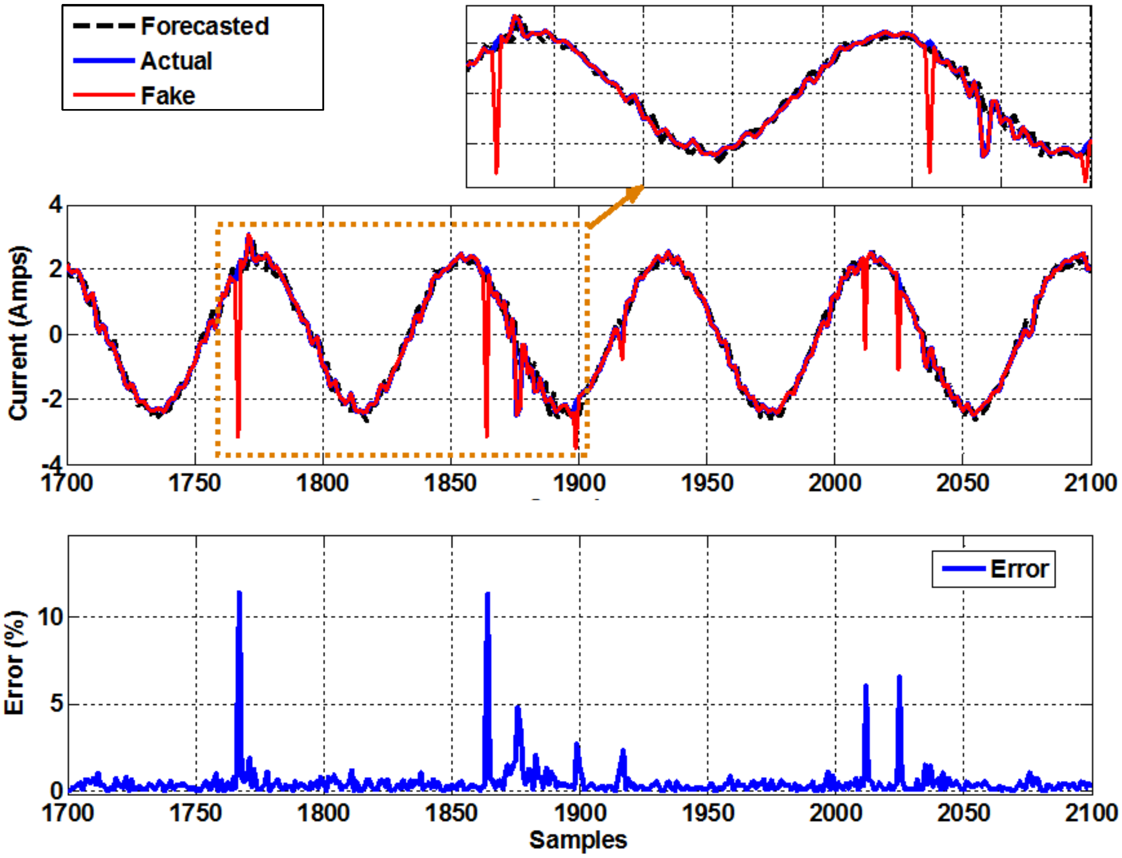


Figure 7.6: Performance of the NN-F.

was ranging between -4 and 4 Amps, which is 1.5 times the rated current of the microgrid under study.

$$fake\ data = (b - a) \times rand() + a \tag{7.1}$$

Where $a = -4$, $b = 4$, and $rand()$ is pseudo-random number generator that produces a random number between 0 and 1. Then, the forecasting error of the neural network was recorded. It was found that a 2% decision threshold produces the highest accuracy in detecting spoofed samples.

Layer 1 was tested on fake SMV packets with current measurement values 1.5 times the rated current and above. As can be seen in Figure 7.6, the NN-F has a high accuracy in forecasting the incoming measurement value. The performed study shows that using a NN-F has good potential in identifying spoofed messages based on learning the system's characteristics; however, over time, forecasters are susceptible to the accumulation of the forecasting error. Knowing that, an attacker could publish spoofed messages such that they don't violate the 2% decision threshold. If the IED uses the fake message, the buffer used to forecast the incoming measurements will have a misleading entry. In this situation, the IED will become indecisive.

To address this issue, an algorithm based on lightweight statistical indicators is proposed.

Layer 2: Enhancing the Resiliency of the NN-F

Based on the results from Layer 1, when the IED receives an SMV with a repeated or out of sequence sample counter, and is unable to distinguish the spoofed message, it will activate Layer 2 of the defense mechanism.

Case 1: Attacker sends one spoofed message

As mentioned earlier, if the IED uses the fake message, the buffer used to forecast the incoming measurements will have a misleading entry. Eventually the accumulation of the forecasting error with time, will lead the NN-F to start discarding legitimate messages, thus, disrupting the control operation of the IED. To avoid this situation, when the IED receives only one spoofed message and it cannot decide on its legitimacy, it will hold onto the suspected samples and will receive 5 new consecutive samples. For every one of the new samples, the error between the forecasted sample and the received one will be recorded. Then the error derivative will be calculated. If the derivative indicates that the error is increasing, a flag is issued. At this stage, thread 2 of the control agent is activated. Thread 2 will create two datasets. The first set contains the one of the suspected samples, five previous samples, and the five new samples, which were monitored. On the other hand, the other dataset contains the other suspected sample, five previous samples, and the five new samples, which were monitored. It will then calculate the mean, the variance and the standard deviation of these datasets. These datasets are all from the measurements received over the network. Simultaneously, the control agent sends the flag and the position of the suspected sample to thread 2 of the MU agent over an out-of-band trusted network. The MU will create a similar dataset composed of the suspected sample, five previous, and five later samples. However, this dataset will be from the local digitized data that hasn't been altered. Similarly, thread 2 of the MU agent will calculate the mean, the variance, and the standard deviation of the created dataset, and will send them to the control agent over the trusted network. Finally, the control agent will compare both sets of statistical indicators. The dataset which has matching statistical indicators is the one bearing the legitimate message.

Case 2: Attackers simulates a fault condition or hides a fault condition by replaying normal condition.

The NN-F is trained to forecast measurement values under fault conditions and other contingencies. The attacker could inject spoofed messages simulating a fake fault and the NN-F could identify them as legitimate. The attacker simulates a fault, one sample after the other. The attacker sends the 1st packet, which doesn't violate the threshold. If the IED uses it, in the next time step, the NN-F will forecast a higher current value (fault) and will discard the legitimate message.

In this case, if two repeated or out-of-sequence samples are detected after each other, then Layer 2 is activated to decide if this is a real fault or not. Similar to Case 1, the IED will create two datasets. One dataset has the samples that correspond to the fake fault, while the other has the normal measurements. The dataset which has matching statistical indicators with the MU will be the legitimate one. Conversely, the attacker could hide an actual fault by replaying measurements corresponding to the normal condition. Following the same procedure, the IED could decide which message stream is valid and which is not.

- *Selection of the Statistical Indicators*: the purpose of the statistical module is to create a small yet indicative feature vector of the two datasets generated by the control IED and the MU. The selected statistical features for the two datasets are the mean, variance, and standard deviation, calculated as shown in equations (7.2), (7.3), and (7.4), respectively.

$$\mu = \sum_{i=1}^{N} \frac{x_i}{N} \tag{7.2}$$

$$var = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{7.3}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{7.4}$$

In order to select the appropriate number of samples for the IED to hold on to, the following study was performed. Consider the current data shown in Figure 7.7. Various spoofed data were injected at the peak of the sine wave ranging between +/-0.5 to +/- 1.5 times of that sample. Next, 5 samples were taken before and after the peak sample and the mean, variance, and standard deviation were calculated. The same were also calculated to the same sample from the original data (i.e. with the actual value instead of the fake sample). The error of both statistical vector indicators was then calculated for all the fake data cases. Next, the dataset size was increased to 10 before and 10 after the fake sample and the error was calculated. The same procedure was then repeated reaching a dataset size of 60 samples before and 60 samples after the fake sample.

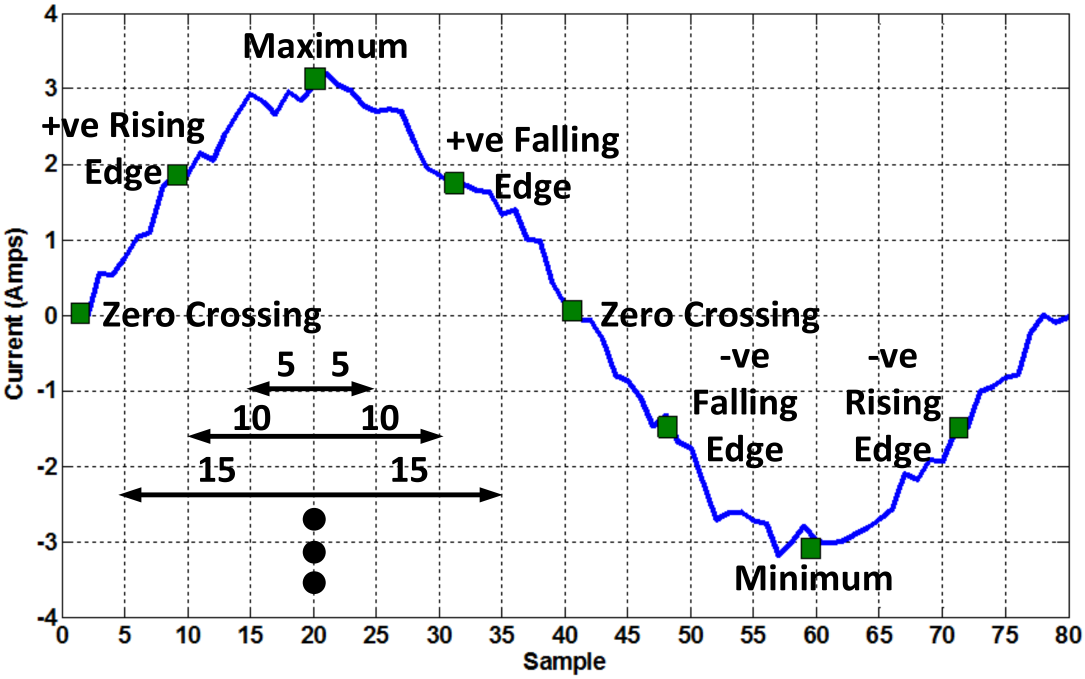


Figure 7.7: Performed fake data injection attack study.

The same procedure was repeated at all the critical locations in the sinusoid: the minimum, negative/positive rising/falling edge, and zero crossing. All the error data were then averaged for each data set size and are plotted in Figure 7.8. As shown in Figure 7.8¸ the most indicative dataset (largest errors) was for 5 samples before the suspected sample and 5 samples after it.

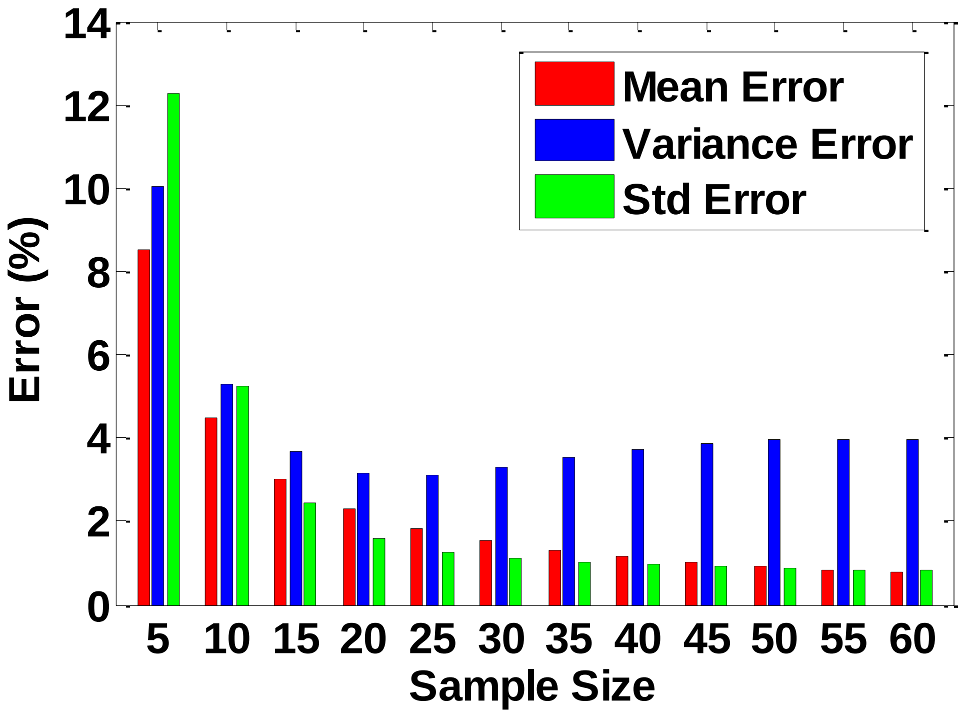


Figure 7.8: Percentage error of the feature vector with increasing sample size.

7.4 Experimental Setup and Model Verification

7.4.1 Hardware Setup

The performance of the proposed security framework was tested on data collected from the hybrid AC/DC cyber-physical microgrid, shown in Figure 7.9 and Figure 7.10. The microgrid has two generation units, Generator 1 and Generator 2, with 13.8 KVA 230 V and 10.3 KVA 230 V, respectively. The two AC loads shown each have 10 levels of parallel resistive loads ranging from 300-W to 3-kW. In this experiment, Load 1 and Load 2 are set at 600 W each. Each of the shown buses has three sets of three-phase inputs and outputs with solid-state relays, and has its own potential and current transformer for measurement data collection. The DC microgrid is connected to the AC microgrid via a bi-directional inverter. A DER is present in the DC microgrid along with a 12-Ω DC resistive pulse load and a 60-Ω DC constant load.

The AC and DC microgrids exchange power to support each other, when necessary, through setting the direct component of the current of the controller shown in Figure 7.11. The controller uses the dq transformation to convert the direct current and quadrature current reference from the dq to the abc reference frame. The results of the transformation are the three-phase reference currents, which are injected or drawn from the AC system to transfer the required active and reactive power. The actual current is generated using hysteresis current controller, which controls a voltage source converter operating in current control mode.

7.4.2 Model Development and Verification

As mentioned earlier, the AI module was trained using datasets for different operational states/contingencies of the microgrid collected from history recorded data or from simulation data. To ensure the fidelity of the simulation model, the simulated microgrid developed in Matlab/Simulink was verified by comparing bus voltages, currents, and power measurements with experimental data. Table 7.1 shows that the simulated microgrid accurately describes the real one. Also, we further investigated the power sharing between the AC and DC microgrids in the simulated and actual system.

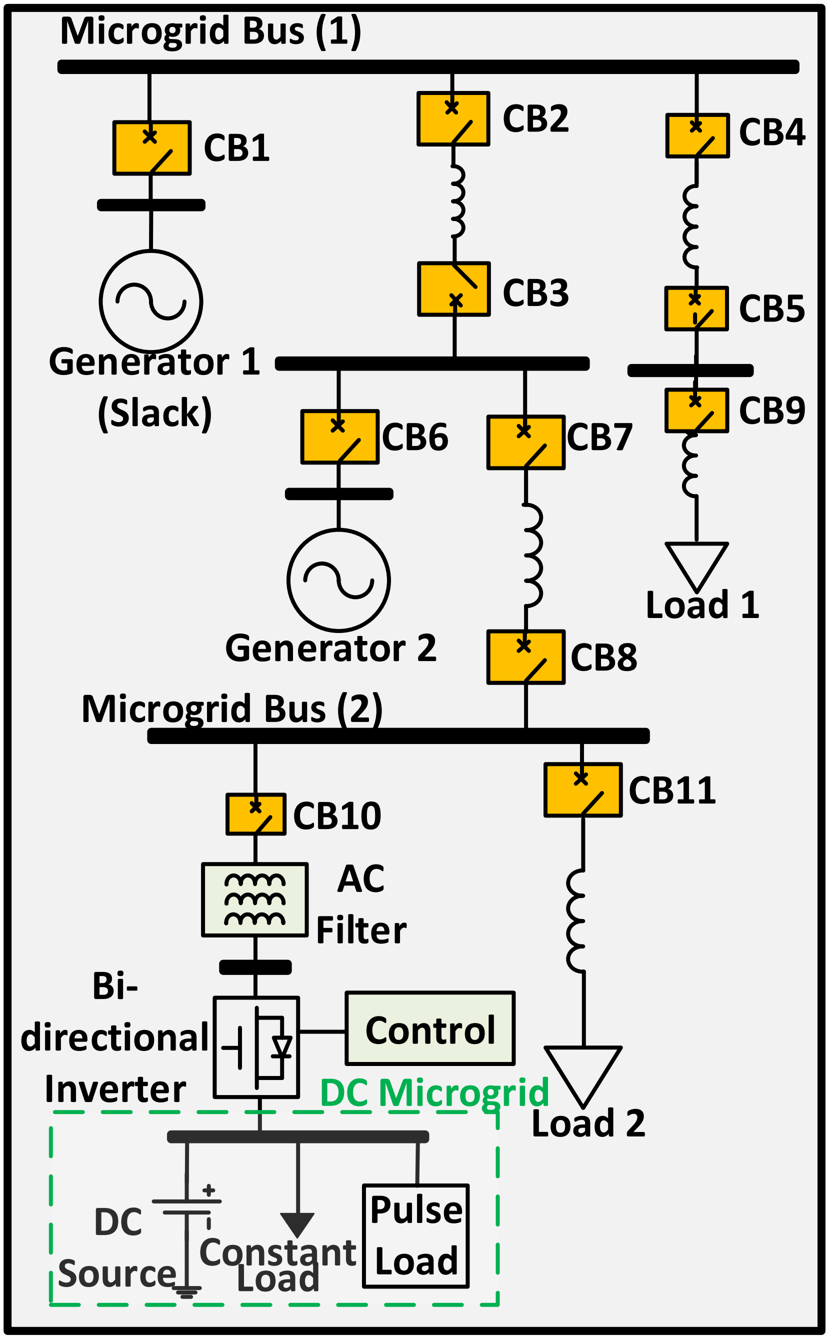


Figure 7.9: General overview of the AC/DC microgrid (CB: Circuit Breaker).

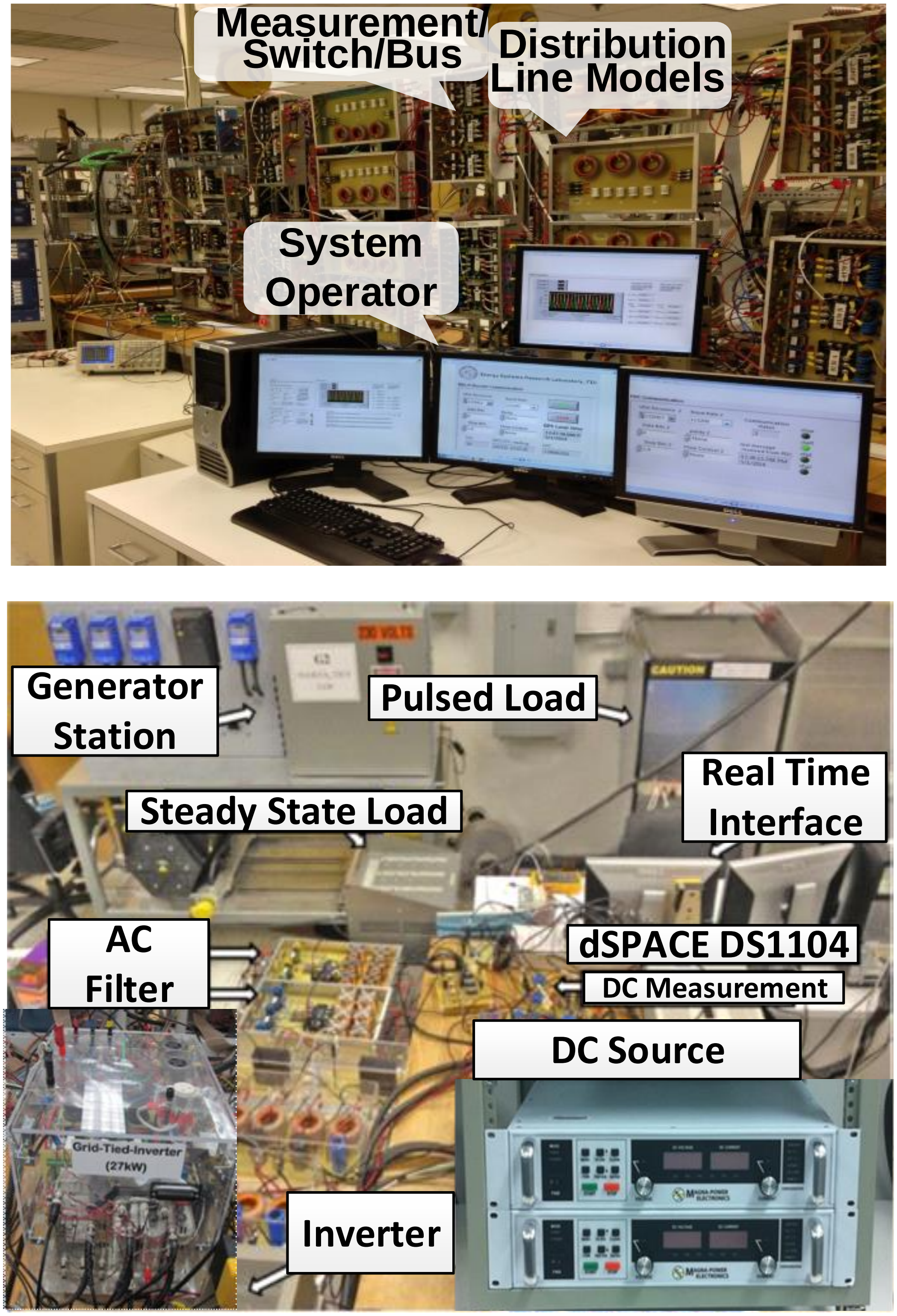


Figure 7.10: Labeled image of the hardware equipment used.

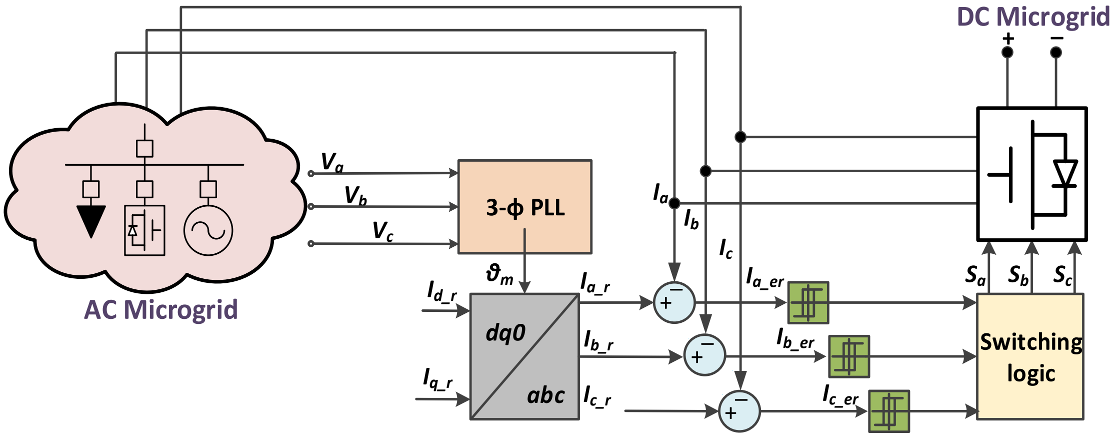
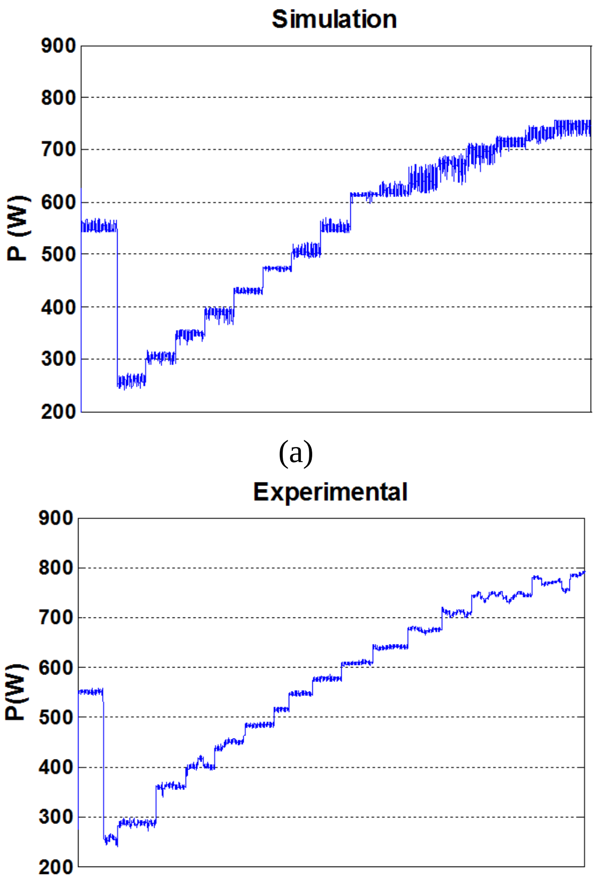


Figure 7.11: Control of the Bi-directional Inverter.

Table 7.1: Simulated Model Verification

	G1 Bus		**G2 Bus**		**AC Load 1**		**AC Load 2**		**DC Microgrid**	
	S	**E**	**S**	**E**	**S**	**E**	**S**	**E**	**S**	**E**
Irms	2.1	2.6	0.7	0.8	1.5	1.2	1.3	1.5	0.4	0.4
Vrms	120	119	120	120	116	116	118	119	118	118
P (W)	761	758	243	255	431	435	441	442	170	170

Figure 7.12 shows the power on the DC microgrid interconnection bus. It can be seen from Figure 7.12 that both systems performed the same, thus, further verifying the fidelity of the simulated model. In practice, each utility or system operator has their own tools to model their power systems for internal studies. Such models and tools can be used to generate training data.



(a)

(b)

Figure 7.12: Model verification against experimental results. (a) Simulation; (b) Experimental.

7.5 Results and Discussion

To study and analyze the effectiveness of the proposed framework, we implemented a MU at Load 1 and an associated control agent that receives current measurements of Load 1 on the other side of the process bus.

Several experiments were performed to assess the effectiveness of the different modules in this framework against spoofed measurement attacks and the results are reported in this section.

- The malware script was used to inject a small perturbation to the current value at sample 1200. In this small perturbation attack, the NN forecasting error started to accumulate. At this stage, the AI module monitored the rate of change of the error of the next few samples. As seen in
- Figure 7.13, the error derivative was increasing indicating that indeed an accumulation of the forecasting error between the received and forecasted samples was occurring. Therefore, the IED sent a flag to the merging unit including the position (i.e. sample 1200) of the suspected sample to activate layer 2 of the proposed framework. Both the control agent and its merging unit performed the statistical study explained earlier. The results of the statistical analyses showed a difference in the calculated mean, variance, and standard deviation. This means that the forecasting error accumulated due the perturbation attack.
- The malware script was used to inject spoofed messages simulating a fake fault. As seen in Figure 7.14, the attacker injected current measurement values corresponding to a fake fault condition. Since the first spoofed sample does not violate the detection threshold, it is assumed that that IED uses it to forecast the next sample. The next forecasted sample will be closer to the fault condition, rather than the legitimate SMV. Therefore, the IED will forecast that there is a fault, and will act accordingly. However, given the proposed cyber and physical defense mechanism, when the IED noticed two SMV messages, one after the other, each with repeated sample counters, it performed the algorithm in Section V B. Based on comparing the statistical indicators it received from the MU and

the ones it calculated, it was able to identify that there is an attack and that this was a fake fault condition.

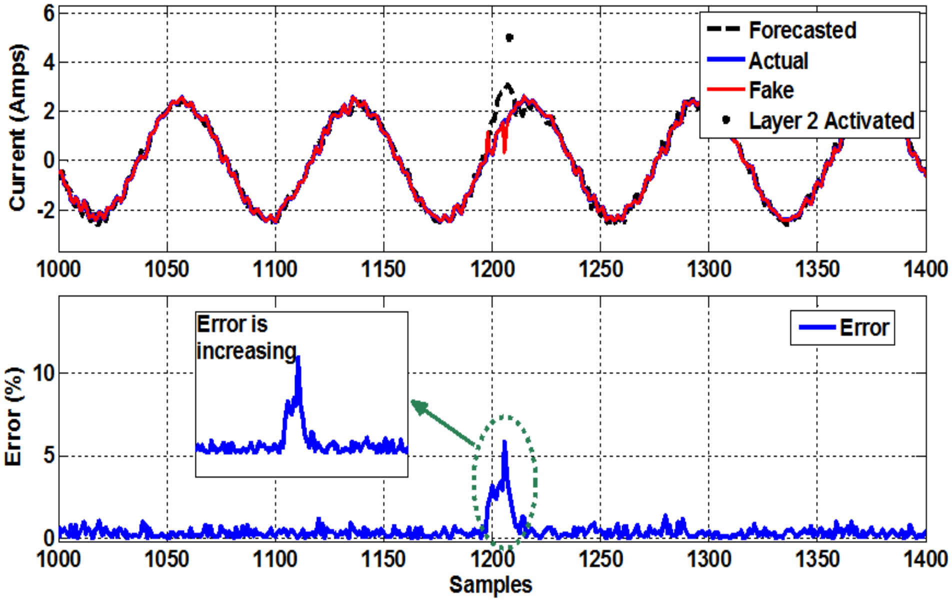


Figure 7.13: Detection of perturbation attack.

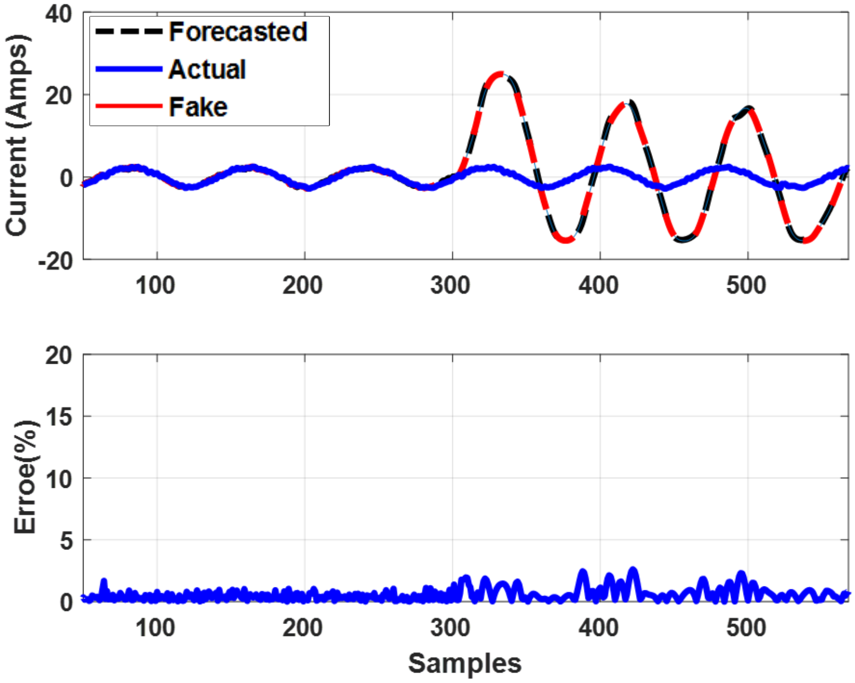


Figure 7.14: Detection of a fake fault.

- Finally, the latency of the complete detection process including the information exchange over the out-of-band network and the hardware time required for packet crafting was assessed. In Figure 7.15, t_1 is the time starting from the instant of issuing the flag, the time to calculate the statistical vector in the control agent, and finally the decision making. t_2 is the time it takes the first packet, which contains the flag, to reach the MU. t_3 is the time for calculating the statistics vector by the MU. Finally, t_4 is the time for the packet, which contains the statistics, to reach the control agent. Therefore, the total detection latency is calculated in equation (7.5):

$$\begin{gathered} t_T = t_1 + t_2 + t_3 + t_4 \\ t_T = 0.4 + 0.25 + 0.25 + 0.3 \\ t_T = 1.2\,ms \end{gathered} \tag{7.5}$$

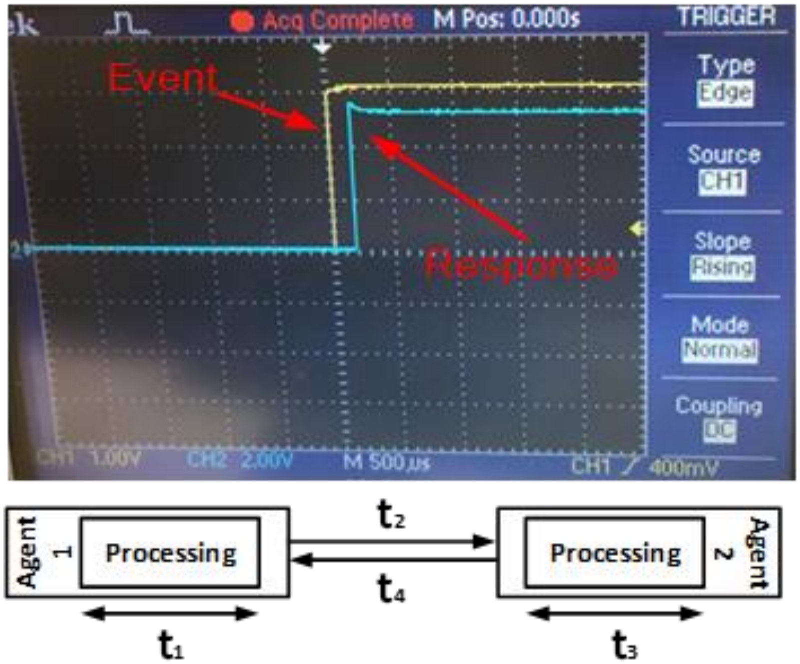


Figure 7.15: Detection of Latency.

We designed the following experiment within the private network to determine the latency of this two-message exchange, while accounting for delays imposed by the hardware. By using one of the General Purpose Input Output (GPIO) header pins on the agent devices, we can instruct the device to craft a packet and send it on the network in response to a High digital input. Then, we can instruct the receiving device to set one of its digital output pins to Low after it has received and processed the GOOSE message. Figure 7.15 shows the results of the experiment, demonstrating that the entire process of sending a single message takes roughly 0.25 ms.

7.6 Conclusion

In this chapter, the reliability of utilizing neural network forecasters to detect spoofed messages was studied. It was found that although they have a high detection accuracy, they could still be susceptible to the accumulation of the forecasting error. Accordingly, this chapter also presented a lightweight algorithm, based on statistical indicators, to enhance the reliability of the neural network forecaster in terms of detecting spoofed sampled measured values. This algorithm is considered as a second line of defense, complementing the cyber security methods, such as encryption and authentication. The studies were conducted on a laboratory-scale hardware microgrid with a commercial network switch to be more representative of the microgrid environment than event-based simulation models. The detection latency of the proposed algorithm was found to be near real-time, in the range of 1-2 ms.

Chapter 8 Utilizing Energy Storage Devices to Ensure Customers' Privacy through Smart Energy Management

The load-demand profiles of customers reveal a lot of information about the customers' daily behavior, such as their sleeping schedules and the time they spend inside or outside their homes. Customers' load-demand data is transferred to utility control centers over wide area networks through an Advanced Metering Infrastructure (AMI). If an attacker intercepts this data, he/she will be able to infer private and sensitive information about the customer. To that end, this chapter discusses the development of an energy management prototype, which utilizes energy storage and local generation devices to camouflage customers' load-demand profiles, ensuring their privacy.

8.1 Introduction

The AMI allows bi-directional information exchange between smart meters and utility control centers. The meters send consumption information, usually every one hour, and receive price information and control commands from the smart meter head-end. Since the smart meters collect accurate load consumption data, a lot of information about the customer can be extracted from the meter data. This information includes private customer behavior, such as the customer's sleeping schedule, the time at which the customer leaves or comes back home, the type of appliances they are using, and even the time at which each appliance is used. Therefore, if an attacker were to access this data, the customer's privacy and safety may be jeopardized.

This privacy problem is not only a concern to residential customers, but is also a big concern for military bases that are present abroad, and are connected to foreign utility grids. The privacy of the load-demand consumption profiles is of high importance in this case.

The technique by which data is extracted from load-demand consumption profiles is called Non-Intrusive Load Monitoring (NILM). NILM was first introduced in [135][136][137]. The idea behind NILM is to disaggregate the consumption data for individual appliances from the total metered data. In order to identify the individual appliances, NILM utilizes current, active power, and reactive power signatures.

In order to ensure the privacy of customers and obfuscate their metered data, several privacy preserving techniques were introduced. A study in [138] categorizes these efforts into three main categories:

1. Anonymization of metering data, which involves the utilization of an ID escrow third-party to separate meter data and ID's of customers.
2. Privacy-preserving metering data aggregation, which involves the aggregation of meter data from several households within a certain district, before sending this data to the utility center.
3. Metering data obfuscation, which involves utilizing energy storage to hide individual load profiles.

This chapter will focus on metering data obfuscation efforts. In [139], a comparison between Battery-based Load Hiding (BLH) techniques and Load-based Load Hiding (LLH) techniques was presented. The main difference between the two techniques is that BLH techniques use battery charging/discharging algorithms to flatten peak demand and eventually obfuscate meter data, whereas LLH techniques use controllable loads to increase the metered load level compared to the net demand. The authors in [140] proposed the usage of electric vehicles for household load-demand profile hiding. Their work achieved a dual purpose of optimal electric vehicle charging and obfuscating metering data. In [141], a power management model, using a rechargeable battery, was introduced under the assumption that electrical power routing can be used to hide usage information of home appliances. Many other similar works are also present in the literature.

In this chapter, the development of an energy management algorithm, which utilizes batteries and local generation devices to obfuscate meter data, is discussed in detail. First, an overview of the energy management system will be presented. Next, details about the system's main module, which is the fuzzy logic controller, are discussed. Then, the optimization problem is formulated along with the details of the implemented exploration simulation approach. Finally, the results of the presented energy management system in terms of hiding customers' load-demand profiles are given for residential load consumption in Miami, Florida, for the year 2014.

8.2 Description of the Energy Management System

The developed energy management system is intended to hide the load-demand profiles of small microgrids, or as they have been referred to in the recent literature "nanogrids". Since the expertise and manpower that are present in large utility systems is not always available for operating small microgrids, energy management systems for small microgrids should be designed for ease of installation, support, and maintenance. Therefore, a robust, resilient, and distributed communication infrastructure, with failover mechanisms, is required. Accordingly, the DDS standard from OMG has been selected as the communication middleware that orchestrates all the energy management system's components together. The DDS is a peer-to-peer data-centric protocol capable of meeting the required performance metrics of the developed energy management for small microgrids. The DDS middleware is explained in detail in Chapter 2.

Figure 8.1 shows the overall topology of the nangorid on which the proposed energy management system is implemented. The controlled nanogrid has its local energy storage (battery in our case), distributed energy resource (a PV system in our case), and local loads. At all times, the nanogrid is assumed to be connected to the utility system through a bidirectional grid-tied inverter. The *Direction of Power Flow signal* is a control command that is issued by the energy management system that sets the percentage of the power that is to bought from or sold to the utility grid.

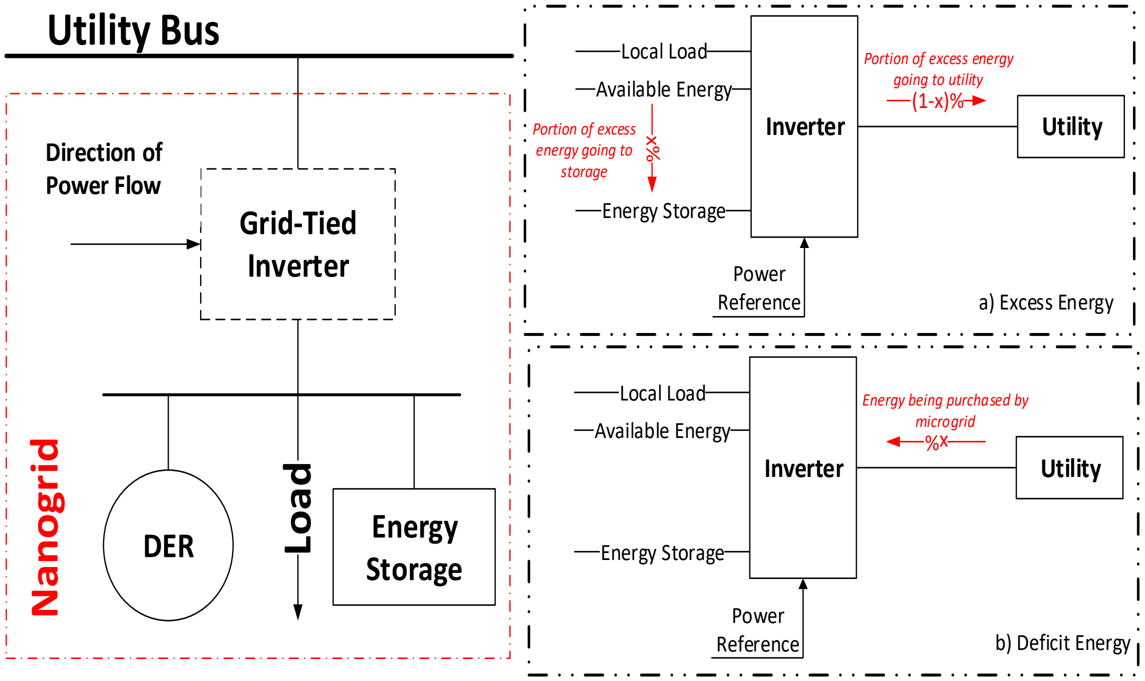


Figure 8.1: Topology of the Nanogrid.

In the case the nanogrid has surplus energy, the energy management system's controller outputs a power reference signal, which dictates the percentage of excess energy that is to be stored in local energy storage devices and the percentage that is to be sold to the utility. This is done only after ensuring that the local load-demand of the nanogrid is satisfied. Contrary to this, if the nanogrid has energy deficiency, the energy management system's controller checks the amount of power that is needed to cover the local load, then it purchases that amount from the utility grid. This process is explained in the diagrams in Figure 8.1.

Figure 8.2 shows the logical relations between the DDS topics and the components of the developed system. In the proposed EMS, in order to forecast the anticipated load-demand for the next operating period, the load forecasting module depends on weather data and previous load-demand profiles. The load forecasting module publishes the anticipated load-demand to the *Forecasting Data* topic. The smart meter, battery system, and the smart loads publish the local measurements, available energy, and local load data to the *Available Energy*, *Load Data*, and *Local Measurements* topics, respectively. The converter controller also publishes to the *Local Measurements* topic. Finally, the energy management system subscribes to the *Available Energy*, *Forecasted Data*, *Price Information*, and *Local Measurements* topics and publishes the control command and price information.

The energy management system controller is based on a set of fuzzy logic rules, and takes its decision based on several input parameters, which are going to be discussed in the next section.

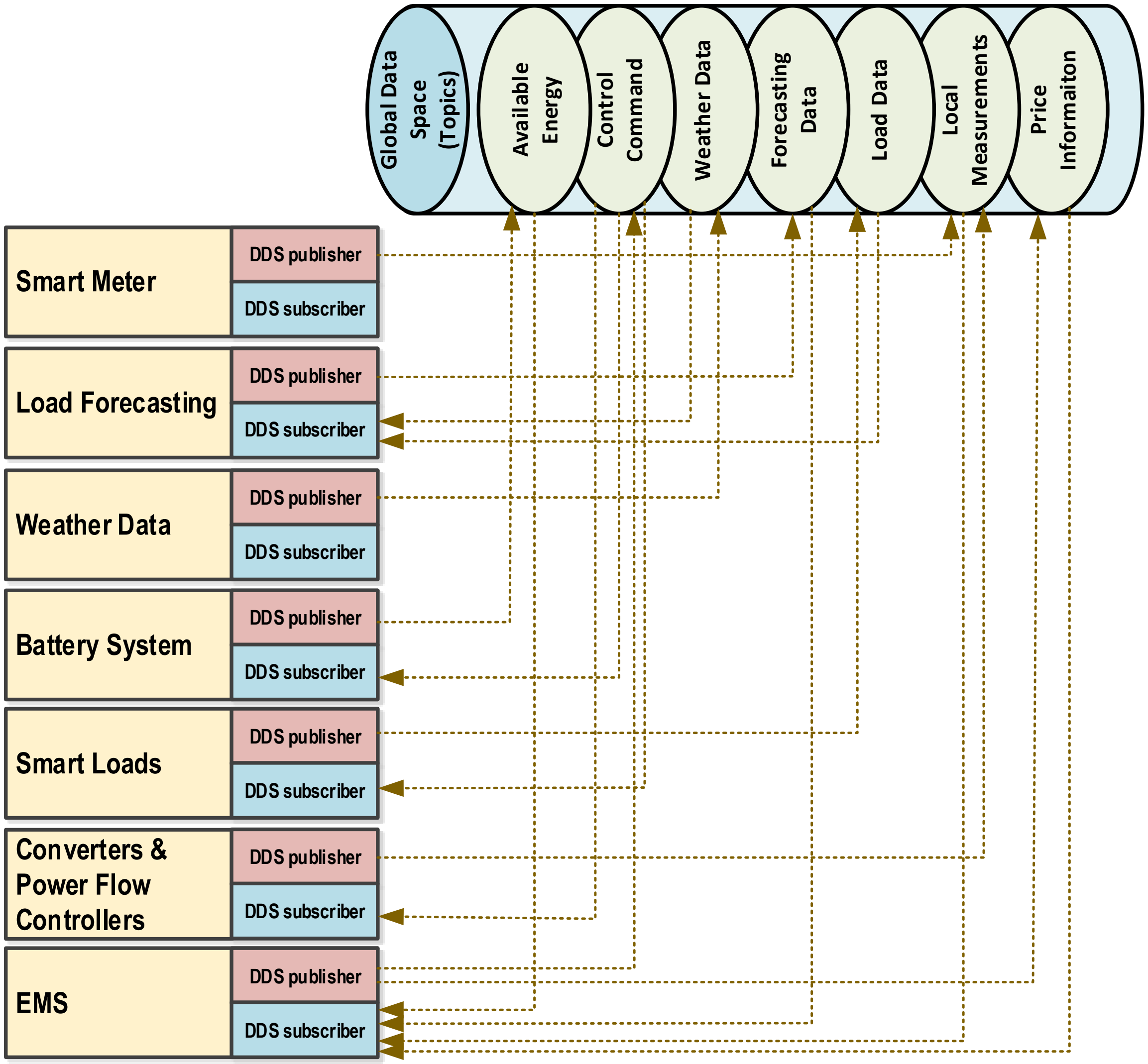


Figure 8.2: Logical Relations between the DDS Topics and the EMS Components.

8.3 The Fuzzy Inference System

The energy management system's controller is a fuzzy inference system built in accordance to the Sugeno-like model. The controller takes the batteries' state-of-charge (SOC), current price (CP), next hour price (NHP), and available energy minus local load-demand (AE - LLD) as inputs, and produces the nanogrid’s reference power (P_ref) as output according to equations (8.1) and (8.2):

Excess Energy Case:

$$P_{ref} = (AE - LLD)x + LLD \tag{8.1}$$

Deficit Energy Case:

$$P_{ref} = (LLD)x \tag{8.2}$$

Where x is the raw output of the controller, and is a number between zero and one. In the case of excess energy, x is thought of as the percentage of extra energy that is going to the local storage devices, and thus, $(1 - x)$ is the percentage of energy that is to be sold to the grid. On the other hand, in the case of deficit energy, x is thought of as the percentage of energy that is required to cover the deficit in local generation, in order to cover the nanogrid’s local load-demand.

Figure 8.3 shows a block diagram of the developed fuzzy inference system. Each input is passed through a set of trapezoidal membership functions and its membership value is evaluated. The ranges for the membership function are tabulated in Table 8.1. It is important to mention here that in order to maintain a good state of health of the battery, the controller does not allow the battery to drop below a state of charge of 40% [142]. Next, the firing strength of each rule is calculated using the minimum operator. Finally, the weighted sum technique is used to calculate the crisp value of the output.

Table 8.1: Ranges of the Membership Functions

Input	Membership Function	Range
Battery State of Charge (%)	*Don't Sell*	$SOC \leq 40$
	Low SOC	$40 < SOC \leq 50$
	Medium SOC	$50 < SOC \leq 60$
	High SOC	$60 < SOC < 100$
	Fully Charged	$SOC = 100$
Current Price (c/kWh)	*Low CP*	$CP = 10$
	High CP	$CP = 14$
Next Hour Price (c/kWh)	*Low NHP*	$CP = 10$
	High NHP	$CP = 14$
Available Energy (%)	*Low AE*	$AE \leq 50$
	High AE	$AE > 50$

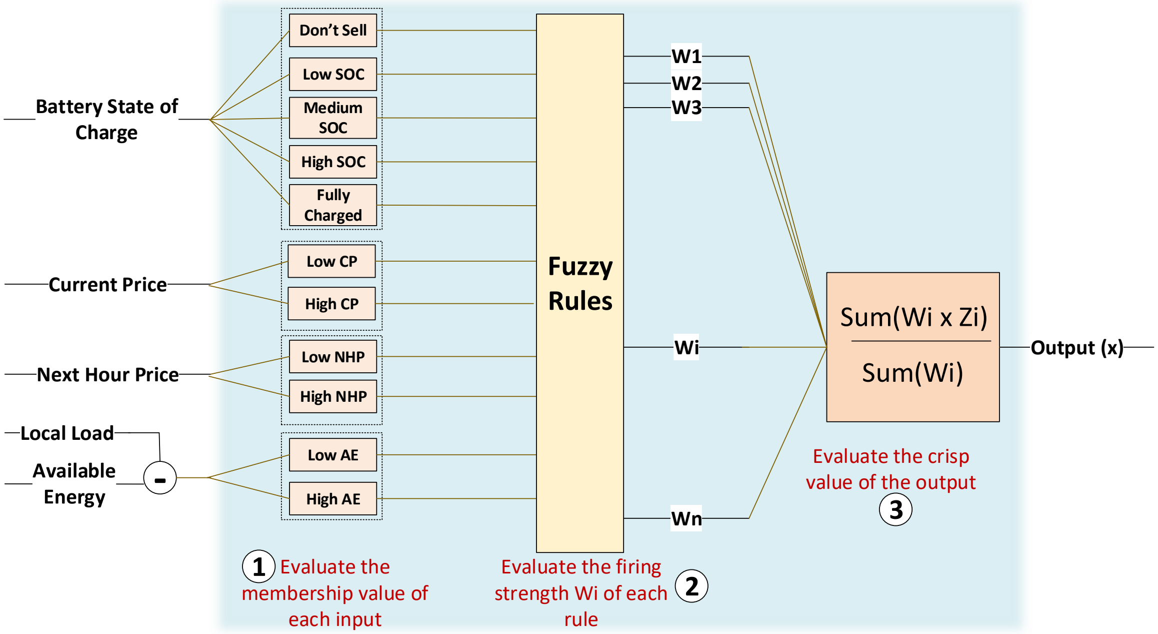


Figure 8.3: The Fuzzy Inference System.

8.3.1 Evaluating the Performance of the Energy Management System

The nanogrids, energy management algorithm, energy storage, renewable resources, and load models are developed on MATLAB/Simulink. Instead of having all the modules exchange data internally within the MATLAB environment, the developed modules are integrated with a DDS communication middleware to exchange the data over a real Ethernet network. This approach was adopted in order to account for networking issues, such as packet drop, latency, and QoS. The merging of real network hardware and simulation software creates a hybrid modelling environment, which allows accurate emulation of the proposed energy management system as an integrated cyber-physical system. Finally, a software module, representing a smart meter head-end, collects all consumption measurements, feeds the data to the utility pricing module, and publishes back the real time energy prices.

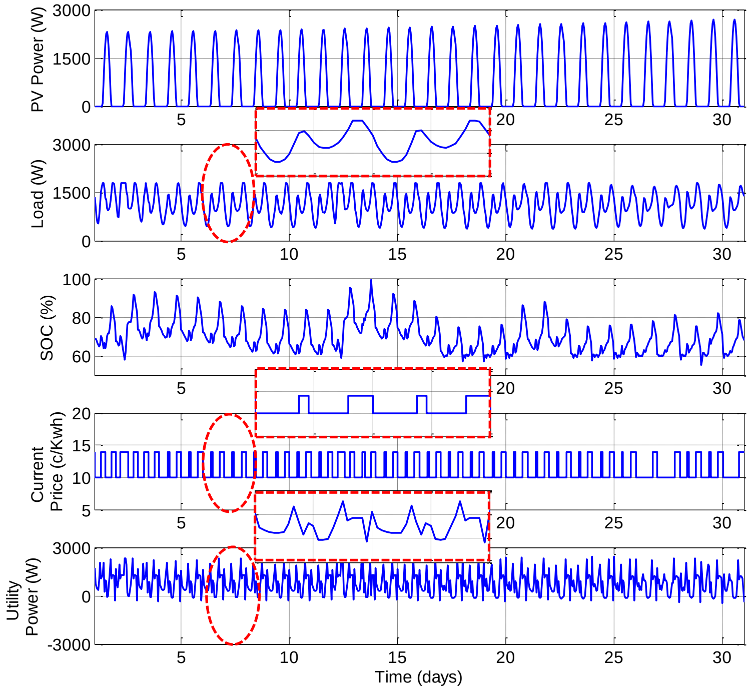


Figure 8.4: Winter Real-time Pricing.

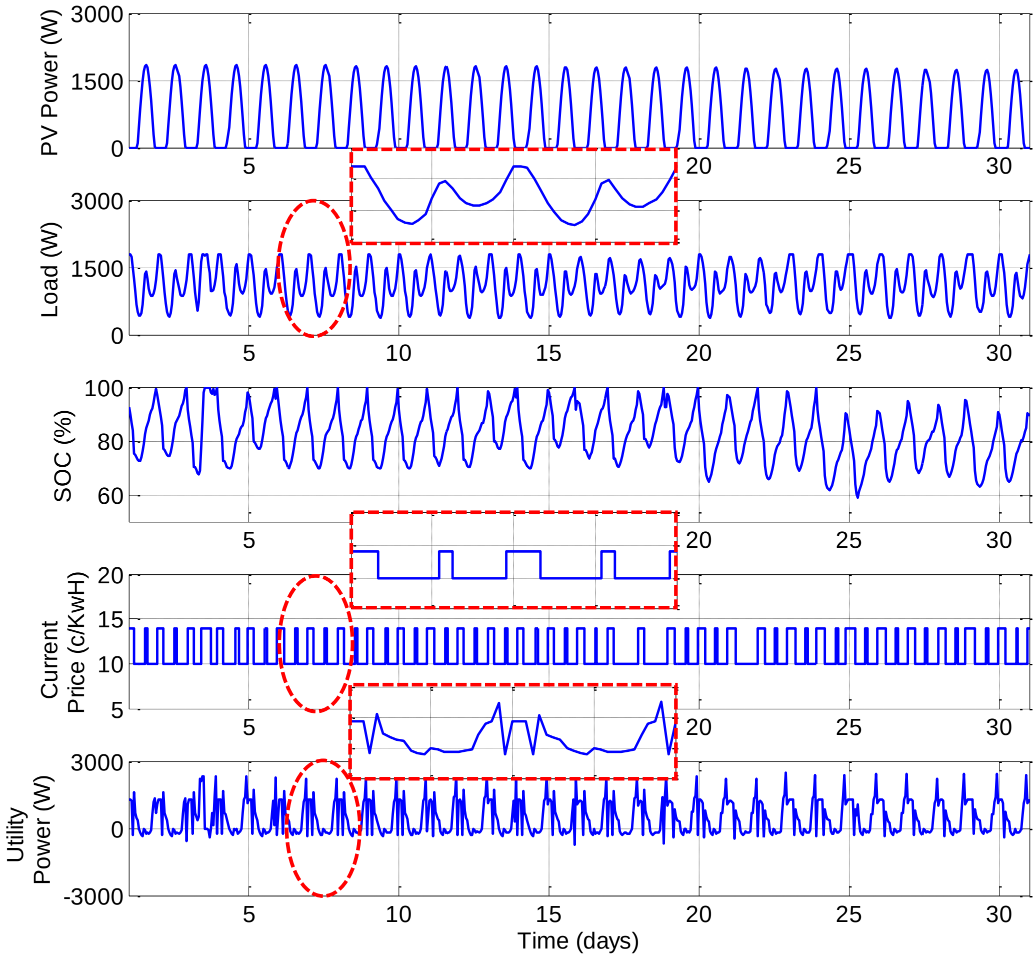


Figure 8.5: Summer Real-time Pricing.

Here, we present the results of applying the proposed energy management system on load-demand profiles and solar irradiance measurements, that were collected for Miami, Florida, USA in the Winter and Summer seasons for the year 2014. The month of January has been selected to represent the Winter season, whereas the month of August was selected to represent the Summer season. Additionally, each case was emulated with real-time pricing profiles.

Figure 8.4 and Figure 8.5 show the result of applying the proposed energy management algorithm with real-time pricing scheme in Winter and Summer seasons, respectively.

Looking at the zoomed parts in Figure 8.4, the original load-demand profile had peaks during high-energy price periods. The proposed energy management system was successful in reducing these peak values by managing the consumption from energy storage and renewable resources. In the Winter season, the peak load-demand at time t = 6.8 days, which corresponds to a high-price period, was reduced from 3600 W to 2571.3 W. Looking at the SOC profile, one notices that the SOC ranges between 60-100%. These results are also asserted in the Summer season, shown in Figure 8.5, where the peak load-demand at t = 7 days dropped from 3600 W to 2628.3 W. Similarly, the SOC of the batteries ranges between 60-100%.

Using the previous discussion, and comparing the load profiles before and after running the energy management system, one notices that the energy management system was clearly able to hide the original load demand profile, ensuring the privacy of the customer's data.

8.4 Optimizing the Controller's Parameters for Better Results

As discussed earlier, the purpose of the energy management system is hiding the customer's load-demand profile as much as possible, to ensure his/her privacy. In order to achieve the best results, an online parameter optimization scheme for the fuzzy logic controller parameters, based on Particle Swarm Optimization (PSO), was developed.

Figure 8.6 shows the chronological order of the optimization process. At the beginning of each day, the collected data from the previous day are fed into the optimizer, and the optimization process is initiated to come up with new optimized controller parameters for that day. It is worth noting that the optimization process in this work falls into the category of exploration simulation. That is, during the second day, while the system is in operation, the optimization process is being executed in the background on a simulated microgrid model to evaluate the performance of the new optimized parameters. Once the processes finalizes, the controller's parameters are updated online without any disturbance to the overall system operation.

Consider the three-day period shown in Figure 8.6. During the previous twenty four hours, measurement units in the microgrid are continuously collecting data about available energy, local load demand, current price, next hour price, and state-of-charge of the nanogrid's energy storage devices. All of the collected data for the past twenty-four-hour window are stored in the nanogrid's database. At the end of the first day, the optimization process initiates. First, a search space is randomly generated by defining a population of varying combinations of membership functions. The population generation process is bounded by vectors of lower bounds (LB) and upper bounds (UB) for each of the vertices of the membership functions. Proper definition of the lower and upper bounds is critical for the success of the optimization process. Figure 8.7, shows the search space for a given trapezoidal membership function. In order to ensure proper operation of the fuzzy controller, the condition that $A < B < C < D$ must be met.

Also, assume that the red membership function corresponds to a low SOC and the green membership function corresponds to a high SOC. It is important, thus, that the green membership function remains to the right of the red one. All these conditions have been taken into account in the setting of the lower bounds' and the upper bounds' vectors. The objective function to be minimized is shown in equation (8.3).

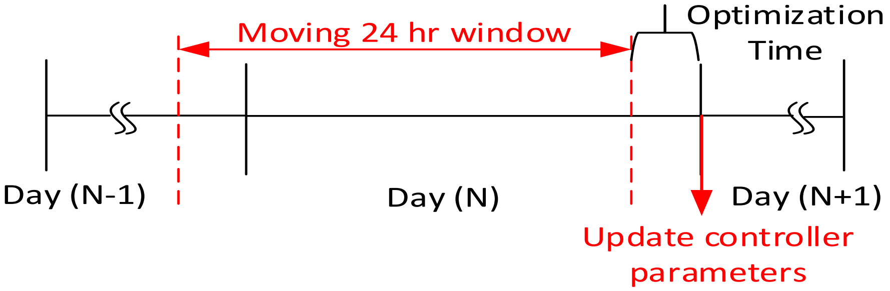


Figure 8.6: Optimization Process.

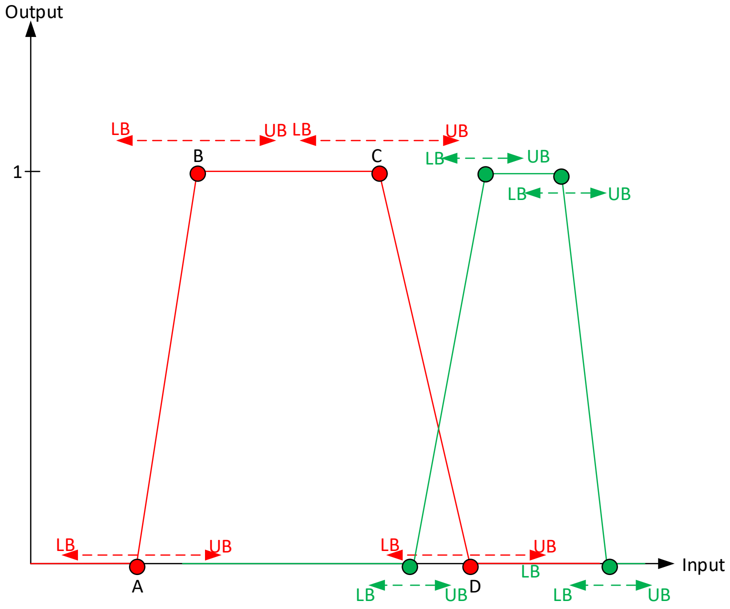


Figure 8.7: Constrained Search Space for Trapezoidal Membership Functions.

$$min\left(\sum_{h=1}^{24} P_{utility}(h) \times Cost(h)\ \right) \tag{8.3}$$

In a unique exploration simulation approach for fitness function evaluation, a software model of the physical nanogrid with its controller was developed and used to simulate the response of the nanogrid to the various particles in the swarm, and evaluate the profits and expenditures. In all situations, the optimization processes is bound by the constraints of checking that the nanogrid is covering its base load and that the energy storage devices are maintained at a proper state of charge that does not deteriorate them. The optimization process is repeated until the combination of membership functions that results in better utilization of the nanogrid's energy storage, to shift peak loads to low price periods, is achieved. This process is summarized in the flowchart of Figure 8.8.

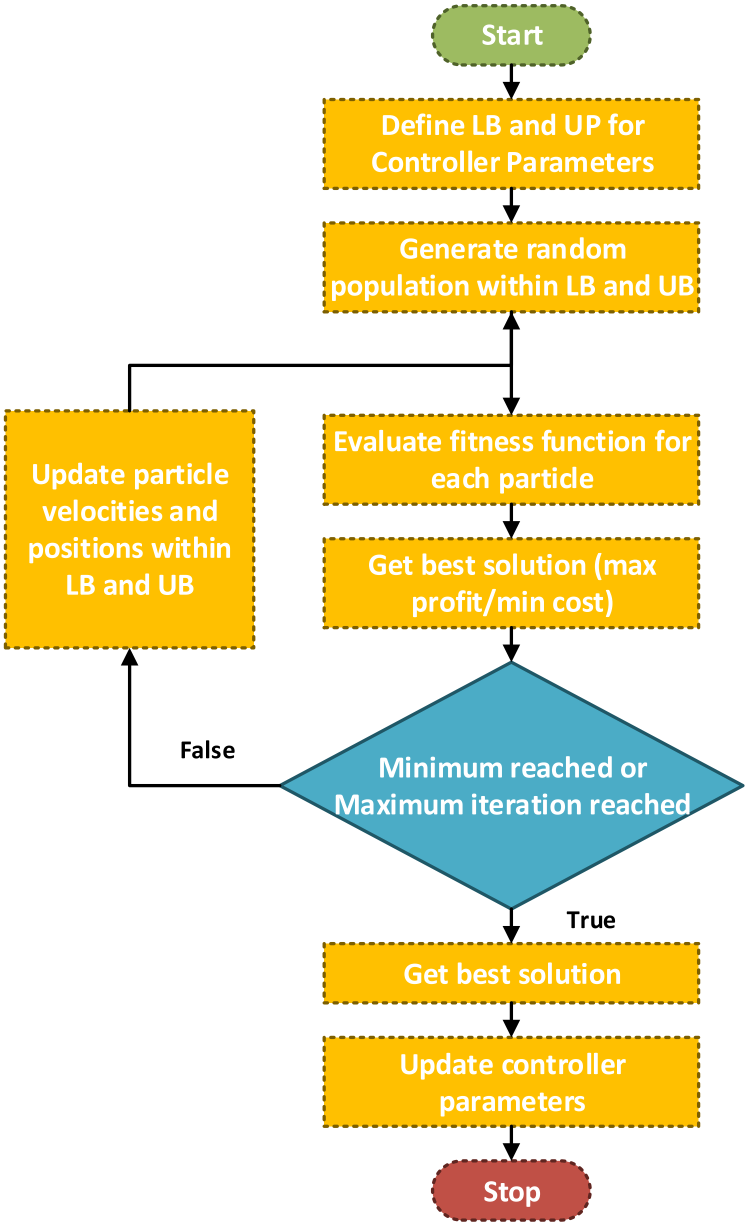

Start
Define LB and UP for Controller Parameters
Generate random population within LB and UB
Evaluate fitness function for each particle
Get best solution (max profit/min cost)
Minimum reached or Maximum iteration reached
False
Update particle velocities and positions within LB and UB
True
Get best solution
Update controller parameters
Stop

Figure 8.8: Flowchart of the PSO problem.

8.4.1 Results of the Online Optimization

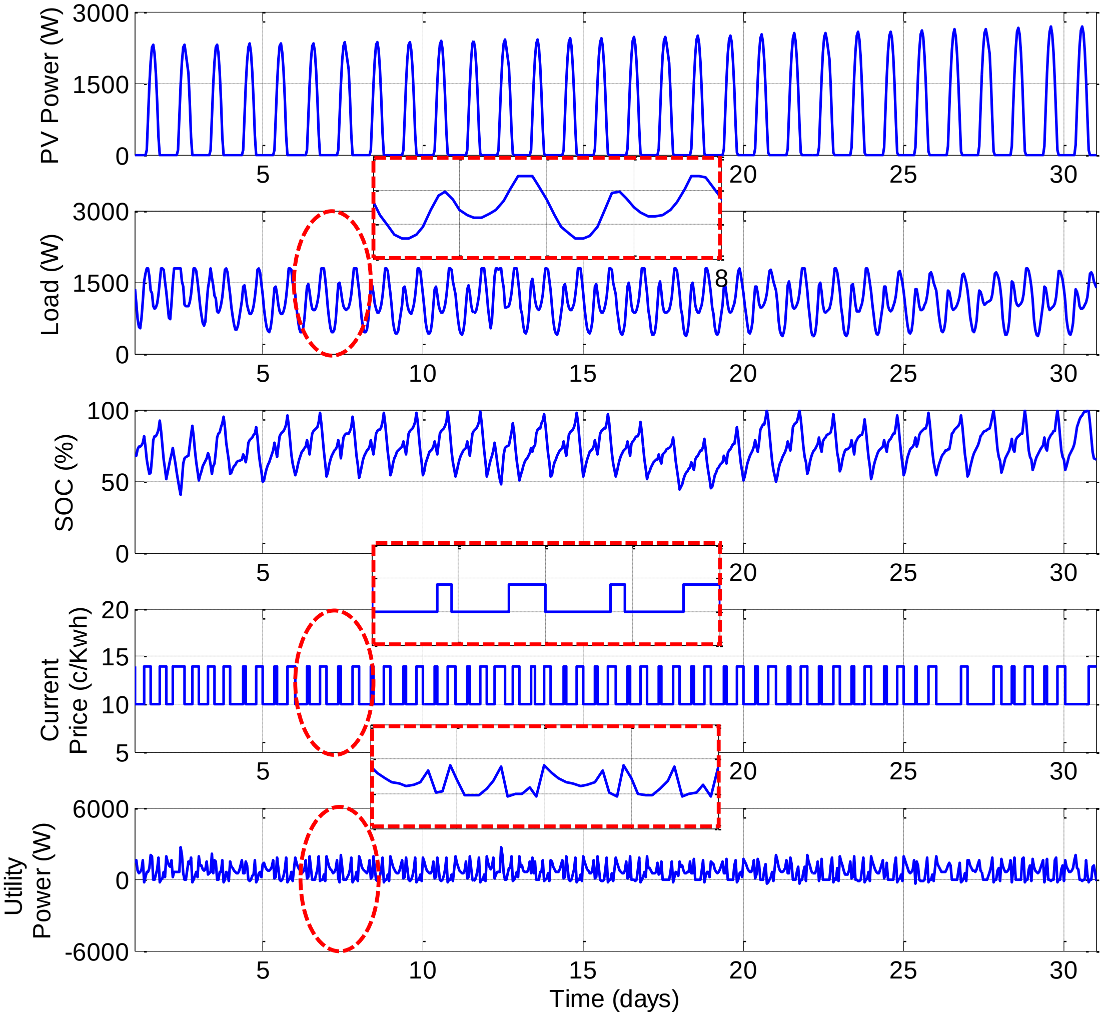


Figure 8.9: Winter Real-time Pricing with Optimized Parameters.

Figure 8.9 and Figure 8.10 show the result of applying the online parameter optimization approach to the proposed energy management algorithm with the same input parameters as in Section 8.3.1. In the winter season, the peak load at time t = 6.8 days, which corresponds to a high-price period, was reduced from 3600 W to 0 W utilizing the optimized parameters. This drastic reduction is because the optimized parameter allows better utilization of the energy storage devices in the nanogrid. Looking at the SOC profile, one notices that the SOC with optimization ranges between 40-100%. These results were similar in the summer season, where the peak load at t = 7 days dropped from 3600 W to 0 W with optimization. Similarly, the SOC with optimization ranges between 40-100%.

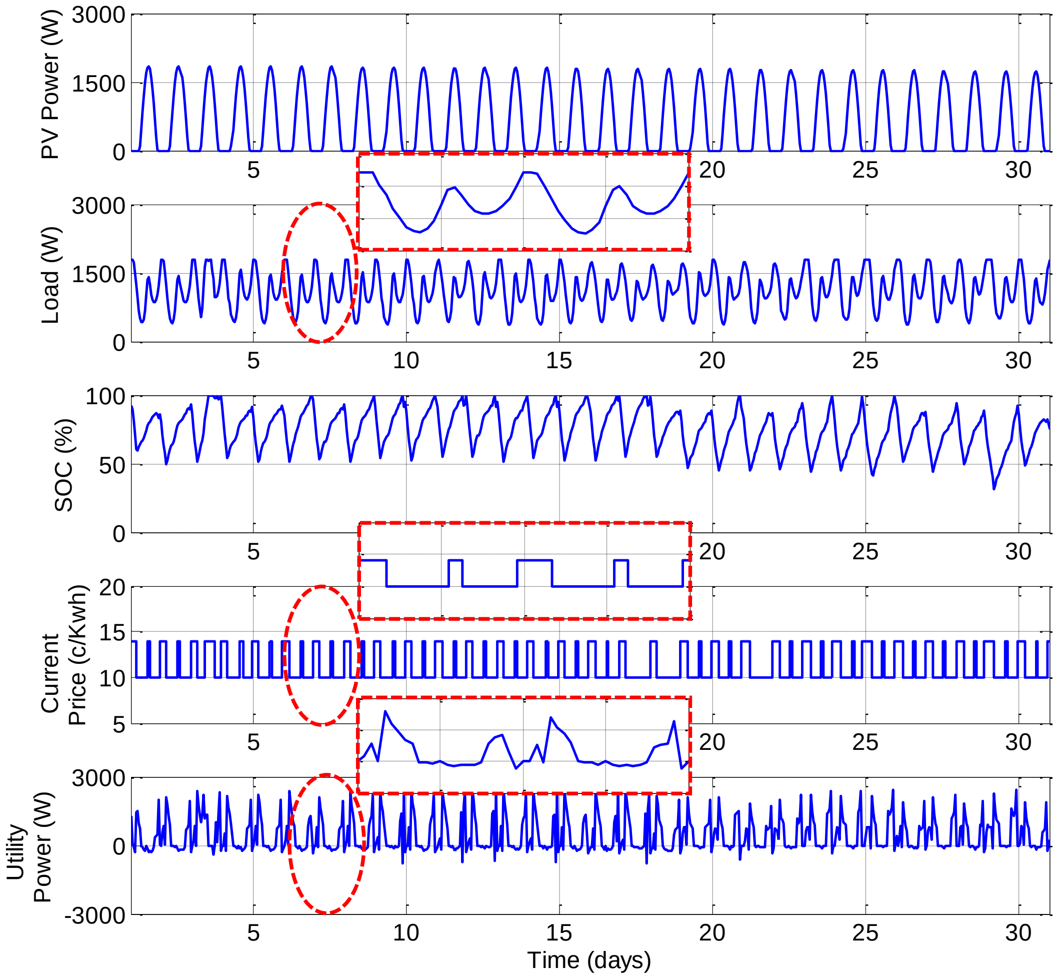


Figure 8.10: Summer Real-time Pricing with Optimized Parameters.

The results of the optimized parameters further assert the robustness of the developed energy management system in hiding the original customer load-demand profile, ensuring his/her privacy.

In addition to hiding the load demand profile, the energy management system provides customers with savings on their energy bills. Table 8.2 shows the anticipated savings that customers will achieve if they implemented the proposed energy management system. As can be seen in Table 8.2 for the real-time pricing scheme, the overall savings without optimization in the winter and summer seasons were 7.5% and 7.3%, respectively. The savings increased with the optimized parameters to 14.86% and 18.56% for the winter and summer seasons, respectively.

Table 8.2: Total Savings

Pricing	**Season**	**Total Cost Without EMS**	**Total Cost with EMS**	**Total Cost with Optimized EMS**
RTP	Winter	$ 145.74	$ 134.76	$ 124.08
	Summer	$ 90.81	$ 84.12	$ 73.95

8.5 Conclusion

In this chapter, an energy management system to hide the load demand profile for small microgrids was developed. An online optimization module accounting for history, current, and future system observations, utilizing particle swarm optimization, was also developed to enhance the results of the energy management system. The DDS middleware was selected as the communication backbone for the proposed framework for its robust failover mechanisms and rich set of QoS profiles. The energy management system was tested on actual residential energy consumption and irradiance data from Miami, Florida, and proved its effectiveness not only in hiding the customer’s load demand profile, but also in reducing the consumer’s bills and achieving flat peak load profiles

Chapter 9 Cyber Security Issues and Measures in Movable Systems

In today's world, EVs are becoming a reality, and therefore, their public charging will be required extensively. Being mobile by nature, EVs require the availability of information over wide geographical areas. Accordingly, this chapter briefly introduces the reader to the EV charging process and discusses both its vulnerabilities and cyber security requirements.

9.1 Introduction

The evolution rate of the power industry is increasing year after year. Over the past decade or two, this progress has been culminated by the emergence of the concept of Smart Grids, which is seen as a power system with real-time communication and control capabilities between energy providers on one hand, and energy consumers on the other. This modern power system model allows facilities to adopt new technologies and consumers to perceive new services. Utilizing communication technologies, the smart grid topology allows optimization of energy usage based on several factors, including environmental, price preferences, and system technical issues [1]. The term Smart Grid may refer to large systems with a large number of interacting energy production sources and energy consumption components. However, the same real-time communication and control capabilities can be applied to smaller scale systems, such as microgrids. The Department of Energy (DoE) defines a microgrid as "a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that act as a single controllable entity with respect to the grid" [143].

The concept of smart grid and microgrids widened the thought horizons and altered the perspectives of other industry sectors. One significant example is the introduction of EV in the transportation industry. EVs have their energy stored in batteries, and thus, have limited operation time. Therefore, EVs need to be recharged regularly. This fact introduces a whole new set of complexities to the operation of the power system, one of which is related to the effect EVs have on load-demand profiles. It is a common practice to divide energy consumers into residential, commercial, and industrial loads with load-demand profiles, which can be forecasted with an acceptable accuracy. Conversely, the energy consumed by EVs is stochastic in nature and is hard to predict. Several factors are incorporated in this issue, including individual driving preferences, manufacturers' specifications, trips' durations, etc. In fact, extensive ongoing research efforts have been placed on EVs, whether plug-in or wirelessly charged, in areas related to design optimization, charge control, efficiency boosting, large scale penetration of EVs into the smart grid, and many others [144][145][146][147].

In the Smart Grid era, EVs are gaining increased attention from various sectors. Given the increased demand for power, researches are looking at energy transaction algorithms between smart grids and electric vehicles, which not only sell power to EVs, but also draw power from them when this power is available and needed. Here, design of a bidirectional power flow model for EVs to support Grid-to-Vehicle (G2V) and Vehicle-to-Grid (V2G) power flow is viable. Looking at an EV as an energy consumer and an energy producer, simultaneously, with grid connection capabilities, leads us to define EVs as mobile microgrids or microgrids on wheels. Therefore, integration of EVs with the electric grid must be handled with extreme care and studied from various perspectives. While bringing many benefits to its owners in terms of cost savings, especially with the work being done on energy management systems, EV charging, when looked at as a cyber-physical process, puts car owners at risk of several attacks. As detailed later in this chapter, these attacks range from fabricating price information to completely shutting off a charging session. Like any other modern process, data exchange is a prerequisite for implementing EV charging. A major issue to look at, therefore, is the cyber-security of EV-Grid interactions.

9.2 On the Security of Electric Vehicle Charging

Today, each EV comes with an onboard Electronic Control Unit (ECU), which holds the vehicle's information and manages communication with charging stations. In fact, there are efforts to standardize EV-Grid communication, such as ISO/IEC 15118 and SAE J9231. However, robustness of these standards lags behind in issues related to identity authentication and data integrity, confidentiality, and privacy [148]. Since the electric vehicle technologies are in the early development stage, researchers could incorporate cyber-security into their design methodologies from the beginning of the development process. Here, it is important to realize the difference between the "functional safety" and "cyber-security," as shown in Figure 9.1. ECUs in the automotive industry are software-intensive units and are used to control a wide range of safety-critical issues, such as emergency brake assist, cruise control, and anti-lock braking systems, to name a few. Functional safety measures, such as IEC 61508, WD 26262, and MISRA Guidelines, were extensively addressed in the literature and are applied in the automotive industry to ensure continuous safe operation of the vehicle. From a cyber-security perspective, this chapter exploits vulnerabilities in the EV charging process and guides researchers into the direction of designing robust and highly secure EV ecosystem, which will safely integrate with the power grid.

It is well known that generation-load balancing in power systems is not an easy task to perform. The introduction of the EVs into this equation complicates things further. Therefore, a lot of feedback control is being developed and implemented on charging

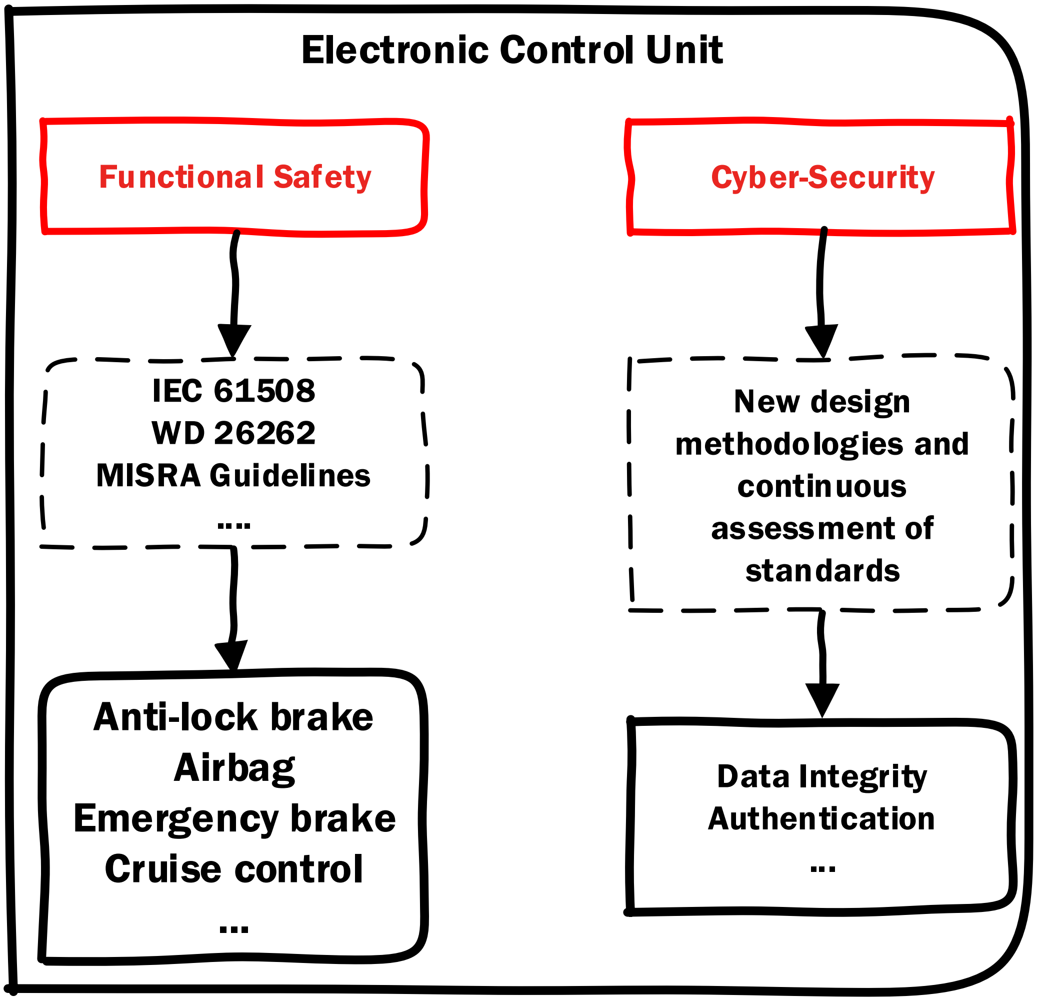


Figure 9.1: Functional Safety versus Cyber-Security.

stations to maintain a stable operation of the power system. That is, when connected to a charging station, the EV's charging profile must always be monitored and the EV itself must properly respond to command signals coming from the charging station. It is imperative that such commands come from an authentic source (the charging station) and reach the intended host (authenticated EV) unaltered. In a typical charging scenario, an EV arrives at a charging station and requests a charging session. As stated by IEC 15118, each EV must have a secret key stored in one of its ECUs. If the control station verifies the car's key through wireless communication, the charging process can be easily compromised by a MITM attack [149].

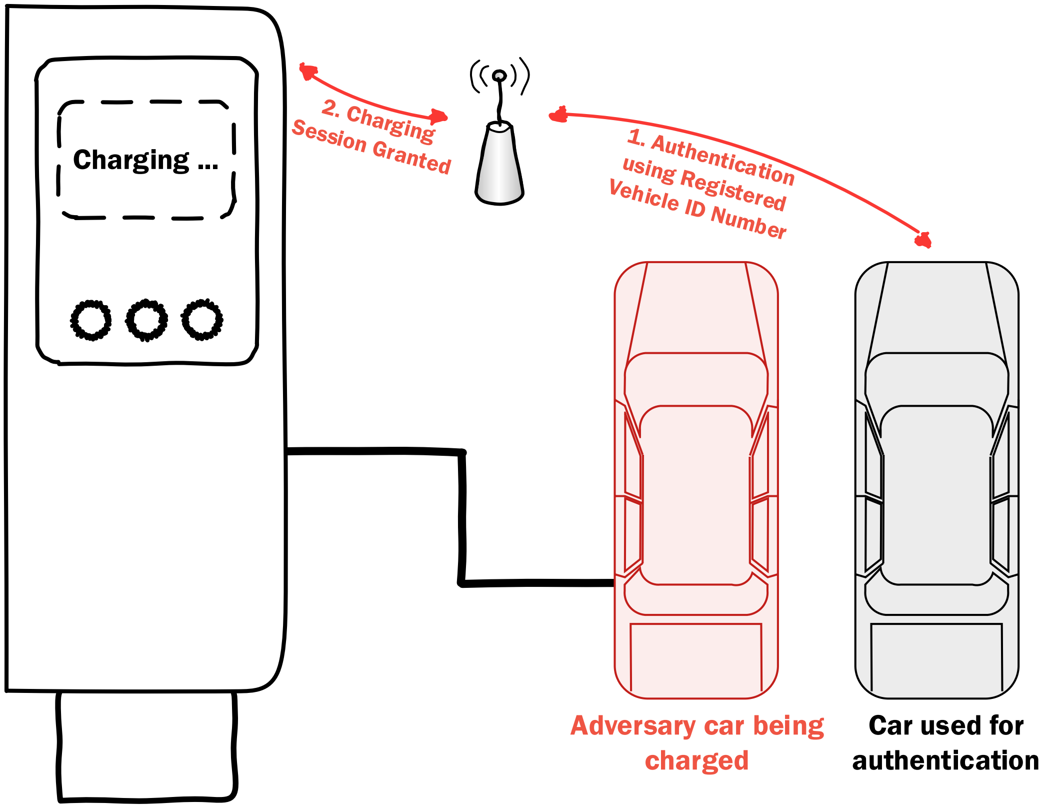

Charging ...
2. Charging Session Granted
1. Authentication using Registered Vehicle ID Number
Adversary car being charged
Car used for authentication

Figure 9.2: Man-in-the-Middle Attack.

As shown in Figure 9.2, a registered car with a valid key performs the authentication process, while the charging cable is connected to a stolen car. Not only do MITM attacks allow charging of unregistered cars, but also the firmware on the stolen car's ECU can be modified, such that the car will not to respond to commands from the charging station. It is to be noted that modifying a vehicle's ECU is not an easy task; however, it is not impossible. In fact, the authors in [150] were able to reprogram the firmware on a 2010 Ford Escape and a 2010 Toyota Prius ECU. Such an attack can tamper with or block vehicle-to-grid services.

Today, EV charging stations can be found in public places, such as airports, malls' parking, and roadsides. They are thus exposed to public and vulnerable networks. Although security practices are being standardized, it is imperative to continuously evaluate these standards to ensure high security levels. In fact, [151] disclosed two weaknesses in the NISTIR 7628 security framework in the context of EV infrastructure: one is related to the device authentication, while the other is related to location privacy. EVs charging process is thus prone to other types of attacks, including eavesdropping, denial of service, and data spoofing. It is important to understand how these attacks are implemented in order to study their consequences on the EV charging ecosystem and to take appropriate defense mechanisms.

1) **Eavesdropping**: In this type of attack, it is assumed that the attacker has access to the LAN of the charging station. After breaching the network, the attacker can sniff and store all the data being communicated between the charging station and the electric vehicle, as shown in Figure 9.3. For example,

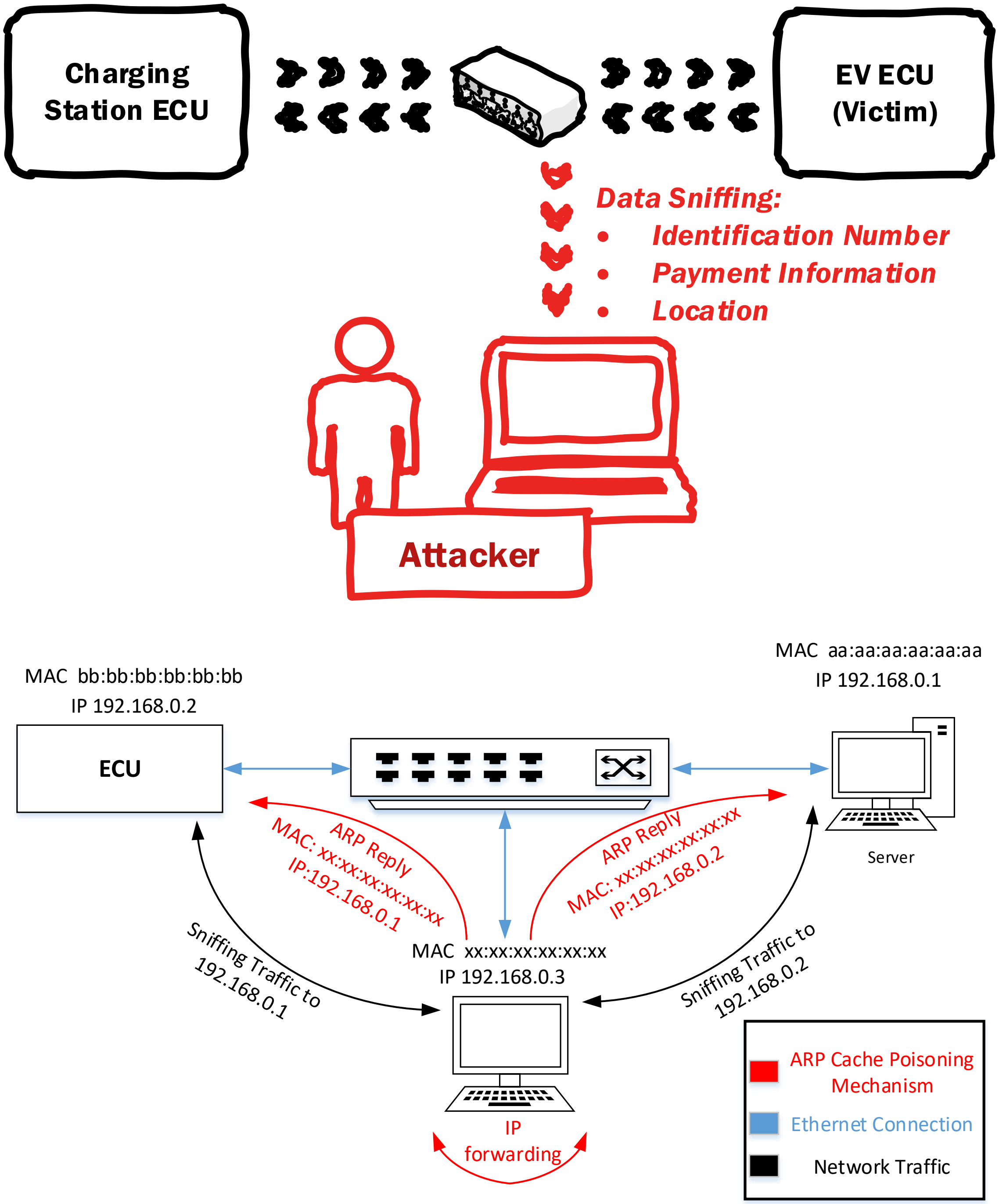


Figure 9.3: Sniffing the Victim's Car Data by ARP Cache Poisoning

the LAN can be penetrated by ARP Cache Poisoning.

IEC 15118 states that before initiating a charging session, an Identification and Authentication process must be performed in order to check the EV's identity. In this process, the car's Identification Number, which is a unique number stored in the vehicle's ECU, is exchanged with the charging station over the LAN. An attacker eavesdropping on the network can get hold of the ID of the EV. Although IEC 15118 established the requirement of using Transport Layer Security (TLS) for communicating data, TLS itself can be compromised [152][153]. With the knowledge of an EV's ID, an attacker can download this ID into an ECU of a stolen or an unregistered car and initiate detestable charging sessions. Not only is the EV's ID compromised, but also is the payment information of the attacked vehicle's owner. Charging of unregistered vehicles with fake ID's could lead to financial losses to both the car owner and to the power company.

2) **Denial of Service**: DoS attacks put the grid and the EV at risk. As shown in Figure 9.4, DoS is when an attacker blocks an entity from accessing a given service. Technically, DoS can be achieved by flooding the network. That is, the attacker attempts to obstruct the communication channels and expend the computational resources of the communicating nodes by delaying message delivery past a critical flooding rate. Here again, blocking a control signal coming from the charging station from reaching the ECU on the EV will prevent the vehicle from responding to serious grid commands, such as load shedding. An attacker might as well block the whole charging process, leading to inconvenience to the car's owner.

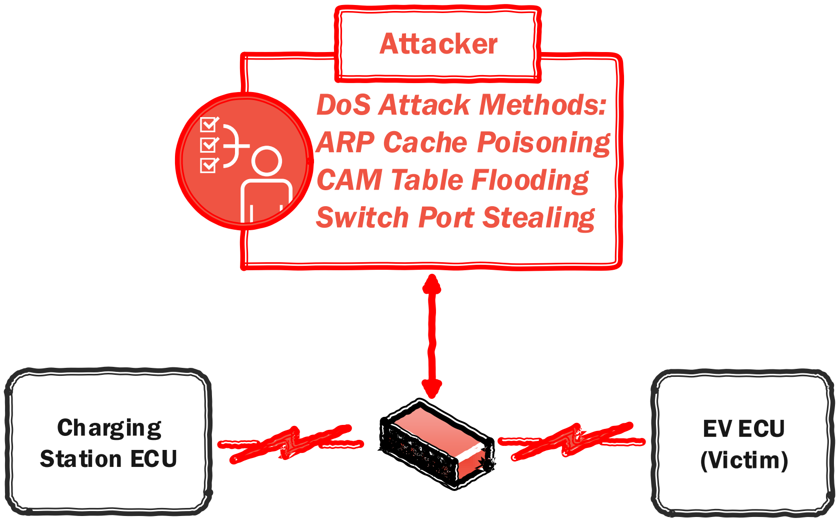


Figure 9.4: Denial of Service Attack.

3) **Data Spoofing**: After blocking the communication link between the charging station and an EV by a successful DoS attack, an attacker might inject false or malicious data both to the charging station and to the EV, as shown in Figure 9.5. With accurate knowledge of data and message types, the probability of a successful data spoofing attack is relatively high, especially in that IEC 15118 relies on the Public Key Infrastructure (PKI) approach. Although PKI helps authenticate identities on a network by a chain of trust of digital certificates, this infrastructure is as strong as the weakest device on the network. Interfering with communication messages in the context of EV charging opens the road to many fraudulent and hazardous acts. Such acts may cause network congestions, delay of successful delivery of legitimate messages, and disruption of services [13]. The authors in [154] also explained that modifying pricing signals sent to EVs to be very low could result in a large number of vehicles requesting charging sessions simultaneously,

leading to unstable grid operations or at a larger scale could possibly cause blackouts.

From a broader viewpoint, the attacks on a vehicle are not only limited to the charging process. The onboard computers, on current vehicles, can be categorized as computers for controlling the car's Entertainment Systems and computers for controlling and monitoring the car operation itself, such as keyless ignition, dynamic breaking, and anti-lock brake control, among others. Rigid security of communication channels between these onboard computers was not an issue when they first came into the market, as they were

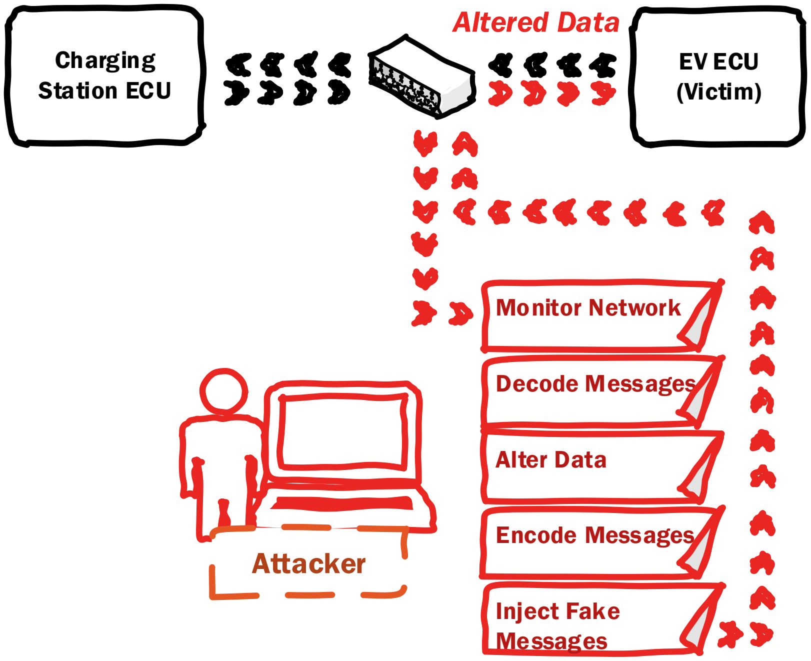


Figure 9.5: Altering Commands Sent to Victim EV by Data Spoofing.

considered to operate within the car's isolated network. However, entertainment systems of new vehicles come with Universal Serial Bus (USB) ports for connecting electronic devices and also internet connectivity over mobile networks (GPRS, 3G, and 4G). The vehicle’s network is no longer isolated. Using the entertainment system’s network as an entry point, an attacker can benefit from open ports to hack into other ECU’s controlling critical operations, such as braking, steering, or even shutting the vehicle off. In fact, researchers Miller and Valasek published their work “Remote Exploitation of an Unaltered Passenger Vehicle” in August 2015, where they were able to hack into a Chrysler Jeep remotely and control any service in the car, which is controllable over the in-car network. The researchers used the U-Connect entertainment system in the Jeep as an entry point to the car network. This astounding work got a lot of attention from the media, as it was applied on a real commercial vehicle, which people were already driving. Yet again, it is highly recommended to look into securing all communication links inside a vehicle starting from the planning and design stages.

The likelihood of electric vehicles charging process to be besieged by various attacks is now evident. Each type of attack has its own objective and critical consequences both on the grid and on the EV itself. These consequences, being grid instability, energy theft, false identification, jeopardizing EV owner’s personal security, and others are not to be belittled. IEC 15118 did publish security standards for communications regarding charging electric vehicles, but its resiliency against the previously mentioned attacks is still debatable. Significant thought is being devoted to designing EV charging stations, and considerations regarding the security of these systems ought to be considered and utilized at early design stages.

Recently, industry and academic research are gaining insight into the importance of securing the EV charging process, and work is being done to defend against some types of attacks. In 2014, the authors in [151] proposed a cyber-physical EV authentication mechanism to eliminate the substitution attack, which is a type of MITM attack. Following the successful vehicle authentication by the charging station occurring over a wireless communication medium, another physical challenge is sent over the control pin of the charging cable to make sure that the claimed authenticated car is actually the one being charged, and not an adversary one. This approach is tailored and well-suited for plug-in electric vehicles and does not account for the next generation wirelessly-charged electric vehicles. In the latter, there is no charging cable, and thus, the proposed defense mechanism is not adequate.

In another recent approach, the authors in [155] looked at a comprehensive demand-side management system, with stringent security mechanisms, for safe integration of electric vehicles into the smart grid. First, the authors identified the importance of power grids having full access control of the charging process in order to ensure stable and reliable grid operation. Then, they started designing their system from a physical and cyber perspective, simultaneously. Key to their work, is the digital identity assurance of electric vehicles for safe integration with the power grid. One might argue that this system uses legacy cryptographic techniques for digital devices' identification and data encryption; however, a big advantage of this work is the thought process followed by the authors, where security was tailored in starting from the planning and system design phases.

The smart grid, as proposed in [156], is divided into five broad components, namely "Smart Power Grid," "Smart Consumer," "Smart Electricity Service," "Smart Renewable," and "Smart Transportation". These are all still envisioned futuristic concepts, which are undergoing extensive research and investigation both nationally and internationally. Contrary to the legacy power systems' communications networks, which were designed to operate in close environments, it is about time that communication security is placed at the top of the table in designing the future smart power grid in all its components, including the EV ecosystem.

9.3 Conclusion

This chapter exploited the shortcomings of current EV-Grid integration systems and investigated the drawbacks of security breaches on the power grid, the EV itself, and the safety of the vehicles' owners, in hope of advocating a new thought philosophy that gives the same importance for system operation and system security. Electric Vehicles are indeed microgrids on wheels, but for them to be widely integrated with the smart grid, their security has to be placed in the front seat, rather than being left behind.

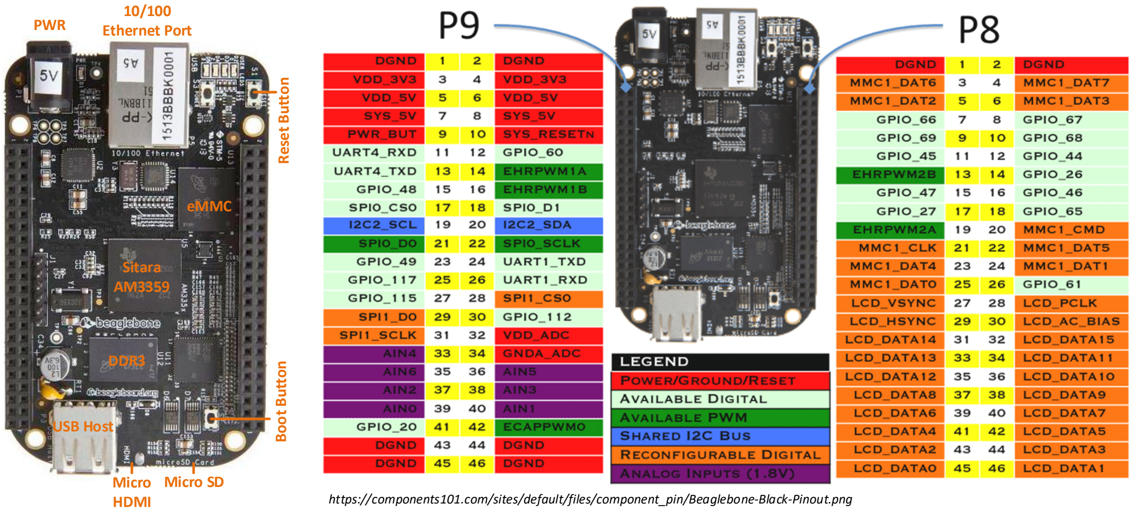
Chapter 10 Working with Embedded Linux

This chapter is aimed to be as a tutorial on important tools and skillsets that will allow the readers to develop technologies related to secure communication and intelligent systems, which facilitate real-time, distributed data processing, and multi-level decision making in the heterogeneous power and energy infrastructure. By the end of this chapter, the reader will be able to develop experimentation with the GOOSE protocol using the libiec61850 open source library and RTI DDS on embedded microcontrollers running under Linux kernels. The readers will be taught how to interface with hardware equipment through analogue inputs and outputs of microntrollers.

10.1 Working with Embedded Microcontrollers Running under Linux.

The Beaglebone Black (BBB) is a powerful microcontroller, which enables researchers to prototype wide variety of technologies related to energy cyber-physical systems. Figure 10.1 is a pictorial of the BBB. The BBB is powered by a 5V DC power supply. It has a 10/100 Ethernet port, a USB host, a micro HDMI, and a micro SD slot. It can be connected to WiFi through a dongle connected to its USB host. The BBB, also, has a 512 MB DDR3 RAM, an eMMC, and a Sitara AM3359 ARM microprocessor.

The BBB comes with a pair of header pins, annotated as P9 and P8 in Figure 10.1. These pair of header pins are of paramount importance to us, since they allow the BBB to interface with the real world equipment through analogue inputs and digital inputs and outputs.



P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	GPIO_112
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND

LEGEND
Power/Ground/Reset
Available Digital
Available PWM
Shared I2C Bus
Reconfigurable Digital
Analog Inputs (1.8V)

P8

DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

https://components101.com/sites/default/files/component_pin/Beaglebone-Black-Pinout.png

Figure 10.1: The Beaglebon Black and its header pins.

10.1.1 Installing the Debian Image on the BBB.

To configure the BBB, we need to setup an operating system on it. To do that, we will need a micro SD card with a minimum capacity of 4 GB.

- Step 1: Download an image flasher software on your Windows or Linux device. One possible flashing software, which could be used, is found in [157].
- Step 2: Download the Debian Linux image for your BBB from [158].
- Step 3: Connect the micrSD card to your computer and burn the image onto it using the software downloaded in Step 1.
- Step 4: Insert the micro SD card, which has the Debian image on it, and hold onto the Boot button. Next, connect the power supply and keep pressing the Boot button until all 4 LEDs on the upper left corner of the BBB are lit.

At this stage, the Beaglebone black is running the Debian image, which is installed on the micro SD card connected to it. To access the BBB, connect it to your computer via the USB that comes with it.

Once the BBB is connected, a new Ethernet interface appears on your computer with a .7 subnet. By default the BBB will have an IP address of 192.168.7.2. We connect to the BBB with a secure shell (SSH) connection through a terminal.

- Step 5: Open a terminal on your computer and connect to the BBB by typing the following command: *ssh user@192.168.7.2*, where *user* is the username you are trying to log in to. User could be root, if you want to log in as root, or debian, which is the default username on the BBB. At this stage, use the root user. The default password for the BBB is *temppwd*.

Once you are logged in, it is preferable that you flash the Debian image onto the eMMC.

- Step 6: To flash the Debian image onto the eMMC of the BBB, change your directory with the following command: *cd /opt/scripts/tools/eMMC/*
  then run the shell script *sudo ./init-eMMC-flasher-v3.sh*
- Step 7: Shutdown the BBB using the PWR button, remove the micro SD card, and turn the BBB back on. You are now running Debian from the eMMC.

10.1.2 Network Preferences.

To connect the BBB to your LAN, you need to setup the network preferences.

- Step 1: Connect the BBB to your computer through the USB cable and login into it as described in Section 10.1.1 (*ssh user@192.168.7.2,*).

- Step 2: Use the following command to open and edit the text file, which contains the network settings: *nano /etc/networks/interfaces*. This step requires root privileges.
- Step 3: To setup a static IP, add the following to the opened text file:

  ```
  auto INTERFACE_NAME
  iface INTERFACE_NAME inet static
  address IP_ADDRESS_OF_CHOICE
  netmask NETMASK
  gateway GATEWAY
  ```

- Step 4: To setup a dynamic IP, add the following to the opened text file:

  ```
  auto INTERFACE_NAME
  iface INTERFACE_NAME inet dhcp
  ```

Once you configured your network interfaces, reboot the BBB to apply the changes. Now you can remotely login into the BBB from any computer on the same network using the newly assigned IP address.

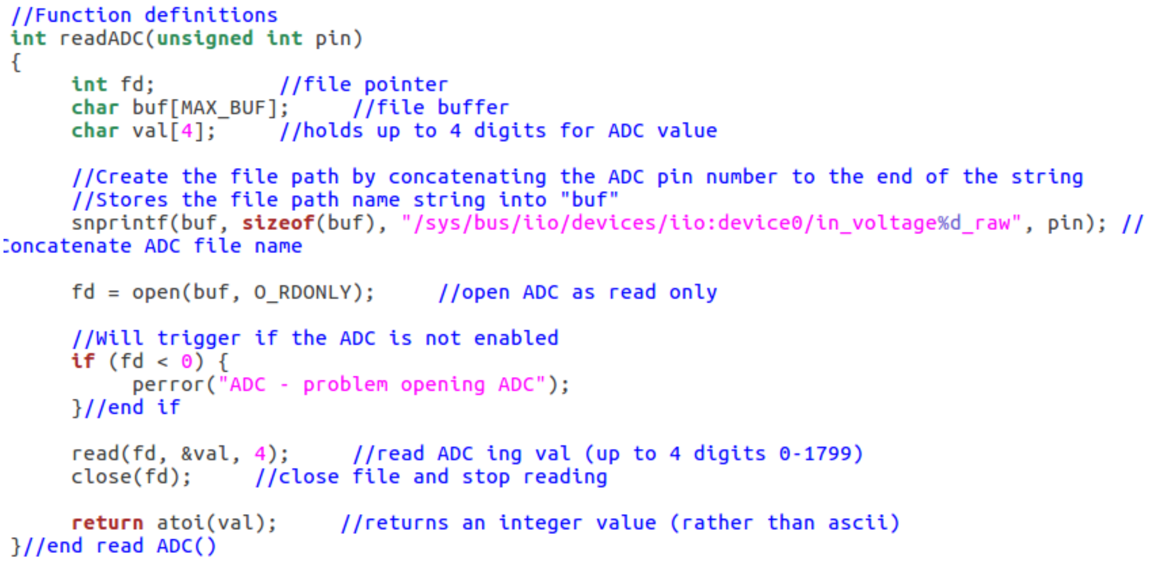
10.1.3 Sampling and Analogue Input.

The BBB has 7 analogue input pins on its P9 header. These are pins 33, 35, 36, 37, 38, 39, and 40. Before using these pins as analogue input pins, we need to let the microprocessor know that these pins will be used as analogue input pins. To do that, we need to define a device tree overlay file and echo it into the BBB’s cape manager. We will go through creating device tree overlay files and building them in later sections. However, for analogue inputs, there already is an environment variable loaded in Debian for BBB that does this. The following command will perform the intended task:

echo BB-ADC > /sys/devices/platform/bone_capemgr/slots

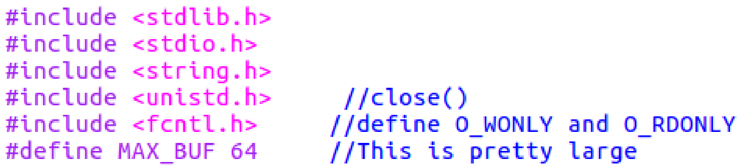
It is very important to note that the analogue input pins read voltages between 0 and 1.8 V. Therefore, any reading from the real world must be preconditioned to be within the specified voltage limits.

To automate data readings and integrate them in your code, the following function could be utilized. This function is written in C.



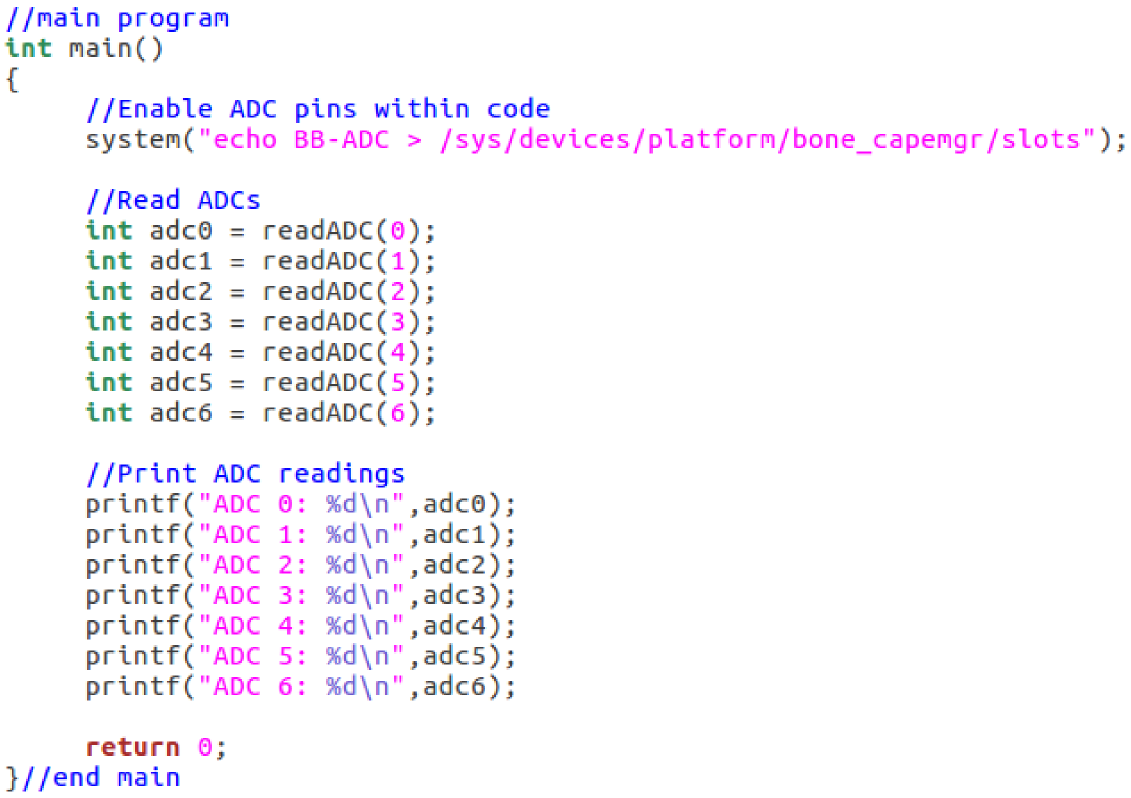
```
//Function definitions
int readADC(unsigned int pin)
{
     int fd;          //file pointer
     char buf[MAX_BUF];     //file buffer
     char val[4];     //holds up to 4 digits for ADC value

     //Create the file path by concatenating the ADC pin number to the end of the string
     //Stores the file path name string into "buf"
     snprintf(buf, sizeof(buf), "/sys/bus/iio/devices/iio:device0/in_voltage%d_raw", pin); //
Concatenate ADC file name

     fd = open(buf, O_RDONLY);     //open ADC as read only

     //Will trigger if the ADC is not enabled
     if (fd < 0) {
          perror("ADC - problem opening ADC");
     }//end if

     read(fd, &val, 4);     //read ADC ing val (up to 4 digits 0-1799)
     close(fd);     //close file and stop reading

     return atoi(val);     //returns an integer value (rather than ascii)
}//end read ADC()
```

The readADC() function takes the number of the ADC pin to read from as input (0 to 6) and returns the read value as output. The function creates a file pointer (fd), a file buffer (buf), and a character variable (4), which holds up to 4 digits of the ADC value read. Next, the ADC pin is treated as a read only file, from which the values can be read through the read() function. Finally, the readADC() function requires the following header files.



```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>      //close()
#include <fcntl.h>      //define O_WONLY and O_RDONLY
#define MAX_BUF 64      //This is pretty large
```

An example C program that utilizes this function to read the ADC values from the 7 ADC pins and print their values is shown below:

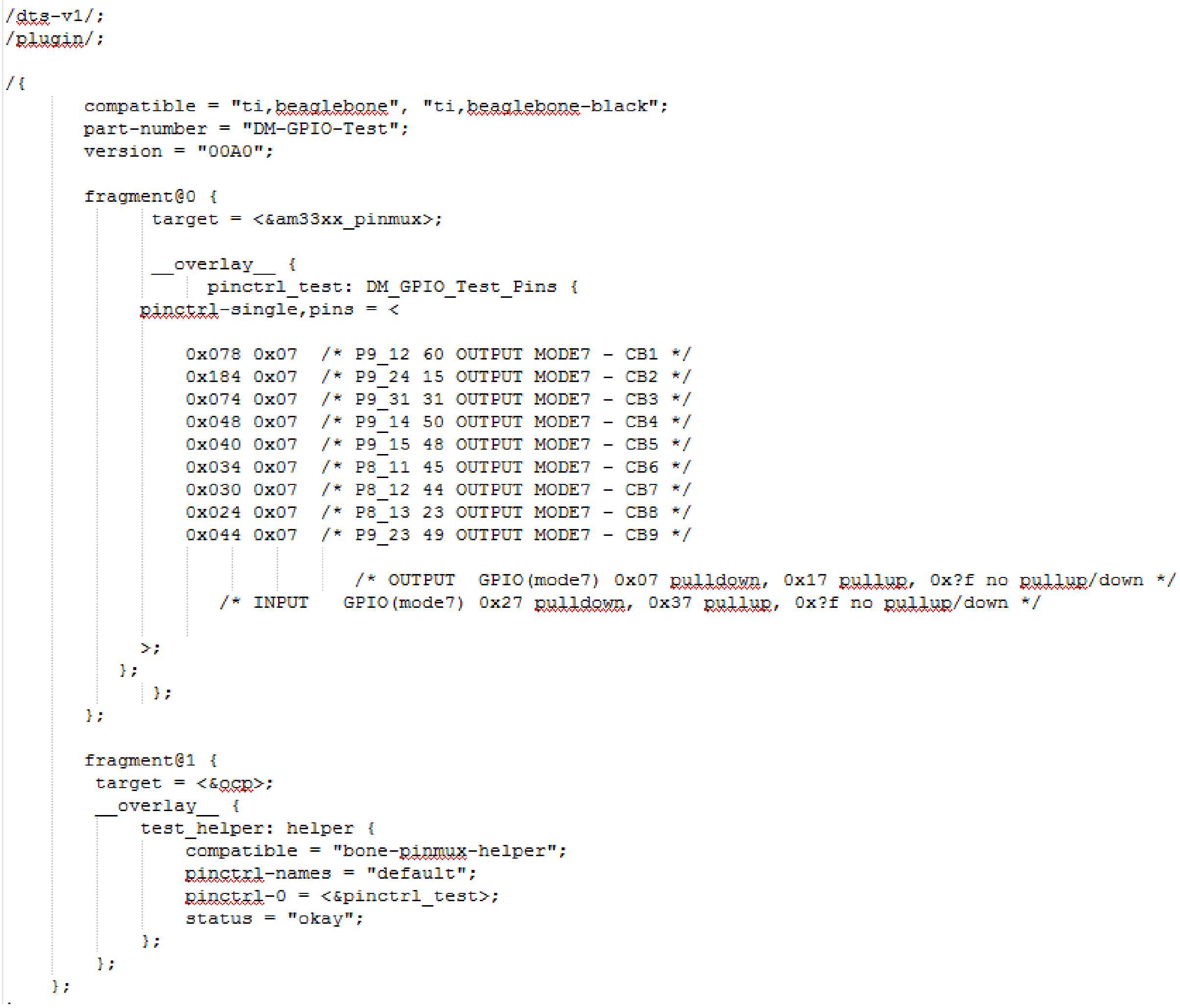


```
//main program
int main()
{
     //Enable ADC pins within code
     system("echo BB-ADC > /sys/devices/platform/bone_capemgr/slots");

     //Read ADCs
     int adc0 = readADC(0);
     int adc1 = readADC(1);
     int adc2 = readADC(2);
     int adc3 = readADC(3);
     int adc4 = readADC(4);
     int adc5 = readADC(5);
     int adc6 = readADC(6);

     //Print ADC readings
     printf("ADC 0: %d\n",adc0);
     printf("ADC 1: %d\n",adc1);
     printf("ADC 2: %d\n",adc2);
     printf("ADC 3: %d\n",adc3);
     printf("ADC 4: %d\n",adc4);
     printf("ADC 5: %d\n",adc5);
     printf("ADC 6: %d\n",adc6);

     return 0;
}//end main
```

10.1.4 Device Tree Overlays and Digital Inputs and Outputs

A Device Tree Overlay, or DTO, is basically data structure of peripherals and their properties that describe non-discoverable hardware [159]. A DTO data structure is used to assign the general purpose input output (GPIO) pins on the BBB's header pins as either digital inputs or outputs. All GPIO pins and their numbers are shown in Figure 10.1. Loading a DTO into the bootloader's runtime-accessible location of the BBB requires creating a .dtc file and compiling it with a dtc compiler to create what is called a device tree blob (.dtb).

An example dts file from [160] is shown below.



```
/dts-v1/;
/plugin/;

/{
       compatible = "ti,beaglebone", "ti,beaglebone-black";
       part-number = "DM-GPIO-Test";
       version = "00A0";

       fragment@0 {
             target = <&am33xx_pinmux>;

             __overlay__ {
                  pinctrl_test: DM_GPIO_Test_Pins {
            pinctrl-single,pins = <

                0x078 0x07  /* P9_12 60 OUTPUT MODE7 - CB1 */
                0x184 0x07  /* P9_24 15 OUTPUT MODE7 - CB2 */
                0x074 0x07  /* P9_31 31 OUTPUT MODE7 - CB3 */
                0x048 0x07  /* P9_14 50 OUTPUT MODE7 - CB4 */
                0x040 0x07  /* P9_15 48 OUTPUT MODE7 - CB5 */
                0x034 0x07  /* P8_11 45 OUTPUT MODE7 - CB6 */
                0x030 0x07  /* P8_12 44 OUTPUT MODE7 - CB7 */
                0x024 0x07  /* P8_13 23 OUTPUT MODE7 - CB8 */
                0x044 0x07  /* P9_23 49 OUTPUT MODE7 - CB9 */

                               /* OUTPUT  GPIO(mode7) 0x07 pulldown, 0x17 pullup, 0x?f no pullup/down */
                   /* INPUT   GPIO(mode7) 0x27 pulldown, 0x37 pullup, 0x?f no pullup/down */

            >;
          };
             };
       };

       fragment@1 {
        target = <&ocp>;
        __overlay__ {
            test_helper: helper {
                compatible = "bone-pinmux-helper";
                pinctrl-names = "default";
                pinctrl-0 = <&pinctrl_test>;
                status = "okay";
            };
        };
    };
```

You can edit the file above to describe the hardware (pins) you wish to use. Set them as input or output, specify whether or not to use the internal pull up/down resistors …. You can use the P9 and P8 header tables, shown in Figure 10.1, to identify the correct addresses of the pins.

To compile the dts file and load it into the BBB:

- Step 1: Set the environment variables. Do this once and for all by adding the following lines of code to the ".bashrc" file in the beagelbone bone black.

```
export SLOTS=/sys/devices/platform/bone_capemgr/slots

export PINS=/sys/kernel/debug/pinctrl/44e10800.pinmux/pins
```

- Step 2: Copy the ".dts" file into the BBB (anywhere, we just need to compile it on it)
- Step 3: Compile the ".dts" file into a ".dtbo" file using:

dtc -O dtb -o xxxxxx-00A0.dtbo -b -0 - @ xxxxxxx.dts (notice we added 00A0 at the end of the name of .dtbo file)

- Step 4: copy the ".dtbo" file into /lib/firmware

```
cp xxx.dtbo /lib/firmware
```

- Step 5: Apply the new changes by:

sudo sh -c "echo xxxxx > $SLOTS" (notice -00A0.dtbo was excluded from file name in this step)

sudo sh -c not always necessary

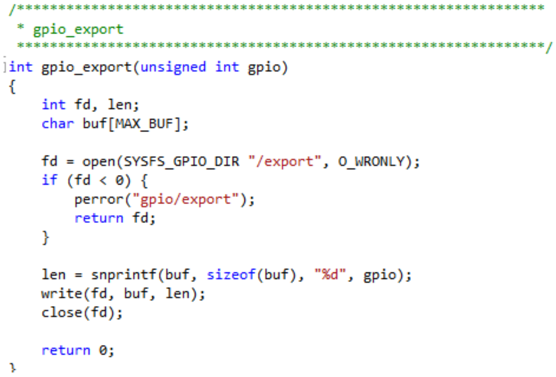
- Step 6: Check if the changes were done by *cat $SLOTS*. A new field will be available with the new configurations.

Once the device tree overlay is loaded, we can start using the assigned pins as output or input digital pins. A library written in C was developed by [161] could be used. This library has 7 subroutines of interest in it.

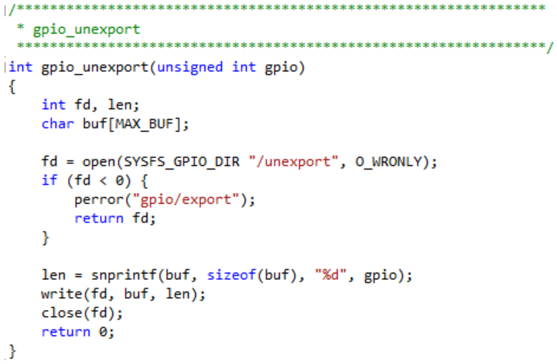
1. gpio_export: this function is used to create a folder in the /sys/class/gpio that has the number on the assigned gpio pin (e.g. gpio60). This folder will then have files that are used to control the state of the pin.
2. gpio_unexport: this function is used to unexport a pin. Once you are done using the pin, it is often a good practice to unexport the pin so that it could be used for other purposes.
3. gpio_set_dir: this function is used to set the direction of the digital pin as either an input pin or an output pin.
4. gpio_set_value: this function is used to the set the value of the pin to HIGH (3.3V) or LOW (0V), if the pin were an output pin.
5. gpio_get_value: this function is used to read the value of the pin, if the pin were an input pin. It returns 1 (True) if the input was HIGH, and 0 (False) if the input was LOW.
6. gpio_fd_open: this function treats the pin as file onto which we can write the pin’s state, or from which we can read the pin’s state.
7. gpio_fd_close: this function closes the file.

Note that the library uses the following headers:

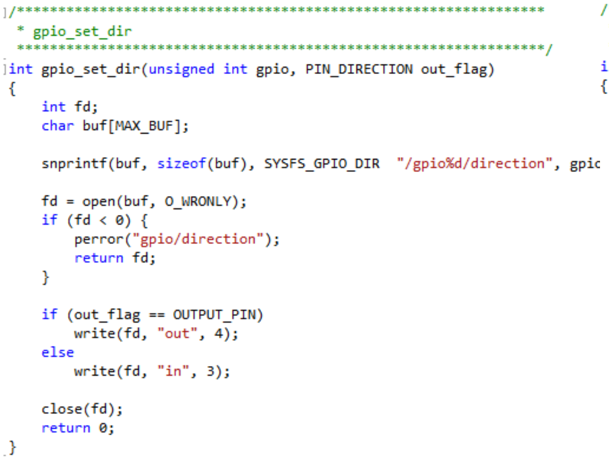
```
#include "SimpleGPIO.h" #include <stdio.h> #include <stdlib.h> #include <string.h>
#include <errno.h> #include <unistd.h> #include <fcntl.h> #include <poll.h>
```



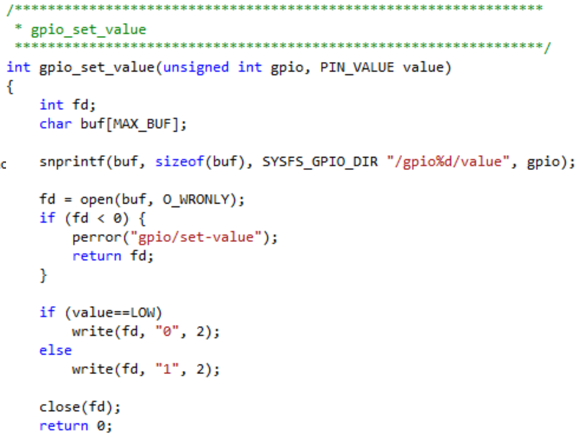
```
/****************************************************************
 * gpio_export
 ****************************************************************/
int gpio_export(unsigned int gpio)
{
    int fd, len;
    char buf[MAX_BUF];

    fd = open(SYSFS_GPIO_DIR "/export", O_WRONLY);
    if (fd < 0) {
        perror("gpio/export");
        return fd;
    }

    len = snprintf(buf, sizeof(buf), "%d", gpio);
    write(fd, buf, len);
    close(fd);

    return 0;
}
```



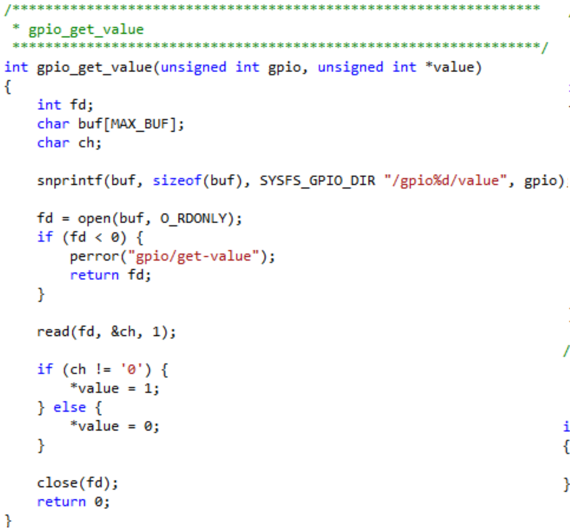
```
/****************************************************************
 * gpio_unexport
 ****************************************************************/
int gpio_unexport(unsigned int gpio)
{
    int fd, len;
    char buf[MAX_BUF];

    fd = open(SYSFS_GPIO_DIR "/unexport", O_WRONLY);
    if (fd < 0) {
        perror("gpio/export");
        return fd;
    }

    len = snprintf(buf, sizeof(buf), "%d", gpio);
    write(fd, buf, len);
    close(fd);
    return 0;
}
```



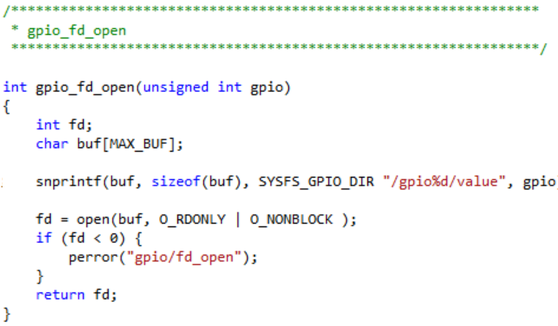
```
/****************************************************************
 * gpio_set_dir
 ****************************************************************/
int gpio_set_dir(unsigned int gpio, PIN_DIRECTION out_flag)
{
    int fd;
    char buf[MAX_BUF];

    snprintf(buf, sizeof(buf), SYSFS_GPIO_DIR  "/gpio%d/direction", gpic

    fd = open(buf, O_WRONLY);
    if (fd < 0) {
        perror("gpio/direction");
        return fd;
    }

    if (out_flag == OUTPUT_PIN)
        write(fd, "out", 4);
    else
        write(fd, "in", 3);

    close(fd);
    return 0;
}
```



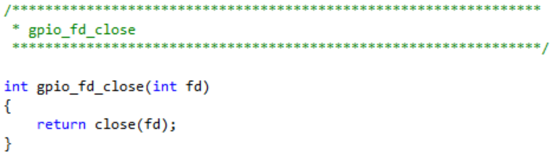
```
/****************************************************************
 * gpio_set_value
 ****************************************************************/
int gpio_set_value(unsigned int gpio, PIN_VALUE value)
{
    int fd;
    char buf[MAX_BUF];

    snprintf(buf, sizeof(buf), SYSFS_GPIO_DIR "/gpio%d/value", gpio);

    fd = open(buf, O_WRONLY);
    if (fd < 0) {
        perror("gpio/set-value");
        return fd;
    }

    if (value==LOW)
        write(fd, "0", 2);
    else
        write(fd, "1", 2);

    close(fd);
    return 0;
```



```
/****************************************************************
 * gpio_get_value
 ****************************************************************/
int gpio_get_value(unsigned int gpio, unsigned int *value)
{
    int fd;
    char buf[MAX_BUF];
    char ch;

    snprintf(buf, sizeof(buf), SYSFS_GPIO_DIR "/gpio%d/value", gpio)

    fd = open(buf, O_RDONLY);
    if (fd < 0) {
        perror("gpio/get-value");
        return fd;
    }

    read(fd, &ch, 1);

    if (ch != '0') {
        *value = 1;
    } else {
        *value = 0;
    }

    close(fd);
    return 0;
}
```



```
/****************************************************************
 * gpio_fd_open
 ****************************************************************/

int gpio_fd_open(unsigned int gpio)
{
    int fd;
    char buf[MAX_BUF];

    snprintf(buf, sizeof(buf), SYSFS_GPIO_DIR "/gpio%d/value", gpio);

    fd = open(buf, O_RDONLY | O_NONBLOCK );
    if (fd < 0) {
        perror("gpio/fd_open");
    }
    return fd;
}
```

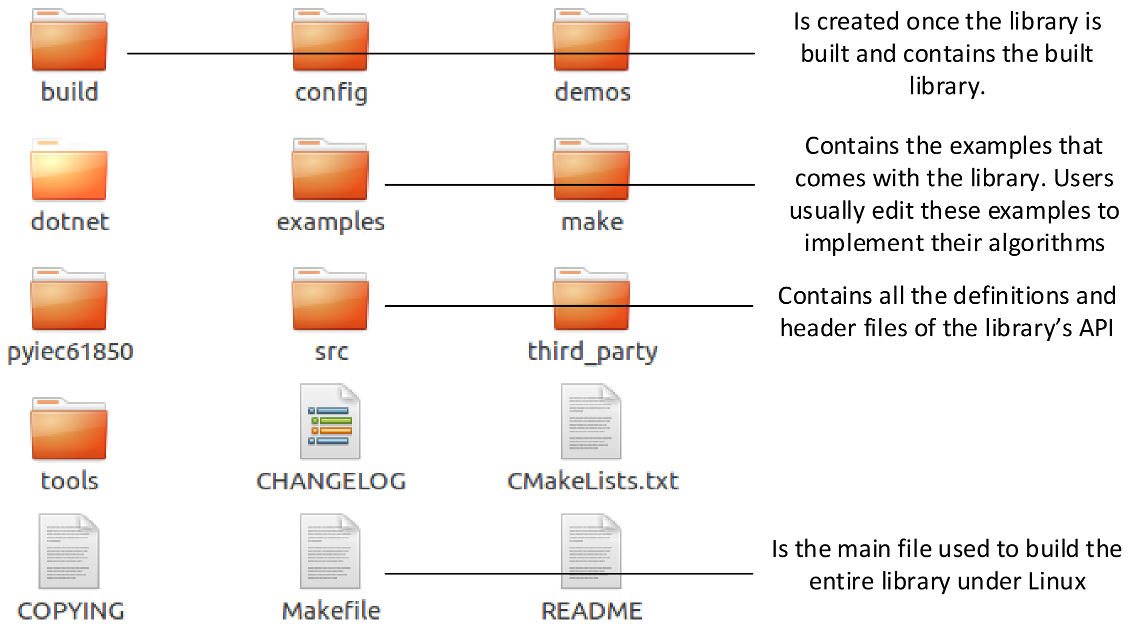


```
/****************************************************************
 * gpio_fd_close
 ****************************************************************/

int gpio_fd_close(int fd)
{
    return close(fd);
}
```

10.2 Understanding the Open Source libiec61850 Library.

The open source libiec61850 library is a useful tool for creating application that need to comply with the IEC 61850 standard. In this section we will focus on creating a GOOSE publisher and subscriber. The complete library can be downloaded from [75].

The home directory of the library is shown in Figure 10.2.



Is created once the library is built and contains the built library.

Contains the examples that comes with the library. Users usually edit these examples to implement their algorithms

Contains all the definitions and header files of the library's API

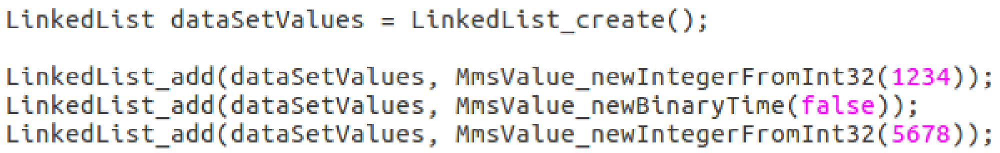
Is the main file used to build the entire library under Linux

Figure 10.2: Home directory of libiec61850.

10.2.1 Creating a GOOSE Publisher.

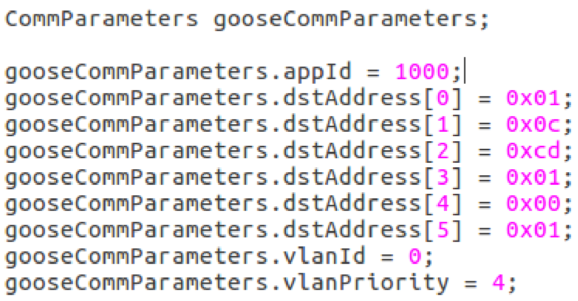
To create a GOOSE publisher using the libiec61850 library, you must follow the following steps.

- Step 1: create an empty linked list data structure and start adding the data fields of the GOOSE message according to their type. A linked lit is linear data structure whose elements are linked by pointers. You can add Boolean, Integer, or Float fields. The proper function definition for each data type can be found in the src/goose folder. The following example creates a linked list structure called dataSetValues, and adds to it two integers and a binary time field.



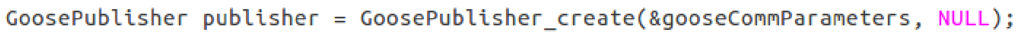
```
LinkedList dataSetValues = LinkedList_create();

LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt32(1234));
LinkedList_add(dataSetValues, MmsValue_newBinaryTime(false));
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt32(5678));
```

- Step 2: configure the parameters of the GOOSE message. To do that, the library defines a data structure called CommParameters. In this structure, you can define the APPID, the destination address, the VLAN ID and the VLAN Priority, as in the example below.



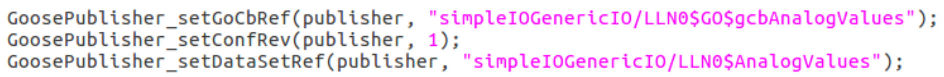
```
CommParameters gooseCommParameters;

gooseCommParameters.appId = 1000;
gooseCommParameters.dstAddress[0] = 0x01;
gooseCommParameters.dstAddress[1] = 0x0c;
gooseCommParameters.dstAddress[2] = 0xcd;
gooseCommParameters.dstAddress[3] = 0x01;
gooseCommParameters.dstAddress[4] = 0x00;
gooseCommParameters.dstAddress[5] = 0x01;
gooseCommParameters.vlanId = 0;
gooseCommParameters.vlanPriority = 4;
```

- Step 3: create the publisher using the parameters that you previously defined.



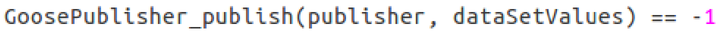
```
GoosePublisher publisher = GoosePublisher_create(&gooseCommParameters, NULL);
```

- Step 4: configure the goCbRef, ConfRev, and DataSetRef fields using the funcitons dedicated to them.



```
GoosePublisher_setGoCbRef(publisher, "simpleIOGenericIO/LLN0$GO$gcbAnalogValues");
GoosePublisher_setConfRev(publisher, 1);
GoosePublisher_setDataSetRef(publisher, "simpleIOGenericIO/LLN0$AnalogValues");
```

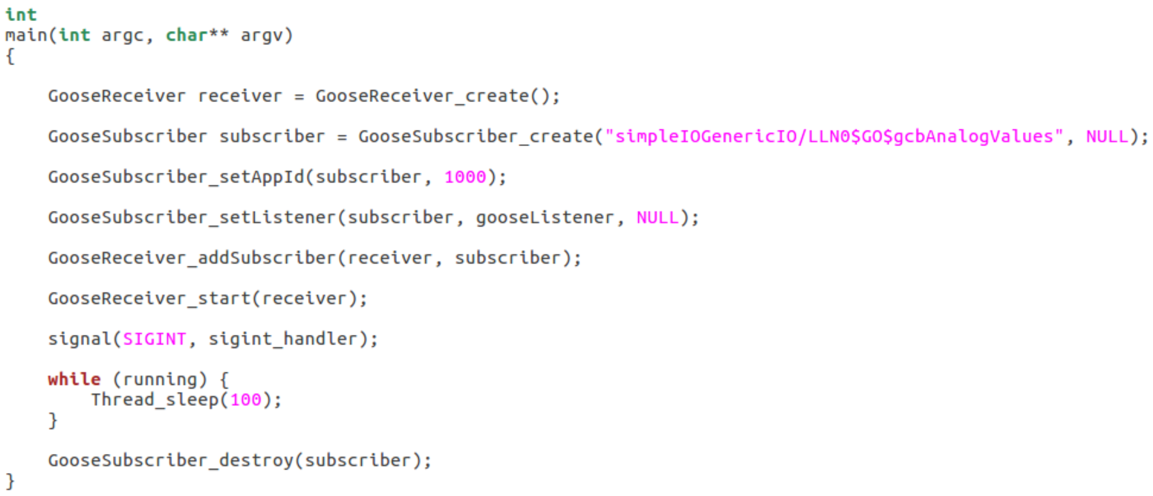
- Step 5: Publish the GOOSE message using the GoosePublisher_publish function.



```
GoosePublisher_publish(publisher, dataSetValues) == -1
```

10.2.2 Creating a GOOSE Subscriber.

Unlike the GOOSE publisher, the GOOSE subscriber is made of two parts. The main program and a thread running in the background. The purpose of the thread is to continuously listen on the network interface for GOOSE message.

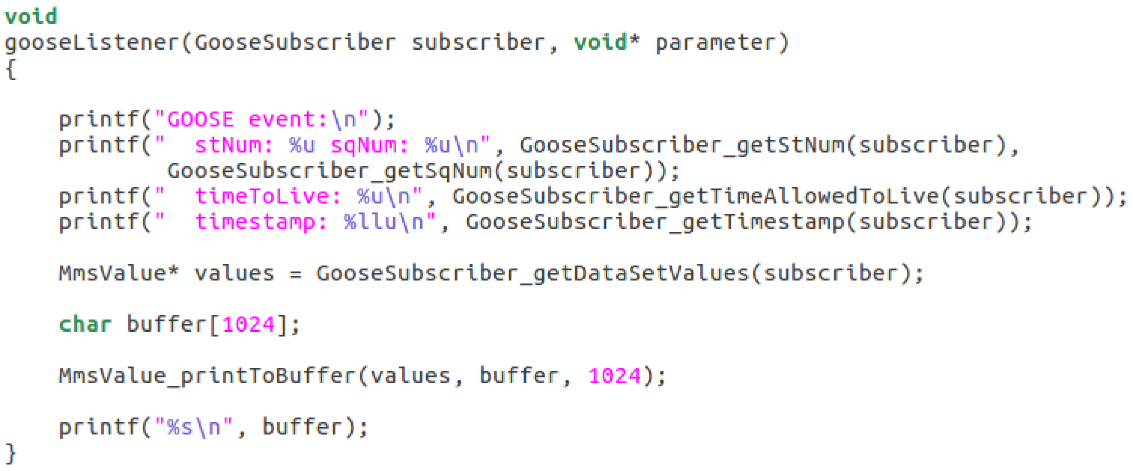


```
int
main(int argc, char** argv)
{

    GooseReceiver receiver = GooseReceiver_create();

    GooseSubscriber subscriber = GooseSubscriber_create("simpleIOGenericIO/LLN0$GO$gcbAnalogValues", NULL);

    GooseSubscriber_setAppId(subscriber, 1000);

    GooseSubscriber_setListener(subscriber, gooseListener, NULL);

    GooseReceiver_addSubscriber(receiver, subscriber);

    GooseReceiver_start(receiver);

    signal(SIGINT, sigint_handler);

    while (running) {
        Thread_sleep(100);
    }

    GooseSubscriber_destroy(subscriber);
}
```

The main program of the GOOSE subscriber should always contain the 6 steps shown above. 1. The receiver should be created. 2. The subscriber should be created. In this step, the created subscriber must have an input string matching that in the DataSetRef of the GOOSE message being subscribe to. 3. The APPID should also be set to match that of the GOOSE message being subscribed to. 4. A listener is assigned to the created subscriber. 5. A subscriber is added to the receiver created in step 1. 6. The receiver is started. That is the thread, which will be defined later, is activated.

In the code above, running is a variable set to True for the program to run continuously.

Next, the gooseListener thread takes the previously defined subscriber as input, as follows:



```
void
gooseListener(GooseSubscriber subscriber, void* parameter)
{

    printf("GOOSE event:\n");
    printf("  stNum: %u sqNum: %u\n", GooseSubscriber_getStNum(subscriber),
            GooseSubscriber_getSqNum(subscriber));
    printf("  timeToLive: %u\n", GooseSubscriber_getTimeAllowedToLive(subscriber));
    printf("  timestamp: %llu\n", GooseSubscriber_getTimestamp(subscriber));

    MmsValue* values = GooseSubscriber_getDataSetValues(subscriber);

    char buffer[1024];

    MmsValue_printToBuffer(values, buffer, 1024);

    printf("%s\n", buffer);
}
```

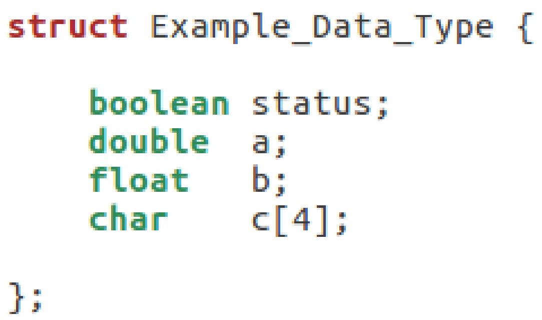
As can be seen in the example code above, each field of the GOOSE message could be accessed through a function defined in the API. All the functions can be found in the src/goose folder.

10.3 The RTI’s Data Distribution Service API

In this section, we will cover the steps to create a DDS publisher and a DDS subscriber on a Linux machine. The tools that we will use are provided by RTI DDS.

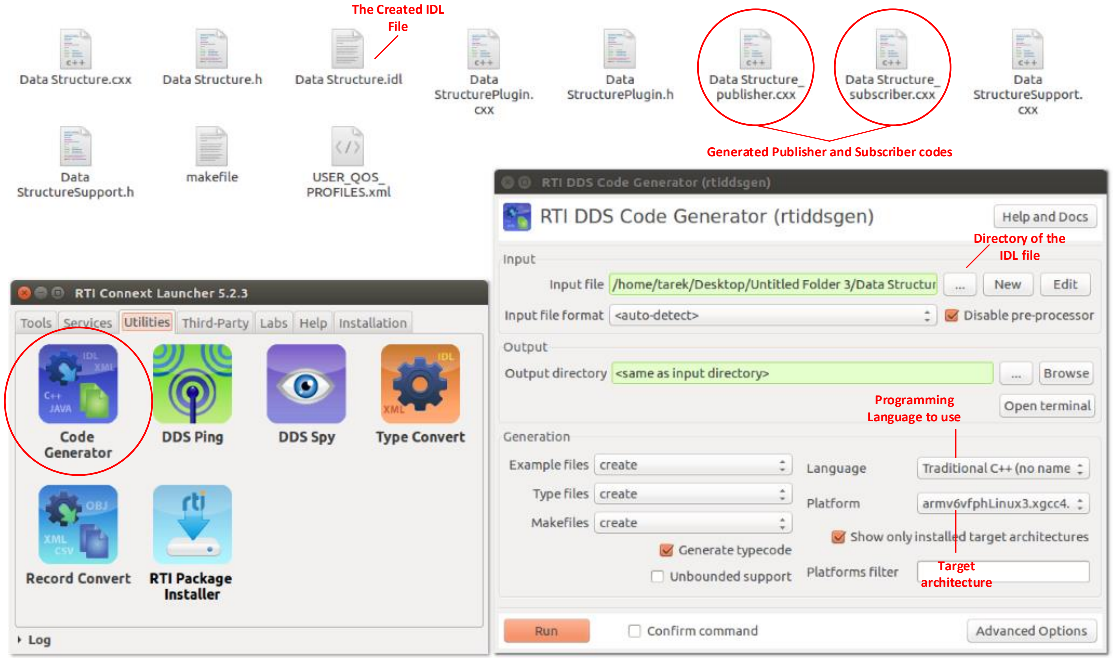
10.3.1 Creating an IDL file and Generating the Draft Code.

The first step in creating a DDS publisher and subscriber applications is to define the data types that you want to publish or subscribe to. This done through creating an IDL file. Inside the IDL file, you will be able to define those data types. The structures is the IDL file looks as follows:



```
struct Example_Data_Type {

     boolean status;
     double  a;
     float   b;
     char    c[4];

};
```

Next, you will use the RTI Code Generator to generate the publisher and subscriber draft codes, which you will edit later to implement your application on, and all other necessary files to compile the generated codes. As seen in Figure 10.3, once you open the Code Generator, you will need to specify the directory of your IDL file, your preferred programming language, and the target architecture (i.e. type of processor, such as arm processor). Once you click run, the publisher and subscriber codes will be generated.



Input
Input file /home/tarek/Desktop/Untitled Folder 3/Data Structur
Input file format <auto-detect>
☑ Disable pre-processor

Output
Output directory <same as input directory>

Generation
Example files create
Type files create
Makefiles create
☑ Generate typecode
☐ Unbounded support
Language Traditional C++ (no name
Platform armv6vfphLinux3.xgcc4.
☑ Show only installed target architectures
Platforms filter

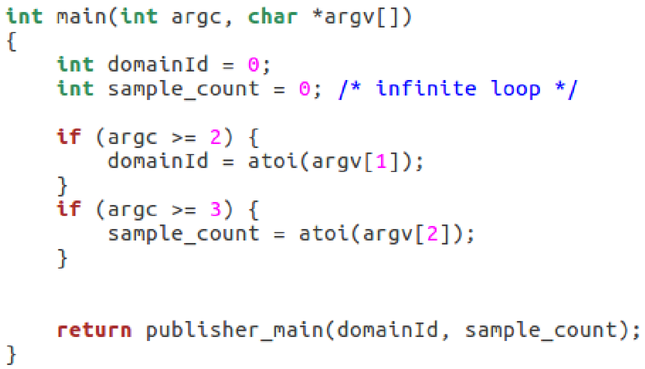
☐ Confirm command

Figure 10.3: Utilizing the code generator.

10.3.2 The DDS Publisher.

The code for the DDS publisher is made of three parts: the main program, the publisher_main, and the publisher_shutdown.

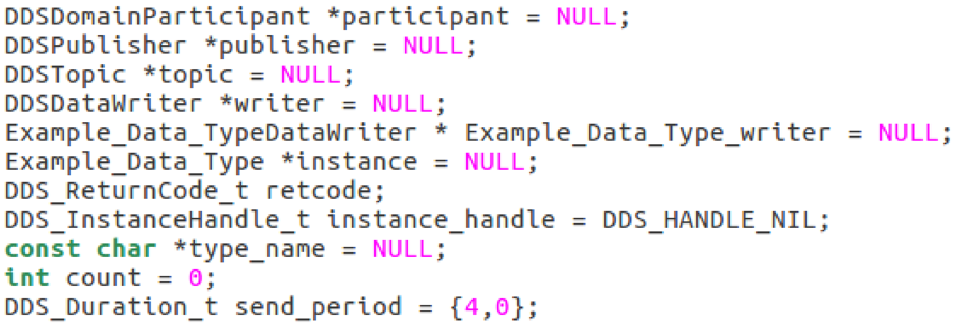
The main program is where the domain ID is defined. Each DDS global data space should be assigned an ID. The main program returns the publisher_main, as shown below.



```
int main(int argc, char *argv[])
{
    int domainId = 0;
    int sample_count = 0; /* infinite loop */

    if (argc >= 2) {
        domainId = atoi(argv[1]);
    }
    if (argc >= 3) {
        sample_count = atoi(argv[2]);
    }


    return publisher_main(domainId, sample_count);
}
```

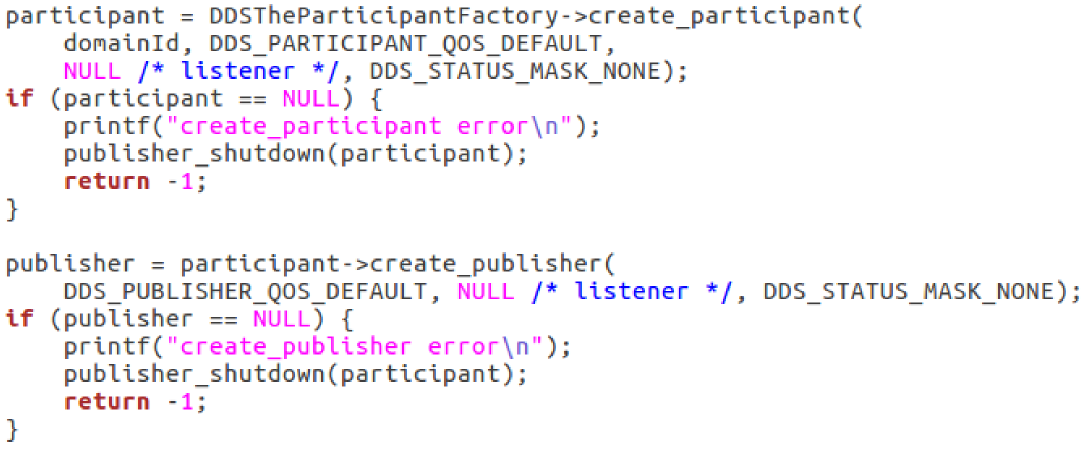
Next, the publisher_main is where all the definitions and configurations are setup. This part is divided into 6 sub parts as follow:

1. Defining the pointers to the DDS participant, publisher, topic, writer, instances, and type names. The error and instance handle structures are also defined here according to the API.



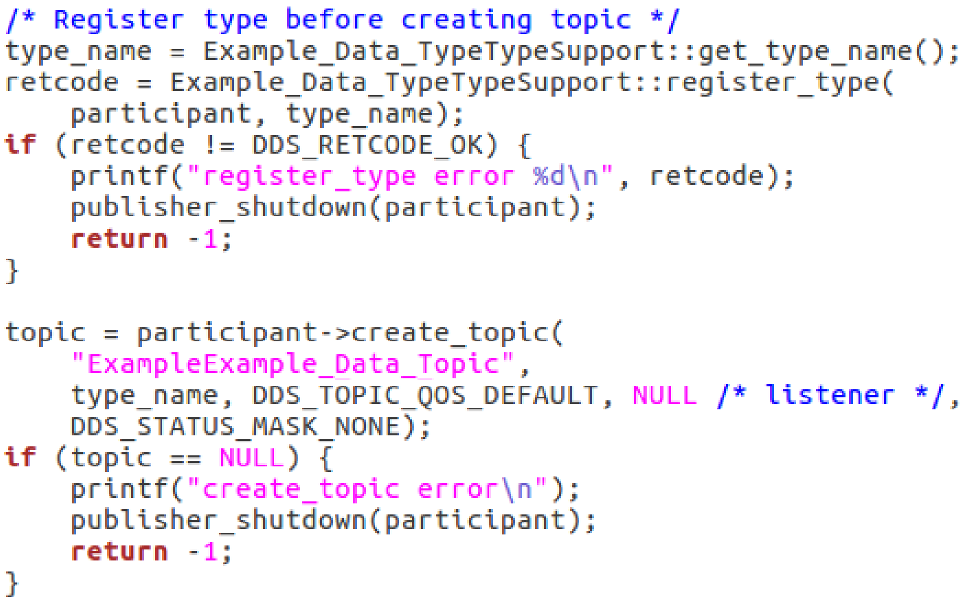
```
DDSDomainParticipant *participant = NULL;
DDSPublisher *publisher = NULL;
DDSTopic *topic = NULL;
DDSDataWriter *writer = NULL;
Example_Data_TypeDataWriter * Example_Data_Type_writer = NULL;
Example_Data_Type *instance = NULL;
DDS_ReturnCode_t retcode;
DDS_InstanceHandle_t instance_handle = DDS_HANDLE_NIL;
const char *type_name = NULL;
int count = 0;
DDS_Duration_t send_period = {4,0};
```

2. Creating the DDS participant and its publisher.



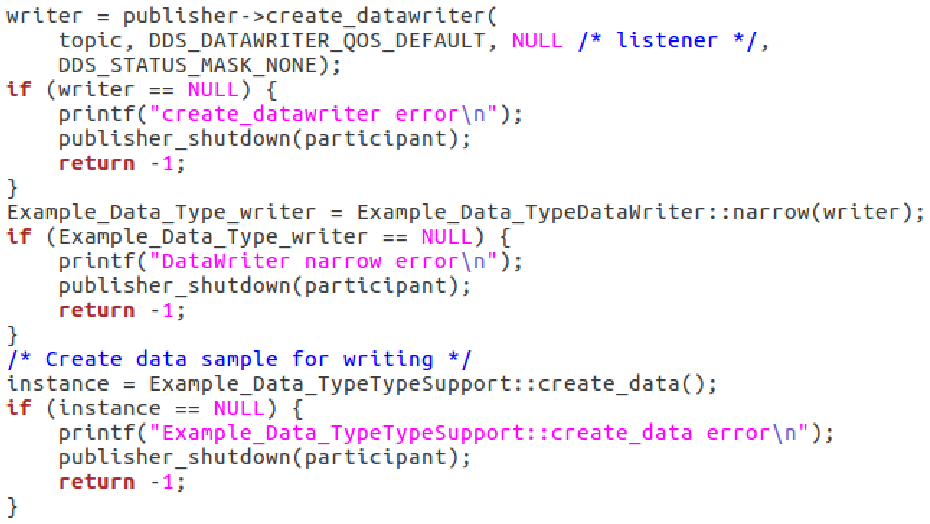
```
participant = DDSTheParticipantFactory->create_participant(
    domainId, DDS_PARTICIPANT_QOS_DEFAULT,
    NULL /* listener */, DDS_STATUS_MASK_NONE);
if (participant == NULL) {
    printf("create_participant error\n");
    publisher_shutdown(participant);
    return -1;
}

publisher = participant->create_publisher(
    DDS_PUBLISHER_QOS_DEFAULT, NULL /* listener */, DDS_STATUS_MASK_NONE);
if (publisher == NULL) {
    printf("create_publisher error\n");
    publisher_shutdown(participant);
    return -1;
}
```

3. Registering the topic and then creating it. In this step, we select the name of the Topic we want to publish out data types into. In this example, the topic name is Example_Data_Topic.



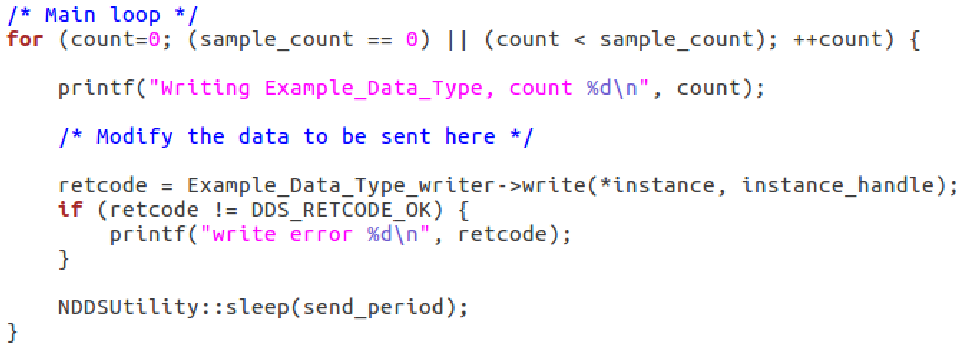
```
/* Register type before creating topic */
type_name = Example_Data_TypeTypeSupport::get_type_name();
retcode = Example_Data_TypeTypeSupport::register_type(
    participant, type_name);
if (retcode != DDS_RETCODE_OK) {
    printf("register_type error %d\n", retcode);
    publisher_shutdown(participant);
    return -1;
}

topic = participant->create_topic(
    "ExampleExample_Data_Topic",
    type_name, DDS_TOPIC_QOS_DEFAULT, NULL /* listener */,
    DDS_STATUS_MASK_NONE);
if (topic == NULL) {
    printf("create_topic error\n");
    publisher_shutdown(participant);
    return -1;
}
```

4. Create the data writer and the data sample for writing.



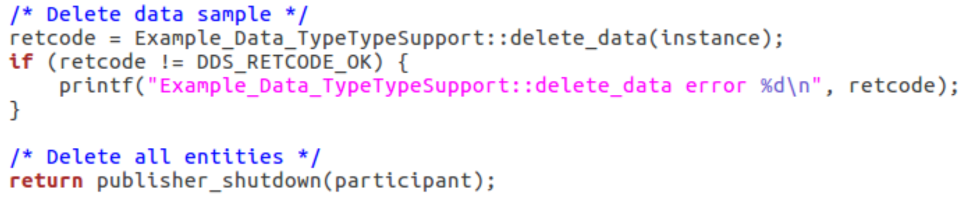
```
writer = publisher->create_datawriter(
    topic, DDS_DATAWRITER_QOS_DEFAULT, NULL /* listener */,
    DDS_STATUS_MASK_NONE);
if (writer == NULL) {
    printf("create_datawriter error\n");
    publisher_shutdown(participant);
    return -1;
}
Example_Data_Type_writer = Example_Data_TypeDataWriter::narrow(writer);
if (Example_Data_Type_writer == NULL) {
    printf("DataWriter narrow error\n");
    publisher_shutdown(participant);
    return -1;
}
/* Create data sample for writing */
instance = Example_Data_TypeTypeSupport::create_data();
if (instance == NULL) {
    printf("Example_Data_TypeTypeSupport::create_data error\n");
    publisher_shutdown(participant);
    return -1;
}
```

5. Modify the main loop and publish the data sample. This the part of the code where you apply all the changes you want to data in your defined data types before publishing it. This is where your algorithm is implemented.



```
/* Main loop */
for (count=0; (sample_count == 0) || (count < sample_count); ++count) {

    printf("Writing Example_Data_Type, count %d\n", count);

    /* Modify the data to be sent here */

    retcode = Example_Data_Type_writer->write(*instance, instance_handle);
    if (retcode != DDS_RETCODE_OK) {
        printf("write error %d\n", retcode);
    }

    NDDSUtility::sleep(send_period);
}
```

6. After publishing the data sample, we need to delete it and finally, shut down the publisher, if the program was ordered to terminate.

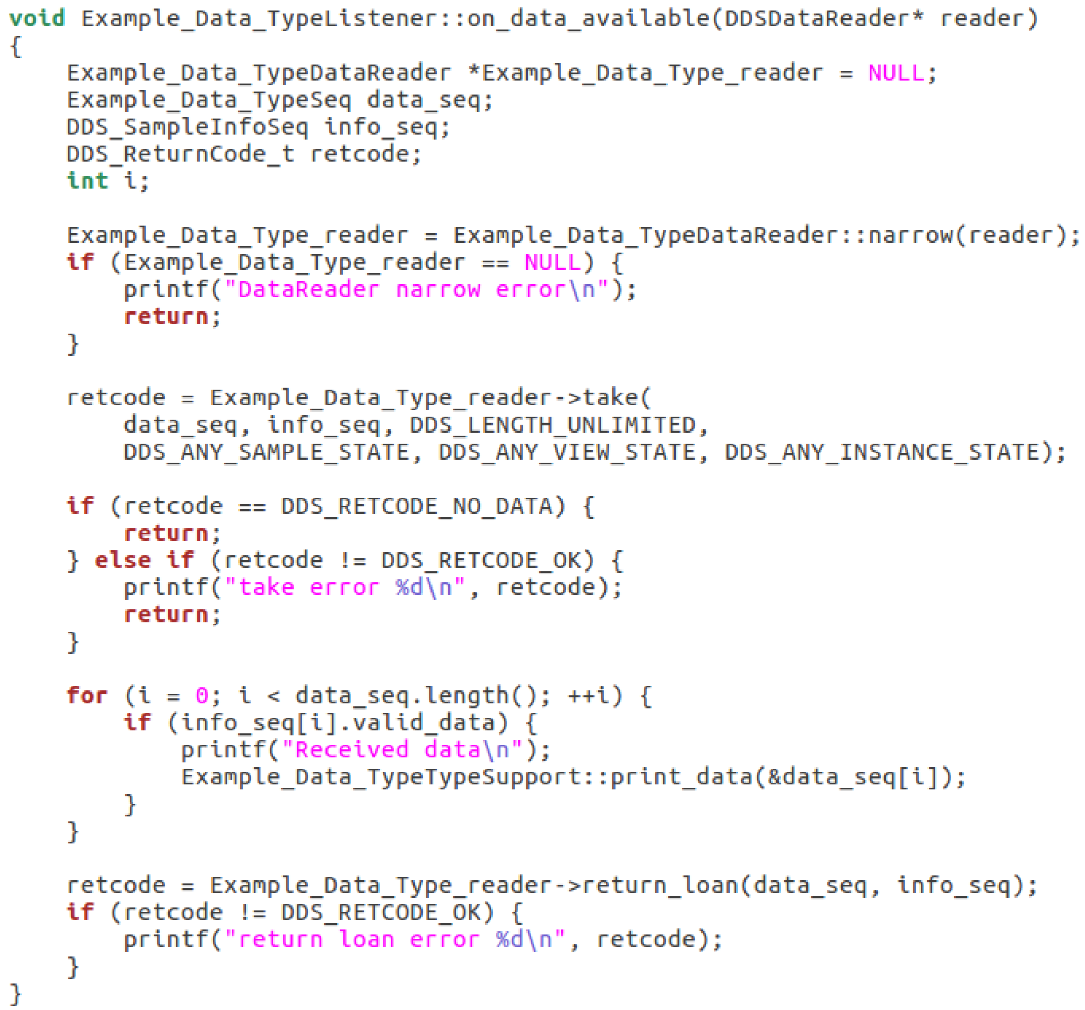


```
/* Delete data sample */
retcode = Example_Data_TypeTypeSupport::delete_data(instance);
if (retcode != DDS_RETCODE_OK) {
    printf("Example_Data_TypeTypeSupport::delete_data error %d\n", retcode);
}

/* Delete all entities */
return publisher_shutdown(participant);
```

10.3.3 The DDS Subscriber.

Similar to the DDS publisher, the code for the DDS subscriber has three main parts: the data listener thread, the subscriber_main, and the subscriber_shutdown.

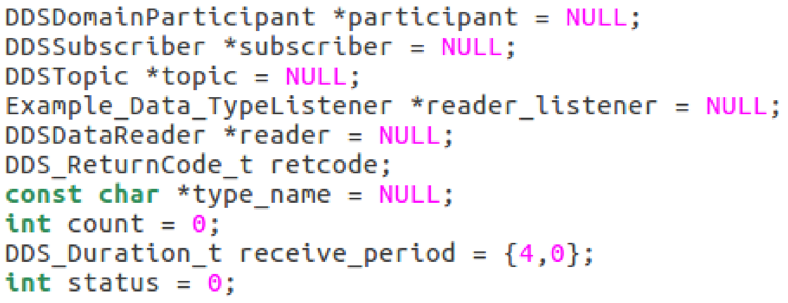
The data listener thread is invoked every time a data sample is available to be read. In this thread, a pointer to the data reader and the structure data_seq, which is where the read data is stored, are created. The data retrieved into the data_seq structure could either be stored in global variables to be used in other parts of the code.



```
void Example_Data_TypeListener::on_data_available(DDSDataReader* reader)
{
    Example_Data_TypeDataReader *Example_Data_Type_reader = NULL;
    Example_Data_TypeSeq data_seq;
    DDS_SampleInfoSeq info_seq;
    DDS_ReturnCode_t retcode;
    int i;

    Example_Data_Type_reader = Example_Data_TypeDataReader::narrow(reader);
    if (Example_Data_Type_reader == NULL) {
        printf("DataReader narrow error\n");
        return;
    }

    retcode = Example_Data_Type_reader->take(
        data_seq, info_seq, DDS_LENGTH_UNLIMITED,
        DDS_ANY_SAMPLE_STATE, DDS_ANY_VIEW_STATE, DDS_ANY_INSTANCE_STATE);

    if (retcode == DDS_RETCODE_NO_DATA) {
        return;
    } else if (retcode != DDS_RETCODE_OK) {
        printf("take error %d\n", retcode);
        return;
    }

    for (i = 0; i < data_seq.length(); ++i) {
        if (info_seq[i].valid_data) {
            printf("Received data\n");
            Example_Data_TypeTypeSupport::print_data(&data_seq[i]);
        }
    }

    retcode = Example_Data_Type_reader->return_loan(data_seq, info_seq);
    if (retcode != DDS_RETCODE_OK) {
        printf("return loan error %d\n", retcode);
    }
}
```

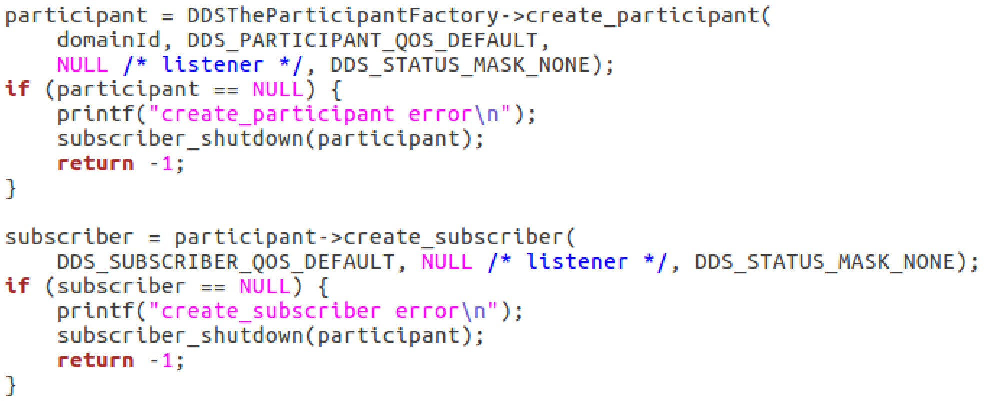
Similar the publisher_main, the subscriber_main is composed on 6 sub parts.

1. Defining the pointers to the DDS participant, reader, topic, listener, instances, and type names. The error and instance handle structures are also defined here according to the API.



```
DDSDomainParticipant *participant = NULL;
DDSSubscriber *subscriber = NULL;
DDSTopic *topic = NULL;
Example_Data_TypeListener *reader_listener = NULL;
DDSDataReader *reader = NULL;
DDS_ReturnCode_t retcode;
const char *type_name = NULL;
int count = 0;
DDS_Duration_t receive_period = {4,0};
int status = 0;
```

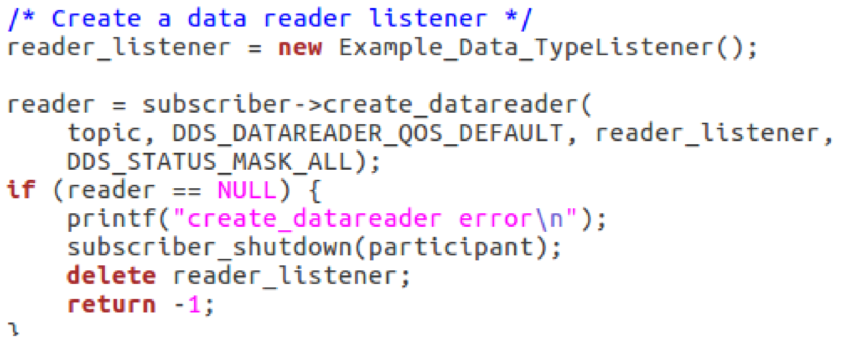
2. Creating the DDS participant and its subscriber.



```
participant = DDSTheParticipantFactory->create_participant(
    domainId, DDS_PARTICIPANT_QOS_DEFAULT,
    NULL /* listener */, DDS_STATUS_MASK_NONE);
if (participant == NULL) {
    printf("create_participant error\n");
    subscriber_shutdown(participant);
    return -1;
}

subscriber = participant->create_subscriber(
    DDS_SUBSCRIBER_QOS_DEFAULT, NULL /* listener */, DDS_STATUS_MASK_NONE);
if (subscriber == NULL) {
    printf("create_subscriber error\n");
    subscriber_shutdown(participant);
    return -1;
}
```

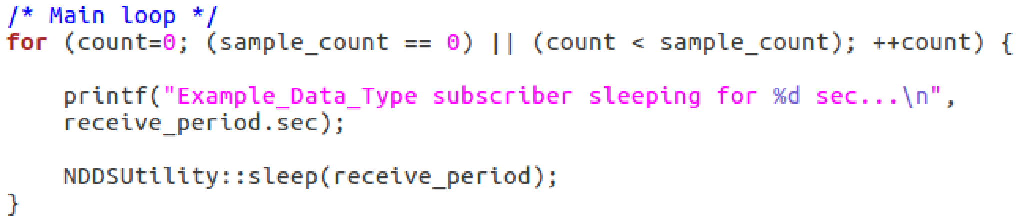
3. Registering the topic and then creating it is the same as in step 3 of the publisher code.

4. Creating the data listener and its subscriber.



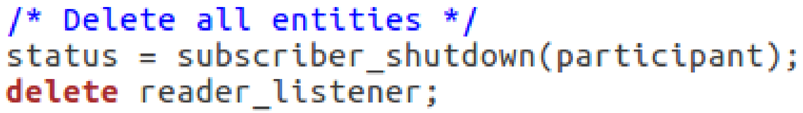
```
/* Create a data reader listener */
reader_listener = new Example_Data_TypeListener();

reader = subscriber->create_datareader(
    topic, DDS_DATAREADER_QOS_DEFAULT, reader_listener,
    DDS_STATUS_MASK_ALL);
if (reader == NULL) {
    printf("create_datareader error\n");
    subscriber_shutdown(participant);
    delete reader_listener;
    return -1;
}
```

5. Modifying the received data in the main loop.



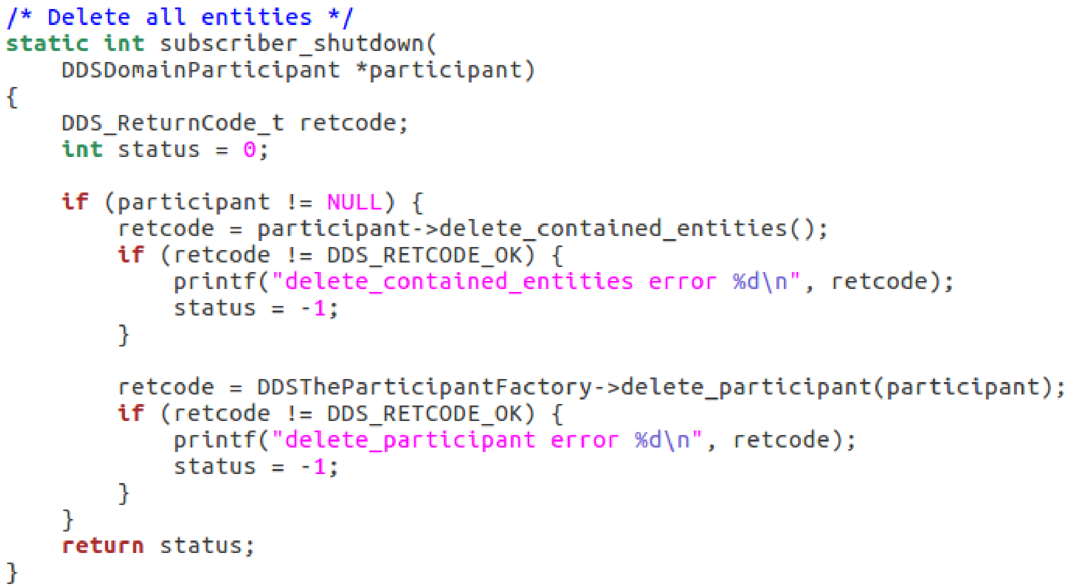
```
/* Main loop */
for (count=0; (sample_count == 0) || (count < sample_count); ++count) {

    printf("Example_Data_Type subscriber sleeping for %d sec...\n",
    receive_period.sec);

    NDDSUtility::sleep(receive_period);
}
```

6. Deleting all entities.



```
/* Delete all entities */
status = subscriber_shutdown(participant);
delete reader_listener;
```

Finally, the subscriber_shutdown is called when the program terminates.



```
/* Delete all entities */
static int subscriber_shutdown(
    DDSDomainParticipant *participant)
{
    DDS_ReturnCode_t retcode;
    int status = 0;

    if (participant != NULL) {
        retcode = participant->delete_contained_entities();
        if (retcode != DDS_RETCODE_OK) {
            printf("delete_contained_entities error %d\n", retcode);
            status = -1;
        }

        retcode = DDSTheParticipantFactory->delete_participant(participant);
        if (retcode != DDS_RETCODE_OK) {
            printf("delete_participant error %d\n", retcode);
            status = -1;
        }
    }
    return status;
}
```

10.4 Conclusion

In this chapter, the reader was introduced the beaglebone black microcrontroller. The reader was taught how install a Debian image onto the microcontroller and interface with real hardware equipment using the analogue inputs and digital inputs and outputs. Next, the read was taught how to create applications using the IEC 61850 GOOSE publishers and subscribers. Finally, the user was presented with an example about creating and dealing with DDS data types, publishers, and subscribers.

Chapter 11 Conclusions and Recommendations for Future Work

11.1 Conclusions

This dissertation presented the concept of complementing cyber security algorithms with information about the physical characteristics of the power system for enhancing the resiliency of the grid against cyber-attacks. An analysis of the recent co-simulation platforms for smart grid applications was performed. It was shown that there is a need for a comprehensive co-simulation platform capable of modeling the relation and interaction between the cyber and physical parts of the smart grid. To that end, a network-in-the-loop co-simulation platform to analyze and understand cyber information flow, physical power system dynamics, and the interrelation between them was developed. A discussion about the importance of interoperability in the smart grid was also presented. As such, the data-centric DDS communication middleware was selected to seamlessly integrate the components of the developed co-simulation platform together and to bridge the different communication protocols and software applications.

To verify the effectiveness of the developed network-in-the-loop co-simulation platform, three case studies were performed, including electric vehicle charging, protocol emulation, and a lost sample forecasting algorithm.

Being the most widely industry-accepted standard, the IEC 61850 GOOSE messaging standard was analyzed. In addition to that, testing of two different implementations of the IEC 61850 GOOSE messaging protocol on commercial IEC 61850-based devices and on the open source libiec61850 library was performed. It was found that different implementations of the same standard might lead to different behaviors even if the devices were present under similar conditions. Deviation from the actual procedures set forth by the IEC 61850 standard and its complementary cyber security IEC 62351 standard were found in the responses of the devices. This vulnerability provides a strong attack surface for attackers to inject malicious activities in power systems, such as the GOOSE poisoning attack demonstrated in this work. The work in this dissertation also proposed guidelines to better enhance utilization of IEC 61850, such as proper processing of a message's source MAC address, better utilization of the time stamp field to check for messages' validity, and the association of new messages' content with a status number increment.

An artificially intelligent physical model-checking approach to detect malicious and erroneous control commands controlling the state of circuit breakers in power systems was developed. The purpose of the work is to push enough intelligence into the agents controlling the power system to enable them to assess the consequences of control commands before taking actions on the physical system. The proposed security framework was tested on a 14-bus IEEE benchmark system. The results showed the accuracy of the AI module in characterizing the system under study and its effectiveness in not allowing the system to go into an insecure state.

To detect spoofed sampled measured values in the smart grid, the feasibility of using neural network forecasters to detect spoofed IEC 61850 Sampled Measured Values was investigated. The system utilized a combined neural network – time series forecasting module to detect fake measurement data injection attacks. The proposed system proved its effectiveness in enhancing the resiliency of modern protection schemes against false data injection attacks. The system was implemented on a real IEC 61850 Ethernet network.

Next, the design and implementation of a targeted data injection attack that simulates real AC waveforms in an attempt to interrupt power flow in a compromised power network was presented. The previously developed neural network forecaster was then tested against the targeted malware to observe how it would handle the smart attack. Experimentation with the malware demonstrated that the NN is yet to be properly trained or fine-tuned enough to detect malicious measurements. The NN showed resistance to less sophisticated data injection methods; however, it is still vulnerable to malicious data injection methods. Also, the NN demonstrated very little resistance to the simulated AC waveform. In order to better prepare the NN, the developed malware was also used as a training tool to identify attack signatures, and allow the user to tune the event thresholds that would result in controlling messages being sent to the IEDs. The malware is configurable, easy to understand, and is simple to use to train predictive intrusion detection systems.

Based on the previous study, it was found that although neural network forecasters have a high detection accuracy, they are susceptible to the accumulation of forecasting errors. As such, a lightweight algorithm, based on statistical indicators, to enhance the reliability of the neural network forecaster in terms of detecting spoofed sampled measured values was developed. This algorithm is considered as a second line of defense, complementing the cyber security methods, such as encryption and authentication. The studies were conducted on a laboratory-scale hardware microgrid with a commercial network switch to be more representative of the microgrid environment than event-based simulation models.

On the application layer of the smart grid, an energy management system to camouflage the load demand profile for small microgrids was developed. An online optimization module accounting for history, current, and future system observations, utilizing particle swarm optimization, was also developed to enhance the results of the energy management system. The energy management system was tested on actual residential energy consumption and irradiance data from Miami, Florida, and proved its effectiveness not only in hiding the customer's load demand profile, but also in reducing the consumer's bills and achieving flat peak load profiles.

The shortcomings of current EV-Grid integration systems was investigated. The drawbacks of security breaches on the power grid, the EV itself, and the safety of the vehicles' owners, were outlined in hope of advocating a new school of thought that gives the same importance for system operation and system security.

Finally, to be able to develop and prototype applications related to energy cyber-physical systems, the reader was taught how to configure and program a Linux-based microcontroller and interface with real hardware equipment using the analogue inputs and digital inputs and outputs. The read was taught how to create applications using the IEC 61850 GOOSE publishers and subscribers, and was presented with an example about creating and dealing with DDS data types, publishers, and subscribers.

11.2 Recommendations for Future Work

This dissertation covered several aspects related to interoperability and security challenges in the smart grid. The smart grid's cyber and physical domains were modeled in a high-fidelity co-simulation platform. Also, artificial intelligence was used to complement the decision of conventional cyber security algorithms, such as authentication and encryption, with physical characteristics of the power system. Nonetheless, since the smart grid is a complex cyber-physical systems covering interdisciplinary topics, it is recommended that this work is expanded by others according to the following research directions:

- Detection of switching related attacks: Conduct a study to optimally allocate the switching-attack-detection agents in the power system taking into consideration constraints from the power system and the communication network sides.
- Detection of anomalous sampled measured values: Compare the performance of the developed neural network forecaster with deep learning techniques in terms of accuracy and detection latency. Expand the work to cover other types of protocols and other applications, such as PMU measurements for wide area monitoring.
- Interoperability: Although there are a couple of standards for data modeling, such as CIM and IEC 61850, they do not cover all the different applications in the smart grid. Therefore, it is necessary to study the development of a unified data model to cope with the new technologies that are being introduced to the smart gird.

LIST OF REFERENCES

[1] Vahid Salehi, Ahmed Mohamed, Ali Mazloomzadeh, Osama A. Mohammed, "Laboratory-Based Smart Power System, Part I: Design and System Development," *IEEE TRANSACTIONS ON SMART GRID,* vol. 3, no. 3, September 2012.

[2] Ravi Akella, Han Tang, Bruce M. McMillin, Analysis of information flow security in cyber–physical systems, International Journal of Critical Infrastructure Protection, Volume 3, Issues 3–4, December 2010, Pages 157-173, ISSN 1874-5482.

[3] Y. Kim, M. Thottan, V. Kolesnikov and W. Lee, "A secure decentralized data-centric information infrastructure for smart grid," in *IEEE Communications Magazine,* vol. 48, no. 11, pp. 58-65, November 2010.

[4] T. A. Youssef, M. E. Hariri, A. T. Elsayed and O. A. Mohammed, "A DDS-Based Energy Management Framework for Small Microgrid Operation and Control," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 958-968, March 2018.

[5] M. P. F. Hommelberg, C. J. Warmer, I. G. Kamphuis, J. K. Kok and G. J. Schaeffer, "Distributed Control Concepts using Multi-Agent technology and Automatic Markets: An indispensable feature of smart power grids," *2007 IEEE Power Engineering Society General Meeting*, Tampa, FL, 2007, pp. 1-7.

[6] R. de Azevedo, M. H. Cintuglu, T. Ma and O. A. Mohammed, "Multiagent-Based Optimal Microgrid Control Using Fully Distributed Diffusion Strategy," in *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1997-2008, July 2017.

[7] Sean Paul McGurk, “Industrial Conrol Systems Security; Protecting the Critical Infrastrucutre”, a Presentation by tge U.S. Department of Homeland Security. Available Online: https://csrc.nist.gov/CSRC/media/Events/ISPAB-DECEMBER-2008-MEETING/documents/ICSsecurity_ISPAB-dec2008_SPMcGurk.pdf (Accessed on: 09/18/2018)

[8] International Electrotechnical Commission. Communication networks and systems in substations—Specificvcommunication service mapping (SCSM). IEC 61850-8, 2008.

[9] Fuloria, S.; Anderson, R.; Mcgrath, K.; Hansen, K.; Alvarez, F. The Protection of Substation Communications. In Proceedings of SCADA Security Scientific Symposium, Miami, FL, USA, 18–19 January 2010.

[10] N. Kush, E. Ahmed, M. Branagan, E. Foo, "Poisoned GOOSE: Exploiting the GOOSE Protocol", Proceedings of the Twelfth Australasian Information Security Conference-Volume 149, vol. 149, pp. 17-22, 2014.

[11] Obermeier S., et al., Assessing the Security of IEC 62351, Conference: Conference: 3rd International Symposium for ICS & SCADA Cyber Security Research 2015

[12] K. Vimalkumar and N. Radhika, "A big data framework for intrusion detection in smart grids using apache spark," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, 2017, pp. 198-204.

[13] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni and H. V. Poor, "Machine Learning Methods for Attack Detection in the Smart Grid," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773-1786, Aug. 2016.

[14] M. A. Faisal, Z. Aung, J. R. Williams and A. Sanchez, "Data-Stream-Based Intrusion Detection System for Advanced Metering Infrastructure in Smart Grid: A Feasibility Study," in *IEEE Systems Journal*, vol. 9, no. 1, pp. 31-44, March 2015.

[15] Y. Yang *et al.*, "Multiattribute SCADA-Specific Intrusion Detection System for Power Networks," in *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092-1102, June 2014.

[16] Y. Kwon, H. K. Kim, Y. H. Lim and J. I. Lim, "A behavior-based intrusion detection technique for smart grid infrastructure," *2015 IEEE Eindhoven PowerTech*, Eindhoven, 2015, pp. 1-6.

[17] J. Hong, C. Liu and M. Govindarasu, "Detection of cyber intrusions using network-based multicast messages for substation automation," *ISGT 2014,* Washington, DC, 2014, pp. 1-5.

[18] J. Hong, C. Liu and M. Govindarasu, "Integrated Anomaly Detection for Cyber Security of the Substations," in *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1643-1653, July 2014.

[19] Lázaro, J.; Astarloa, A.; Araujo, J.A.; Moreira, N.; Bidarte, U. MACsec Layer 2 Security in HSR Rings in Substation Automation Systems. *Energies* **2017**, *10*, 162.

[20] B. Vaidya, D. Makrakis and H. T. Mouftah, "Authentication and authorization mechanisms for substation automation in smart grid network," in *IEEE Network*, vol. 27, no. 1, pp. 5-11, January-February 2013.

[21] W. Fangfang, W. Huazhong, C. Dongqing and P. Yong, "Substation Communication Security Research Based on Hybrid Encryption of DES and RSA," *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Beijing, 2013, pp. 437-441.

[22] Raciti Massimiliano and Nadjm-Tehrani Simin, "”Embedded Cyber-Physical Anomaly Detection in Smart Meters,” Critical Information Infrastructures Security, Springer Berlin Heidelberg, pp. 34-45, 2013, 978-3-642-41485-5.

[23] G. Koutsandria, V. Muthukumar, M. Parvania, S. Peisert, C. McParland and A. Scaglione, "A hybrid network IDS for protective digital relays in the power transmission grid," *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm),* Venice, 2014, pp. 908-913.

[24] M. S. Rahman, M. A. Mahmud, A. M. T. Oo and H. R. Pota, "Multi-Agent Approach for Enhancing Security of Protection Schemes in Cyber-Physical Energy Systems," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 436-447, April 2017.

[25] S. Pan, T. Morris and U. Adhikari, "Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems," in *IEEE Transactions on Smart Grid,* vol. 6, no. 6, pp. 3104-3113, Nov. 2015.

[26] U. C. Netto, D. C. Grillo, I. D. Lonel, E. L. Pellini, D. V. Coury,"An ANN based forecast for IED network management using the IEC61850 standard," Electric Power Systems Research, Volume 130, 2016, Pages 148-155, ISSN 0378-7796.

[27] F. Leu, K. Tsai, Y. Hsiao and C. Yang, "An Internal Intrusion Detection and Protection System by Using Data Mining and Forensic Techniques," in *IEEE Systems Journal*, vol. 11, no. 2, pp. 427-438, June 2017.

[28] O. A. Beg, T. T. Johnson and A. Davoudi, "Detection of False-Data Injection Attacks in Cyber-Physical DC Microgrids," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2693-2703, Oct. 2017.

[29] M. S. Rahman, A. M. T. Oo, M. A. Mahmud and H. R. Pota, "A multi-agent approach for security of future power grid protection systems," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, 2016, pp. 1-5.

[30] H. Lin, A. Slagell, Z. T. Kalbarczyk, P. W. Sauer and R. K. Iyer, "Runtime Semantic Security Analysis to Detect and Mitigate Control-Related Attacks in Power Grids," in *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 163-178, Jan. 2018.

[31] S. Sridhar and M. Govindarasu, "Model-Based Attack Detection and Mitigation for Automatic Generation Control," in *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580-591, March 2014.

[32] B. Talha and A. Ray, "A framework for MAC layer wireless intrusion detection &amp; response for smart grid applications," *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, Poitiers, 2016, pp. 598-605.

[33] Maëlle Kabir-Querrec, Stéphane Mocanu, Pascal Bellemain, Jean-Marc Thiriet, Eric Savary. Corrupted GOOSE Detectors: Anomaly Detection in Power Utility Real-Time Ethernet Communications. *GreHack 2015*, Nov 2015, Grenoble, France.

[34] Youssef, Tarek, "Co-design of Security Aware Power System Distribution Architecture as Cyber Physical System" (2017). *FIU Electronic Theses and Dissertations*. 3210.

[35] Chen, T.M., Swansea Univ., Swansea, UK ; Abu-Nimeh, S." Lessons from Stuxnet" IEEE computer magazine, vol. 44, issue. 4, pp. 91-93, April 2011.

[36] Available Online: https://scadahacker.com/ (Accessed on: 09/17/2018).

[37] Y. Liu, M. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," in Proc. 16th ACM Conf. Comput. Commun. Security, Nov. 2009.

[38] G. Celli, P. A. Pegoraro, F. Pilo, G. Pisano and S. Sulis, "DMS Cyber-Physical Simulation for Assessing the Impact of State Estimation and Communication Media in Smart Grid Operation," in IEEE Transactions on Power Systems, vol. 29, no. 5, pp. 2436-2446, Sept. 2014.

[39] Xichen Jiang , Jiangmeng Zhang , Harding, B.J. , Makela, J.J. , Dominguez-Garcia, A.D. "Spoofing GPS Receiver Clock Offset of Phasor Measurement Units" IEEE Transactions on Power Systems, Vol.28 , No. 3 , pp. 3253 – 3262, Aug. 2013

[40] Daniel P. Shepard and Todd E. Humphreys "Evaluation of the Vulnerability of Phasor Measurement Units to GPS Spoofing Attacks" International Journal of Critical Infrastructure Protection, Vol. 5, Dec. 2012.

[41] El Hariri, M.; Youssef, T.A.; Mohammed, O.A. On the Implementation of the IEC 61850 Standard: Will Different Manufacturer Devices Behave Similarly under Identical Conditions? Electronics 2016, 5, 85.

[42] T. A. Youssef, M. E. Hariri, N. Bugay and O. A. Mohammed, "IEC 61850: Technology standards and cyber-threats," 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, 2016, pp. 1-6.

[43] Kelly Zieglar, “Grid, PhD: The Smart Grid,Cyber Security, and the Future of Keeping the Lights On”, 19th USENIX Security Symposium, Washington, DC, August 11-13, 2010.

[44] Ravi Akella, Han Tang, Bruce M. McMillin, Analysis of information flow security in cyber–physical systems, International Journal of Critical Infrastructure Protection, Volume 3, Issues 3–4, December 2010, Pages 157-173, ISSN 1874-5482.

[45] Nguyen, V.H.; Besanger, Y.; Tran, Q.T.; Nguyen, T.L. On Conceptual Structuration and Coupling Methods of Co-Simulation Frameworks in Cyber-Physical Energy System Validation. *Energies* **2017**, *10*, 1977.

[46] André N. Albagli, Djalma M. Falcão, José F. de Rezende, Smart grid framework co-simulation using HLA architecture, Electric Power Systems Research, Volume 130, January 2016, Pages 22-33, ISSN 0378-7796.

[47] Dhananjay Bhor, Kavinkadhirselvan Angappan, Krishna M. Sivalingam, Network and power-grid co-simulation framework for Smart Grid wide-area monitoring networks, Journal of Network and Computer Applications, Volume 59, January 2016, Pages 274-284, ISSN 1084-8045.

[48] K. Boroojeni, M. H. Amini, A. Nejadpak, T. Dragičević, S. S. Iyengar and F. Blaabjerg, "A Novel Cloud-Based Platform for Implementation of Oblivious Power Routing for Clusters of Microgrids," in IEEE Access, vol. 5, no. , pp. 607-619, 2017.

[49] G. Celli, P. A. Pegoraro, F. Pilo, G. Pisano and S. Sulis, "DMS Cyber-Physical Simulation for Assessing the Impact of State Estimation and Communication Media in Smart Grid Operation," in IEEE Transactions on Power Systems, vol. 29, no. 5, pp. 2436-2446, Sept. 2014.

[50] E. Sharma, C. Chiculita and Y. Besanger, "Co-simulation of a low-voltage utility grid controlled over IEC 61850 protocol," 2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), Changsha, 2015, pp. 2365-2372.

[51] Lin, H.; Sambamoorthy, S.; Shukl, S.; Thorp, J.; Mili, L. A study of communication and power system infrastructure interdependence on PMU-based wide area monitoring and protection. In Proceedings of the Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012.

[52] Lin, H.; Veda, S.; Shukla, S.; Mili, L.; Throp, J. GECO: Global Event-Driven Co-Simulation Framework for Interconnected Power System and Communication Network. IEEE Trans. Smart Grid 2012, 3, 1444–1459.

[53] Nutaro, J.; Kuruganti, P.; Miller, L.; Mullen, S.; Shankar, M. Integrated hybrid-simulation of electric power and communication systems. In Proceedings of the IEEE PES General Meeting 2007, Tampa, FL, USA, 24–28 June 2007.

[54] Li, W.; Monti, A.; Luo, M.; Dougal, R. VPNET: A co-simulation framework for analyzing communication channel effects on power systems. In Proceedings of the IEEE Electric Ship Technologies Symposium, Alexandria, VA, USA, 10–13 April 2011; pp. 143–149.

[55] Anderson, D.; Zhao, C.; Hauser, C.; Venkatasubramanian, V.; Bakken, D.; Bose, A. A virtual smart grid–real-time simulation for smart grid control and communication design. IEEE Power Energy Mag. 2012, 10, 49–57.

[56] Liberatoire, V.; Al-Hammouri, A. Smart Grid Communication and Co-Simulation. In Proceedings of the IEEE Energytech, Cleveland, OH, USA, 25–26 May 2011; pp. 1–5.

[57] Bergmann, J.; Glomb, C.; Götz, J.; Heuer, J. Scalability of Smart Grid Protocols: Protocols and Their Simulative Evaluation for Massively Distributed DERs. In Proceedings of the 1st IEEE International Conference on Smart Grid Communication, Gaithersburg, MD, USA, 4–6 October 2010; pp. 131–136.

[58] Babazadeh, D.; Chemine, M.; Kun, M.; Al-Hammouri, A.; Nordstrom, L. A platform for Wide Area Monitoring and Control System ICT analysis and Development. In Proceedings of the IEEE PowerTech, Grenoble, France, 16–20 June 2013; pp. 1–7.

[59] Babazadeh, D.; Nordstrom, L. Angent-based control of VSC-HVDC Transmission Grid—A Cyber Physical System Perspective. In Proceedings of the IEEE MSCPES 2014, Berlin, Germany, 14 April 2014.

[60] Godfrey, T.; Mullen, S.; Dugan, R.; Rodine, C.; Griffith, D.; Golmie, N. Modeling smart grid applications with co-simulation. In Proceedings of the 1st IEEE International Conference on Smart Grid Communication, Gaithersburg, MD, USA, 4–6 October 2010; pp. 291–296.

[61] Mallouhi, M.; Al-Nashif, Y.; Cox, D.; Chadaga, T.; Hariri, S. A testbed for analyzing security of SCADA control systems (TASSCS). In Proceedings of the 2011 IEEE PES Innovative Smart Grid Technologies Conference, Anaheim, CA, USA, 17–19 January 2011.

[62] Wei, M.; Wang, W. Greenbench: A benchmark for observing power grid vulnerability under data-centric threats. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 2625–2633.

[63] Oh, S.-J.; Yoo, C.-H.; Chung, I.-Y.; Won, D.-J. Hardware-in-the-Loop Simulation of Distributed Intelligent Energy Management System for Microgrids. Energies 2013, 6, 3263-3283.

[64] M. S. Almas and L. Vanfretti, "RT-HIL Implementation of the Hybrid Synchrophasor and GOOSE-Based Passive Islanding Schemes," in IEEE Transactions on Power Delivery, vol. 31, no. 3, pp. 1299-1309, June 2016.

[65] David Goughnour, Joe Stevents, Testing Intelligent Device Communications in Distributed System. Available Online: http://trianglemicroworks.com/docs/default-source/referenced-documents/testing-intelligent-device-communications-in-a-distributed-system.pdf?sfvrsn=2 (Accessed on 01/04/2017).

[66] Kianoosh G. Boroojeni, M. Hadi Amini, and S. S. Iyengar, Smart Grids: Security and Privacy Issues, Springer International Publishing, 2016.

[67] [IEEE 90] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.

[68] Frances Cleveland, Forrest Small, and Tom Brunetto, “Smart Grid: Interoperability and Standards an Introductory Review”, Utility Standards Board, September 2008.

[69] Oxford Dictionaries. Available online: http://blog.oxforddictionaries.com/2012/09/world-gratitude-day/ (Accessed on 12/21/2016).

[70] Youssef, T.A.; Elsayed, A.T.; Mohammed, O.A. Data Distribution Service-Based Interoperability Framework for Smart Grid Testbed Infrastructure. Energies 2016, 9, 150.

[71] Alkhawaja A. R., Ferreira L. L., and Albano M., Message Oriented Middleware with QoS Support for Smart Grids, InForum 2012 - Conference on Embedded Systems and Real Time, Caparica, Portugal, 6 and 7 of September 2012.

[72] RTi Connext DDS Professional. Available Online: https://www.rti.com/products/dds/omg-dds-standard.html (Accessed on: 12/21/2016).

[73] Angelo Corsaro, DDS & OPC-UA Explained. Available Online: http://www.prismtech.com/events/dds-and-opc-ua-explained-live-webcast (Accessed on: 04/01/2017).

[74] American National Standard for Electrical Power Systems and Equipment, “ANSI C84.1-2006, Voltage Ratings (60 Hertz),” 2006.

[75] LibIEC61850 Open Source Library. Available online: http://libiec61850.com/libiec61850/.

[76] M. G. Kanabar and T. S. Sidhu, "Performance of IEC 61850-9-2 Process Bus and Corrective Measure for Digital Relaying," in IEEE Transactions on Power Delivery, vol. 26, no. 2, pp. 725-735, April 2011.

[77] J. C. Tournier and O. Goerlitz, "Strategies to secure the IEEE 1588 protocol in digital substation automation," 2009 Fourth International Conference on Critical Infrastructures, Linkoping, 2009, pp. 1-8.

[78] A. Abdolkhalig and R. Zivanovic, "Evaluation of IEC 61850-9-2 samples loss on total vector error of an estimated phasor," 2013 IEEE Student Conference on Research and Developement, Putrajaya, 2013, pp. 269-274.

[79] R. Cimadevilla, Í. Ferrero and J. M. Yarza, "IEC61850-9-2 Process Bus implementation on IEDs," *2014 IEEE PES T&D Conference and Exposition*, Chicago, IL, USA, 2014, pp. 1-5. doi: 10.1109/TDC.2014.6863506

[80] Le Borgne Y., Santini S. Bontemp G., Adaptive model selection for time series prediction in wireless sensor networks, in Elsevier Signal Processing Journal Special Section: Information Processing and Data Management in Wireless Sensor Networks, Volume 87, Issue 12, December 2007, Pages 3010–3020

[81] Javier Moriano et al., A New Approach to Detection of Systematic Errors in Secondary Substation Monitoring Equipment Based on Short Term Load Forecasting, in Sensors 2016, 16(1), 85; doi:10.3390/s16010085

[82] Buchholz, B.M.; Brunner, C.; Naumann, A.; Styczynski, A. Applying IEC standards for communication and data management as the backbone of Smart Distribution. In Proceedings of the IEEE PES General Meeting 2012, San Diego, CA, USA, 22–26 July 2012.

[83] Brunner, C.; Naumann, A. The link between IEC 61850 and CIM/IEC 61968/61970—Experience from a smart grid project. In Proceedings of the Distributech, San Antonio, TX, USA, 24–26 January 2012.

[84] Naumann, A.; Bielchev, I.; Voropai, N.; Styczynski, Z. Smart grid automation using IEC 61850 and CIM standards. *Control Eng. Pract.* **2014**, *25*, 102–111.

[85] Das, N.; Islam, S. Analysis of power system communication architectures between substations using IEC 61850. In Proceedings of the 5th Brunei International Conference on Engineering and Technology (BICET 2014), Bandar Seri Begawan, Brunei, 1–3 November 2014; pp. 1–6.

[86] Hoyos, J.; Dehus, M.; Brown, T.X. Exploiting the goose protocol: A practical attack on cyber-infrastructure. In Proceedings of the 2012 IEEE Globecom Workshops (GC Wkshps), Anaheim, CA, USA, 3–7 December 2012; pp. 1508–1513.

[87] International Electrotechnical Commission. Communication networks and systems in substations—Specific communication service mapping (SCSM). IEC 61850-90-5, 2012.

[88] Wen, J.; Hammond, C.; Udren, E.A. Wide-area Ethernet network configuration for system protection messaging. In Proceedings of the 2012 65th Annual Conference for Protective Relay Engineers, College Station, TX, USA, 2–5 April 2012; pp. 52–72.

[89] Dehalwar, V.; Kalam, A.; Kolhe, M.L. Zayegh, A. Review of IEEE 802.22 and IEC 61850 for real-time communication in Smart Grid. In Proceedings on the 2015 International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, 16–19 December. 2015; pp. 571–575.

[90] Cintuglu, M.H.; Ma, T.; Mohammed, O. Protection of Autonomous Microgrids using Agent-Based Distributed Communication. *IEEE Trans. Power Deliv.* **2016**, doi:10.1109/TPWRD.2016.2551368.

[91] Cintuglu, M.H.; Mohammed, O.A. Multiagent-based decentralized operation of microgrids considering data interoperability. In Proceedings of the 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), Miami, FL, USA, 2–5 November 2015; pp. 404–409.

[92] Kumar, S.; Das N.; Islam, S. Performance evaluation of a process bus architecture in a zone substation based on IEC 61850-9-2. In Proceedings of the 2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Brisbane, QLD, Australia, 15–18 November 2015; pp. 1–5.

[93] Das, N.; Modi, H.; Islam, S. Investigation on architectures for power system communications between substations using IEC 61850. In Proceedings of the 2014 Australasian Universities Power Engineering Conference (AUPEC), Perth, WA, USA, 28 September–1 October 2014; pp. 1–6.

[94] Serra, M.; Castro, F. Using IEC 61850 for Teleprotection. In Proceedings of the 19th International Conference on Electricity Distribution, Vienna, Austria, 21–24 May 2007.

[95] Lo, B.T.; Mendes, F.M.; Samaniego, L.H.; Oliveira, S.R. Cloud IEC 61850: Architecture and Integration of Electrical Automation Systems. In Proceedings of the 2014 Brazilian Symposium on Computing Systems Engineering (SBESC), Manaus, Brazil, 3–7 November 2014; pp. 13–18.

[96] Cintuglu, M.H.; Youssef, T.; Mohammed, O.A. Development and Application of a Real-Time Testbed for Multiagent System Interoperability: A Case Study on Hierarchical Microgrid Control. *IEEE Trans. Smart Grid* 2016.

[97] Python. Available online: https://www.python.org/ (accessed on 16 June 2016).

[98] Scapy. Available online: http://www.secdev.org/projects/scapy/ (accessed on 16 June 2016).

[99] M. El Hariri, T. A. Youssef, H. Habib, and O. A. Mohammed, “Online False Data Detection and Lost Packet Forecasting System Using Time Series Neural Networks for IEC 61850 Sampled Measured Values,” in IEEE PES Innovative Smart Grid Technologies North America, Wishington DC, USA, 2017.

[100] Y. Zhu, J. Yan, Y. Tang, Y. L. Sun, and H. He, “Resilience Analysis of Power Grids Under the Sequential Attack,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 12, pp. 2340–2354, Dec. 2014.

[101] Y. Zhu, J. Yan, Y. Sun, and H. He, “Revealing Cascading Failure Vulnerability in Power Grids Using Risk-Graph,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 12, pp. 3274–3284, Dec. 2014.

[102] N, Falliere, L. O. Murchu, and E. Chein, "W32.Stuxnet Dossier", Symantic Security Response Report, Version 1.4, February 2011.

[103] DRAGOS INC. Report Version 2.20170613 "Crashoverride: Analyses of the Threat to Electric Grid Operation".

[104] M. E. Hariri, E. Harmon, H. F. Habib, T. Youssef and O. A. Mohammed, "A targeted attack for enhancing resiliency of intelligent intrusion detection modules in energy cyber physical systems," 2017 19th International Conference on Intelligent System Application to Power Systems (ISAP), San Antonio, TX, 2017, pp. 1-6.

[105] Y. Wu, Y. Xiao, F. Hohn, L. Nordström, J. Wang and W. Zhao, "Bad Data Detection Using Linear WLS and Sampled Values in Digital Substations," in IEEE Transactions on Power Delivery, vol. 33, no. 1, pp. 150-157, Feb. 2018.

[106] A. Anwar, A. N. Mahmood and Z. Tari, "Ensuring Data Integrity of OPF Module and Energy Database by Detecting Changes in Power Flow Patterns in Smart Grids," in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3299-3311, Dec. 2017.

[107] J. Ruan, H. Wang, S. Aziz, G. Wang, B. Zhou and X. Fu, "Interval state estimation based defense mechanism against cyber attack on power systems," 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, 2017, pp. 1-5.

[108] J. J. Q. Yu, Y. Hou and V. O. K. Li, "Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3271-3280, July 2018.

[109] M. S. Rahman, H. R. Pota, and M. J. Hossain, “Cyber vulnerabilities on agent-based smart grid protection system,” in 2014 IEEE PES General Meeting | Conference Exposition, 2014, pp. 1–5.

[110] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar and K. Poolla, "Smart grid data integrity attacks: characterizations and countermeasures," 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), Brussels, 2011, pp. 232-237.

[111] R. Mangoon, "Transmission Planning Guidelines (Revision 1)", A Report by Orange and Rockland Transmission and Substation Engineering Department, May 2008.

[112] Data Sheet for IEEE 14 Bus System. Available Online: http://shodhganga.inflibnet.ac.in/bitstream/10603/5247/18/19_appendix.pdf. Accessed on: 07/28/2017.

[113] V. Salehi, A. Mohamed, A. Mazloomzadeh and O. A. Mohammed, "Laboratory-Based Smart Power System, Part II: Control, Monitoring, and Protection," in IEEE Transactions on Smart Grid, vol. 3, no. 3, pp. 1405-1417, Sept. 2012.

[114] S. Fries, H. J. Hof and M. Seewald, "Enhancing IEC 62351 to Improve Security for Energy Automation in Smart Grid Environments," 2010 Fifth International Conference on Internet and Web Applications and Services, Barcelona, 2010, pp. 135-142. doi: 10.1109/ICIW.2010.28

[115] Available Online: https://cconell2858.wordpress.com/packet-sniffing-attack-prevention/

[116] International Electrotechnical Commission. Security for IEC 61850 profiles. IEC 62351-6.

[117] Pubudu Weerathunga, Security Aspects of Smart Grid Communication, The School of Graduate and Postdoctoral Studies Western University London, Ontario, Canada, 2012.

[118] Intel DPDK Validation team, DPDK Intel Cryptodev Performance Report, Release 17.11, Nov 2017.

[119] Peyrin Thomas, Sasaki Yu, Wang, Lei, Wang Xiaoyun, Sako Kazue, Generic Related-Key Attacks for HMAC, Advances in Cryptology – ASIACRYPT 2012, pp. 580-597, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[120] P. Jesu Jayarin, Visumathi, Srilakshmi, and Madhuri Pendyala, "A Secured Key Distribution for Effective File Transfer Using HMAC-SHA Algorithm with Self-Healing Property," Journal of Applied Security Research, vol. 10, no. 2, pp. 221–237, 2015.

[121] Fouque Pierre-Alain, Leurent, Gaëtan, Nguyen Phong Q, Menezes Alfred, Full Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5, Conference Proceedings Advances in Cryptology – CRYPTO 2007, Springer Berlin Heidelberg Berlin, Heidelberg, 2007.

[122] Su Sheng, Duan Xianzhong, W.L. Chan, Li Zhihuan, Erroneous measurement detection in substation automation system using OLS based RBF neural network, International Journal of Electrical Power & Energy Systems, Volume 31, Issues 7–8, September 2009, Pages 351-355, ISSN 0142-0615

[123] B. Chen, Y. Liu and Y. Zhang, "FS-LSSVM Based Fake Measurement Detection in Substation Automation System Using IEC 61850," 2010 Asia-Pacific Power and Energy Engineering Conference, Chengdu, 2010, pp. 1-5.

[124] Pengyuan Wang and M. Govindarasu, "Multi intelligent agent based cyber attack resilient system protection and emergency control," 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Minneapolis, MN, 2016, pp. 1-5.

[125] John DeDad, Looking For Sources of Transient Overvoltages. Available Online: http://ecmweb.com/contractor/looking-sources-transient-overvoltages (Accessed on: 03/27/2017).

[126] Shruti Hanumanthaiah and Srinvas NVNS, Design Considerations for Electrical Fast Transient (EFT) Immunity. Available Online: http://www.cypress.com/file/138636/download (Accessed on: 03/27/2017).

[127] Y. Yuan, Z. Li and K. Ren, "Modeling Load Redistribution Attacks in Power Systems," in IEEE Transactions on Smart Grid, vol. 2, no. 2, pp. 382-390, June 2011.

[128] R. Macwan et al., "Collaborative defense against data injection attack in IEC61850 based smart substations," 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, 2016, pp. 1-5.

[129] M. T. A. Rashid, S. Yussof, Y. Yusoff and R. Ismail, "A review of security attacks on IEC61850 substation automation system network," *Proceedings of the 6th International Conference on Information Technology and Multimedia*, Putrajaya, 2014, pp. 5-10.

[130] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh and J. C. Tan, "An Intrusion Detection System for IEC61850 Automated Substations," *in IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2376-2383, Oct. 2010.

[131] Nick Ismail, "How artificial intelligence is aiding the fight against cybercrime". Available Online: http://www.information-age.com/artificial-intelligence-aiding-fight-cybercrime-123461911/ (Accessed 04/07/2017).

[132] D. M. E. Ingram, P. Schaub, R. R. Taylor and D. A. Campbell, "Performance Analysis of IEC 61850 Sampled Value Process Bus Networks," *in IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1445-1454, Aug. 2013.

[133] J. M. Guerrero, M. Chandorkar, T. L. Lee and P. C. Loh, "Advanced Control Architectures for Intelligent Microgrids—Part I: Decentralized and Hierarchical

Control," in *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1254-1262, April 2013.

[134] J. M. Guerrero, P. C. Loh, T. L. Lee and M. Chandorkar, "Advanced Control Architectures for Intelligent Microgrids—Part II: Power Quality, Energy Storage, and AC/DC Microgrids," in *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1263-1270, April 2013.

[135] G. Hart, "Nonintrusive appliance load monitoring," Proceedings of the IEEE, vol. 80, no. 12, pp. 1870–1891, 1992.

[136] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," IEEE Trans. Consum. Electron., vol. 57, no. 1, pp. 76 –84, february 2011.

[137] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," Energy Policy, vol. 52, no. 0, pp. 213 – 234, 2013.

[138] Security is not enough! On privacy challenges in smart grids. Available Online: https://smartgridawareness.org/privacy-and-data-security/smart-grid-vulnerabilities-a-more-detailed-review/security-is-not-enough-on-privacy-challenges-in-smart-grids/ (Accessed on 09/17/2018).

[139] D. Egarter, C. Prokop and W. Elmenreich, "Load hiding of household's power demand," *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, 2014, pp. 854-859.

[140] Y. Sun, L. Lampe and V. W. S. Wong, "Combining electric vehicle and rechargeable battery for household load hiding," *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Miami, FL, 2015, pp. 611-616.

[141] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis and R. Cepeda, "Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures," *2010 First IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, 2010, pp. 232-237.

[142] D. G. Vutetakis and H. Wu, "The effect of charge rate and depth of discharge on the cycle life of sealed lead-acid aircraft batteries," *IEEE 35th International Power Sources Symposium*, Cherry Hill, NJ, 1992, pp. 103-105

[143] U.S. DOE MEG: https://building-microgrid.lbl.gov/microgrid-definitions

[144] S. Deilami, A. S. Masoum, P. S. Moses and M. A. S. Masoum, "Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile," in IEEE Transactions on Smart Grid, vol. 2, no. 3, pp. 456-467, Sept. 2011.

[145] Y. Cao et al., "An Optimized EV Charging Model Considering TOU Price and SOC Curve," in IEEE Transactions on Smart Grid, vol. 3, no. 1, pp. 388-393, March 2012.

[146] L. Pieltain Fernandez, T. Gomez San Roman, R. Cossent, C. Mateo Domingo and P. Frias, "Assessment of the Impact of Plug-in Electric Vehicles on Distribution Networks," in IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 206-213, Feb. 2011.

[147] O. Sundstrom and C. Binding, "Flexible Charging Optimization for Electric Vehicles Considering Distribution Grid Constraints," in IEEE Transactions on Smart Grid, vol. 3, no. 1, pp. 26-37, March 2012.

[148] S. Lee, Y. Park, H. Lim and T. Shon, "Study on Analysis of Security Vulnerabilities and Countermeasures in ISO/IEC 15118 Based Electric Vehicle Charging Technology," IT Convergence and Security (ICITCS), 2014 International Conference on, Beijing, 2014, pp. 1-4.

[149] A. C. F. Chan and J. Zhou, "Cyber–Physical Device Authentication for the Smart Grid Electric Vehicle Ecosystem," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 7, pp. 1509-1517, July 2014. doi: 10.1109/JSAC.2014.2332121

[150] Charlie Miller and Chris Valasek, “Adventures in Automotive Networks and Control Units”, http://www.ioactive.com/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf

[151] A. C. F. Chan and J. Zhou, "On smart grid cybersecurity standardization: Issues of designing with NISTIR 7628," in IEEE Communications Magazine, vol. 51, no. 1, pp. 58-65, January 2013. doi: 10.1109/MCOM.2013.6400439

[152] N. J. Al Fardan and K. G. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols," Security and Privacy (SP), 2013 IEEE Symposium on, Berkeley, CA, 2013, pp. 526-540. doi: 10.1109/SP.2013.42

[153] K. Bhargavan, A. D. Lavaud, C. Fournet, A. Pironti and P. Y. Strub, "Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS," 2014 IEEE Symposium on Security and Privacy, San Jose, CA, 2014, pp. 98-113. doi: 10.1109/SP.2014.14

[154] M. A. Mustafa, N. Zhang, G. Kalogridis and Z. Fan, "Smart electric vehicle charging: Security analysis," Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES, Washington, DC, 2013, pp. 1-6. doi: 10.1109/ISGT.2013.6497830

[155] A. C. F. Chan and J. Zhou, "A Secure, Intelligent Electric Vehicle Ecosystem for Safe Integration With the Smart Grid," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3367-3376, Dec. 2015. doi: 10.1109/TITS.2015.2449307

[156] S. Lee, Y. Park, H. Lim and T. Shon, "Study on Analysis of Security Vulnerabilities and Countermeasures in ISO/IEC 15118 Based Electric Vehicle Charging Technology," IT Convergence and Security (ICITCS), 2014 International Conference on, Beijing, 2014, pp. 1-4. doi: 10.1109/ICITCS.2014.7021815

[157] Avilable Online: https://sourceforge.net/projects/win32diskimager/

[158] Avilable Online: https://beagleboard.org/boards

[159] Avilable Online: https://source.android.com/devices/architecture/dto

[160] Avilable Online: github.com/jadonk/validation-scripts/blob/master/test-capemgr/

[161] Available Online: http://www.derekmolloy.ie/

VITA

MOHAMAD EL HARIRI

Born, Saida, Lebanon

2009-2012	B.Sc., Mechatronics Engineering, Hariri Canadian Univeristy, Mechref, Lebanon
2012-2014	M.Sc., Mechatronics Engineering, Rafik Hariri University, Mechref, Lebanon
2014-2015	Proposal Engineer, National Oilwell Varco, Jebel Ali Free Zone, Dubai, UAE
2015-2016	Research Assistant, American University of Beirut, Beirut, Lebanon
2016-2018	Research Assistant, Florida International University, Miami, Florida, USA
2018	Dissertation Year Fellowship, Florida International University, Miami, Florida, USA

SELECTED PUBLICATIONS AND PRESENTATIONS

Format: Journal [J-*n*], Conference [C-*n*], Patent [P-*n*]

[J-1] Mohamad El Hariri, Youssef TA, Mohammed OA. On the Implementation of the IEC 61850 Standard: Will Different Manufacturer Devices Behave Similarly under Identical Conditions? *Electronics*. 2016; 5(4):85.

[J-2] Mohamad El Hariri, Samy Faddel, and Osama Mohammed, "Physical-Model-Checking to Detect Switching-Related Attacks in Power Systems." Sensors Special Issue: Smart Grid Networks and Energy Cyber Physical Systems 2018, 18, 2478.

[J-3] Tarek A. Youssef, Mohamad El Hariri, Ahmad ElSayed, and Osama A. Mohammed, "A DDS-Based Energy Management Framework for Small Microgrid Operation and Control," in IEEE Transactions on Industrial Informatics, vol. PP, no. 99, pp. 1-1.

[J-4] Hany F. Habib, A.A.S. Mohamed, Mohamad El Hariri, Osama A. Mohammed, Utilizing supercapacitors for resiliency enhancements and adaptive microgrid protection against communication failures, Electric Power Systems Research, Volume 145, April 2017, Pages 223-233, ISSN 0378-7796.

[J-5] H. Habib, Mohamad El Hariri, A. Elsayed and O. A. Mohammed, "Utilization of Supercapacitors in Protection Schemes for Resiliency against Communication Outages: A Case Study on Size and Cost Optimization," in IEEE Transactions on Industry Applications, vol. PP, no. 99, pp. 1-1.

[C-1] Mohamad El Hariri, Tarek A. Youssef, Hany F. Habib, and Osama Mohammed, "Online False Data Detection and Lost Packet Forecasting System Using Time Series Neural Networks for IEC 61850 Sampled Measured Values", in IEEE Innovative Smart Grid Technologies North America (IEEE ISGT 2017), Washington DC, USA, April 23-25 2017.

[C-2] Mohamad El Hariri, E. Harmon, H. F. Habib, T. Youssef and O. A. Mohammed, "A targeted attack for enhancing resiliency of intelligent intrusion detection modules in energy cyber physical systems," 2017 19th International Conference on Intelligent System Application to Power Systems (ISAP), San Antonio, TX, 2017, pp. 1-6.

[C-3] Mohamad El Hariri, S. Faddel and O. Mohammed, "An artificially intelligent physical model-checking approach to detect switching-related attacks on power systems," 2017 IEEE 7th International Conference on Power and Energy Systems (ICPES), Toronto, ON, 2017, pp. 23-28.

[C-4] Mohamad El Hariri, Tarek Youssef, Hany F. Habib, and Osama Mohammed, "A Network-in-the-Loop Framework to Analyze Cyber and Physical Information Flow in Smart Grids", in the 2018 IEEE Innovative Smart Grid Technologies (ISGT) Asia. May 22-24, 2018.

[C-5] Tarek A. Youssef, Mohamad El Hariri, Nicole Bugay, and Osama A. Mohammed, "IEC 61850: Technology Standards and Cyber-Security Threats," In Proceedings of the 16th IEEE International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016.

[C-6] H. F. Habib, Mohamad El Hariri, A. Elsayed and O. Mohammed, "Utilization of supercapacitors in adaptive protection applications for resiliency against communication failures: A size and cost optimization case study," 2017 IEEE Industry Applications Society Annual Meeting, Cincinnati, OH, 2017, pp. 1-8.

[C-7] Abla O. Hariri, Mohamad El Hariri, Tarek Youssef, and Osama Mohammed, "Decentralized Multi-Agent System for Management of En Route Electric Vehicles," in the 2018 IEEE SouthEeastCon, St. Petersburg, Tampa, Florida, April 19-22 2018.

[P-1] Tarek Youssef, Mohamad El Hariri., and Osama Mohammed. Sequence hopping algorithm for securing goose messages. US Patent number US 15/285,126.