

Automated Microgrid Security Monitoring and Threat Hunting

Mark Karanfil

A Thesis

in the

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December 2025

© Mark Karanfil, 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: *Mark Karanfil*

Entitled: *Automated Microgrid Security Monitoring and Threat Hunting*

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (*Information Systems Engineering*)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____	Chair
<i>Dr. Hoi Dick Ng</i>	
_____	External Examiner
<i>Dr. Khalil El-Khatib</i>	
_____	Examiner
<i>Dr. Mohsen Ghafouri</i>	
_____	Examiner
<i>Dr. Lingyu Wang</i>	
_____	Arms-Length Examiner
<i>Dr. Otmane Ait Mohamed</i>	
_____	Supervisor
<i>Dr. Mourad Debbabi</i>	
_____	Co-supervisor
<i>Dr. Marthe Kassouf</i>	
_____	Co-supervisor
<i>Dr. Elnasser Abdelhafez</i>	
_____	Co-supervisor
<i>Dr. Aiman Hanna</i>	

Approved by

Dr. Chun Wang, Graduate Program Director

2025

Dr. Mourad Debbabi, Dean,
Gina Cody School of Engineering and Computer Science

Abstract

Automated Microgrid Security Monitoring and Threat Hunting

Mark Karanfil, Ph.D.

Concordia University, 2025

The microgrid is a valuable part of the growing smart grid critical infrastructure, generating its own electric power for local consumers while engaging in energy exchange with the main grid. As with other smart grid systems, the microgrid is reliant on the security of its Information Technology (IT) and Operational Technology (OT) networks for safe operation. Threat actors possessing large amounts of smart grid domain knowledge are a major threat to microgrid cybersecurity. Failure to protect the microgrid against these threat actors can lead to major loss of generation, data destruction, and equipment damage. Achieving extensive security coverage against microgrid cyberattacks requires consideration of numerous potential attack entry points, ranging from microgrid line equipment to hosts in the microgrid control centre. With this in mind, this thesis proposes a framework for automated microgrid security monitoring and threat hunting. The proposed automated framework incorporates several novel contributions related to microgrid cybersecurity. Among these contributions is a study on the effectiveness of various machine learning models for anomaly detection on IEC 62351-7:2017 Network and System Management (NSM) active monitoring data and for passive monitoring of power measurements to detect false data injection of line fault readings. Another contribution is a threat hunting approach for generating attack hypotheses based on available Cyber Threat Intelligence (CTI) and attributing the hypotheses to particular Advanced Persistent Threats (APTs). The final contribution is a risk-based methodology for predicting the subsequent actions of an attack campaign. A microgrid co-simulation platform is used to evaluate the impact of compromises to various microgrid components and to collect data in near-real time.

Acknowledgments

I would like to thank everyone who supported me throughout my academic studies and who contributed to the completion of this thesis.

I am grateful to my supervisors, Dr. Mourad Debbabi, Dr. Elnasser Abdelhafez, Dr. Marthe Kassouf, and Dr. Aiman Hanna, for guiding my research and providing valuable feedback on my thesis contents. Their feedback, backed by their deep knowledge and experience in my research field, has been very constructive.

I would like to extend my gratitude to Dr. Otmame Ait Mohamed, Dr. Lingyu Wang, and Dr. Mohsen Ghafouri for their time as members of my doctoral thesis examination committee alongside my supervisors. I would also like to deeply thank my external examiner, Dr. Khalil El-Khatib, for taking part in the examination committee, as well as my committee chair, Dr. Hoi Dick Ng.

I wish to acknowledge the collaboration and support that I have received from my colleagues Mr. Dhiaa Rebbah, Mr. Afshin Ebtia, Dr. Abolfazl Rahiminejad, Dr. Andrei Soeanu, and Dr. Mohammad Ali Sayed. Additionally, I wish to recognize the insights shared by Dr. Mohsen Ghafouri regarding the microgrid and power system simulation. Also, I very much appreciate the support and encouragement I have received from my other colleagues and professors within the smart grid security research group within the Concordia Institute of Information Systems Engineering.

Lastly, I am very thankful for the support and encouragement I have received from my family members. They have been with me throughout my academic journey and have shown patience and understanding towards the time I have spent completing this thesis. I love you all.

Contents

List of Figures	x
List of Tables	xii
List of Acronyms	xiv
1 Introduction	1
1.1 Problem Statement	3
1.2 Contributions	4
1.3 Thesis Organization	6
2 Background	7
2.1 Microgrid Architecture	7
2.1.1 Process Level Devices	8
2.1.2 Protection and Control Devices	8
2.1.3 Control Centre Devices	10
2.2 Microgrid Protection and Control Operations	10
2.2.1 Primary Control Layer	11
2.2.2 Secondary Control Layer	12
2.3 NSM According to IEC 62351-7	13
2.4 Basics of SNMP	14
2.5 Threat Hunting	15
2.6 MITRE ATT&CK Framework	16

3	Literature Review	19
3.1	Detection of Smart Grid FDI Attacks	19
3.1.1	Statistical Analysis	20
3.1.2	Spatial and Temporal Relationships	22
3.1.3	Deep Learning Models	23
3.1.4	Other Machine Learning Models	25
3.2	Threat Hunting	27
3.2.1	Leveraging of CTI on Known Attack Indicators	27
3.2.2	Application of Provenance Graphs	30
3.2.3	Leveraging the MITRE ATT&CK Framework	31
3.3	Attack Prediction	34
3.3.1	Response	35
3.3.2	Forecasting	37
3.3.3	Risk Assessment	39
3.4	Power System Simulation	40
4	Microgrid Model	43
4.1	Microgrid Co-simulation Testbed	43
4.1.1	Microgrid Power Simulation Model	44
4.1.1.1	Microgrid Control Network	48
4.2	Microgrid Threat Hunting System Model	51
4.2.1	System Model Topology	52
4.2.2	Mapping Microgrid Assets to MITRE ATT&CK	54
4.2.3	Simulation of the Microgrid Network	55
5	Active and Passive Security Monitoring	57
5.1	Anomaly Detection Model Design	58
5.2	Active Monitoring of Microgrid Telecontrol Data	59
5.2.1	Active Monitoring through IEC 62351-7 NSM	60
5.2.2	Testbed Implementation of NSM	62

5.2.2.1	NSM Agent Implementation	62
5.2.2.2	NSM Manager Implementation	63
5.2.3	Scalability	65
5.2.4	Prediction Model Training for NSM Anomaly Detection	68
5.2.5	Simulating Microgrid Attacks of Interest for Active Monitoring	71
5.2.6	Prediction Model Comparison	74
5.3	Passive Monitoring of Power Measurement Data	78
5.3.1	Attack Model	79
5.3.2	Passive Monitoring Setup	79
5.3.2.1	Passive Monitors	80
5.3.2.2	Central Security Manager	80
5.3.3	Passive Monitoring on the Microgrid Testbed	81
5.3.4	Data Model	84
5.3.5	Prediction Model Training for Power Quality Prediction Error	87
5.3.6	Anomaly Classification	88
5.3.7	Simulated Fault Spoofing FDI Attack Scenario	90
5.3.8	Evaluation of FDI Attack Detection	91
6	Threat Hunting	94
6.1	Microgrid Security Alerts	95
6.1.1	Alert Characteristics	96
6.1.2	Alerts from Active and Passive Monitoring	97
6.2	Overview of the Threat Hunting Approach	97
6.3	Security Alert Grouping	98
6.4	CTI Knowledge Graph	100
6.4.1	Construction of the KG	100
6.5	Matching Alerts to TTPs	104
6.5.1	Converting KG Nodes to Vector Embedding Space	104
6.5.2	Success Rate of Vector Embedding	107

6.6	Attack Hypothesis Construction	108
6.6.1	Unified Kill Chain	109
6.6.2	Security Alert Group Correlation	110
6.6.3	Considerations for Real-World Deployment	112
6.7	APT Attack Sequence Generation	113
6.7.1	Attack Sequence Generation Methodology	113
6.7.2	Usage of the Tuning and Validation Sets	115
6.8	APT Identification Method	116
6.9	Attack Hypothesis Severity	117
6.10	Managing the Number of Hypotheses with MCDM	121
6.11	Managing Noisy Alerts with the APT Verification Step	126
6.11.1	APT Verification Step	126
6.11.2	Considerations for Novel Attacks	127
6.12	Experimental Evaluation	128
6.12.1	Security Alert Group Correlation Accuracy	128
6.12.2	APT Identification Accuracy	129
6.12.3	Hypothesis Validation Statistics	132
7	Attack Step Prediction	134
7.1	Likelihood of Subsequent Attack Steps According to CTI	137
7.1.1	Bayesian Network	138
7.1.2	Preparing the Attack Sequence Dataset	139
7.1.3	Bayesian Network Structure	141
7.1.4	Measuring Information Gain from Attack Sequence Evidence	143
7.1.5	Use of Bayesian Network for Risk Assessment	146
7.2	Path Exploitability	148
7.2.1	Exploit Resistance	149
7.2.2	Path Distance Factor	152
7.2.3	Computing Path Exploitability	154

7.2.4	Use of Path Exploitability for Risk Assessment	155
7.3	Impact Evaluation	155
7.3.1	Impact Criteria	156
7.3.2	Impact Scoring for Asset Compromise	157
7.3.2.1	Simulated Attack Example: Disconnecting the Wind Farm	161
7.3.2.2	Impact Scores Across All Microgrid Assets	163
7.3.3	Use of Impact Score for Risk Assessment	165
7.4	Measuring Risk	166
7.4.1	Example of Risk Value Calculation	168
7.4.2	Risk Value Statistics Across All APTs and Microgrid Assets	169
7.4.3	Alignment of Risk Value with Attacker Intent	172
8	Conclusion and Future Work	176
8.1	Summary of Contributions	177
8.2	Insights	178
8.3	Future Research	179
	References	194

List of Figures

Figure 2.1	Relationships within the ICS MITRE ATT&CK CTL.	17
Figure 2.2	MITRE ATT&CK Matrix for ICS.	18
Figure 3.1	Related work on smart grid FDI.	20
Figure 3.2	Related work on cyber threat hunting.	27
Figure 3.3	Work related to attack prediction.	35
Figure 4.1	High-level view of microgrid power simulation model.	45
Figure 4.2	Local ESS control scheme for PV and PQ mode.	47
Figure 4.3	Frequency deviation after closing PCC based on phase angle difference.	51
Figure 4.4	Voltage phasor angle θ according to the phasor quadrant.	52
Figure 4.5	Devices involved in primary and secondary microgrid control.	54
Figure 4.6	Devices in the microgrid control centre.	54
Figure 4.7	Network topology of the microgrid system model.	55
Figure 5.1	Design of the ANN detection process.	60
Figure 5.2	Architecture of the recurrent neural network prediction model.	61
Figure 5.3	General scheme of NSM.	62
Figure 5.4	Flow of microgrid operation traffic and SNMP for NSM monitoring.	64
Figure 5.5	SNMP delay as a function of number of agents and MIBs requested.	67
Figure 5.6	SNMP and snapshot creation delay as a function of number of agents.	68
Figure 5.7	Standard linear correlation coefficient across all features.	69
Figure 5.8	Frequency waveform during Attack I.	74
Figure 5.9	Active power consumption of load 1 during Attack I.	74

Figure 5.10	Microgrid frequency during Attack II.	75
Figure 5.11	Frequency of microgrid during Attack III.	75
Figure 5.12	Design of the FDI attack detection and response system	82
Figure 5.13	Effect of shunt compensation fault at ESS 3 and ESS 2.	83
Figure 5.14	Placement of the passive monitors and relays in the microgrid.	84
Figure 5.15	Effect of a fault near the PCC at various microgrid locations.	85
Figure 5.16	Impact of FDI attack on microgrid power line.	91
Figure 6.1	Summary of the threat hunting methodology.	98
Figure 6.2	Steps taken to find total similarity score from data components.	107
Figure 6.3	Composition of attack hypotheses from grouped security alerts.	110
Figure 6.4	Adaptation of the Unified Kill Chain tactics from Enterprise to ICS.	111
Figure 6.5	Example of UKC order distance from the Persistence tactic.	112
Figure 6.6	Structure of the generated APT attack sequences.	115
Figure 6.7	Illustrative example of AHP for three attack hypotheses.	124
Figure 6.8	Number of attack hypotheses with and without MCDM.	127
Figure 6.9	APT identification results for different severity thresholds.	132
Figure 6.10	Tactic reached before hypothesis is validated.	133
Figure 7.1	Overview of the components involved in calculating risk value.	137
Figure 7.2	Dividing of an attack sequence into a truncation set.	141
Figure 7.3	Connectivity of attack features within the Bayesian Network.	143
Figure 7.4	Impact on microgrid frequency due to WTG tripping attack.	162
Figure 7.5	Impact on PV, WTG, and CHP due to WTG tripping attack.	163
Figure 7.6	Impact on ESSs and loads due to WTG tripping attack.	164
Figure 7.7	Relationship between truncation set elements and predictions.	167
Figure 7.8	Risk value histograms for (a) Profile A (b) Profile B (c) Profile C.	168
Figure 7.9	Average percentile of end goal asset risk values for different weights.	174
Figure 7.10	Average percentile of end goal asset risk values for different APTs.	175

List of Tables

Table 2.1	Arrangement of microgrid devices and hosts into security levels	9
Table 4.1	Mapping of microgrid system asset to ICS ATT&CK assets	56
Table 5.1	MIB objects collected for microgrid active monitoring	63
Table 5.2	Validation loss on single hidden layer neural network models	71
Table 5.3	Validation loss on two hidden layer neural network models	72
Table 5.4	Validation loss on three hidden layer neural network models	72
Table 5.5	Comparison between different DNN prediction models	76
Table 5.6	Comparison between classifiers and anomaly-based approach	78
Table 5.7	FDI attack detection results for different machine learning models	93
Table 6.1	MIB objects collected for microgrid active monitoring	98
Table 6.2	Technique precision and recall for data component and asset combinations .	109
Table 6.3	Weights and assets assigned to microgrid functions	119
Table 6.4	Severity levels and weights for hypothesis feature value ranges	120
Table 6.5	Pairwise comparison table of criteria weights for the AHP	124
Table 6.6	Impact of MCDM filtering on the size of the active hypothesis pool	126
Table 6.7	Correctly correlated security alerts groups	130
Table 6.8	APT identification success rate	130
Table 7.1	Features extracted for the Bayesian network	142
Table 7.2	Prior probabilities of the next asset to be targeted	145
Table 7.3	Entropy of prediction features with varying amounts of evidence	145
Table 7.4	Overall next target posterior probability for all ATT&CK ICS assets	149

Table 7.5	APT next target posterior probability for all ATT&CK ICS assets	150
Table 7.6	Exploit resistance values for Profile B and Profile C	152
Table 7.7	Network hop distance between various microgrid assets of interest	153
Table 7.8	Path distance factor computed for different parameter combinations	154
Table 7.9	NESCOR impact criteria used for proposed microgrid risk assessment	157
Table 7.10	Ranges of conditions for each impact criteria value	159
Table 7.11	Assets associated with the microgrid attacks being simulated	161
Table 7.12	Impact scores assigned to assets directly associated with attacks	165
Table 7.13	Average risk values for APT	170
Table 7.14	Average risk values for ATT&CK assets	171

List of Acronyms

AC	Alternating Current
AHP	Analytic Hierarchy Process
AMI	Advanced Metering Infrastructure
ANN	Artificial Neural Network
ANP	Analytic Network Process
APDU	Application Protocol Data Unit
APT	Advanced Persistent Threat
ASDU	Abstract Service Data Unit
ASN.1	Abstract Syntax Notation One
C&C	Command and Control
CB	Circuit Breaker
CHP	Combined Heat and Power
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
DC	Direct Current
DER	Distributed Energy Resource
DNN	Deep Neural Network
DNP3	Distributed Network Protocol 3

DoS	Denial-of-Service
DRTS	Digital Run-Time Simulator
ESS	Energy Storage System
FDI	False Data Injection
FPR	False Positive Rate
FRT	Fault Ride Through
GOOSE	Generic Object-Oriented Substation Event
GRU	Gated Recurrent Unit
HIL	Hardware-in-the-Loop
HMI	Human-Machine Interface
HMM	Hidden Markov Model
ICS	Industrial Control System
IDS	Intrusion Detection System
IEC-104	IEC 60870-5-104
IED	Intelligent Electronic Device
IoC	Indicator of Compromise
IP	Internet Protocol
IT	Information Technology
KG	Knowledge Graph
KNN	K-Nearest Neighbor
LLM	Large Language Model

LSTM	Long Short-Term Memory
MAC	Media Access Control
MAUT	Multi-Attribute Utility methods
MCDM	Multi-Criteria Decision Making
MGCC	Microgrid Central Controller
MIB	Management Information Base
MitM	Man-in-the-Middle
MLP	Multi-Layer Perceptron
MU	Merging Unit
NESCOR	National Electric Sector Cybersecurity Organization Resource
NIDS	Network Intrusion Detection System
NLP	Natural Language Processing
NSM	Network and System Management
NSM DO	Network and System Management Data Object
OT	Operational Technology
PCA	Principal Component Analysis
PCC	Point of Common Coupling
PI	Proportional Integral
PLC	Programmable Logic Controller
PV	Photovoltaic
RAS	Remote Access Server
RDP	Remote Desktop Protocol

RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SCS	Semantic Connectivity Score
SEI-ETF	Securing Energy Infrastructure Executive Task Force
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SNMPv2	Simple Network Management Protocol Version 2
SNMPv3	Simple Network Management Protocol Version 3
SoC	State of Charge
SV	Sampled Value
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TNR	True Negative Rate
TPR	True Positive Rate
TTP	Tactic, Technique, or Procedure
UDP	User Datagram Protocol
UKC	Unified Kill Chain
VM	Virtual Machine
VPN	Virtual Private Network
WTG	Wind Turbine Generator

Chapter One

Introduction

The smart grid refers to the electric power grid that has been enhanced with digital technologies for improved automation, monitoring, and flexibility. It also uses digital technology to integrate renewable energy sources [1]. Being responsible for supplying electric power to large populations, the smart grid makes up part of the critical infrastructure. Among the innovative technologies associated with the smart grid is the microgrid. The microgrid is a localized energy generation and distribution system that integrates multiple Distributed Energy Resources (DERs), both renewable and non-renewable, to serve nearby consumer loads [2]. Given its close proximity to consumers, the microgrid reduces power transmission losses and thus improves on power efficiency [3]. The microgrid has two operation modes: grid-connected mode and islanded mode. The microgrid is coupled to the main grid in grid-connected mode to allow for energy exchange between them, whereas islanded-mode has the microgrid operate independently from the main grid [4]. Islanding allows the microgrid to operate safely when the main grid is suffering from instability, thus enhancing grid reliability [5], [6], [7]. However, the islanded microgrid has lower inertia and is thus more susceptible to frequency instability under changing operational conditions compared to when it is connected to the main grid [8].

The microgrid is reliant on both Information Technology (IT) and Operational Technology (OT) to operate effectively. This includes the Industrial Control System (ICS) communication protocols used to monitor and control field devices that provide real-time visibility and control over the power system. The use of such technology puts the microgrid at risk of cyberattacks that can cause

significant harm to physical processes. This risk is made apparent by recent cyberattacks from state-sponsored threat groups that targeted the energy sector. The Ukrainian power grid has faced multiple cyberattacks from Advanced Persistent Threat (APT) groups in the past decade, as recently as 2022 [9], [10], [11]. Meanwhile, it was reported in February of 2021 that India had its power sector targeted by a concerted attack campaign, with concerns that the campaign had been responsible for the blackout that occurred in October of 2020 [12].

There are various entry points from which cyberattacks can originate. Attackers can compromise individual Intelligent Electronic Devices (IEDs) that manage local DERs, establish themselves within the communication network used for distance communication, or intrude on the control centre that monitors and controls the microgrid as a whole [8]. Depending on their entry point, cyberattacks can use different methods to inflict damage on the microgrid. A layered defense approach is necessary to achieve extensive security coverage for the safe operation of the microgrid.

This thesis proposes a framework for automated microgrid security monitoring and threat hunting. The proposed security monitoring involves both passive monitoring and active polling to collect microgrid telemetry data and uses machine learning models to detect anomalies within the collected data. The threat hunting uses the microgrid system topology and Cyber Threat Intelligence (CTI) on attacks from APTs that target ICS networks to automate the correlation of microgrid security alerts into an attack hypothesis. The attack hypothesis, upon crossing a configured severity threshold, is analyzed to identify the APT responsible for the alerts. This thesis also proposes a method to use details of a given attack hypothesis to assess predictions on what attack actions will follow as the APT propagates further to achieve its end goal in the microgrid. In order to assess the impact of potential attacks and evaluate the experimental results of the proposed framework, the microgrid co-simulation testbed developed previously by the PhD Candidate in [13] is extended to support the reading of simulated line measurements in near-real time and to test more attack scenarios. The testbed integrates a microgrid power system simulated on HYPERSIM [14] with an OpenStack network emulation server [15] containing Virtual Machines (VMs) that host microgrid applications and attack simulation tools.

1.1. Problem Statement

The purpose of this thesis is to design and develop a novel microgrid cybersecurity framework to protect the microgrid against malicious threat actors, including APTs. An APT can be described as a sophisticated and persistent cyber threat that operates stealthily for long periods [16]. The proposed framework spans multiple levels of the microgrid control hierarchy to provide a layered defense against APTs. The framework includes learning-based anomaly detection for microgrid telemetry, automated threat hunting capabilities that generate and validate attack hypotheses, and a method to predict the next actions of an APT.

The nature of the microgrid poses a challenge for model-based approaches. Notably, the microgrid exhibits bi-directional power flow across DERs and has a non-linear relationship between line fault current and the distance from the fault [17], [18]. This makes learning-based methods more suitable for classifying microgrid power quality data. However, such methods need to ensure that they are aligned with the availability and real-time performance needs of the microgrid measurement and command payloads transmitted over ICS protocols. Failure to meet these needs can result in the instability of the microgrid power system. Learning methods must also contend with the lack of sample data for attacks targeting the microgrid. The anomaly detection scheme proposed in this thesis uses a data-driven, unsupervised learning approach to identify anomalies within microgrid telemetry. It trains prediction models on normal operation data, acquired from a simulated microgrid model, in order to accommodate for the lack of microgrid attack samples.

The IEC 62351 suite of standards gives a wide range of specifications for power system security and management. IEC 62351 Part 7 (IEC 62351-7:2017) [19] specifies the power system data to monitor through Network and System Management (NSM) to support security objectives. This includes the monitoring of the IEC 60870-5-104 (IEC-104) [20] communication used for microgrid telecontrol. Security monitoring tools involving machine learning can leverage NSM data for detection. However, IEC 62351-7:2017 does not give any specifications or guidelines on how to analyze the NSM data for detecting intrusions. Addressing the gap in analysis of NSM data, this thesis explores the effectiveness of its proposed anomaly detection scheme on NSM data across various machine learning model architectures and hyperparameters. In doing so, it highlights the strengths

and limitations surrounding the use of NSM data for security monitoring.

Confidently identifying the presence of an APT in the network, let alone a particular APT, is not a straightforward task. Existing Intrusion Detection Systems (IDSs) often rely on detecting known Indicators of Compromise (IoCs) [21]. However, relying on IoCs for detection is likely insufficient to detect APT activity due to APTs regularly changing their Internet Protocol (IP) addresses and malware hashes to ones not yet recognized as IoCs [22], [23]. An IDS may also employ anomaly detection methods, which are useful for identifying misuse and generating alerts that signal the presence of a potentially malicious actor. However, a security alert considered in isolation can be hard to distinguish between an APT and the many false positives that anomaly detection methods tend to generate [22], [24]. It is through threat hunting that alerts can be correlated into attack hypotheses that facilitates the identification and assessment of APT attack campaigns. Yet, the evaluation of attack hypotheses is largely a manual process carried out by analysts [25]. The lack of prompt, automated evaluation of attack hypotheses puts the system at risk of the threat actor executing the final stages of their attack before a proper response can be mounted following proper analysis of the hypothesis. The threat hunting component of the proposed framework accounts for the microgrid system topology when correlating security alerts. It also leverages CTI on APT attack behaviour to help quantify the hypothesis severity and automate the identification of candidate APTs for a hypothesis. From there, predictions for what the APT will do next are made based on attack hypothesis features and the risk that the hypothesized attack presents to microgrid operations. The CTI used in this thesis is sourced from the MITRE ATT&CK® framework [26] and the APT campaign database published by Di Tizio et al. [27].

1.2. Contributions

The contributions of this thesis are as follows:

- A study on the most suitable Deep Neural Network (DNN) models and configurations for detecting anomalies within IEC 62351-7:2017 NSM data collected through active monitoring.

The use of DNN for anomaly detection on microgrid IEC 62351-7:2017 NSM data for microgrid cyberattack detection is proposed in the master's thesis of the PhD candidate [13]. A corresponding conference paper is published in [28]. A journal paper presenting more extensive evaluation of the proposed anomaly detection regarding the choice of machine learning model and the scalability of the monitoring solution are published in [29].

- The design of a passive monitoring scheme involving distributed monitors communicating with a central security manager to detect fault-spoofing FDI within physical microgrid power measurements in real-time.

The proposal of using DNNs to generate prediction errors from passive monitoring data of power quality measurements, derived from the instantaneous current and voltage readings, that are subsequently compared against each other to detect False Data Injection (FDI) of fault measurements is published as a conference paper in [30].

- A threat hunting approach that uses available CTI and microgrid system information to determine the relative severity of generated attack hypotheses and to identify the APT group behind an ongoing microgrid attack campaign.

A manuscript detailing the microgrid threat hunting approach is in the process of being submitted to a journal venue. Included in this manuscript are the methods used to correlate security alerts into attack hypotheses based on available CTI and a kill chain model adapted to ICS. It also presents the severity assessment, validation process, and APT identification steps applied on the attack hypotheses.

- A risk-based methodology for predicting the next actions of a threat actor behind a given attack hypothesis in the microgrid.

1.3. Thesis Organization

Several research directions are explored in this thesis in order to address the research gaps mentioned above. This thesis is organized as follows; Chapter 2 provides background information on several topics, including microgrid operations, NSM, and threat hunting. Chapter 3 presents a literature review on smart grid FDI detection, threat hunting, attack prediction, and power system modeling. Chapter 4 is dedicated to describing the microgrid co-simulation testbed used for this thesis research. Chapter 5 presents contributions related to the proposed microgrid security monitoring framework employing DNNs for both active and passive monitoring. Chapter 6 covers the aspects of the proposed threat hunting approach involving the association of alerts to Tactics, Techniques, and Procedures (TTPs), the correlation of alerts into attack hypothesis, and the identification of threat actors most likely to be the source of an observed attack campaign. Chapter 7 presents a risk-based methodology that assesses predictions for the next steps of an attack campaign, with the methodology complementing the threat hunting activities described in the preceding chapter. Finally, Chapter 8 concludes the thesis and discusses possible directions for future work.

Chapter Two

Background

This chapter elaborates on key aspects related to the research topics and proposed solutions covered in this thesis. First, the overall microgrid architecture is presented along with the relevant microgrid protection and control devices. The protection and control needs of the microgrid are then discussed, as are the communication protocols used to transmit the data needed to meet these needs. Next, details are given on the IEC 62351-7:2017 standard for the use of NSM in power systems. This is followed by an explanation of Simple Network Management Protocol (SNMP), which is specified in IEC 62351-7:2017 for the collection of NSM data. An overview on threat hunting is then presented. Finally, the MITRE attack framework is presented in the last section of this chapter.

2.1. Microgrid Architecture

The U.S. Department of Energy describes the microgrid as a set of interconnected loads and DERs, each with their own defined electrical boundaries, that is controllable as a single entity by the utility grid [31]. The microgrid power system integrates various DERs into a shared power system, with each DER being a source of renewable or non-renewable electrical power generation. Energy sources include Direct Current (DC) sources, such as solar Photovoltaic (PV) arrays and Energy Storage Systems (ESSs), and Alternating Current (AC) sources, such as Wind Turbine Generators (WTGs) and Combined Heat and Power (CHP) generators [7]. The main microgrid bus shared by the DERs can be DC or AC, but any power line connecting the microgrid to the main grid through

a Point of Common Coupling (PCC) must be AC to match the AC power of the main grid [3]. Inverters are used to convert DC power from the DERs to AC power, or vice-versa, as necessary.

The microgrid architecture has a hierarchical structure and can loosely be described as having a star topology, where the DERs and loads can be viewed as spokes connecting to the microgrid control centre. The microgrid control centre consists of an IT network infrastructure made up of workstations and servers that observe, manage, and transmit OT data to the rest of the microgrid. The availability and integrity of the microgrid OT data and devices must be maintained in order to ensure safe and reliable power generation and delivery.

The Securing Energy Infrastructure Executive Task Force (SEI-ETF) [32] provides a reference architecture for the DER profile associated with microgrid operation. The profile breaks down the DER management into security levels, which is itself inspired by the Purdue model [33]. It describes the devices typically found within each security level, as well as the function security features of each level. Table 2.1 sorts the devices of interest for this thesis by security level. Compared to enterprise IT networks, the microgrid DER network is relatively heterogeneous when it comes to the types of devices involved in its operation. The roles of these devices are briefly discussed in the subsections to follow.

2.1.1. Process Level Devices

Among the microgrid protection and control devices, the Merging Units (MUs) and Circuit Breakers (CBs) found in each DER and load lie closest to the actual microgrid power lines. The MUs digitize the analog current and voltage signals they receive from the measurement current and voltage transformers, transmitting the digital measurements to the local protection relays [8]. The CBs disconnect the power line from the rest of the microgrid to help isolate a line fault that is causing unsafe voltage and current levels. The CBs do not necessarily act on their own. They will disconnect the line in response to a trip signal received from protection relays.

2.1.2. Protection and Control Devices

The PCC is the junction point where the microgrid connects with the rest of the main grid [34]. When the PCC line is closed and power flows through it, the microgrid is said to be in grid-connected

Table 2.1: Arrangement of microgrid devices and hosts into security levels

Level	Devices	Functions
3.5	Remote Access Server	Access Point
3	Workstations, Database Servers, Application Servers, Telecontrol Servers, Microgrid Central Controller	Internal Operational Communication
2	Supervisory Controllers	Process Data Conversion, Local Control, Asset Monitoring
1	Point of Common Coupling, Protection Relays, Load Controllers, Energy Storage System Controllers	Data Acquisition, Local Control
0	Merging Units, Circuit Breakers	Physical Process Interface

mode. The microgrid is said to be in islanded mode when it is disconnected from the main grid, with the line being disconnected at the PCC. The PCC is responsible for managing the transition to islanded mode upon receiving the disconnection signal from the control centre. A controller is needed at the PCC to synchronize the microgrid to the main grid when transitioning from islanded mode to grid-connected mode. The PCC cannot safely reconnect to the main grid if the phase angle and frequency of the main grid and microgrid are not sufficiently close [35].

Protection relays are responsible for quickly responding to unsafe line conditions by sending a trip signal to the corresponding CB so that the CB disconnects the line from the rest of the grid. They read the measurement stream transmitted by the local MU and determine if the conditions have been met to send a trip signal. The trip settings for the relays need to be selected carefully to meet the protection needs of its line. A trip threshold that is too high will prevent line faults from being properly isolated, while thresholds that are too low risk causing unnecessary disconnections [36].

Within the context of this thesis, the load controller refers to the device that acts on the load setpoint signals, sent from the supervisory controller on behalf of the microgrid control centre, to dynamically adjust the amount of power being consumed at the load. Similarly, the ESS controllers act on setpoint signals to adjust the amount of power provided, or absorbed, by their corresponding ESS. However, the ESS setpoint signals are influenced by both the local control loop of the ESS and the corrective signals sent from their supervisory controller, again on behalf of the microgrid control

centre. The supervisory controllers each act as an Remote Terminal Unit (RTU) associated with a single DER, load, or PCC. They communicate with the microgrid control centre clients, acting on commands they receive and transmitting IT and OT telemetry back to the control centre.

2.1.3. Control Centre Devices

The microgrid control centre workstations host a human-operable computer operating system, for example a laptop or a desktop. Operators use them to perform various tasks. The Remote Access Server (RAS) is a network jump point that allows authorized users to connect to their workstations remotely. As such, the RAS is an entry point of high concern from a security standpoint. Access to the RAS should be closely managed and monitored to help mitigate intrusions from potential threat actors passing through it. Database servers store pertinent data for microgrid operation and system management. Examples of pertinent data would be telemetry history, past alerts, and static device specifications. Application servers provide access to software tools, such as data visualization dashboards, that can assist operators in accomplishing their tasks.

For this thesis, a distinction is made between the telecontrol server and the Microgrid Central Controller (MGCC). The telecontrol server acts as a gateway between control centre operators and the various devices that manage the microgrid DERs and loads. The telecontrol server considered in this thesis is comparable to an OPC server [37], ensuring interoperability between control centre applications and the ICS applications and protocols used by the microgrid DER equipment. Meanwhile, the MGCC is an automated application running in the control centre that is involved in microgrid control loops and protection responses. It receives measurement and confirmation messages from the DERs supervisory controllers and sends commands and setpoints to the DER controllers as appropriate.

2.2. Microgrid Protection and Control Operations

Microgrid control can be separated into three layers: primary control, secondary control, and tertiary control [38], [39]. Primary control refers to the primary control loops that the DERs use to maintain grid stability at the local level. Secondary control refers to the actions that the microgrid

control centre takes on the DERs and loads, with the control centre having visibility that spans the entire microgrid. Tertiary control involves the coordination of multiple microgrids to achieve long term goals. The tertiary control level is outside of the scope of this thesis.

Cyberattacks that breach the primary and secondary control levels can have adverse effects on the microgrid and can even cause catastrophic damage under certain circumstances. It is therefore essential to detect and act on cyberattacks that threaten the primary and secondary control levels.

2.2.1. Primary Control Layer

Each DER and load in the microgrid has its own local protection measures, and possibly a local power generation control loop as well. In the microgrid system under consideration for this thesis, each line has a MU that reports the line current and voltage measurements to the local protection relay over a local Ethernet network. Specifically, the MU broadcasts three-phase current and voltage measurements as a stream of IEC 61850 Sampled Value (SV) Ethernet frames [40] at 4,800 frames per second. The Ethernet network is shared between the MU, protection relay, and the CB. The protection relay subscribes to the IEC 61850 SV stream of the MU to monitor the line current and voltage levels. If the relay senses that current or voltage exceeds its configured safety thresholds, the relay broadcasts a trip command over the Ethernet network using the IEC 61850 Generic Object-Oriented Substation Event (GOOSE) protocol [41]. The local CBs subscribe to the IEC 61850 broadcasts. While IEC 61850 SV and GOOSE are protocols originally designed to fulfill these roles in the IEC 61850 automated substation, their easy integration and high interoperability has led to them being recommended for use in microgrid systems as well [8].

Note that the term “frame” is used for Ethernet messages instead of “packet” to distinguish them from messages sent over IP protocols, with the major difference between them being that the Ethernet frames lack the IP header fields that specify the IP source and destination addresses. Ethernet frames are forwarded across the network according to the source and destination Media Access Control (MAC) addresses contained in their Ethernet header fields.

An important function of microgrid primary control is local frequency regulation performed by dispatchable DERs. Frequencies that deviate from the nominal value are a result of power imbalance between generators and loads [2]. In this context, a dispatchable DER refers to an energy source

whose power output undergoes granular adjustment by the power system control while online. For example, the ESS is a dispatchable DER whose power output can be increased or reduced in order to maintain power balance across the microgrid energy sources and loads. To this end, the ESS can even be made to absorb power from the microgrid, storing the absorbed energy for future use. In contrast, non-dispatchable energy sources like the solar PV array or the wind farm have their power output largely determined by the environment, though portions of the PV array or wind farm can be disconnected under more severe circumstances. The dispatchable DERs engaging in local frequency control determine the frequency by reading the measurements from the local MU.

Specific guidelines on the allowable tolerances for microgrid voltage and frequency deviations are given in the IEEE 1547 standard for DER interconnection and interoperability [42]. For the North American grid, the nominal frequency is 60 Hz. Under relatively safe conditions, the dispatchable DERs will adjust their active power output to maintain the nominal frequency. Line faults will result in more significant voltage and frequency deviations, resulting in unsafe conditions. However, the microgrid should avoid disconnecting in the event of a fault that is merely transient, as short-lived faults do not necessarily warrant the disconnection of an entire line. As such, the IEEE 1547 standard specifies that the microgrid is to inject reactive power upon detecting large voltage or frequency deviations as part of a Fault Ride Through (FRT) response. This response helps stabilize the grid during a transient fault. If the microgrid does not stabilize within the designated fault clearing time, then the microgrid will disconnect. The IEEE 1547 standard specifies a minimum fault clearing time of 0.16 seconds. The FRT response in microgrid systems is typically performed by the ESS and solar PV array, with the amount injected reactive power being proportional to the nominal voltage and observed voltage [43].

2.2.2. Secondary Control Layer

The MGCC is responsible of maintaining microgrid secondary control. It has monitoring and control capabilities on the DER, PCC, and loads. The MGCC communicates with the supervisory controllers over a distance network. Through this communication, the MGCC is informed of the microgrid frequency, the state of the CBs, and the State of Charge (SoC) of the ESS. Based on this telemetry, the MGCC sends commands to adjust the ESS power setpoints and load consumption

setpoints to support frequency regulation alongside the primary control loops. The MGCC can also manually open line switches. For example, the MGCC can disconnect some number of WTGs from the rest of the wind farm to curtail power in cases of over-frequency.

This thesis considers a microgrid system where communications between the MGCC and the supervisory controllers is done using IEC-104 over Transmission Control Protocol (TCP) [20]. The IEC-104 protocol is often used for communications between the Supervisory Control and Data Acquisition (SCADA) network and remote devices that are in closer proximity to the line equipment, though Distributed Network Protocol 3 (DNP3) [44] is a suitable alternative [45]. The IEC-104 protocol follows a client/server model, with the MGCC acting as the client and the supervisory controllers acting as servers. This protocol sends Application Protocol Data Units (APDUs), which have one or more data fields. Each data field in the APDU has an Abstract Service Data Unit address to use when referencing. The fields are configured with one of several possible data type. For example, the CB status of a DER can assign its data field with a binary value, while a float value is more appropriate for the data fields associated with the ESS power setpoints. A notable shortcoming of IEC-104 as defined in its original standard is its lack of security features, having no mechanisms for authentication, encryption, or integrity checking [46], [47].

2.3. NSM According to IEC 62351-7

IEC 62351 [48] is a suite of standards that specify security measures to protect power system data and communication. Together, these standards give actionable recommendations to support the end-to-end security of the power system against both deliberate and inadvertent threats. Examples of deliberate threats accounted for by the standard are disgruntled employees, industrial espionage, hackers, and terrorist activity. Though compliance with IEC 62351 is non-compulsory, it is highly relevant due to its specifications covering many essential communication protocols and environments within the power system.

This thesis explores the effectiveness and practicality of IEC 62351 Part 7 (IEC 62351-7) [19]. The IEC 62351-7:2017 standard defines NSM within the power system to support active monitoring of system health, system performance, and potential intrusions. Specifically, it defines Network and

System Management Data Objects (NSM DOs) to be polled and analyzed at regular intervals. In terms of microgrid relevancy, network statistics related to the IEC-104 protocol are among the NSM DOs classes included in IEC 62351-7:2017. However, it is up to security operators to decide on how to interpret the NSM DOs. The IEC 62351-7:2017 standard specifies that SNMP [49] is to be used for collecting NSM DOs. Indeed, the NSM DOs are all defined within Management Information Base (MIB) files [50], allowing them to be readily integrated into SNMP deployments.

2.4. Basics of SNMP

The role of the SNMP digital communication protocol is to collect data related to the state of managed network devices. The SNMP messages are transmitted over User Datagram Protocol (UDP) without acknowledgment. Given the nature of UDP, there is no guarantee that SNMP requests will be adequately served by the request sender. Managed devices are installed with SNMP agent applications that report on the state of the managed device and the traffic it has generated and observed. The scope of the information that is accessible through the SNMP agent is specified within the MIBs configured on the agent. The standard version of SNMP supports the MIBs defined in RFC-3418 [50]. Each MIB contains one or more objects that each correspond particular piece of information about the device. Each MIB object has an underlying implementation that retrieves or maintains record of the value representing the information. The SNMP agents are lightweight in that they do not maintain a history of past values for the MIB objects, but instead simply overwrite the MIB object value whenever the information it corresponds to has changed, essentially treating the MIB objects as variables. The SNMP agents provide requested information about its managed device upon receiving an SNMP *GET* request message, with the request specifying the MIB object to be read. The agents can also send a set of MIB object values within a single SNMP *TRAP* message to a configured recipient. The possible *TRAP* messages that can be sent by the agent are defined in the MIB, specifying the set of other MIB objects that compromise it. The *TRAP* messages are unsolicited. They are sent whenever the value of one of its MIB objects changes. Devices that send SNMP requests, and optionally process SNMP *TRAP* messages, are referred to as SNMP managers within the context of this thesis.

Multiple versions of SNMP exist, namely SNMP, SNMPv2, and SNMPv3 [51]. Among other changes, SNMPv2 enhances SNMP by allowing network nodes to assume the role of SNMP agent and SNMP manager at the same time. This allows the network to establish a hierarchy of SNMP nodes that can act as agents to nodes above them in the hierarchy and can take the role of SNMP manager when requesting information from a subordinate agent. The SNMPv2 protocol also adds support for *GETBULK* requests that retrieve multiple MIB object values. SNMPv3 improves on SNMP and SNMPv2 by adding encryption and message integrity checking mechanisms for all messages sent. The lack of such features in UDP makes it all the more essential that cybersecurity measures for SNMP are present in its application layer, as is the case for SNMPv3.

2.5. Threat Hunting

Threat hunting refers to the process of searching for evidence of attack behaviour within system telemetry and generating attack hypotheses that describe potential attack campaigns and novel threats [25], [52]. Threat hunting leverage IoCs, CTI, and system knowledge to uncover and correlate alerts and TTPs from which to construct attack hypotheses. Alert correlation methods typically fall under similarity-based, statistics-based, or knowledge-based correlation approaches [53]. Hypotheses formed from alert correlation can then be validated and attributed to an APT group [22] when relevant. However, the correlation of events and TTPs into a stealthy APT attack campaign is challenging due to such campaigns remaining passive for long periods between steps, thus requiring the correlation to have long term memory. Furthermore, APT groups regularly replace the malware hashes and IP addresses that they use in order to evade IDSs that attempt to match system artifacts with known IoCs [22], [23].

Methods to analyze attack hypotheses generated from threat hunting can be described as falling into four categories: (i) machine learning, (ii) graphs, (iii) rules, and (iv) statistics [25]. Machine learning methods are trained on large amounts of sample data to make conclusions about a given hypothesis without extensive expert knowledge. Graph-based methods attempt to match sequences of events or process flows with sequences known to be benign or malicious. Rules-based methods leverage expert knowledge to understand the exploits used by attackers and codify them in ways

that allow observations to be readily compared against them. Statistics-based methods refer to large amounts of relevant threat data to make decisions on incoming events. In contrast to machine learning approaches, statistical-based approaches focus on understanding and presenting relationships between variables, as well as modeling the uncertainty surrounding the data [54], [55].

2.6. MITRE ATT&CK Framework

The MITRE ATT&CK Matrix is a framework established by the MITRE Corporation as a public resource for compiling the TTPs used by threat actors against enterprise, mobile, and ICS networks [26]. The information it presents is sourced from trusted industry collaborators. The MITRE ATT&CK framework is a valuable source of CTI that is utilized by security experts and research groups. Separate ATT&CK matrices are defined for enterprise networks, ICS networks, and Android/iOS mobile networks. The CTI contained in the framework is available in a structured format, namely JSON.

The MITRE ATT&CK Matrix enumerates attack techniques and classifies the techniques as part of an attack tactic e.g., $\langle$ Initial Access $\rangle$, $\langle$ Persistence $\rangle$. It should be noted that some techniques are common to multiple tactics. Also, some Enterprise ATT&CK Matrix techniques contain subtechniques that describe more specific examples of the parent technique. However, the techniques included in the ICS ATT&CK Matrix do not have subtechniques assigned to them. As of now, there are 12 unique tactics and 83 unique techniques in the ICS ATT&CK Matrix. The Enterprise ATT&CK Matrix, meanwhile, has 14 tactics, 211 techniques, and 468 subtechniques [26]. To note is that there is a large overlap of attack techniques and tactics between the Enterprise and ICS matrices. This thesis focuses on the ICS MITRE ATT&CK Matrix, as it provides information that is more pertinent to the microgrid system network.

The extent of the CTI made available by the MITRE ATT&CK framework extends beyond simply listing attack tactics and techniques. The framework also lists the malware and APT groups that are known to use each technique, along with short descriptions on how they use the techniques. In addition, it lists the data source elements that can be monitored to detect instances of the techniques. More specifically, it associates data sources with multiple data components, and data components

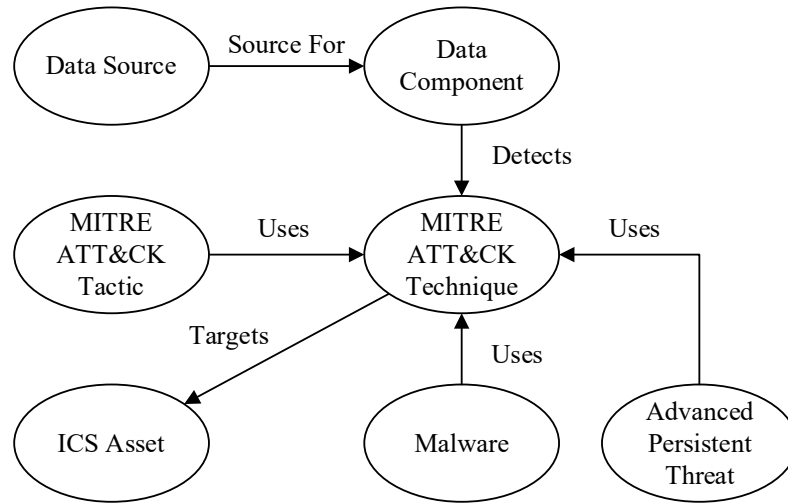


Figure 2.1: Relationships within the ICS MITRE ATT&CK CTI.

are associated with techniques. The framework provides short descriptions explaining how the data components can be utilized to detect their associated techniques. Figure 2.1 summarizes the types of associations between entities defined in the MITRE ATT&CK framework for ICS. With the exception of the one-to-many relationship between data source and data component, the relationships between the entities in the framework are many-to-many. In the context of the ICS ATT&CK Matrix specifically, the techniques are associated with ICS assets that they can potentially target. Asset entities do not appear in the Enterprise ATT&CK Matrix and Mobile ATT&CK Matrix due to the types of devices found in those environments being much more homogeneous.

To give examples of some relationships found among the ICS MITRE ATT&CK entities, the `<File>` data source has `<File Creation>`, `<File Deletion>`, `<File Access>`, `<File Modification>`, and `<File Metadata>` as data components. The `<File Creation>` data component is a monitored event that can potentially detect the use of the `<Drive-by Compromise>`, `<Lateral Tool Transfer>`, `<Replication Through Removable Media>`, or `<Spearphishing Attachment>` techniques. For reference, the ICS MITRE ATT&CK Matrix of tactics and techniques is given in Figure 2.2.

The MITRE ATT&CK framework provides other useful information as well. It lists preventative mitigation options for each technique. It also describes a number of notable attack campaigns according to the ATT&CK techniques and malware they used. The 2022 Ukraine electric power attack launched by Sandworm Team [11] is among the campaigns included in MITRE ATT&CK.

Initial Access	Execution	Persistence	Privilege Escalation	Evasion	Discovery
Drive-by Compromise	Change Operating Mode	Hardcoded Credentials	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration
Exploit Public-Facing Application	Command-Line Interface	Modify Program	Hooking	Exploitation for Evasion	Network Sniffing
Exploitation of Remote Services	Execution through API	Module Firmware		Indicator Removal On Host	Remote System Discovery
External Remote Services	Graphical User Interface	Project File Infection		Masquerading	Remote System Information Discovery
Internet Accessible Device	Hooking	System Firmware		Rootkit	Wireless Sniffing
Remote Services	Modify Controller Tasking	Valid Accounts		Spoof Reporting Message	
Replication Through Removable Media	Native API				
Rogue Master	Scripting				
Spearphishing Attachment	User Execution				
Supply Chain Compromise					
Transient Cyber Asset					
Wireless Compromise					

Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Default Credentials	Adversary-in-the-Middle	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploitation of Remote Services	Automated Collection	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Hardcoded Credentials	Data from Information Repositories	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
Lateral Tool Transfer	Data From Local System		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Program Download	Detect Operating Mode		Block Serial COM	Unauthorized Command Message	Loss of Control
Remote Services	I/O Image		Change Credential		Loss of Productivity and Revenue
Valid Accounts	Monitor Process State		Data Destruction		Loss of Protection
	Point & Tag Identification		Denial of Service		Loss of Safety
	Program Upload		Device Restart / Shutdown		Loss of View
	Screen Capture		Manipulate I/O Image		Manipulation of Control
	Wireless Sniffing		Modify Alarm Settings		Manipulation of View
			Rootkit		Theft of Operational Information
			Service Stop		
			System Firmware		

Figure 2.2: MITRE ATT&CK Matrix for ICS.

Chapter Three

Literature Review

The contributions of this thesis are positioned against existing research works discussed in this chapter. Cybersecurity research works whose topics relate to the contents of this thesis, in consideration of their methodology and domain, are presented and placed into a taxonomy. This chapter presents a separate taxonomy for articles related to (i) smart grid FDI attacks, (ii) threat hunting, and (iii) attack prediction. These works are reviewed and examined to identify their research gaps that are addressed by the contributions presented in this thesis. Following this, a brief discussion is given on how the microgrid co-simulation testbed used for this thesis compares to other simulation testbeds developed within the smart grid domain.

3.1. Detection of Smart Grid FDI Attacks

The microgrid system contains various IT and OT assets that communicate over Ethernet and IP networks. These communication channels transmit critical measurement and command data whose integrity and authenticity must be preserved to maintain safe microgrid operation. The anomaly detection presented in this thesis focuses on detecting attacks that involve the primary or secondary control levels. A taxonomic breakdown of existing research on detection of attacks in these control layers, as classified by their detection strategy, is given in Figure [3.1](#).

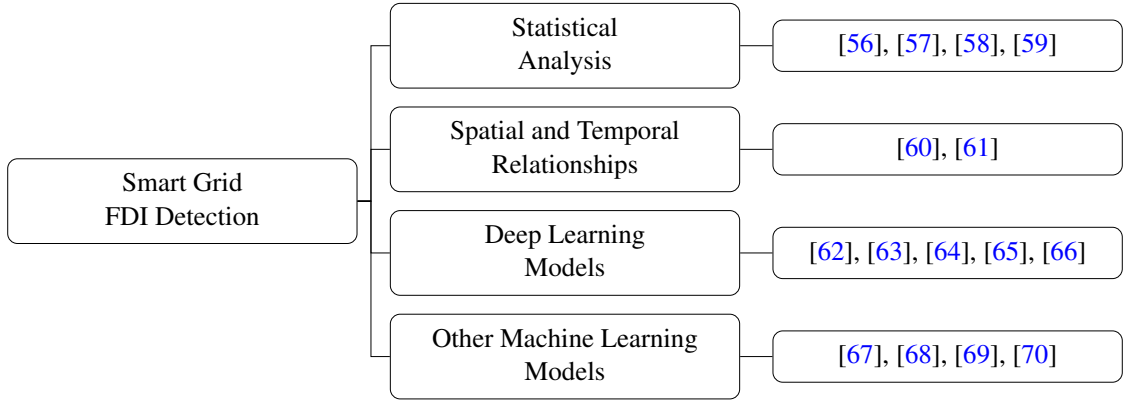


Figure 3.1: Related work on smart grid FDI.

3.1.1. Statistical Analysis

Statistical approaches detect smart grid FDI attacks by comparing sensor measurements to the distribution of sensor values across the grid. Bad data detection schemes are in effect, which are originally designed to identify sensor faults and noisy measurements. However, attackers can craft measurement data that bypasses bad-data detection when they perform FDI attacks [57], [58]. The statistics-based FDI detection approaches discussed here fill the cyberattack detection gap in the bad data detection method.

Rigatos et al. [56] propose the use of a Kalman Filter and statistical decision-making criteria to identify power grid sensors whose measurements have been compromised by an attacker. The Kalman Filter output is modeled after the fault-free sensor outputs and is used to detect anomalies in the outputs of the real sensors that result from a sensor being attacked. The residual error between the Kalman Filter output and the real sensor output follows the χ^2 distribution, which is used to set the anomaly thresholds. Statistical tests are administered to multiple sensors. Any sensor whose output deviation exceeds the χ^2 threshold is labeled as anomalous. This detection scheme is capable of distinguishing between attacks and faulty sensors. It can also account for attacks against multiple sensors by applying the statistical test on combinations of two or more sensors.

Manandhar et al. [57] also use the Kalman Filter and χ^2 detector to identify attacks on smart grid systems, including FDI and Denial-of-Service (DoS) attacks. The authors consider the case where attackers perform FDI attacks whose measurement modifications are crafted to evade detection

from the Kalman filter and χ^2 detector. They address this possibility by proposing the addition of a Euclidean distance detector that calculates the difference between the measured data and the estimated sinusoidal data. Distances that exceed a set threshold are treated as attacks.

Ruben et al. [58] propose a hybrid detection approach that uses a physics-based non-linear model combined with multi-agent data-driven detectors to identify smart grid FDI attacks. The physics-based components perform state estimation according to the weighted least squares method to calculate the error in monitored sensor measurements. The measurement error is compared against the χ^2 distribution for bad data detection. Meanwhile, the data-driven components use a set of ensemble correlation-based detectors that each learn the background statistics of a local bus in the power system under normal conditions. Decision scores are computed for the state estimation and the ensemble correlation-based detectors. An FDI attack is detected if the sum of these scores crosses a set fusion score threshold, with the threshold optimized for maximizing F1-score.

Lei and Mohammadi [59] propose a security management architecture for AC-DC microgrids based on communication quality characteristics measured within the Advanced Metering Infrastructure (AMI). Specifically, their architecture analyses stable Received Signal Strength (RSS) collected from AMI meters, fitting them to a normal distribution. It performs sequential hypothesis testing to determine if incoming RSS measurements are a sufficiently good fit for the hypothesis of an FDI attack compared to the null hypothesis. Their approach is capable of identifying Sybil and masquerade attacks. However, it only uses one node feature, namely RSS, in its prediction model. RSS is notable for being related to communication quality rather than transmitted data. The anomaly detection scheme proposed in this thesis uses multiple features in its prediction related to both the communication quality and the transmitted data, allowing it to detect a wider range of anomalies.

Like the statistical approaches, the proposed anomaly detection scheme is trained on samples that only contain measurements taken during normal, fault-free grid operation, however the proposed deep-learning models can incorporate multiple disparate features from potentially different data sources into one model. In this way, the proposed scheme addresses the shortcomings of these works in regard to the minimal scope of their data sources. The wider range of data sources allows the proposed scheme to find anomalies by learning the correlations between data pertaining to different microgrid functions and data types.

3.1.2. Spatial and Temporal Relationships

Some research works propose microgrid attack detection methods that leverage the spatial and temporal relationships within microgrid measurement data. These approaches are an alternative to those relying on statistical data. The microgrid contains multiple DERs spread out across the microgrid power network. Each of these DERs has local sensors and control loops deployed on its line. A central controller can monitor the various microgrid DERs and loads together and thus can verify that constraints related to the spatial and temporal relationships between sets of microgrid elements are satisfied.

Beg et al. [60] propose the use of signal temporal logic to detect DoS and FDI attacks in DC microgrids. The signal temporal logic represents acceptable microgrid states as logical conditions. These conditions take the form of constraints on the values of the microgrid signals maintained over a specified period. The conditions can involve individual signals, a trace containing multiple signals, or a hyperproperty that applies across multiple traces. An attack impact measure is calculated to determine the extent that a signal violates the signal temporal logic, with negative impact measures being considered evidence of a cyberattack. The authors also evaluate the use of low-pass filtering and saturation limiting to mitigate FDI attacks. While the signal temporal logic conditions are explicit, additional conditions are needed if other signals, such as active power or phase angle, are introduced in an effort to improve detection results. The anomaly detection scheme proposed in this thesis has the advantage of being able to learn temporal relationships through its DNN models while being able to expand its set of input features with minimal complexity. However, the models would still need to be retrained if the input feature set changes.

Zhang et al. [61] consider the case of FDI attacks against DC microgrid local voltage measurements that coincide with the spoofing of measurements collected by operators to hide the FDI. To detect such concurrent attacks, the authors propose a method to decompose signals into intrinsic mode functions using model-free ensemble empirical mode decomposition. Through this decomposition, their method can compare the energy flow between nodes to determine if any node is producing abnormal amounts of energy. The authors also propose a mitigation approach that replaces signals from a compromised node with the signals of its trusted neighbours. In this thesis,

finer details about the microgrid power quality, such as phase angle, are extracted as input features for detection. This allows the anomaly detection scheme proposed in this thesis to identify FDI that do not necessarily involve discrepancies in energy flow.

3.1.3. Deep Learning Models

Moving on to works that also employ deep learning models, the capabilities of proposed microgrid anomaly detection and its DNN models are compared against other works that leverage deep learning. It is important to make these comparisons in order to highlight the novelty of the anomaly-based detection method proposed in this thesis.

Zhang et al. [62] propose an unsupervised deep stack autoencoder to uncover cyberattacks targeting market and measurement data within Transactive Energy Systems (TES). They introduce a rule-based mechanism for distinguishing between different types of anomalies. Notably, it can distinguish between anomalies caused by cyberattacks and ones caused by non-malicious activity. Among the cyberattacks it can identify are parameter manipulation attacks and DoS attacks. The authors attribute this success to the higher predictability of TES application data compared to IT network data. To an extent, the same is also true for the microgrid and its power control communications, though Zhang et al. do not consider the microgrid in their work. When comparing to their work, the proposed anomaly detection scheme does not directly distinguish between attack types. It does, however, single out the anomalous features and the network nodes that experienced them. This leaves room to apply a rule-based method to identify attack types from the set of anomalous features and the extent of the deviation from the model prediction.

Siniosoglou et al. [63] present a smart grid IDS that uses an Autoencoder Generative Adversarial Network (Autoencoder-GAN) to detect anomalies in Modbus/TCP and DNP3 traffic. The Autoencoder-GAN is trained on normal message samples that are generated according to a normal distribution. The samples contain features derived from time series electricity measurements as well as network flow data. The authors propose two Autoencoder-GAN models. One of them is for anomaly detection, and the other is for classification. A sample is deemed anomalous if the difference between the real sample and the output generated by the model exceeds a set threshold. The classification model uses a single DNN to produce three ground-truth label points that describe

the following: (i) the validity of the sample, (ii) the anomaly approximation, and (iii) the anomaly class of the sample. The anomaly detection model in particular follows a similar approach as the anomaly detection proposed in this thesis, though the IP traffic it considers uses Modbus and DNP3 traffic instead of IEC-104 NSM traffic. The success of the Autoencoder-GAN on network traffic statistics solidifies the autoencoder DNN as a potential candidate for processing NSM data.

Presekal et al. [64] use novel DNN models to detect attacks in the early stages of the cyber kill chain and generate attacks graphs in near real-time to assist in attack mitigation. The proposed DNN model contains a Graph Convolutional LSTM (GC-LSTM) and deep convolutional network that engages in anomaly detection on power system OT traffic. The OT traffic they consider includes IEC-104 and DNP3 telecontrol traffic exchanged between MUs, IEDs and RTUs, which contain measurement data and commands. The authors note that OT data is more deterministic and homogenous than traditional IT traffic, which supports the viability of OT data anomaly detection. The outputs node classifications of the anomaly detection can then be used to construct an attack graph online. The NSM data and power measurement data analyzed by the anomaly detection scheme proposed in this thesis is distinct from the telecontrol traffic considered by Presekal et al. in their work. However, the NSM data is an aggregate of IEC-104 telecontrol traffic, which makes their work closely aligned with the research presented in this thesis.

Kavousi-Fard et al. [65] use unsupervised learning to establish a Prediction Interval (PI) for microgrid consumer load consumption data for the purpose of detecting integrity attacks on the microgrid AMI. They train a neural network using the Lower and Upper Bound Estimation (LUBE) method to establish a PI for forecasting data. They label forecast data that exceeds the PI bounds as being anomalous. In terms of optimization, fuzzy logic and multi-model aggregation is used to minimize the number of legitimate datapoints that fall outside the PI. They also use the Symbiotic Organisms Search (SOS) evolutionary algorithm to optimize the LUBE parameters. A notable difference between their work and the anomaly detection scheme proposed in this thesis is that their work is centered on AMI rather than telecontrol. Their work also does not incorporate time series elements as input.

Inoue et al. [66] explore the use of unsupervised learning to detect anomalies in Secure Water

Treatment (SWaT) ICS data. Their dataset contains seven days of continuous normal SWaT operation and four days of operations impacted by network attack scenarios. This dataset is used to train a large DNN, based on Long Short-Term Memory (LSTM), with 100 hidden layers to perform outlier detection. The LSTM is compared against a one-class Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel. The authors found that the DNN had better precision than the SVM, while the SVM had better recall. Although this work is not directly tied to microgrids, it does apply DNN models to detect anomalies in operational data of an ICS, demonstrating the potential for LSTM models in detecting anomalies within smart grid domains like the microgrid.

This thesis uses machine learning models to detect anomalies and inconsistencies within microgrid operation data. It focuses on detecting attacks through IEC 62351-7 NSM data and the curves for instantaneous voltage and current. Rather than simply classify incoming feature set inputs, it also identifies the anomalous feature, giving better granularity when identifying the nature of the anomaly and the affected devices.

3.1.4. Other Machine Learning Models

Ahmed et al. [67] propose the use of the Isolation Forest classifier to detect covert data integrity attacks on the smart grid power system. The proposed classifier uses state estimation data as its input, with Principal Component Analysis (PCA) being used to reduce the dimensionality of the data. Similar to the anomaly detection proposed in this thesis, Ahmed et al. use an unsupervised learning approach and train their machine learning models on unlabeled measurement samples. They note that acquiring the amount of labeled measurement data required for supervised learning tends to be difficult.

Parizad and Hatziaioniu [68] propose a machine learning framework that uses a combination of supervised and unsupervised learning methods to detect FDI attacks in the power system. The framework has five key stages: (i) pre-processing, (ii) statistics-based feature selection, (iii) supervised learning feature selection, (iv) unsupervised PCA for final feature selection, and (v) DBSCAN classification and clustering of FDI attacks. The proposed framework is trained and evaluated on data from the IEEE 14-bus system. The supervised learning is done by observing the changes to

active and reactive power that result from certain FDI attacks on the phase angle and voltage magnitudes of the power line. The use of collaborative PCA-density methods for unsupervised learning was found to outperform other classifiers such as SVM, Empirical Covariance, and Random Isolation Forest. Though the power dynamics of the IEEE 14-bus system differ from those found in the microgrid, the potential for PCA to enhance the performance of the microgrid anomaly detection being proposed in this thesis should nevertheless be considered.

Hodo et al. [69] study the effectiveness of various machine learning classifiers for detecting FDI and DoS attacks on IEC-104 packets that can occur as a result of a Man-in-the-Middle (MitM) attack. The classifiers tested in their work are among those found in the WEKA workbench [71] and include Naïve Bayes, Random Forest, Decision Table, J48 Decision Tree, and OneR. For training and testing data, they generated a labeled dataset of normal and attacked IEC-104 packets. A total of nine features are extracted from these packets: (i) *Length*, (ii) *Total_Len*, (iii) *Frame_Length_on_wire*, (iv) *DeltaTime_prev_capt_frame* (time difference from previous capture), (v) *RTT* (Round Trip Time), (vi) *Value*, (vii) *Month*, (viii) *Packets_s* (packets per second), and (ix) *Window_size_value* (from TCP header). They find that the J48 and Decision Table classifiers have the highest accuracy. The relevance of the IEC-104 protocol for microgrid telecontrol frames the work of Hodo et al. as an important reference point in the context of microgrid security monitoring that is the topic of this thesis. In particular, the thesis also compares the proposed prediction models to existing ones found in the WEKA workbench.

Ma et al. [70] also explore the use of supervised learning for detecting MitM attacks that target power system control packets. Specifically, they evaluate machine learning classifiers on data integrity attacks against voltage reference setpoint commands transmitted over unencrypted TCP. The tested classifiers include Naïve Bayes, Decision Tree, K-Nearest Neighbor (KNN), SVM, Multi-Layer Perceptron (MLP), and Random Forest. To evaluate these classifiers, Ma et al. generate a labeled dataset of normal and attacked control packets using an experimental setup containing hardware devices acting as controllers. For learning features, they use packet latency, neighboring voltage measurements, and neighboring voltage setpoints. The results of their evaluation show the Random Forest classifier as having the highest detection accuracy when applied on their dataset.

These works present a wide range of machine learning models that can potentially be used for

security monitoring of microgrid telemetry. The effectiveness of the DNN architectures used by the anomaly detection scheme proposed in this thesis is compared against many of these competing machine learning models. When monitoring NSM data and power quality measurements, the use of DNN models is found to either outperform the competing models or be much more practical in the case of where attack sample data is lacking.

3.2. Threat Hunting

Interest in threat hunting solutions has increased in recent years as the threat posed by cyber-attacks has become more pronounced. However, there is still much to explore when it comes to adapting these solutions to ICS environments such as the microgrid. The microgrid control centre contains elements in common with the traditional IT environments found in enterprise systems. It is therefore possible that existing threat hunting proposals are at least partially viable in the microgrid, even if they are not necessarily optimal. As such, it is important to consider recent advances in threat hunting within enterprise networks in addition to ICS networks. Figure 3.2 shows the taxonomic breakdown of the related work on threat hunting covered here. The topics making up this taxonomy include IoCs, provenance graphs, and models that follow the MITRE ATT&CK framework.

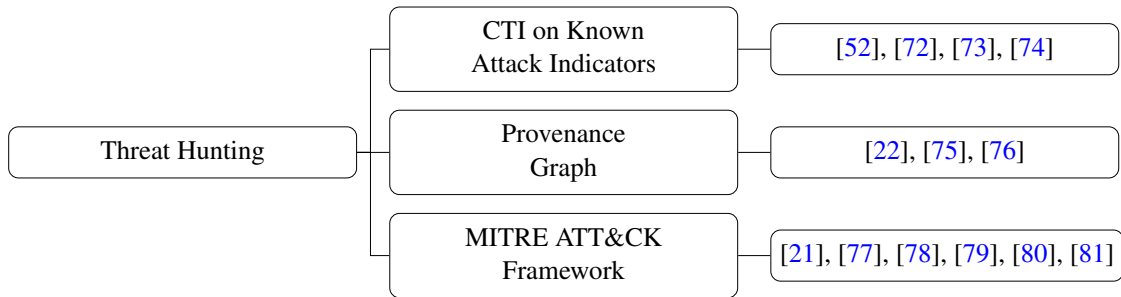


Figure 3.2: Related work on cyber threat hunting.

3.2.1. Leveraging of CTI on Known Attack Indicators

To start with, threat hunting research exploring the detection of IoCs is discussed. An IoC is a tangible system artifact that can be treated as an indicator of a cyberattack [52], [74]. Examples of IoCs include malicious process names, virus signatures, botnet IP addresses, and registry

objects. The Pyramid of Pain describes IoCs as being relatively easy to detect, but which change frequently [21], [23]. As such, attack detection tools that aim to detect IoCs must be updated to match the current state of CTI on a regular basis in order to keep up with the latest IoC information. It should be noted that the existing body of available CTI extends far beyond IoCs. For example, CTI may contain attack behaviours expressed as attack TTPs and how they relate to attacker goals [78].

Gao et al. [52] propose a threat hunting framework, called ThreatRaptor, that makes use of a novel query language constructed from open-source CTI to analyze system audit logs. The ThreatRaptor query language is built through the application of Natural Language Processing (NLP) on unstructured CTI reports. Threat behaviour is first modeled as a graph. Then, a query subsystem, along with a corresponding query language, is synthesized from this graph. Gao et al. evaluate ThreatRaptor on a dataset of attack samples based on the Cyber Kill Chain [82] and the Common Vulnerabilities and Exposures (CVE) database [83]. In contrast to ThreatRaptor, the threat hunting approach proposed in this thesis makes use of structured CTI to construct a more generalized threat profile. It also takes the microgrid model into consideration when generating a sample dataset of malicious activity following an attack kill chain.

Xu et al. [72] propose an adaptive real-time safeguard framework for uncovering malware within Internet of Things (IoT) devices. Their framework, called MIDAS, constructs a behavior paradigm based on behaviour extracted from IoT malware on Linux systems. MIDAS engages in real-time auditing that monitors for break-in program behaviour, traces their homology with the malicious behaviour of malware submodules, summarizes their operations into abstract behaviour pairs, and assembles the pairs into malware stages. The malware stages they consider are (i) infection, (ii) reconnaissance, (iii) persistence, and (iv) impacts. However, their work does not consider the behaviour of threat groups, whose capabilities extend beyond a single malware. The threat hunting approach proposed in this thesis models the behaviour of APTs into attack stages, which gives it an overall bigger scope than what is reached from malware analysis.

Nadeem et al. [73] propose a sequential learning framework that derives attack graphs directly from intrusion alerts rather than from expert opinion or specific system topology knowledge. Their framework, called SAGE, uses a Suffix-based Probabilistic Deterministic Finite Automaton to identify attack paths for infrequent and severe alerts. It aggregates alerts into sequences based on

sources, targets, and time window. It also distinguishes between sequences based on temporal and probabilistic differences. For example, alerts at the start of a sequence may be associated with reconnaissance while alerts in the middle of a sequence are associated with lateral movement. The sequences are combined into attack episodes, each of which corresponds to one attacker action according to alert frequency patterns. Regarding datasets, they use the Collegiate Penetration Testing Competition dataset from 2017 and 2018 (CPTC-2018, CPTC-2018), as well as the 2018 Collegiate Cyber Defense Competition dataset (CCDC-2018). Through their experiments, they determine that attackers usually follow the shortest path through a given network to exploit a target asset. The threat hunting approach proposed in this thesis also makes use of time windows and alert sequencing when correlating and analyzing alerts. However, it leverages available CTI to further the threat attribution of its attack hypothesis. The use of CTI to support threat hunting activities is absent in the work of Nadeem et al.

Asiri et al. [74] explore the detection of IoCs within the ICS domain. They highlight the importance of maintaining the availability of ICS devices and ensuring their continuous operation. In particular, attacks that impact the operation of ICS devices put people at risk of physical harm. The authors mention how defining IoCs within ICS operations faces several challenges. Examples of such challenges include the use of proprietary software, lack of data monitoring features within legacy devices, and the use of default passwords within field devices. Attempting to address these issues, Asiri et al. provide classes of IoCs to consider in ICS contexts, such as unexpected usage of controller resources and mismatches between controller behaviour and historian logs.

In regard to threat hunting, the research presented in this thesis explores how the nature and priorities of attacks targeting the microgrid may be distinct from attack behaviour observed in enterprise IT environments. There are a number of collaborating assets within the microgrid that coordinate to perform particular microgrid functions related to power system stability, making use of particular ICS protocols. Also, certain devices within the network are more prone to certain types of exploits over others. This thesis takes the microgrid network topology, from the control centre to the field devices, into account when modeling microgrid attack paths. Though this thesis focuses on the microgrid, the use of ICS elements for threat hunting contributes to making the proposed threat hunting approach adaptable to other ICS contexts as well.

3.2.2. Application of Provenance Graphs

Distinguishing themselves from threat hunting research on IoCs is the research centered on the use of provenance for threat hunting. The provenance graph of a system describes the series of actions related to data processing and the flow of data between processes [84]. Existing research has explored the construction of provenance graphs within the IT domain and their application for attack detection. Within the context of this thesis, detection approaches based on provenance graphs are treated as an alternative to threat hunting strategies that rely on correlations between security events and alerts.

Milajerdi et al. [22] propose a threat hunting approach, called Poirot, that reads kernel logs to construct provenance graphs, which are subject to analysis capable of uncovering attack campaigns. The constructed provenance graphs describe the causal links and information flow found within the system, as deduced from kernel logs. Their approach involves building query graphs from IoC relationships extracted from CTI reports and then using inexact graph matching to map kernel log provenance graphs onto the query graphs. A notable feature of their approach is its ability to thwart attackers attempting to obfuscate their actions with activity that mimics other APTs. Their approach does so by prioritizing flows based on attack cost, and thus identifying obfuscation efforts that require low effort to execute on the part of the attacker. Attack cost is measured according to the number of root processes that must be compromised in order to perform the attack. The authors assume that attackers are less likely to compromise a larger number of unique processes in pursuit of their objective.

Wang et al. [75] propose a novel anomaly-based threat detection system that trains a graph neural network to detect stealthy host-level cyber threats in real-time. Their detection system, called ThreaTrace, trains its neural network on the system provenance graph behaviour of benign nodes. In order to improve scalability, the provenance graph is divided into subgraphs of similar nodes that are used as separate training sets for corresponding models, thus forming a multi-model framework. The subgraphs are formed through an iterative application of a GraphSage-based training method that classifies nodes together if there is sufficient confidence that they are related, then begins forming new classifications from the remaining nodes until all nodes are classified. ThreaTrace detects

malicious activity by identifying the number of nodes that are misclassified by the trained multi-model framework. If the number of misclassified nodes exceeds a certain threshold, an alert is raised and the corresponding activity is treated as malicious. Though effective at detecting suspicious activity, the lack of attack samples in its training data limits ThreaTrace from identifying the nature of the underlying attack.

Wang et al. [76] present ProvDetector, which is a proposed provenance-based approach for detecting stealthy malware. It is also trained on benign provenance data with the intention of identifying abnormal causal paths. The training data they use is generated by building a Word2vec neural embedding of provenance graph paths represented as word token sentences. The Word2vec embedding allows the representation of provenance graph paths as n-dimensional vectors that are suitable for use as input for machine learning models. Wang et al. use the Local Outlier Factor (LOF) density-based method to detect anomalies within a given provenance graph. A graph is labeled as malicious if the number of anomalous paths in the graph exceeds a configured threshold.

The existing research works on provenance graphs discussed above suggest that there is potential to adopt provenance graph inspection to detect novel forms of system misuse. However, they have yet to adapt their approaches to the ICS environment and the particular ICS assets found within. Moreover, there are scalability concerns with maintaining detailed records of system provenance that stretch far enough to observe patient APT who may remain dormant for long periods of time before advancing to the next stage of their kill chain. However, provenance graph analysis can be useful as part of a larger monitoring solution when it comes to detecting novel forms of asset misuse across hosts and network channels that warrant raising an alert.

3.2.3. Leveraging the MITRE ATT&CK Framework

The IoCs and provenance explored in the works above constitute low-level descriptors of attack behaviour. The MITRE ATT&CK framework, meanwhile, describes more generalized threat actor behaviour by defining the TTPs they use. The TTPs they define can be leveraged for the purpose of threat hunting. The work of Liu et al. [77] describes how MITRE ATT&CK can serve as the foundation for a cybersecurity ontology. Such an ontology can act as the basis of a cybersecurity Knowledge Graph (KG), such as the CyGraph threat hunting tool proposed by MITRE themselves

for system protection and attack visualization. However, there are alternative CTI sources that can be used to guide the creation of a cybersecurity ontology. Examples given by Liu et al. include Computer Emergency Response Teams (CERTs), Common Weakness Enumeration (CWE), the National Vulnerability Database, and Common Attack Pattern Enumeration and Classification (CAPEC).

The work of Li et al. [78] highlights the value of the MITRE ATT&CK framework as a source of CTI. They propose a CTI aggregation approach, called AttacKG, that extracts attack elements from natural language CTI reports and structures them into a technique KG, which describes attacks as a complete chain of attack techniques. The technique associations in their KG are derived from the TTPs described in the MITRE ATT&CK framework. In this way, they bridge the gap between unstructured CTI reports written in natural language and structured CTI information that can be readily used for intrusion detection and threat hunting.

Kaiser et al. [21] demonstrate the usefulness of KGs for the purpose of correlating attack activity. As part of their research, they construct a threat knowledge database, called AttackDB, that contains information derived from several CTI sources, with the MITRE ATT&CK framework being among them. They use this database to generate attack hypotheses from ATT&CK techniques detected within enterprise networks. To detect techniques, they use a multinomial Naïve Bayes classifier to determine the probability that a given system observable, such as anomalous telemetry or an IoC, corresponds to a particular technique. They refine the attack hypotheses according to the co-occurrence of techniques found in records of previous attacks. The threat hunting approach proposed in this thesis also filters out weaker hypotheses over time to improve performance, but adds the dimension of the microgrid system model topology and the MITRE ATT&CK ICS assets to better align its alert correlations with attacks targeting the microgrid environment.

Nour et al. [79] propose a threat hunting approach, named AUTOMA, that constructs attack hypotheses from the sequences of MITRE ATT&CK techniques associated with system telemetry. It then filters out hypotheses by relevancy to determine the most impactful hypotheses. The determination of relevancy is guided by a knowledge base of known attacks that are expressed as sequences of MITRE ATT&CK TTPs. Factors that influence the relevance of a hypothesis include similarity to known attack sequences, success score based on frequency and importance of the hypothesis techniques in the knowledge base, the likelihood of the hypothesis being valid, and the criticality

of the hypothesis calculated by multiplying the success score with the likelihood. In addition, hypotheses with low similarity to existing attack sequences are compared with possible attack variants to determine their relevancy. The proposed AUTOMA approach assumes that there are monitoring tools present in the system that can readily associate system events and alerts to MITRE ATT&CK techniques. It also does not attempt to predict future attack steps. By contrast, the threat hunting approach proposed in this thesis presents a generalized method for associating security alerts with ATT&CK techniques, and analyzes hypotheses to make predictions on what the APT will do next in the microgrid. In addition, the threat hunting proposed in this thesis arranges the TTPs in a hypothesis according to their placement in the cyber kill chain rather than filtering them based on similarity to known attack sequences.

On the topic of associating system telemetry to TTPs, Abdeen et al. [80] propose a publicly available tool, called SMET, that automatically maps CVE entries to ATT&CK techniques based on textual similarity. Attack labels are extracted from the textual CVE entries descriptions with an Allenlp semantic role labeling model. These labels are then embedded into vector space with a transformer DNN that is fine-tuned using MITRE ATT&CK data passed through a SIAMESE network. Semantically similar labels have similar vectors in the embedding space with respect to cosine similarity. The SIAMESE network is optimized such that two attack labels associated with the same ATT&CK technique have similar embeddings, while two labels from different techniques produce dissimilar embeddings. Logistic regression is then used to find the probability of a given vector corresponding to each ATT&CK technique. Logistic regression is chosen in order to reduce overfitting. The SMET mapping is found to be more accurate than Sentence BERT, Term Frequency - Inverse Document Frequency (TF-IDF), and Large Language Model (LLM) chatbots. The suitability of SMET for online alert-to-technique mapping is dependent on having accurate textual descriptions given to the alert, which may not be available if the event is generated from anomaly detection. Thus, it is not a full solution for the task of associating alerts to TTPs performed by the threat hunting approach proposed in this thesis. It can, however, be leveraged in cases where detailed textual descriptions of alerts are available.

The MITRE ATT&CK framework can also be used as a basis for attack modeling. This is seen in the work of Choi et al. [81], who propose a method for automatically generating sequences of

ATT&CK TTPs while satisfying a target set of requirements. Their proposed method uses a Hidden Markov Model (HMM) to derive ATT&CK sequences. The model treats ATT&CK tactics as states and ATT&CK techniques as observations. Sequences transition from one tactic to another according to state transmission probability and emit a technique as an observable according to emission probabilities corresponding to each state. The probabilities for tactic-to-tactic transmissions and for tactic-to-technique emissions are assigned according to the frequency in which they are found within ICS reports. Their approach has the major benefit of generating many permutations of ATT&CK TTPs sequences without having to resort to a manual process. However, it should be noted that their work does not cover real-time detection or mitigation of attacks.

The threat hunting approach proposed in this thesis has elements that are similar to the works discussed above. Most notably, it attempts to associate observables with ATT&CK TTPs and it puts forth a methodology to automatically generate sequences of MITRE ATT&CK TTPs. However, it makes use of alert grouping prior to TTP matching in order to improve precision. It also considers the microgrid network topology when generating potential attack sequences while arranging the sequence of TTP according to a version of the cyber kill chain that is adapted for ICS domains. In addition, the proposed threat hunting approach associates sets of correlated security alerts, structured as attack hypotheses, to ICS APTs while imposing a validation process to help limit the number of false positives.

3.3. Attack Prediction

This section discusses works related to the topic of attack prediction, with an emphasis on works pertaining to power systems. These works are compared against the attack step prediction methodology proposed in this thesis, which analyzes validated attack hypothesis generated during threat hunting. Moreover, the APT attack sequence dataset generation method elaborated in this thesis is also assessed in relation to these works. Within the context of cybersecurity, the term “prediction” does not exclusively refer to the prediction of the next step in a given sequence. It can also refer to anticipating the cyberattacks that have yet to occur against a given target system, i.e., forecasting. Having predictions on future threats to a system allows system defenders to devise prevention

measures accordingly. In addition, one can perform a risk assessment on the power system to better understand where the system might be vulnerable. The related work on attack prediction is classified as (i) a response approach, where action is taken according to observed system activity, (ii) forecasting, where attack possibilities are considered offline based on characteristics of the target system itself rather than from a feed of system telemetry, and (iii) risk assessment, where the likelihood of an attack occurring is considered alongside the severity of its impact to help compare the severity of potential threats. The classification of these works is shown in Figure 3.3.

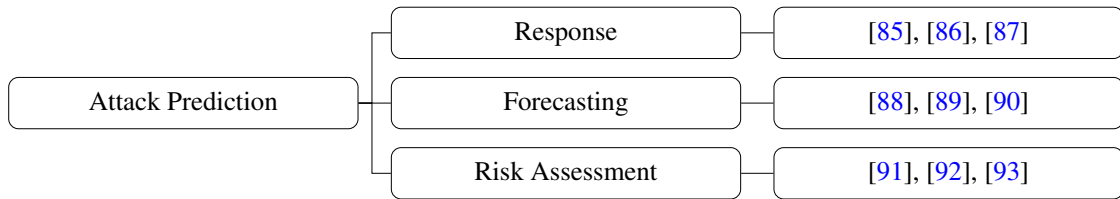


Figure 3.3: Work related to attack prediction.

3.3.1. Response

The responsive attack prediction works discussed here establish models of attack behaviour within a system that allow security operators to make predictions on incoming attack activities given the attack activity already observed. These predictions are typically not extrapolated from individual events or alerts reported by IDSs due to the very high percentage of false positives they tend to give [24], [53]. Rather, predictions are based on a series of security-related observables that have been correlated. As such, the success of responsive attack prediction is largely dependent on the effectiveness of deployed threat hunting solutions at providing valid attack hypotheses.

The use of the HMM for attack prediction is worth considering given its separation of the model state and its transitions from the observables that the model actually generates, with each transition having a certain probability of transmitting each of the observables. It is reflective of the position that security operators find themselves in, where the past actions of the attacker are not directly known and must instead be inferred from what is actually observed in the system. Zhang et al. [85] propose a dynamic HMM for predicting attack activity within the power system, with the HMM generating system alarms as observables. Their model is dynamic in that it adjusts the transition

probabilities of the HMM describing the attacker’s behaviour to account for changes in the strategies adopted by the attacker and defender as their knowledge of the situation increases over the course of the attack. Their model also accounts for transition probabilities between observables, which is neglected in the traditional HMM. The authors address the issue of an overabundance of alarm data by first using the Random Forest algorithm to find key features and then reducing the feature dimensions of the alarms accordingly. Afterwards, the alarms are grouped using an attribute-similarity algorithm and combined into events included in the HMM, with the number and complexity of events being much more manageable in terms of computational needs.

In a similar vein, Chen et al. [86] propose models that use the Markov Decision Process (MDP) to determine the policy followed by an APT aiming to control the electric substation as their attack progresses. They also use the MDP to determine the policy followed by system defenders in response to observables. The success rate of each attack step is modeled by a Poisson distribution over the number of vulnerabilities the APT tries to exploit with respect to the average number of vulnerability exploits needed for success. The authors note that the APT must engage in more reconnaissance whenever they want to find more vulnerabilities to exploit. As for the defender, the probability that an alarm is generated as a result of a successful attack step is determined using a Bayesian model conditioned on the number of logs generated by the vulnerability exploit attempts. The APT and defender compete against each other according to the MDP, whose reward function quantifies the amount of control the attacker has gained or lost over the substation after an action. The attacker optimizes their policy to maximize reward, while the defender tries to minimize it.

In contrast to HMM models, Villarreal-Vasquez et al. [87] propose an anomaly detection framework that uses LSTM models to learn event patterns in computer systems. This framework is called LADOHD, which stands for LSTM-based Anomaly Detector Over High-dimensional Data. It collects and analyzes high volumes of system events with the intent of detecting insider threats who perform multi-stage persistent attacks. They base their model on LSTM in order to overcome the order-aware recognition problem facing event sequencing approaches that are limited to a certain max length. Their dataset contains 38.9 million events collected from 30 machines over 28 days. Their data processing approach filters out events that are not of interest, then converts the remaining events into a finite vocabulary of event data and then arranges them in sequence. The LADOHD

framework generates a probability distribution for every possible event that can follow (at time t) from a given input event sequence (up to time $t - 1$). The top k events in the distribution are treated as normal. Regarding anomaly detection, the next observed event is treated as anomalous if it falls outside the top k events predicted when given the sequence of events preceding the newly observed one. The LADOHD framework is found to have a True Positive Rate (TPR) of 97.29% and False Positive Rate (FPR) of 0.38%. While its effectiveness is only demonstrated on IT data, the OT traffic, found in smart grid systems like the microgrid, is expected to be more deterministic overall [64], thus making such an approach especially valuable to consider.

The attack prediction methodology proposed in this thesis makes predictions on future attack steps of an ongoing attack campaign based on known information about the campaign. This information includes the assets already believed to be compromised, the TTPs believed to have been used, and the suspected APT group. The predictions are made by combining the Bayesian probability of the various attack possibilities according to an APT attack sequence training dataset and the risks associated with those possibilities. Though not dynamic in the same way as the HMM, the proposed prediction methodology makes more confident predictions as more information about the campaign becomes evident.

3.3.2. Forecasting

In contrast to responsive prediction, attack forecasting refers to making predictions on attacks that will occur through analysis of information that does not originate from data or traffic feeds. That is to say, the predictions are made from relatively static information about the target system rather than from telemetry. Such predictions are relevant for planning out system hardening efforts that set out to improve the attack prevention capabilities of the system or to introduce new mitigation capabilities within the system.

Khalil et al. [88] propose a threat modeling methodology, based on STRIDE, for cyberphysical systems, with the microgrid being presented as a case study. STRIDE stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation. It represents the system as a Data Flow Diagram (DFD) that presents the system state according to six security properties: i) confidentiality, ii) integrity, iii) availability, iv) authentication, v) authorization, and vi)

non-repudiation. The proposed threat modeling methodology consists of nine steps: 1) initial attack taxonomy creation, 2) information and system assets identification, 3) system mapping into DFD, 4) security context definition, 5) trust boundary determination, 6) threat elicitation and attack taxonomy update, 7) threat consequences and losses identification, 8) threat prioritization, and 9) security requirement selection. When considering the microgrid use case, the authors account for the physical control processes and their analog connections alongside digital components. The methodology also accounts for several different types of administrative users that influence the microgrid access control opportunities and challenges faced by attackers. The attack scenarios elaborated on the microgrid use case are divided into “threat precondition” and “threat final harm”. The final impacts on the microgrid are distinguished between immediate consequences and losses. These impacts are derived from CTI provided by the MITRE ATT&CK ICS framework. This work is however limited in its scope of covered threat actors. The attackers are neither assumed to be state-sponsored, nor are they considered to potentially be part of a terrorist groups. Attacks that originate from the supply chain are also not accounted for in this work.

Ling and Ekstedt [89] present a threat modeling language, called sasLang, for describing cyber threat attack graphs in IEC 61850 substations [94]. The proposed sasLang is extended from Meta Attack Language (MAL), which defines the assets involved and success likelihood of attack and defense steps. The sasLang modeling language encompasses the substation system component types, as well as the attack types that are considered effective against the substation. It covers the process, bay, and station levels of the substation. Its asset classes originate from studies on the substations of industry leaders such as Hitachi ABB Power Grids, Siemens Energy, General Electric, Cisco, and Schneider Electric. Among the attacks covered in sasLang are FDI to disconnect CBs and malicious manipulation of transformers. As for impacts, sasLang adopts the impacts described by the MITRE ATT&CK ICS framework.

Katina et al. [90] propose a method of forecasting cyber attacks in the ICS domain that decides on cybersecurity strategies based on r-Interdiction Median Problem with Fortification (RIMF). The RIMF model identifies and evaluates vulnerabilities in the system to optimize security implementation solutions according to cost. Accounting for various possible attack vectors in the smart grid, Katina et al. are able to forecast cyber attacks and natural disasters with a high degree of precision.

3.3.3. Risk Assessment

Responsive attack prediction acts on system observables, while forecasting prediction considers the state of the system and threat landscapes. Both of these prediction methods can form part of a risk assessment that attempts to quantify or qualify the danger being posed to the system at a given moment. Devising a method to perform risk assessment within the context of the smart grid is a topic that has continues to receive attention. However, accurately assessing the risks to the smart grid can prove challenging. Some of these challenges include the wide variety of equipment that can be found in a single system, the way that impacts can propagate along the power network, the diversity of the digital network, and the multidisciplinary knowledge required [91], [95]. Nevertheless, existing works have endeavored to develop a framework for smart grid risk assessment.

Langer et al. [91] propose a cybersecurity risk assessment method that integrates Information and Communication Technology (ICT) with legacy systems. They demonstrate their method on a smart grid reference architecture model developed through the Austrian Smart Grid Security Guidance research project. The risk assessment process is divided into conceptual-based steps and an implementation-based risk assessment steps. The conceptual steps identify the threats that are relevant to the reference architecture, assigns probabilities and impact to these threats to form a risk matrix, and elaborates mitigation strategies against the risks highlighted in the risk matrix. A semi-quantitative approach is used to assign appropriate probabilities and impacts for the risk matrix according to a five-level scale. The implementation steps find a real-world smart grid implementation that is close enough to the reference architecture to specify as the system under test, identify its potential vulnerabilities, perform security audits with respect to the potential vulnerabilities, and develop specific security measures for the implemented system.

Yan et al. [92] present a novel risk assessment model for cyberphysical power systems that considers the danger posed by software and network vulnerabilities within the SCADA, along with the impacts on the physical system that can result from malicious activity. They use an attack graph to model the maximum exploitation probability of the SCADA based on the software vulnerabilities found within. This model accounts for the security defense measures present in the system, as well as the characteristics of the attack propagation and knowledge acquisition. Through the attack graph,

the vulnerability network can be found. The authors measure the impact of the attacks according to the minimum load shedding resulting from a transmission line being disconnected by the attack. The risk is calculated as the product of the network vulnerability and the impact. Their risk assessment model is demonstrated on a simulated IEEE 14-bus and IEEE 18-bus system.

Mishra et al. [93] focus on quantifying the resilience of the microgrid. They derive resilience from risk assessments that combine mathematical approaches with expert knowledge. They define a metric, called the context aware resilience metric, that is quantified according to the ability of the microgrid to absorb, restore, and adapt to low-probability-high-impact events while sustaining critical loads. To measure risk, they compute a residual resilience score based on semi-quantitative values for threat level, the threat probability, the impact vulnerability, and the probability that the vulnerability is exploited. A Monte Carlo approach is used to sample ranges of residual resiliency scores to find the mean risk for various threat and vulnerability combinations. The mean risk for the combinations are then aggregated to determine the overall resilience score. The authors present a case study using a grid-tied microgrid serving a coastal town in New England to demonstrate the steps of their approach.

The attack prediction methodology proposed in this thesis takes inspiration from these works in its use of a semi-quantitative approach to describe the negative impacts to the microgrid that can be realized through the compromise of critical assets, with these impacts factoring into the risk assessment. However, the impacts are measured using a microgrid co-simulation testbed that is subjected to simulated attacks. Also, this thesis calculates the likelihood component of the risk assessment based on an ICS APT campaign dataset derived from CTI, as well as the expected exploitability of the attack path leading up to the critical assets. The proposed model aims to be parameterized in order for system operators to tune the risk assessment to their security priorities, system knowledge, and assumptions.

3.4. Power System Simulation

The microgrid co-simulation testbed used for this thesis research combines microgrid power simulation with IT network emulation. The microgrid power system is implemented using the

HYPERSIM power modeling software and simulated on an OPAL-RT Digital Run-Time Simulator (DRTS) capable of simulating power systems with a precision of $50\mu s$ [14]. The I/O ports of the DRTS are coupled to an OpenStack network emulation server [15], where the inputs and outputs of the simulation, contained in Ethernet frames or IP packets, can be provided and read, respectively. The HYPERSIM model is also capable of simulating line-to-ground and line-to-line faults on the microgrid. This section briefly discusses other testbeds for smart grid systems that have been developed. It is important to consider such works in relation to the microgrid co-simulation testbed used in this research, even if they are not necessarily intended for evaluation of smart grid cyberattacks.

Du et al. [96] present a Hardware-in-the-Loop (HIL) real-time testbed for microgrid distributed control applications that are compliant with the IEEE 2030.7 Standard for the Specification of Microgrid Controllers [97]. In its own words, the standard defines “the functional requirements of the microgrid controller in a manner that can be universally adopted”. In other words, it specifies the minimum functional requirements that are needed for microgrid control to ensure proper operation. The real-time simulation setup presented by the authors is intended to validate any microgrid controller that meets the IEEE 2030.7 requirements. It uses an OPAL 5607 DRTS for switching components and an Opal 5031 for non-switching components such as power transformers. The HIL includes SEL 451 Relays, microcontrollers from Texas Instruments, and BeagleBone Blackboards. It also uses the Resilient Information Architecture Platform for Smart Grid (RIAPS) as a software platform for modeling distributed computing.

Ma et al. [98] propose a microgrid controller that allows a microgrid system to be integrated into an existing distribution automation system. Specifically, it supports the integration of a microgrid with a dynamic boundary, which refers to a microgrid that can connect and disconnect its DERs during operation. The proposed controller acts as an intermediary between the MGCC and local DER controllers. The proposed controller manages the dynamic boundary with the use of online topology assessment and balancing of active/reactive (P/Q) power. A simplified Kruskal’s algorithm, adopted from graph theory, is used to generate the microgrid topology online. The topology assessment steps are (i) identify independent connected graphs from the switch status readings, (ii) identify boundary switches for each grid area, (iii) create a grid area for an unenergized load, and (iv) report the P/Q balance. The authors use MATLAB/Simulink [99] to model a real feeder system, and use an

OPAL-RT real-time HIL platform to simulate the microgrid. The platform communicates with the real microgrid controllers over DNP3.

Danilczyk et al. [100] propose a digital twin framework, called ANGEL, that supports microgrid security visualization. As a digital twin, ANGEL runs in real-time alongside the corresponding physical power system. In doing so, it provides operational data in real-time to system operators, which they can use to aid in microgrid diagnostics and recovery. ANGEL contains a physics-based simulation, a physical system model, and an adaptive GUI. The physics-based simulation is validated on the IEEE 39-bus system. The framework runs on MATLAB/Simulink, which runs quickly enough to keep up with the actual physical environment. The authors note that their proposed framework can be used to help detect FDI in real time by observing discrepancies between the framework output and the real system output. However, their framework does not implement machine learning features that can be leveraged for cyberattack detection. It also simulates microgrid communications rather than implementing real communication protocols such as those defined in IEC 61850.

Regarding testbeds used to demonstrate cyberattacks, Qassim et al. [47] demonstrate the impact of false command injection attacks against IEC-104 traffic in power systems managed by a SCADA. Such attacks exploit the lack of authentication and encryption in IEC-104 traffic. The authors use an OPAL-RT OP5600 DRTS to simulate an IEEE 39-bus power system. They also integrate Human-Machine Interfaces (HMIs), an RTU, and engineering workstations. They use the proprietary Real Time Application Platform (RTAP) to model the RTU. An example of a harmful command injection attack that modifies the state of a CB is demonstrated on the testbed.

As evidenced by these works, the fidelity requirements of a microgrid simulation platform depend on the purpose of the simulation. In the case of this thesis, the primary purpose of the microgrid simulation is to assess the impact of potential attacks, including cyberattacks, on the microgrid power dynamics. The simulated microgrid model incorporates the various protection and control functionalities found within real microgrid systems to help ensure that the observed attack impacts closely correspond to what would occur if the attack were to be successfully replicated on a real microgrid with similar power system architecture and device configurations.

Chapter Four

Microgrid Model

This chapter details the microgrid co-simulation testbed used to collect realistic microgrid OT data in near-real time and assess the impacts on various cyberattacks on the microgrid power system. The testbed includes various control and protection functions that help maintain microgrid power stability and ensure that faulted lines are safely isolated. The microgrid cybersecurity framework proposed in this thesis identifies anomalies within power measurement data and within IEC 60870-5-104 (IEC-104) telecontrol data. As such, it is important that this thesis uses an appropriate power reference model when proposing anomaly detection methods that accurately distinguish between normal and anomalous microgrid telemetry. This chapter also presents the microgrid system model used to assess the capabilities of the threat hunting approach proposed in this thesis, including the attack step prediction process. The microgrid system model consists of a network of connected devices spanning from the microgrid control centre to the physical process level.

4.1. Microgrid Co-simulation Testbed

The description of the microgrid co-simulation testbed given here is divided into discussion on the microgrid power simulation model and on the microgrid control network. The microgrid power simulation and control centre are deployed on a power Digital Run-Time Simulator (DRTS) and emulated network server, respectively. The term “co-simulation” used in this thesis refers to the integration of the simulation elements provided by the DRTS with the capability to monitor and

manipulate the protocol messages found in actual smart grid power systems like the microgrid.

Note that the microgrid co-simulation model presented here is an extension of the one presented in our master's thesis [13]. The additions made to the testbed can be summarized as follows:

- a Fault Ride Through (FRT) control scheme is added that is engaged at the ESSs upon a significant voltage drop or spike;
- a subset of microgrid buses have IEC 61850 SV frames transmitted from one I/O port of the power DRTS into another, passing through a bridge host in the emulated network;
- multiple line fault locations are added to the simulation, which can be triggered on an individual basis if the user so chooses; and
- a voltage synchronization procedure is added to the simulation that allows for the phase angle difference between the microgrid and the main grid to be brought into alignment prior to closing the PCC switch during the process of transitioning to grid-connected mode.

4.1.1. Microgrid Power Simulation Model

The microgrid model is developed to simulate and observe the impacts on microgrid electric power generation that may result from line faults, line disconnections, and changes to the data transmitted over digital communications. This microgrid power model is implemented using the HYPERSIM power system simulation software [14]. To simulate the microgrid power system, the model is deployed on an OPAL-RT DRTS, which has a real-time precision of $50\mu s$.

The microgrid power simulation model contains a number of DERs and loads connected to a shared high voltage AC bus, rated at 25 kV. The PCC connects the high-voltage bus of the microgrid to a voltage source. The voltage source is used to simulate the power that the microgrid receives from the main grid when the microgrid PCC switch is closed, i.e., when the microgrid is operating in grid-connected mode. The set of connected DERs includes a solar PV array, three ESSs, a wind farm containing three WTGs, and a CHP generator. The simulated microgrid uses these DERs, and the main grid in the case of grid-connected operation, to serve the electric power needs of three consumer loads. The simulated microgrid model is depicted in Figure 4.1. Each DER and load line

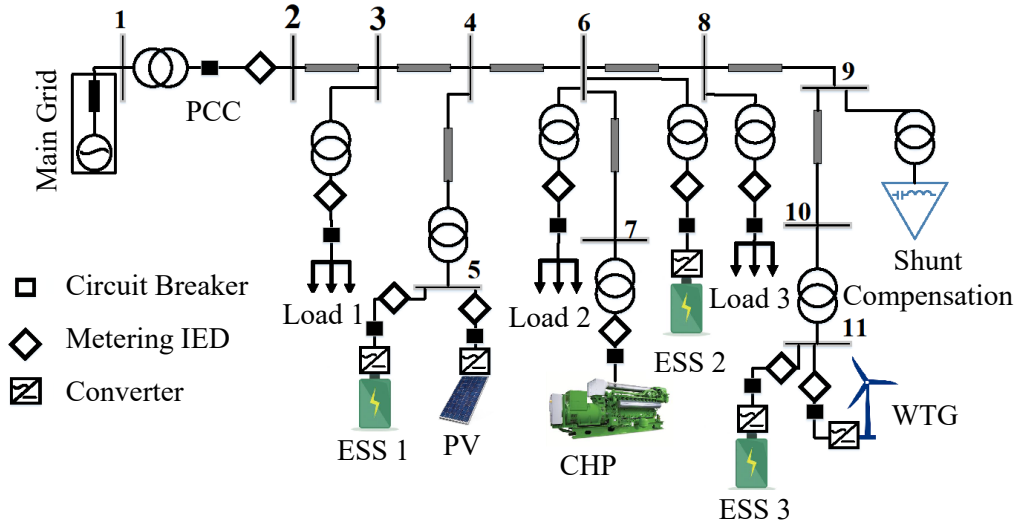


Figure 4.1: High-level view of microgrid power simulation model.

has a CB and a metering IED performing the role of the Merging Unit (MU). The ESSs and the solar PV array are DC sources, and so they require DC-AC converters that turn their DC generation into AC before they can be integrated into the microgrid power network. The wind farm, despite being an AC source, also needs a converter to synchronize its output to the 60 Hz microgrid. The wind farm uses an AC-DC-AC converter to ensure that the power it generates flows into the microgrid at the correct frequency. The CHP does not need a converter because it already generates AC at the correct frequency. The microgrid model also has a shunt compensation line, which is a capacitor bank that supplies reactive power to maintain grid stability when power demand increases [101].

The CHP is rated at 10 MW and is the DER with the highest energy generation capacity in the modeled microgrid. The next largest source of generation is the wind farm, whose three WTGs are each rated at 2.2 MW and thus have a combined rating of 6.6 MW. The solar PV array is rated at 1.5 MW. The rated power for ESS 1, ESS 2, and ESS 3 are 1 MW, 2 MW, and 2 MW, respectively. The three ESSs are the dispatchable DERs of the microgrid model. Their power setpoints are adjusted continuously according to their local frequency control loop, which adjusts the active power setpoint proportionally with the frequency deviation. The ESSs also receive setpoint adjustment commands sent by the Microgrid Central Controller (MGCC) that do more fine-tuned adjustments of the power balance to maintain stable frequency. Frequency control via the ESSs is primarily

important during islanded mode operation. In grid-connected mode, the main grid provides enough inertia to make the ESS power output adjustment far less impactful. The loads each consume 1 MW under normal simulation conditions. They are, however, implemented as dynamic loads whose power consumption can be reduced in accordance with the load consumption setpoint input. In cases where there is significant overfrequency above the safe operation conditions, the MGCC can attempt to balance power by sending a command to the wind farm to disconnect one of the three WTGs, curtailing the total power generation of the wind farm by one third.

Regarding protection functions, the microgrid power simulation model implements protection relays that send trip signals to its corresponding CBs in the case of sustained over/under voltage or over/under current. The relays determine if the tripping conditions are met by reading the voltage and current measurements transmitted by the local MU. How quickly they send a trip command depends on the extent of the deviation. There is also a frequency relay near the PCC that sends a signal to disconnect the entire microgrid upon detecting sustained frequency deviations. Accounting for the specifications of IEEE 1547 [42], the frequency relay tolerates frequencies within a 5% margin from the nominal 60 Hz, which corresponds to an acceptance range of 57 Hz to 63 Hz. The relay disconnects the microgrid if the frequency falls outside of this range for 8 seconds. This provides the MGCC with a time window within which it attempts to stabilize the grid. The frequency relay has a step-wise tripping configuration that causes it to send trip commands even earlier if frequency thresholds for higher severity deviations are crossed.

The ESSs also have a protection role in the microgrid model. Under normal operation, the ESSs operate in PV mode, meaning they generate power according to reference setpoints for active power and voltage. However, they have a FRT response when the line voltage deviates below 0.80 times the nominal voltage or above 1.1 times the nominal voltage. This response follows from the microgrid FRT response described by Ahmed et al. [102]. Such drops are typically the result of line faults. During FRT, the ESSs switch from PV mode to PQ mode, where their active power (P) and reactive power (Q) output are controlled by setpoints. The ESS control mode switch is depicted in Figure 4.2. The ESS control is configured to ignore all setpoint commands sent from the MGCC while the ESS is in PQ mode. The ESS injects reactive power in proportion to the voltage deviation (V) from the nominal value. This injected reactive power helps stabilize the microgrid during fault conditions

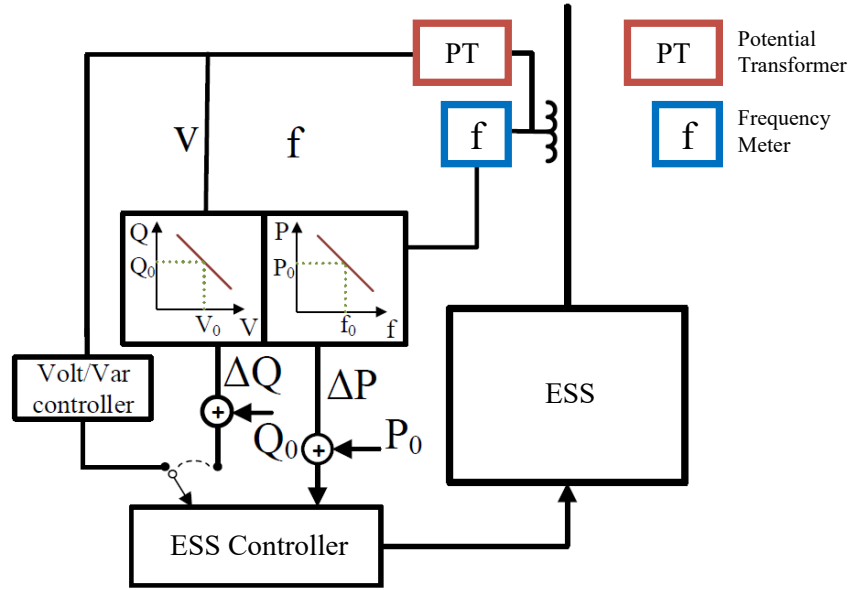


Figure 4.2: Local ESS control scheme for PV and PQ mode.

until the fault clears on its own or until the faulted line is isolated. The FRT response is implemented in HYPERSIM using C-code blocks.

HYPERSIM has I/O drivers that support a number of communication protocols intended for transmission of OT data. This thesis research makes use of its drivers for IEC 61850 SV and GOOSE as well as its IEC-104 server driver. With these drivers, HYPERSIM can have the simulated microgrid send and receive Ethernet frames and IP packets through one of several I/O ports on the OPAL-RT DRTS. The I/O ports are Ethernet interfaces that can be connected to an external emulated network. By doing so, applications deployed in the external network can read the protocol messages transmitted over the course of the simulation by the DRTS, as well as send messages back into the DRTS to be read by its simulated devices. The OPAL-RT DRTS and the external network are connected via a physical network switch. It is noteworthy that a host in the external network can act as a bridge that forwards messages transmitted from one DRTS I/O port into another, creating a closed communication loop. This thesis makes use of such bridge hosts to launch FDI attacks. The physical switch has a VLAN trunk connecting it to the emulated network, and the interfaces connecting the switch to the OPAL-RT DRTS I/O ports are configured to have different VLANs. This ensures that messages transmitted from the I/O ports must first travel through the emulated

network via the VLAN trunk before they can reach a different I/O port on the DRTS.

4.1.1.1 Microgrid Control Network

The secondary control of the microgrid model used for this thesis is implemented through digital communication between the HYPERSIM simulation model and an MGCC application hosted on a Virtual Machine (VM) in an emulated network managed through an OpenStack hypervisor [15]. Specifically, IEC-104 servers are simulated using HYPERSIM I/O drivers and the MGCC application establishes a TCP connection with the servers to communicate over IEC-104. The simulated IEC-104 servers all have the same IP address but have their own TCP port. As such, the MGCC establishes a TCP connection with multiple TCP ports at the same IP address in order to connect to the IEC-104 servers simultaneously. The packet rate of IEC-104 observed in the testbed ranges from 2 to 4 packets per second. The MGCC used in this thesis runs custom code written in C that uses the lib60870 library to implement the IEC-104 client functionality [103]. In addition to hosting the MGCC, the emulated network hosts VMs that act as bridges used to complete the IEC 61850 SV forwarding loop of the co-simulation.

The MGCC uses IEC-104 to communicate with the rest of the simulated microgrid to perform the following functions:

- Frequency control through control of the dispatchable DERs i.e, the ESSs
- Load shedding through control of the allowable consumption at the loads
- Power curtailment through disconnection of a WTG in the wind farm
- Managing the State of Charge (SoC) of the ESSs
- Initiating transition from islanded mode to grid-connected mode and vice versa

The frequency control loop has the MGCC receive a stream of frequency measurements from the PCC. The deviation, or error, between the measured frequency at time t and the desired frequency of 60 Hz is calculated as $e(t)$ and passed through a Proportional Integral (PI) controller. The PI controller computes the output signal o_{pi} and sends it to the ESSs as a power setpoint signal through a command message. The PI controller output is calculated as follows [104]:

$$o_{pi}(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (4.1)$$

Where K_p is the gain for the proportional component and K_i is the gain for the integral component. The Laplace transform applied on 4.1 yields the following transfer function O_{pi} :

$$O_{pi} = K_p + \frac{K_i}{s} \quad (4.2)$$

Essentially, the PI controller sends incrementally larger control signals if the measured frequency error does not correct itself. It is important that the microgrid frequency and ESS power output are updated in a timely manner to ensure that the microgrid can properly converge to the desired frequency. The frequency control loop performed by the MGCC has more impact during islanded mode operation, where the microgrid has an overall lower inertia compared to traditional synchronous generator power systems [8], [105]. The microgrid can also send signals to the ESSs to switch between a droop control scheme and a centralized control scheme. In the droop control scheme, the setpoint signals received by the ESS from the local frequency control and the MGCC frequency control are summed and applied to set the power output of the ESS. In the centralized control scheme, only the setpoint signals from the MGCC are applied on the ESS.

Load shedding is initiated when the microgrid frequency is well below 60 Hz (underfrequency), which occurs when power generation is struggling to meet the power consumption demands. In such cases, the MGCC gradually decreases the consumption setpoint of the dynamic loads until the frequency returns to an acceptable range. In the overfrequency direction, power curtailment is initiated when the frequency is far above 60 Hz, which occurs when the power generation exceeds what the loads can consume. As previously mentioned, the power curtailment involves sending a switch disconnection command to the wind farm to remove one of its WTGs from the generation line, cutting the power output of the wind farm by one third.

The MGCC monitors the SoC of the ESS to ensure that they are not operating beyond capacity. Specifically, the MGCC will override the frequency control signals of the PI controller to ensure that an ESS that is drained of energy does not attempt to inject more active power into the microgrid. In a similar fashion, the MGCC prevents an ESS that has reached its energy storage capacity from

absorbing more energy from the microgrid.

The MGCC, through user input, can initiate a transition from islanded mode to grid-connected mode. Once initiated, the MGCC will begin to read the measurements for the voltage phase angle difference between the microgrid and the main grid at the PCC. In islanded mode, the PCC is disconnected, and thus the microgrid phase angle and frequency will not necessarily be synchronized with the main grid. It is unsafe to engage the PCC switch and enter grid-connected mode where there is a large disparity between the microgrid and the main grid phase angle and frequency [35]. Within the microgrid simulation, a sudden PCC connection when the microgrid and main grid are desynchronized results in a frequency spike, with the spike magnitude increasing as the phase angle difference at the moment of connection increases. The largest frequency deviations observed across possible phase differences at the moment the PCC switch is closed are shown in Figure 4.3 for the simulated microgrid model. Before closing the PCC switch to connect the microgrid to the main grid, the microgrid will begin a synchronization process. This involves setting the target frequency to 60.1 Hz, which is a slight deviation from the nominal 60 Hz. It maintains the deviated frequency until the phase angles of the microgrid and the main grid are aligned at the PCC. The risks posed by entering grid-connected mode while desynchronized mean it is important that the phase angle difference measurements are accurate and untampered. By contrast, the transition to islanded mode from grid-connected mode does not require a synchronization process. Within the context of the microgrid model, the PCC switch can be disengaged at will via a command from the MGCC. In a real setting, this allows the microgrid to quickly isolate itself from the main grid in response to instability within the main grid.

The phase angles of the microgrid and main grid are calculated within the HYPERSIM simulation model and reported to the MGCC. A C-code block is implemented in the simulation model that keeps track of the observed voltage amplitude V_{amp} and the instantaneous real voltage V_r . The value for V_{amp} is updated to be equal to the peak voltage observed during the last half-cycle. The C-code block uses V_{amp} and V_r to calculate the phase angle by first computing the angle θ_o using the following equation:

$$\theta_o = \cos^{-1}\left(\frac{V_r}{V_{amp}}\right) \quad (4.3)$$

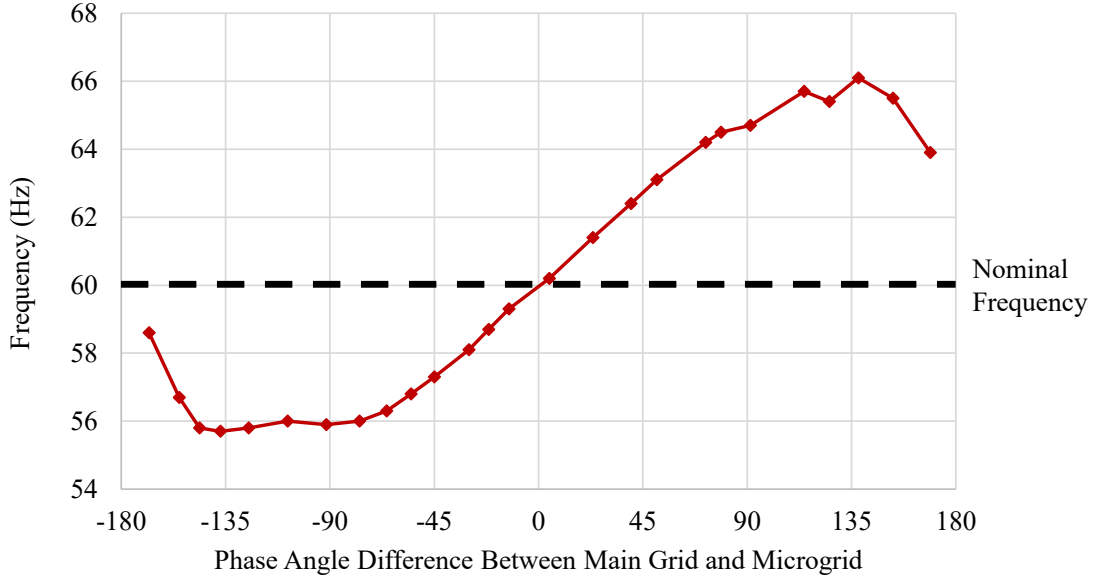


Figure 4.3: Frequency deviation after closing PCC based on phase angle difference.

The true phase angle θ , whose range is $[-180^\circ, 180^\circ]$, is then determined according to the formula below:

$$\theta = \begin{cases} \theta_o & \text{If real voltage } V_r \text{ is decreasing} \\ -\theta_o & \text{Otherwise} \end{cases} \quad (4.4)$$

A visualization of how the phase angle of the voltage phasor can be found from θ_o at each quadrant of the complex plane is given in Figure 4.4. Note that the voltage phasor rotates counter-clockwise within the complex plane.

4.2. Microgrid Threat Hunting System Model

This section describes the microgrid system network model used to demonstrate the capabilities of the threat hunting approach proposed in this thesis. It presents the topology of the system model, as well as a usable mapping between the modeled device assets and the ICS assets associated with MITRE ATT&CK techniques. Though the design of the system model has some overlap with the network of simulated and emulated devices that make up the microgrid co-simulation testbed, the

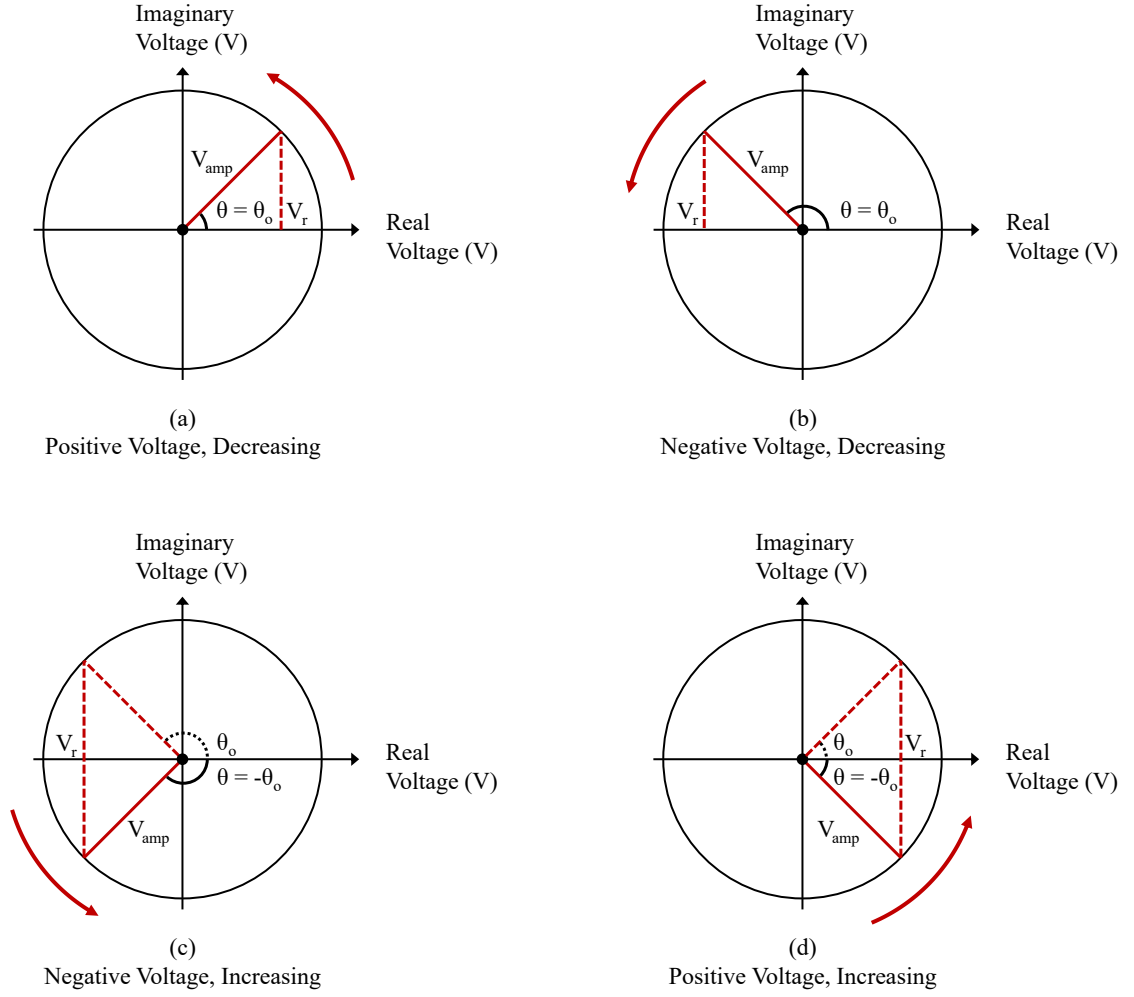


Figure 4.4: Voltage phasor angle θ according to the phasor quadrant.

system network model is currently independent of the co-simulation testbed.

4.2.1. System Model Topology

The design of the microgrid system model takes inspiration from the Securing Energy Infrastructure Executive Task Force (SEI-ETF) reference model for the DER profile when it comes to the selection and arrangement of microgrid devices [32]. It also takes inspiration from the ICS Purdue model [33] when assigning system levels to microgrid devices, henceforth referred to as Purdue levels. The system model has a single control centre that manages multiple DERs and loads, as well as the PCC. However, the DERs and loads have a hierarchical network of their own. The topology

of the devices involved in the primary control, secondary control, and maintenance of the microgrid is summarized in Figure 4.5. At the lowest level of the model are MUs and CBs. They share a process bus with the protection relay. The MUs, CBs, and protection relays within a load or DER are all managed by a supervisory controller. The supervisory control also manages the controllers that interface with a DER or load to dynamically alter their operation state. The supervisory controller is itself managed by the telecontrol server and the MGCC that lie in the control centre. Within the control centre is an operator workstation used by microgrid personnel, which interacts with the telecontrol server with the intent of delivering messages to the supervisory controller via the telecontrol server. In addition, there is a maintenance workstation within the control centre that connects to the supervisory controllers and relays through a separate communication network dedicated to management tasks. Maintenance personnel use this network for tasks such as device health monitoring and software updates.

The topology of the microgrid control centre is shown in Figure 4.6. As already stated, the control centre hosts the telecontrol server, MGCC, and user workstations for operation and maintenance tasks. The system model specifically includes two operator workstations to reflect how the microgrid may be managed through the cooperation of multiple operators. It also hosts a database server and an application server. Access to the workstations is managed by RASs, which allow users at the enterprise level of the power utility to connect to the microgrid control centre. There is also an Internet firewall put in place for the maintenance workstation such that it can access reference material and software when needed.

Drawing from the security levels defined in the SEI-ETF reference architecture for the DER profile and the levels in the Purdue Model, the microgrid system network model includes assigned Purdue levels for its microgrid assets. The levels describe logical and/or physical boundaries between assets in different levels. The microgrid assets are also described as falling into different network zones depending on their functional role, with certain Purdue levels having multiple zones. A breakdown of the microgrid system assets into Purdue levels and functional zones is given in Figure 4.7. To reduce visual clutter, the communication link between the supervisory controller and the CB is omitted in the figure, as is the one between the supervisory controller and the MU. The proposed threat hunting approach takes into consideration the Purdue level boundaries that are

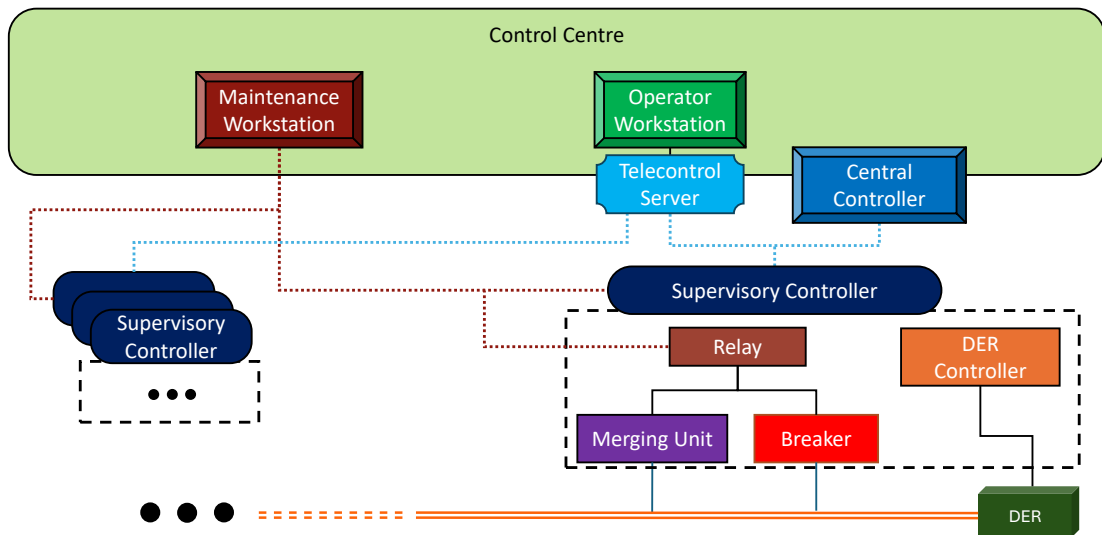


Figure 4.5: Devices involved in primary and secondary microgrid control.

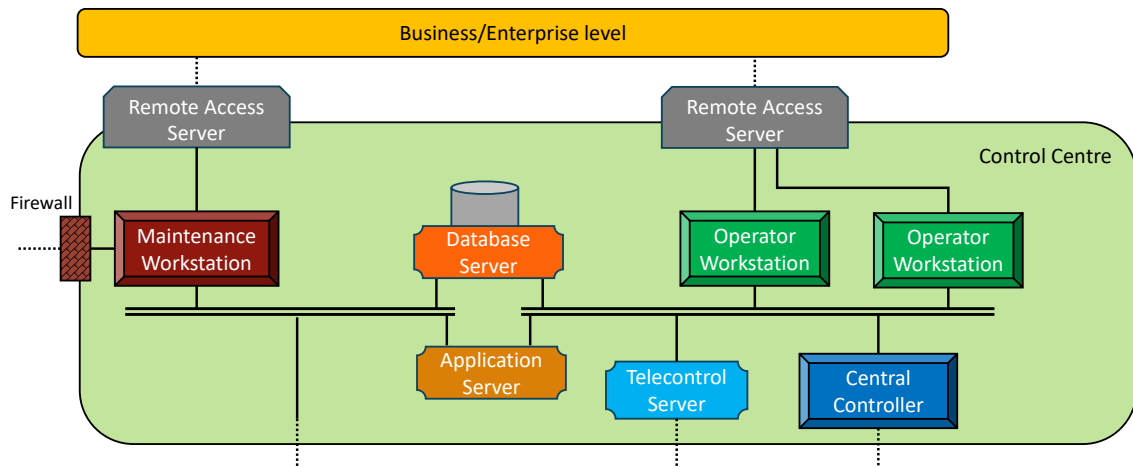


Figure 4.6: Devices in the microgrid control centre.

believed to have been crossed when assessing the severity of a given attack hypothesis.

4.2.2. Mapping Microgrid Assets to MITRE ATT&CK

When considering how a threat actor might attack a given system, it is important to understand the role and characteristics of each asset in the system. By doing so, the potential capabilities and intentions of the attacker can be determined for a given set of privileges. In addition, the added value

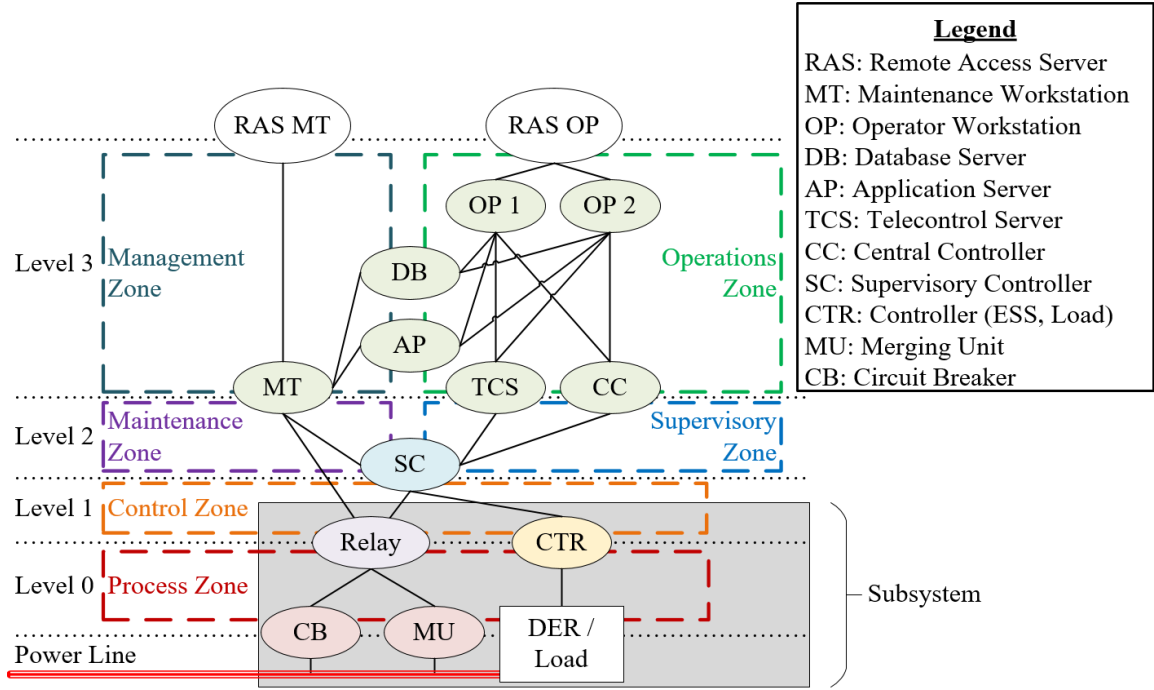


Figure 4.7: Network topology of the microgrid system model.

that the attacker gains by exploiting the asset can be identified. The MITRE ATT&CK framework is a source of CTI that gives insights about the Tactics, Techniques, and Procedures (TTPs) that have been used on various kinds of assets. Specifically, the techniques defined in the ICS MITRE ATT&CK Matrix are associated with ICS assets that they have been known to target. This thesis research leverages this CTI by specifying a direct mapping between the assets that comprise the modeled microgrid system and the ICS assets defined in MITRE ATT&CK. This mapping is given in Table 4.1, which also mentions the Purdue level of the asset. This mapping is used to filter the set of TTP that can potentially target a given microgrid asset when designing the APT attack model.

4.2.3. Simulation of the Microgrid Network

The microgrid system model described in this section is converted into a network graph using Python scripting. The graph treats assets as nodes and direct communication channels between assets as edges. Each microgrid asset is represented by (i) a node corresponding to its operating system or control logic and (ii) one or more additional nodes that each correspond to a network interface on the device. An asset connected to two different IP subnets or network switches can act

Table 4.1: Mapping of microgrid system asset to ICS ATT&CK assets

Microgrid Asset	Purdue Level	MITRE ATT&CK Asset
Remote Access Point	3.5	Virtual Private Network Server
Operator Workstation	3	Workstation
Maintenance Workstation	3	Workstation
Database Server	3	Data Historian
Application Server	3	Application Server
Telecontrol Server	3	Control Server
Microgrid Central Controller	3	Control Server
Supervisory Controller	2	Control Server
Protection Relay	1	Intelligent Electronic Device
Point of Common Coupling	1	Programmable Logic Controller
Controller (ESS, Load)	1	Programmable Logic Controller
Merging Unit	0	Field I/O
Circuit Breaker	0	Safety Controller

as an intermediary that must be crossed before an attack affecting one sub-network can reach the other. Each node and edge is represented by a Python object containing its name, direct connections, and Purdue level. The `NetworkX` Python module is used to process the graph nodes and edges into a single query object. The `NetworkX` query object is used to automatically compute the shortest path between every pair of assets. The shortest path is equal to the fewest number of edges that need to be crossed to reach a given target asset from a given starting asset. Each asset has a neighbourhood of radius H , where the shortest path length between the asset and any of the assets in its neighbourhood is less than or equal to H .

Chapter Five

Active and Passive Security Monitoring

This chapter presents the microgrid security monitoring contributions that have resulted from the research undertaken as part of this thesis. One of these contributions is a study on how best to use IEC 62351-7:2017 Network and System Management (NSM) active monitoring for detecting attacks on the microgrid telecontrol network. This study involves the collection of NSM data over SNMP and a proposed anomaly detection scheme for analyzing the NSM data using Deep Neural Network (DNN). The other contribution is the proposal of a passive monitoring solution for detecting fault-spoofing False Data Injection (FDI) attacks. The passive monitoring solution uses distributed monitors that report to a central security manager that identifies discrepancies in power measurement data. Both of these contributions involve the use of Artificial Neural Networks (ANNs) to uncover attacks through anomaly detection. More specifically, they make predictions on the next set of monitored values to arrive based on a window of the contiguous values preceding it, then raise an alert if any values deviate from thresholds set during model validation. However, they have a different monitoring infrastructure and process different sets of telemetry.

In the context of this thesis, “active monitoring” refers to data collection based on request/response interactions between a security manager and an agent. The collected data may provide information on system health and performance, which has value to the security manager but is not used by the system to meet its functional requirements. Such monitoring requires that network devices devote computational resources to cooperate with the security manager. By contrast, “passive monitoring” refers to observing essential network traffic without having the means to directly disrupt the traffic

or the devices. Network sniffing, in cases where the traffic reaching the sniffer is duplicated rather than intercepted, is an example of passive monitoring. The proposed passive monitoring solution positions its distributed monitors as network sniffers for power measurements.

5.1. Anomaly Detection Model Design

The proposed microgrid security monitoring framework uses two main data sources: (i) IEC 62351-7:2017 NSM data collected via active SNMP monitoring, and (ii) IEC 61850 SV power measurement data collected through passive monitoring. These data sources are analyzed by ANN machine learning models for the purpose of detecting anomalies resulting from cyberattacks. These machine learning models all use a window of contiguous time series data points as input to make a prediction of the next data point, with the prediction being directly compared to the actual data point retrieved from the system. The actual data point is labeled as an anomaly if it differs sufficiently from the prediction. The ANNs used in this thesis have multiple hidden layers and thus are considered to be Deep Neural Networks (DNNs).

The machine learning models for the proposed anomaly detection learn the state of the microgrid with respect to the time series of states that preceded it. The state of the microgrid at a given time t is represented by a snapshot S^t . These snapshots are an aggregate of values that correspond to data elements monitored at time t , collected from a set of K monitors. To perform anomaly detection, the snapshots are arranged in a time series window $W = [S^{t-1}, S^{t-2}, \dots, S^{t-N}]$ containing the sequence of the last N snapshots. The window is then given as an input to a DNN model, which makes a prediction for snapshot S^t . The feature set, containing M features, is common across all the snapshots. Each feature corresponds to an ANN input node to which it is fed. Each hidden layer has twice as many nodes as there are input features, i.e., $2M$ nodes. The output of the DNN is a prediction for each of the M feature values in the newest snapshot S^t , with each output node corresponding to a specific feature value. The actual feature values of S^t are compared to those of the predicted snapshot to determine the feature losses $L = [L_1, L_2, \dots, L_M]$. Loss L_i is equal to the difference between the predicted value and the actual value for the i^{th} feature. An alert signaling an anomaly is generated if there is a feature i whose feature loss L_i exceeds the corresponding

threshold, Th_i . The thresholds for the features are determined during the training of the DNN. Figure 5.1 depicts the usage of the DNN prediction models for anomaly detection. The prediction model input is a sliding window that is shifted every time a new snapshot is observed. That is to say, once snapshot S^{t+1} arrives, snapshot S^t is added to W and snapshot S^{t-N} is removed, with the resulting window being used as input to predict S^{t+1} .

The DNN models explored for anomaly detection in this thesis include Long Short-Term Memory (LSTM) [106] and Gated Recurrent Unit (GRU) [107], which are recurrent neural networks. In addition to these DNN models, a feed forward Multi-Layer Perceptron (MLP) is explored as well. In the case of the MLP, the time series window is flattened into a single vector of size $N \times M$ to use as input for the MLP. Meanwhile, the LSTM and GRU essentially have a 2-dimensional input, with one dimension being the snapshot size $i \in [1, M]$ and the other being window slot $j \in [1, N]$. The 2-dimensional input reflects how the LSTM and GRU pass the input snapshots into their DNNs sequentially. A depiction of the recurrent neural network prediction process is given in Figure 5.2.

5.2. Active Monitoring of Microgrid Telecontrol Data

This section discusses the methodology, scalability concerns, and experimental results of a proposed active security monitoring platform based on IEC 62351-7:2017 NSM. This platform is used to detect anomalies that emerge due to attacks launched against the microgrid telecontrol traffic sent over IEC-104 TCP/IP. The Network and System Management Data Objects (NSM DOs) defined in IEC 62351-7:2017 are intended to assist in health monitoring and intrusion detection within the power system. However, IEC 62351-7:2017 does not give specifications or instructions on how to process the NSM DOs to detect potential security breaches. To address the gap in security specifications, this thesis evaluates the effectiveness of DNN models for identifying anomalies within the NSM DOs, collected through active monitoring, that follow from microgrid cyberattacks. The active security monitoring platform presented in this thesis is an extension of the security monitoring platform proposed in the master's thesis of the PhD candidate [13]. The extensions made through this thesis research are as follows:

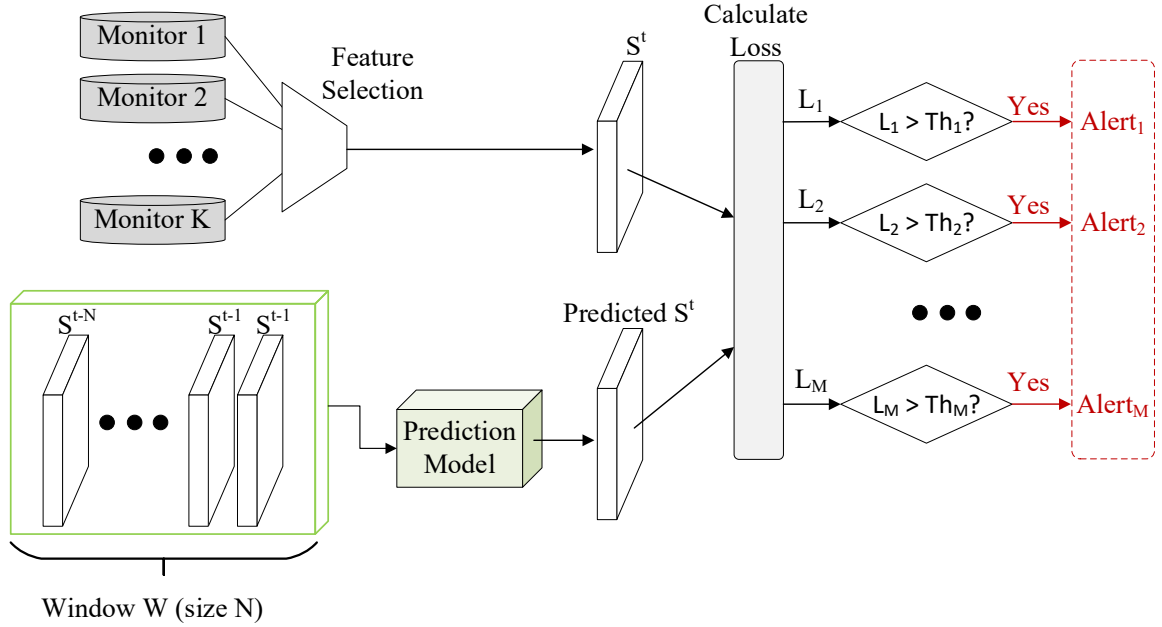


Figure 5.1: Design of the ANN detection process.

- The scalability of the active monitoring platform is evaluated by observing how its performance is impacted when the number of agents is increased and when the number of polled SNMP MIB objects is increased.
- The effectiveness of the anomaly detection is evaluated for different DNN architectures and training hyperparameters.
- The success of the anomaly detection based on DNN is compared against other machine learning models, including classifier models.

5.2.1. Active Monitoring through IEC 62351-7 NSM

The security monitoring framework proposed in this thesis includes an active monitoring platform that has an NSM manager poll agents at various sections of the microgrid for security management data at regular intervals. The polled sections include the DER, the loads, the PCC, and the Microgrid Central Controller (MGCC). Within the model, the NSM agents are assumed to be installed on the supervisory controllers, treating the controller as a monitored device. The agents

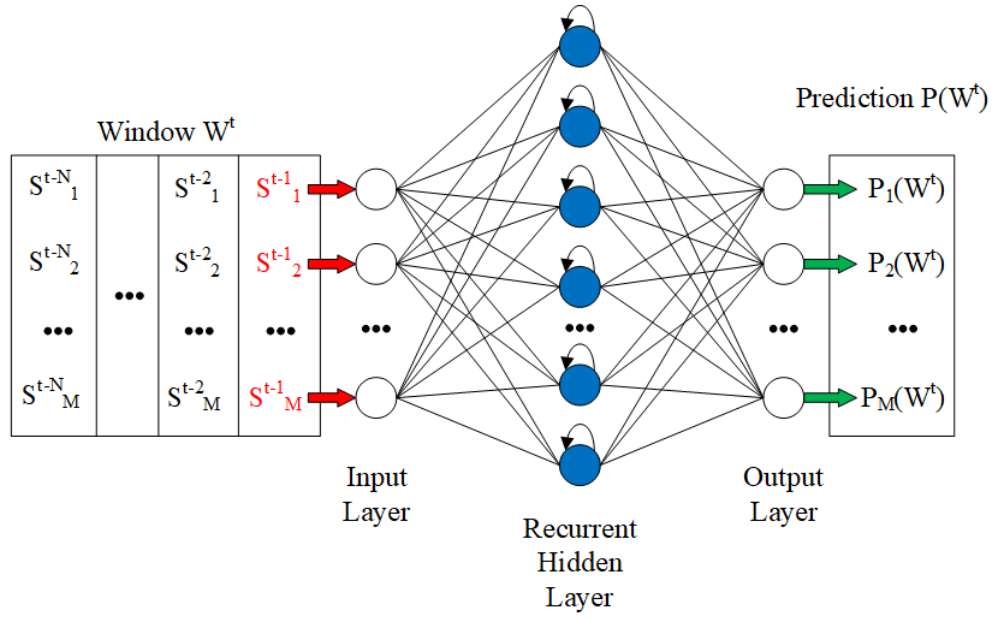
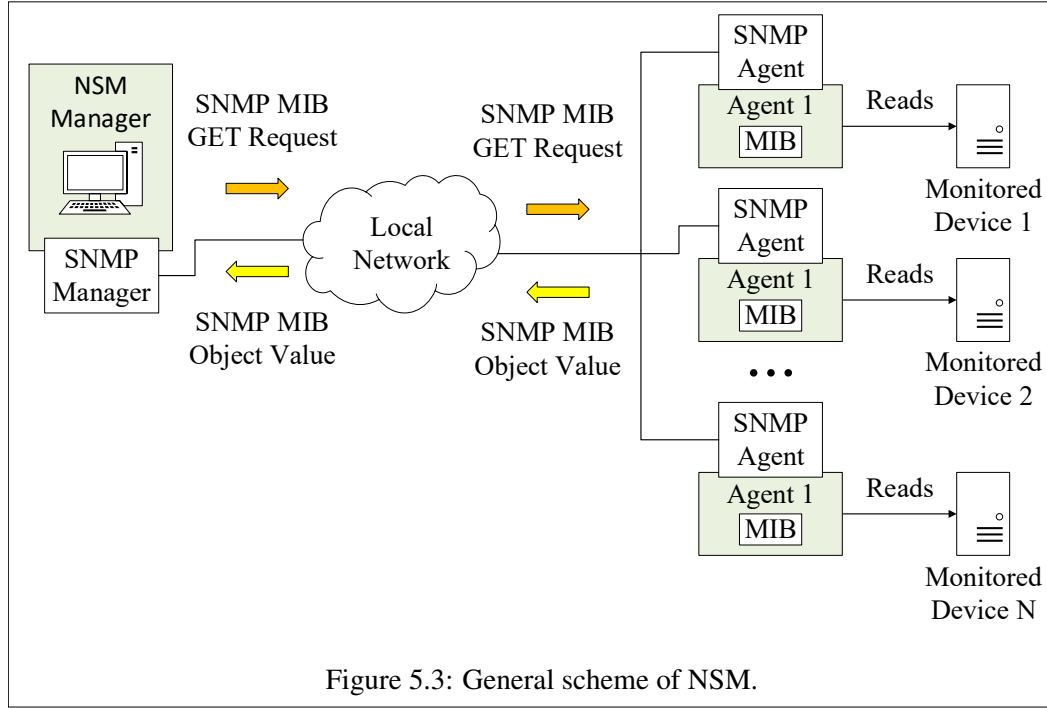


Figure 5.2: Architecture of the recurrent neural network prediction model.

report on the network flow of the telecontrol communication between the MGCC and the supervisory controller they monitor. The MGCC has its own NSM agent installed as a separate process on its host machine. The NSM manager collects data from each of the agents using SNMP, as shown in Figure 5.3. Specifically, it uses Simple Network Management Protocol Version 3 (SNMPv3) to authenticate and encrypt the SNMP traffic to protect the confidentiality and integrity of SNMP messages. The NSM manager shares unique cryptographic keys, specifically SHA-1 and AES-128, with each of the NSM agents to use for SNMPv3. The agents having their own key is a precaution that prevents an attacker who has compromised the keys of one agent from simultaneously compromising the remaining agents. The attacker would need to compromise each agent separately.

The data being polled contains MIB object values defined in RFC-3418 [50] and IEC 62351-7:2017 [19] for SNMP. In regards to the NSM anomaly-based detection proposed in this thesis, the two MIBs of interest are (i) the IEC 62351-7:2017 MIB for IEC-104 and DNP3 (IEC-62351-IEC60870-5-IEEE1815-MIB) and (ii) the SNMP MIB for TCP (TCP-MIB). These two MIBs allow the NSM manager to collect statistics on the IEC-104 protocol messages used for telecontrol, which is sent over TCP. Within these MIBs are a number of MIB objects that are the subject of a rules-based security check that results in an alert whenever their value changes. The full list of monitored



objects from the IEC-104 and TCP MIBs is given in Table 5.1.

5.2.2. Testbed Implementation of NSM

The components that make up the NSM active monitoring are all implemented on Virtual Machines (VMs) within the OpenStack emulated network. These include the NSM agents and the NSM manager.

5.2.2.1 NSM Agent Implementation

Each NSM agent is deployed on a separate VM, with the MGCC and its agent being deployed on the same VM. The VMs run Ubuntu Linux 16.04 as an operating system. The agents run a customized version of NetSNMP [108]. The customized software implements C code for many of the MIB objects defined in IEC 62351-7:2017. The implementations for each MIB object are largely identical. Upon receiving an SNMP *GET* request for its MIB object, the agent code simply retrieves the appropriate value from its local SQLite database [109] and sends the value, over SNMP, back to the address that sent the request. The values in the SQLite database are constantly updated by a continuously running Linux Bash script. The script uses tshark to parse the trace files

Table 5.1: MIB objects collected for microgrid active monitoring

MIB Object	Description	Detection Method
IEC-104 MIB objects (IEC-62351-IEC60870-5-IEEE1815-MIB)		
tCInterPDUTime	Average time between received APDU	Anomaly-based
tCPDURTT	Average IEC-104 packet round trip time	Anomaly-based
tCRxCnt	Number of retransmitted IEC-104 packets	Anomaly-based
tCRxPdu	Number of IEC-104 packets received	Anomaly-based
tCRxCritical	Number of critical IEC-104 packets received	Anomaly-based
tCRxUnsolicitedReq	Number of unsolicited IEC-104 packets received	Anomaly-based
tCTxPdu	Number of IEC-104 packets transmitted	Anomaly-based
tCTxCritical	Number of critical IEC-104 packets transmitted	Anomaly-based
tCTxUnsolicitedReq	Number of unsolicited IEC-104 packets transmitted	Anomaly-based
tCOutErrCnt	Number of IEC-104 transmitted in error	Rule-based
tCInErrCnt	Number of IEC-104 packets received in error	Rule-based
tCPDUSizeFail	Number of IEC-104 packets with wrong length	Rule-based
TCP MIB objects (TCP-MIB)		
tcpInSegs	Number of TCP segments received	Anomaly-based
tcpOutSegs	Number of TCP segments transmitted	Anomaly-based
tcpRetransSegs	Number of retransmitted TCP segments	Anomaly-based
tcpActiveOpens	Number of TCP connections opened from the closed state	Anomaly-based
tcpOutRsts	Number of TCP segments with the RST (reset) flag set to 1	Rule-based
tcpInErrs	Number of TCP segments received in error	Rule-based

containing the captured IEC-104 traffic, process the relevant statistics, and update the database values accordingly with SQLite write commands. Each agent has its own version of the script that is configured to read from the TCP port associated with the monitored device, or the entirety of the IEC-104 communication in the case of the MGCC agent.

5.2.2.2 NSM Manager Implementation

The NSM manager is deployed on a separate VM in the OpenStack emulated network. The NSM manager lies outside the OT communication pathway, but communicates with the NSM agents that observe OT traffic. The setup of the NSM manager within the microgrid co-simulation testbed and the IEC-104 traffic flow is given in Figure 5.4. Unlike the NSM agents, the manager uses custom Python scripting with the `pysnmp` module to implement SNMP. The manager launches a separate instance of a custom script for each agent that polls the agent for MIB object values at a set interval.

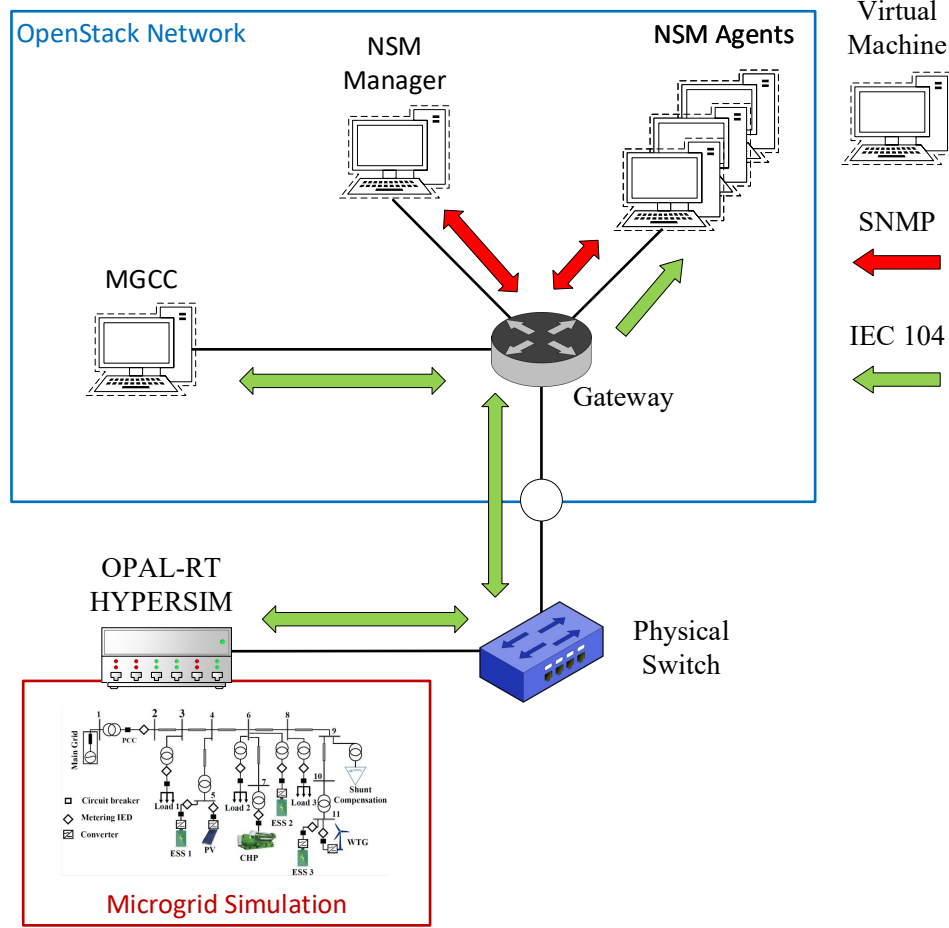


Figure 5.4: Flow of microgrid operation traffic and SNMP for NSM monitoring.

Specifically, the script sends a *GETBULK* request over SNMPv3 to collect all the needed object values for a single MIB within one response message. The script also creates a separate execution thread for each MIB in which *GETBULK* requests for the MIB are sent.

The NSM manager host machine is installed with an instance of the Elasticsearch service [110]. Elasticsearch is a search and analytics engine that stores data within indexes and provides search and visualization tools to inspect the indexed data. The NSM manager saves each agent response to Elasticsearch, assigning each agent to its own Elasticsearch index for storing its MIB object values. Also within the NSM manager host machine is a Kibana web application that, by design, allows users to visualize the data saved to Elasticsearch. The Kibana application is used to build an NSM security monitoring dashboard that has visualizations for information and statistics related to the

alerts generated through the NSM anomaly detection.

Within the testbed, a polling period of 5 seconds provides acceptable performance with low risk of failing to process the data collected for one polling period before sending the next poll request. However, the need to space out the polling periods by several seconds means that the active monitoring proposed in this thesis cannot be considered a true real-time detection capability. At best, it can be seen as a near real-time solution.

A separate custom script, written in Python, is run continuously on the NSM manager to read the MIB object values saved to Elasticsearch and then aggregate them into snapshots. Each snapshot has a timestamp that corresponds to the polling period where its values were collected. The snapshots are saved to a separate Elasticsearch index.

5.2.3. Scalability

The proposed active monitoring platform has a central NSM manager collecting data from multiple hosts over an IP network at regular intervals. As such, the collected data is subject to network delays. This is especially true given the request/response paradigm used by the platform, with network delays being experienced on both the SNMP request sent by the manager and the response sent by the agents.

Naturally, the total network bandwidth used by the platform increases as the amount of requested data increases and as the number of polled agents increases. The NSM polling rate should be chosen so as not risk impacting the performance of microgrid operations. In addition to increasing bandwidth, the time needed to preprocess and analyze the received data increases as the size of the data increases. The notable sources of delay during the anomaly detection process of the active monitoring platform are as follows:

- (1) The time spent waiting to receive all agent SNMP responses (Δ_{snmp}).
- (2) The time needed to construct the snapshot from the agent data (Δ_{sh}).
- (3) The time taken by the anomaly detection model to analyze the snapshot (Δ_d).

The total delay Δ_T between the sending of the SNMP *GETBULK* requests and determining if the resulting snapshot has anomalies is expressed as:

$$\Delta_T = \Delta_{snmp} + \Delta_{sh} + \Delta_D$$

The intention of the active monitoring platform is to have the anomaly detection processing of the data collected from a polling period be completed before the next polling period begins. The chosen polling rate should therefore be sufficiently larger than Δ_T .

Recall that the NSM manager is a single host machine that opens a separate polling process for each agent, and each of those processes runs a separate thread for each MIB being requested. As such, the multiple *GETBULK* requests of the manager during each polling period are not sent simultaneously. Limitations in the computing power of the manager can result in substantial delay between the first and last SNMP request sent as the total number of requests increases. To quantify this impact, the Δ_{snmp} is measured for different numbers of polled MIBs and agents. Figure 5.5 shows a graph of Δ_{snmp} in seconds for one, three, and five MIBs collected from one to ten agents. Though the proposed NSM anomaly detection only reads snapshots containing values from two MIBs, multiple MIBs are implemented on the NSM agents. Examples of these MIBs are IEC-62351-DEV-MIB for IED health monitoring, IEC-62351-ENV-MIB for monitoring device access events, and IEC-62351-INT-MIB for monitoring the Ethernet interfaces of the monitored device. The graph shows the curves for the average, maximum, and minimum delay for each combination of MIB count and number of agents. It can be noted that the delay spikes considerably upon moving to five MIBs, which becomes exacerbated by the increase in the number of agents. One explanation for this spike is that the delays incurred by the Python threading functions delays are larger than the ones incurred by increasing the number of Python processes to scale to a higher number of agents. The observed increases in delay highlight the value of carefully selecting the total set of monitored NSM data features to minimize the Δ_{snmp} .

The creation of microgrid snapshots from collected MIB data depends on the execution time of the process that reads the data from Elasticsearch. Each snapshot contains data collected from all NSM agents, with each MIB object in the snapshot having samples from each agent. As such, the snapshot delay Δ_{sh} increases as the number of agents to aggregate increases. The impact observed on Δ_{sh} when increasing the number of agents, with three polled MIBs, is given in Figure 5.6. Also

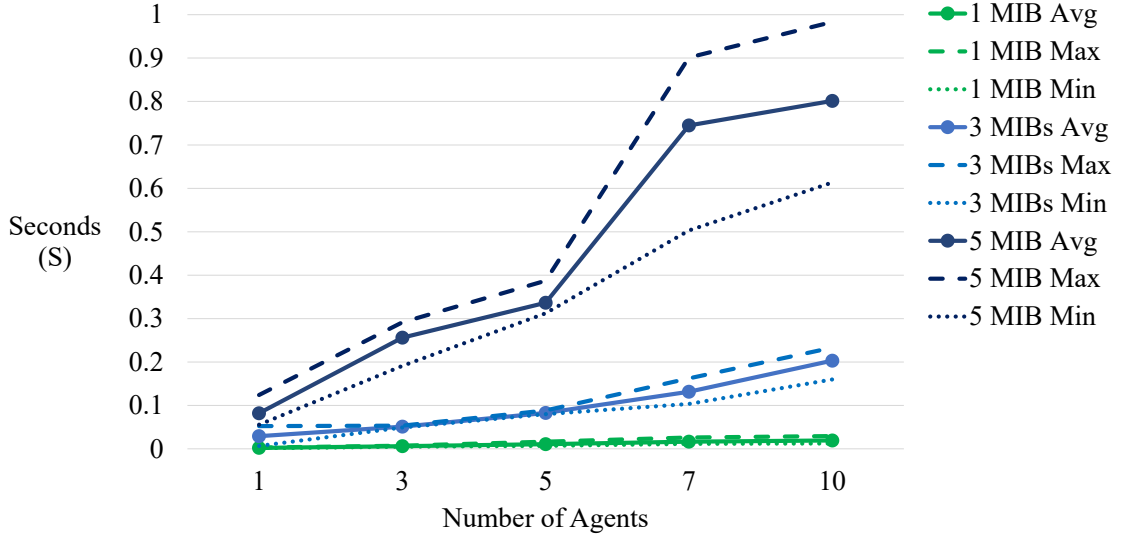


Figure 5.5: SNMP delay as a function of number of agents and MIBs requested.

shown for comparison purposes is SNMP delay Δ_{snmp} as the number of agents increases.

As for Δ_D , the impact of increasing data sizes can mostly be mitigated by predicting the next snapshot in parallel with the polling process such that the prediction of the next snapshot is already computed once the next snapshot is constructed. This leaves the computation of the prediction error for each feature and checking the errors against the corresponding thresholds. Such computation scales linearly with the number of features, but is nonetheless expected to be fast compared to Δ_{snmp} and Δ_{sh} . Running the anomaly detection in parallel can be done by having the detection computations performed on a separate host machine within the microgrid network. This is the case for the microgrid co-simulation testbed presented in this thesis, where the anomaly detection computations are performed on a workstation. As a result, Δ_D is mostly impacted by the time needed by the workstation to read the next snapshot from the appropriate Elasticsearch index over the network.

Overall, these performance results suggest that, should the microgrid monitoring scope be too large, there may be a need to make adjustments to the proposed active monitoring architecture. For example, multiple NSM managers, each responsible for a subset of agents, can be deployed within the microgrid management network. These NSM managers can be part of a hierarchy where they simultaneously act as agents for a manager at a higher level.

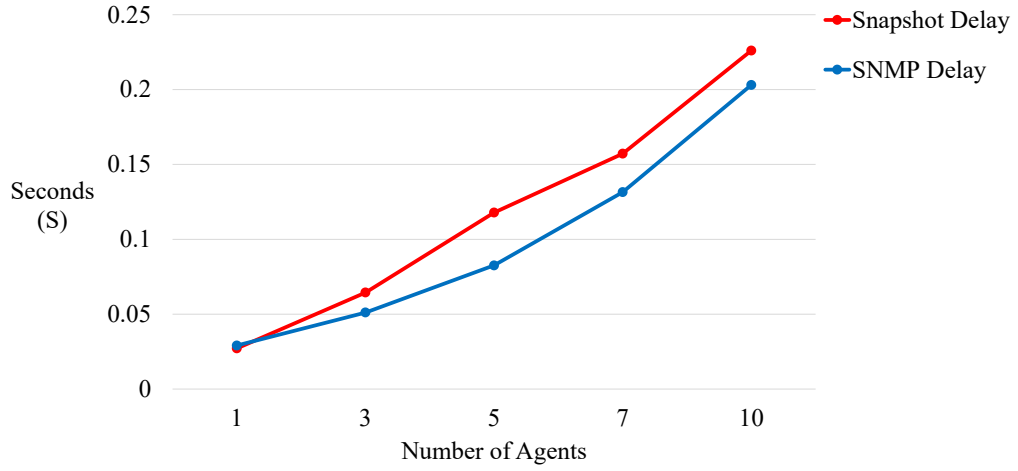


Figure 5.6: SNMP and snapshot creation delay as a function of number of agents.

5.2.4. Prediction Model Training for NSM Anomaly Detection

The proposed anomaly detection for NSM data uses a DNN trained on data collected during normal microgrid operation. The result of the training is a model that takes a sliding window of NSM data snapshots as input and yields a prediction for each feature in the next snapshot as output. Regarding feature selection, the set of features used to evaluate the DNN models for NSM is mostly taken from the IEC-104 MIB. Specifically, the following four MIB objects are used for the training and evaluation:

- tCInterPDUTime - Average time between received IEC-104 packets
- tCPDURTT - Average IEC-104 packet round trip time
- tCRxPdu - Number of IEC-104 packets received
- tCTxPdu - Number of IEC-104 packets transmitted

These MIBs are chosen by hand so as to minimize the redundancy of the feature set. They are collected from nine agents, which adds up to 36 features taken directly from MIBs object values. These agents includes the ones deployed for the PCC, the three ESSs, the three loads, the wind farm, and the MGCC. The cross-correlation of these 36 features is shown in Fig. 5.7. The highest

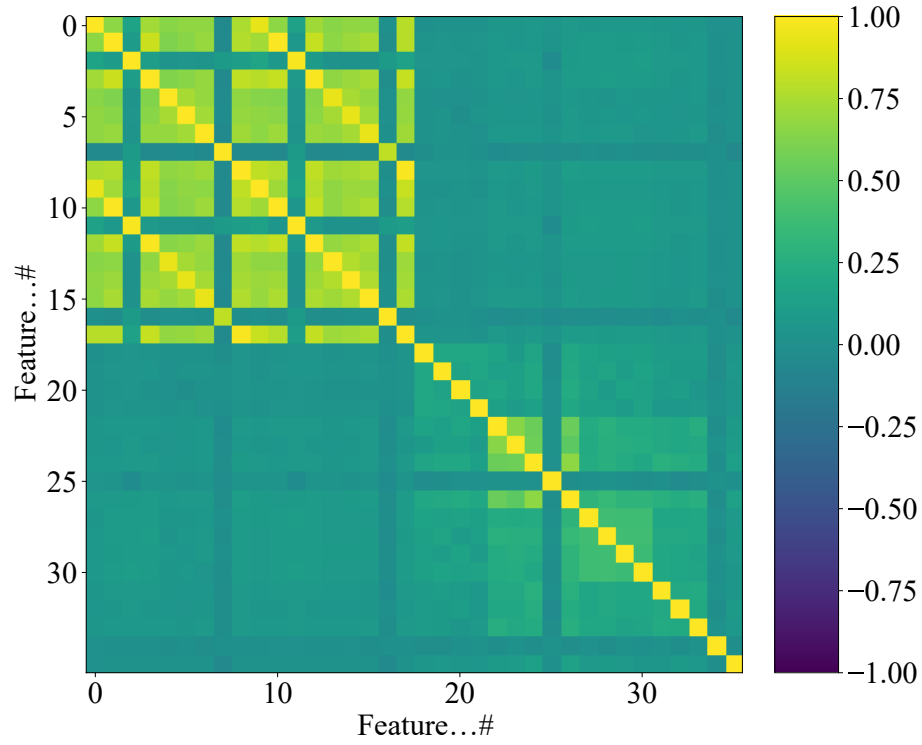


Figure 5.7: Standard linear correlation coefficient across all features.

cross-correlation is seen in the rates of packets being received and transmitted by the DERs (ESS, wind farm) and the MGCC. The cross-correlation map suggests the possibility of further reducing the number of features used for the proposed DNN prediction models without lowering the anomaly detection accuracy. Additionally, two other features are added to the input feature set that are based on the other features: (i) the difference between the MGCC packet transmission rate and the total rate of packets received by the IEC-104 servers, and (ii) the difference between the total packet transmission rate of the IEC-104 servers and the rate of packets received by the MGCC. These two features are used to measure packet losses within the IEC-104 communication. Adding these features gives a total of 38 features used by the DNN models for NSM anomaly detection that are presented in this thesis.

A dataset of NSM data is collected from the microgrid co-simulation testbed under normal operating conditions in the absence of attacks. This dataset is arranged as a contiguous sequence of 4093 snapshot samples. Each snapshot contains the 38 NSM features described earlier, ordered by increasing timestamp. The dataset is split into (i) a training set containing 2/3 of the full dataset,

and (ii) a validation set containing the remaining samples. The training and validation datasets contain 2729 samples and 1364 samples, respectively. The training and validation sets maintain a contiguous ordering of samples, which is necessary for training the models on sliding windows.

The DNN prediction models are implemented in Python using the Keras module for building neural networks. The DNNs are trained to minimize the mean absolute error of the output with respect to the snapshot features to predict. The training weights are updated using the gradient descent backpropagation of the training loss. Regarding the training hyperparameters, the learning rate and decay are fixed at 0.001 and 0.0001, respectively. The output layer uses linear activation for its nodes, meaning it outputs the results of its node calculations as is. A grid search approach is used to find the best combination of training batch size, number of epochs, and window size for various DNN architectures. After the training phase, the models undergo a validation phase. The validation phase uses the validation set to find the mean absolute error μ_j , standard deviation of the error σ_j , and maximum absolute error mae_j for node o_j in the output layer after the final training epoch. The average of the mean absolute error (μ), standard deviation (σ), and maximum absolute error (mae) across all output nodes is then calculated for comparison purposes.

The training results for the LSTM and GRU architectures with a single hidden layer, two hidden layers, and three hidden layers are presented in Table 5.2, Table 5.3, and Table 5.4, respectively. For brevity, the training results for the batch size of 256 is omitted for the two hidden layer and three hidden layer models. For all the models, the lowest maximum absolute error was found for a combination of hyperparameters with a batch size of 128. Highlighted are the models with the lowest μ , as these are the models chosen as candidates for evaluating the NSM anomaly detection on attack samples. The most noticeable trends are the decrease in μ as the batch size decreases and as the number of training epochs increases.

Samples are labeled as anomalies if the prediction error for any of the snapshot features exceeds the threshold set for that feature during the validation phase. A common approach for setting the threshold is to use 3-sigma [111], which would mean setting the threshold Th_j to $\mu_i + 3\sigma_j$ for a given output node o_j . However, this thesis instead sets the threshold Th_j to be equal to mae_j multiplied by a constant factor. This is due to the presence of numerous outlier data points found in the dataset of normal operation data that greatly exceed 3-sigma. Basing the threshold on mae_j

Table 5.2: Validation loss on single hidden layer neural network models

Model	Batch Size	Epoch	Window Size N	μ	σ	Max Absolute Error (mae)
1 Hidden Layer LSTM	128	50	10	0.0638	0.0638	0.493
	256	50	10	0.0717	0.0656	0.5026
	128	100	10	0.062	0.0633	0.4793
	256	100	10	0.0703	0.0622	0.4815
	128	50	25	0.0638	0.0637	0.4883
	256	50	25	0.0641	0.0631	0.4775
	128	100	25	0.0631	0.062	0.4772
	256	100	25	0.0646	0.0629	0.4829
	128	50	50	0.0697	0.0639	0.486
	256	50	50	0.0735	0.0638	0.4866
	128	100	50	0.0648	0.0626	0.4781
	256	100	50	0.062	0.0635	0.4858
1 Hidden Layer GRU	128	50	10	0.0652	0.0633	0.4938
	256	50	10	0.0668	0.0636	0.4903
	128	100	10	0.0584	0.0609	0.4723
	256	100	10	0.0619	0.0612	0.4723
	128	50	25	0.0645	0.0635	0.4853
	256	50	25	0.0684	0.0622	0.4797
	128	100	25	0.0589	0.0616	0.4722
	256	100	25	0.0695	0.0606	0.4812
	128	50	50	0.0611	0.0626	0.4779
	256	50	50	0.0802	0.063	0.4892
	128	100	50	0.0605	0.0604	0.4764
	256	100	50	0.066	0.0617	0.4797

results in fewer false positives.

5.2.5. Simulating Microgrid Attacks of Interest for Active Monitoring

Three different attack scenarios (Attack I, Attack II, Attack III) are briefly presented here to demonstrate the impact that cyberattacks against IEC-104 traffic can have on the stability of the microgrid. The microgrid is operating in islanded mode during these attacks. It uses the droop control scheme for microgrid frequency during Attack I and Attack II, but uses the centralized control scheme for Attack III.

In Attack I, the threat actor establishes a Man-in-the-Middle (MitM) connection between the PCC supervisory controller and the MGCC. The MitM connection is used to intercept the IEC-104

Table 5.3: Validation loss on two hidden layer neural network models

Model	Batch Size	Epoch	Window Size N	μ	σ	Max Absolute Error (mae)
2 Hidden Layer LSTM	128	50	10	0.0692	0.0668	0.5171
	128	100	10	0.0655	0.0658	0.5065
	128	50	25	0.066	0.0656	0.5065
	128	100	25	0.0656	0.0656	0.4951
	128	50	50	0.0693	0.0671	0.5139
	128	100	50	0.0679	0.0671	0.5097
2 Hidden Layer GRU	128	50	10	0.0692	0.066	0.5134
	128	100	10	0.0652	0.0652	0.5086
	128	50	25	0.0731	0.0628	0.481
	128	100	25	0.063	0.0641	0.5025
	128	50	50	0.0643	0.0647	0.4979
	128	100	50	0.0626	0.0632	0.4934

Table 5.4: Validation loss on three hidden layer neural network models

Model	Batch Size	Epoch	Window Size N	μ	σ	Max Absolute Error (mae)
3 Hidden Layer LSTM	128	50	10	0.0664	0.0675	0.5123
	128	100	10	0.067	0.0671	0.5172
	128	50	25	0.0695	0.0675	0.5221
	128	100	25	0.0689	0.0674	0.5191
	128	50	50	0.067	0.0674	0.505
	128	100	50	0.0664	0.0675	0.5151
3 Hidden Layer GRU	128	50	10	0.0679	0.0663	0.5055
	128	100	10	0.0666	0.0652	0.498
	128	50	25	0.0621	0.0635	0.4973
	128	100	25	0.063	0.0652	0.5057
	128	50	50	0.0675	0.066	0.5119
	128	100	50	0.0718	0.0632	0.4953

packets that contain the microgrid frequency measurements. The MitM automatically modifies the frequency measurement in the intercepted packets to a very low value, then forwards the modified packet to the MGCC. Such modification is straightforward due to the lack of integrity protection measures implemented in IEC-104. In response to the low frequency values being reported, the MGCC compensates by greatly increasing the power output setpoints sent to the ESSs so that they can inject extra power in an attempt to stabilize the microgrid frequency. The MGCC will also initiate load shedding by sending load consumption setpoints to the dynamic loads. The actual

microgrid power system will be driven to a state of overfrequency as the ESSs inject excessive power while the load consumption is simultaneously diminished. As shown in Figure 5.8, the frequency quickly reaches 66 Hz, at which point the microgrid frequency relay will send a system-wide signal to disconnect the loads and DERs. This disconnection of the microgrid is done to help avoid physical damage to the power equipment. The effect of the load shedding prior to the full microgrid disconnection is shown in Figure 5.9.

Attack II is similar to Attack I. Again, a MitM is established between the PCC and the MGCC that intercepts the frequency measurement packets, then modifies the reported frequency measurements before forwarding the packets to the MGCC. However, Attack II only reduces the reported microgrid frequency by 4.0 Hz. The MGCC responds by raising the ESSs power setpoints up until the actual microgrid reaches 64 Hz, which corresponds to when the MGCC observes the microgrid as having returned to 60 Hz. A frequency of 64 Hz lies above the acceptable frequency specified in IEEE.1547 for an islanded microgrid [42]. As such, the frequency relay will disconnect the microgrid after some time. Unlike in Attack I, this attack does not lead to load shedding being performed prior to the microgrid disconnection, due to the microgrid appearing to have stabilized at the MGCC. The frequency of the microgrid as Attack II progresses is shown in Figure 5.10.

In Attack III, the threat actor is able to delay the arrival time of the frequency measurement packets sent from the PCC to the MGCC without making any modifications to the packet contents. Specifically, the threat actor delays the IEC-104 packets containing the frequency measurements by 0.5 seconds. This leads to the MGCC calculating the ESSs power setpoints on consistently outdated measurements. As a result, the microgrid frequency begins to oscillate, with the MGCC continuously failing to stabilize the frequency while the delay attack is maintained. The frequency oscillation is shown in Figure 5.11. Note that this attack can be mitigated by having the microgrid operate with a droop control scheme, where local frequency control counteracts the destabilizing setpoints of the MGCC that stem from the delay attack. This, however, is not an option if the microgrid in question lacks the equipment and configuration to perform droop control.

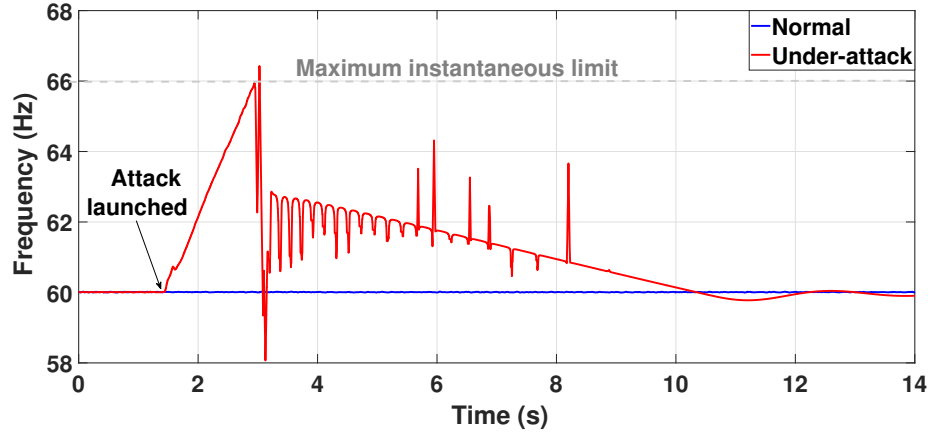


Figure 5.8: Frequency waveform during Attack I.

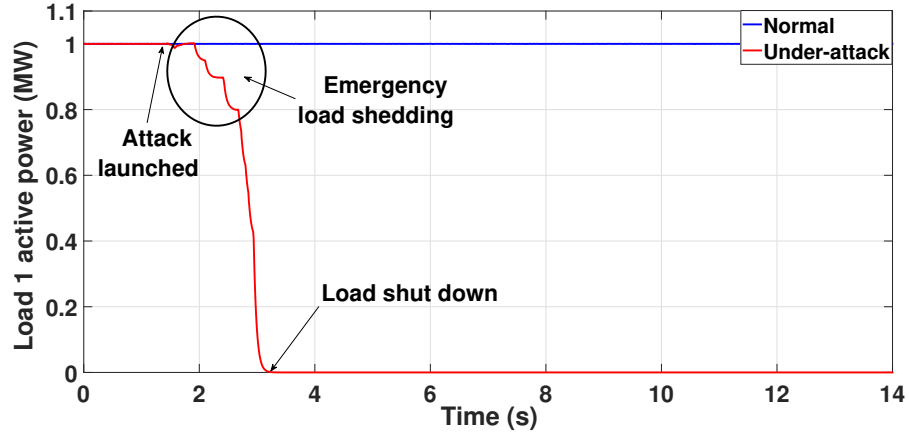


Figure 5.9: Active power consumption of load 1 during Attack I.

5.2.6. Prediction Model Comparison

The effectiveness of the various DNN models trained for NSM anomaly detection is evaluated on a testing dataset containing NSM data during both normal operation and attacked operation. The testing dataset contains 1796 snapshots of NSM data collected from the microgrid co-simulation testbed. Within the dataset are samples of normal operation data mixed with samples that reflect the impact of packet delay, packet dropping, and packet payload corruption. In this context, packet corruption refers to changing the payload in such a way as to cause an error in the application layer of the IEC-104. In the case of a packet drop or a packet corruption error, the IEC-104 TCP connection is lost and the client must reconnect with the server to resume IEC-104 communication. The MGCC attempts to reconnect automatically when the connection is lost.

The precision, recall, and F1 score of the trained LSTM and GRU models, all trained for 100

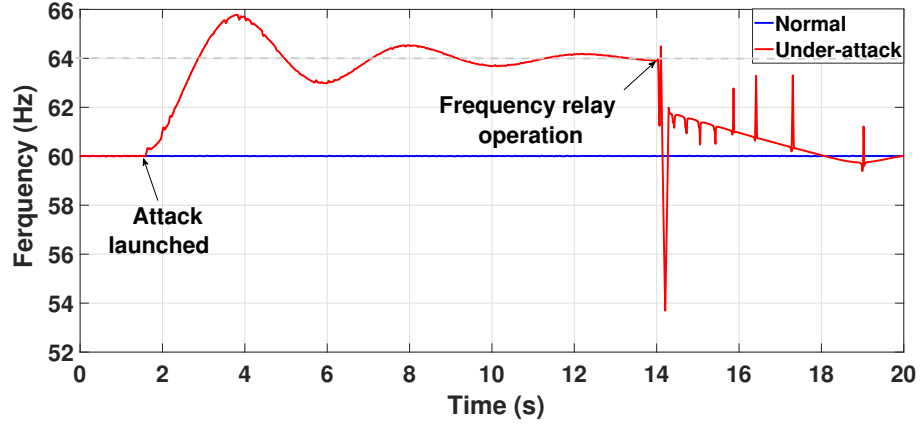


Figure 5.10: Microgrid frequency during Attack II.

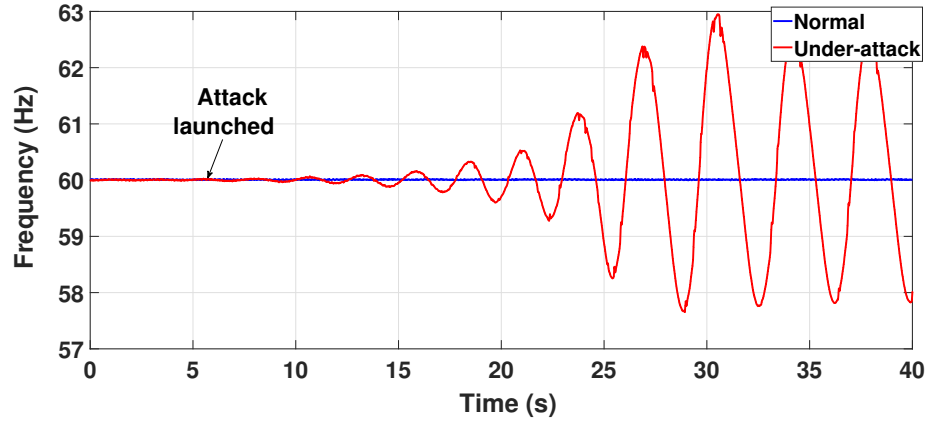


Figure 5.11: Frequency of microgrid during Attack III.

epochs and with batch size of 128, is given in Table 5.5. The results are separated across two selections for the threshold Th_j for the prediction error of output o_j . The threshold selections are (i) $Th_j = 1.5 * mae_j$ and (ii) $Th_j = 1.35 * mae_j$. The sensitivity of the anomaly detection increases as the scaling factor between the threshold and the mean absolute error decreases. The results show that the DNN models using two hidden layers and three hidden layers (3 layer) have better F1 scores than the single hidden layer models. The single hidden layer GRU has a notably poor F1 score relative to all the other models. Increasing the sliding input window size leads to an increase in F1 score for the model with three hidden layers, but a decrease in F1 score for the ones with two hidden layers. The models with more layers require more computational time. However, the computation of the predicted snapshot for a given sliding window can be performed while waiting for the next snapshot to which the prediction is compared. Thus, the size of the DNN does not

Table 5.5: Comparison between different DNN prediction models

Model	Window Size N	$Th_j = 1.5 \times mae_j$			$Th_j = 1.355 \times mae_j$		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
1 Hidden Layer LSTM	10	0.9816	0.9521	0.9666	0.9752	0.9621	0.9686
	25	0.9802	0.9354	0.9573	0.9761	0.9555	0.9657
	50	0.9826	0.9454	0.9637	0.9773	0.9577	0.9674
2 Hidden Layer LSTM	10	0.9786	0.9666	0.9725	0.9732	0.9710	0.9721
	25	0.9796	0.9621	0.9708	0.9742	0.9688	0.9715
	50	0.9795	0.9599	0.9696	0.9764	0.9666	0.9715
3 Hidden Layer LSTM	10	0.9786	0.9666	0.9725	0.9689	0.9722	0.9705
	25	0.9786	0.9655	0.9720	0.9732	0.9710	0.9721
	50	0.9797	0.9655	0.9725	0.9743	0.9710	0.9727
1 Hidden Layer GRU	10	0.9785	0.9120	0.9441	0.9785	0.9120	0.9441
	25	0.9506	0.9009	0.9251	0.9506	0.9009	0.9251
	50	0.9176	0.9176	0.9176	0.9176	0.9176	0.9176
2 Hidden Layer GRU	10	0.9795	0.9577	0.9685	0.9795	0.9577	0.9685
	25	0.9750	0.9543	0.9645	0.9750	0.9543	0.9645
	50	0.9783	0.9555	0.9668	0.9783	0.9555	0.9668
3 Hidden Layer GRU	10	0.9829	0.9621	0.9724	0.9829	0.9621	0.9724
	25	0.9807	0.9621	0.9713	0.9807	0.9621	0.9713
	50	0.9786	0.9666	0.9725	0.9786	0.9666	0.9725

impact the timeliness of the anomaly detection results, assuming that the model computation time is significantly smaller than the NSM polling duration. This is a realistic assumption if the polling duration is in the order of seconds while the prediction computation time is less than a second, as is the case in the proposed NSM active monitoring platform.

In the interest of comparing the detection effectiveness of the proposed anomaly detection models to existing classifiers, separate testing datasets containing labeled attack data snapshots are generated from the microgrid co-simulation testbed. Each of these testing datasets have a total of 500 samples, with 2/3 of each dataset containing labeled snapshots for normal operation and the rest of the dataset containing labeled attack snapshots. There are three such datasets:

- (1) A dataset containing snapshots collected during a packet delay attack on the frequency measurements sent by the PCC.
- (2) A dataset containing snapshots collected during a packet delay attack on the ESSs setpoint

commands sent by the MGCC

- (3) A dataset containing snapshots collected during a packet dropping attack on the ESSs setpoint commands sent by the MGCC.

A major advantage of the proposed anomaly detection method is that its DNN models do not need attack samples to be trained. They can effectively distinguish between normal and attack samples even in the absence of attack data in their training set. This is an important feature, as the amount of labeled attack data for the OT traffic of a given microgrid system is expected to be low and not very diverse. Even so, for evaluation purposes, it is assumed that there is a usable dataset of labeled attack data for a particular attack. Specifically, a training dataset of 1000 labeled snapshot samples is prepared from testbed data, with 1/3 of the samples being labeled as attacks and which reflect the impacts of a frequency delay attack. This dataset is used to train a number of commonly used classifiers associated with the WEKA workbench [71]. The selection of the WEKA classifier models for comparison is inspired by the work of Hodo et al. [69], who previously evaluated the effectiveness of these models at detecting attacks on IEC-104 messages. The specific set of tested classifiers include MLP, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM). As previously mentioned, the MLP has $N \times M$ nodes in its input layer. The KNN uses $K=5$ for the number of nearest neighbours. The Decision Tree and Random Forest separate data according to Gini impurity, with the Random Forest using 100 trees. The SVM uses the Radial Basis Function (RBF) kernel with $C = 1$ and $\text{Gamma} = 1/(M + \sigma^2)$, where σ^2 is the dataset variance. The WEKA classifiers are implemented in Python and leverage the scikit-learn library. Also, the classifier inputs are normalized to fall between 0 and 1 before being passed into the model, as this improved the classifier performance overall. Table 5.6 shows the precision and recall of these classifiers compared to the LSTM with two hidden layers and trained over 100 epochs and with a batch size of 128, whose detection results are summarized earlier in Table 5.5.

As expected, the classifiers are much more effective at detecting frequency delay attacks than the attacks on the ESS setpoints due to having seen frequency delay attacks during training. Even so, the LSTM with window size $W = 10$ has higher precision and recall than most of the classifiers. The classifiers struggle greatly to detect the ESS setpoint attacks, as these are novel attacks from

Table 5.6: Comparison between classifiers and anomaly-based approach

Model	Frequency Delay		ESS Setpoint Delay		ESS Setpoint Drop	
	Precision	Recall	Precision	Recall	Precision	Recall
LSTM (W=10)	0.9986	0.9848	0.9670	0.9832	0.9514	0.9832
LSTM (W=25)	0.9985	0.9661	0.9667	0.9721	0.9565	0.9832
LSTM (W=50)	0.9984	0.9443	0.9718	0.9609	0.9565	0.9832
MLP	0.9953	0.9772	0.0063	0.0112	0.089	0.095
KNN	1.0	0.9819	0.0057	0.0112	0.5241	0.4246
Decision Tree	0.9954	0.9795	0.1970	0.4469	0.4944	0.9944
Random Forest	1.0	0.9821	0.0057	0.0112	0.9834	0.9944
SVM	0.9956	0.9912	0.0112	0.0223	0.7028	0.9777

their perspective due to not being trained on data reflecting their impact. A notable exception is the Random Forest, which outperforms even the LSTM models at detecting ESS setpoint drop attacks. The shortcomings of the Random Forest are nevertheless exposed when considering its near-zero precision and recall for detecting ESS setpoint delay attacks. Overall, it is evident that the classifiers cannot be relied upon to detect attacks that have not yet been demonstrated on the microgrid data used to train them.

5.3. Passive Monitoring of Power Measurement Data

This section presents a novel proposal for passive monitoring of power measurement data and the experimental results when using the passive monitoring data for detecting FDI that spoofs fault conditions. The FDI detection uses DNN prediction models that are designed and trained in largely the same way as the prediction models used for anomaly detection in NSM data. The passive monitoring models differ in their input data and in how the model outputs are used to identify attack conditions. A key difference between the active and passive monitoring presented in this thesis is that the passive monitoring is performed directly on the microgrid OT traffic responsible for microgrid protection and control functions. The passive monitoring is designed to avoid introducing any noticeable delay in the transit time of the traffic. This is in contrast to the active monitoring that collects management data that is generated specifically for the purpose of NSM and is not used by the microgrid to meet its functional objectives.

5.3.1. Attack Model

The passive monitoring solution presented here is focused on protecting the microgrid from FDI attacks that invoke a line trip command from the protection relay. This attack is achieved by compromising a Merging Unit (MU) to perform a replay-based FDI attack that presents line faults conditions to the relay. The relay detects faults by checking if there are current spikes or voltage drops reaching a level that is indicative of a fault. An FDI that convincingly presents faulted voltage or current measurements will cause the relay to respond as it does during a real fault, i.e., it would send a line trip command to the Circuit Breaker (CB) to disconnect the line. This attack is referred to as a fault spoofing attack within the context of this thesis. The outcome of this attack is a non-faulted microgrid line needlessly being disconnected from the microgrid. Alternatively, an FDI attack on MU measurements can mask the presence of an actual fault by replaying safe line conditions, which endangers the line equipment. A relay that observes its own line measurements without having visibility of the rest of the microgrid cannot reasonably distinguish between a real fault and an FDI attack that closely matches actual fault behavior.

5.3.2. Passive Monitoring Setup

The passive monitoring proposed in this thesis reads the instantaneous current and voltage reported by the MU for the 3-phase AC microgrid and infers power quality measurements from the current and voltage curves. The power quality measurements are then used as features for machine learning models that make decisions on whether or not the measurements are abnormal. A central security manager assesses the extent of abnormality across different locations of the microgrid. If it detects a discrepancy, the central manager will intervene to prevent an unwanted line disconnection. The IEEE 1547 standard [42] specifies a minimum Fault Ride Through (FRT) time of 0.16 seconds for islanded microgrids before it can begin to disconnect lines. The goal of the proposed passive monitoring solution is to be able to detect a fault spoofing attack and respond to prevent the relay from disconnecting the line within the minimum 0.16 seconds allotted for FRT.

5.3.2.1 *Passive Monitors*

Each DER and load line contains a local MU and protection relay that share an Ethernet network alongside the line CB. The MU broadcasts IEC 61850 SV frames that contain the instantaneous voltage and current measurements at a rate of 4,800 frames per second. More specifically, each frame contains values representing the instantaneous current and the instantaneous voltage for each of the three microgrid phases. This makes a total of six relevant fields contained in each frame. The protection relay reads the stream of IEC 61850 SV frames and acts to protect the line if it reads unsafe measurements. In the context of IEC 61850 SV, the MU and protection relay take the role of publisher and subscriber, respectively.

The proposed microgrid passive monitoring infrastructure involves deploying a number of passive monitors across the microgrid. Each monitor is placed in one of the local Ethernet networks shared by the MU, protection relay, and CB. The local Ethernet networks each have a switch that forwards the traffic sent by the local devices to their intended destination. The passive monitors connect to this switch, and the switch mirrors the IEC 61850 SV stream published by the MU so that copies of the packets arrive at both the protection relay and the passive monitor.

The passive monitors preprocess the current and voltage measurements contained in the IEC 61850 SV stream to convert the readings into a sliding window of contiguous power quality snapshots, as is done for the NSM data. The sliding window is passed into a DNN prediction model, deployed on the passive monitor, that computes the prediction error for each feature.

5.3.2.2 *Central Security Manager*

The security manager is a device that connects to the microgrid passive monitors over an IP network. The prediction errors calculated by the passive monitors for each power quality snapshot feature are arranged in a vector L of size M , where M is the number of features in the power quality snapshots. The monitors encode the values of L into Abstract Syntax Notation One (ASN.1) [112] and transmit the encoding to the central security manager over UDP.

The central security manager compares the loss vectors sent by the passive monitors to determine if they are in agreement on whether the microgrid is behaving abnormally. If a passive monitor

is suspected of being subject to FDI that is misrepresenting the microgrid as being under fault conditions, the central manager reaches out to the corresponding protection relay to override the line trip response that would otherwise result from the FDI. Figure 5.12 shows the overall design of the passive monitoring infrastructure and fault spoofing countermeasure. It presents an infrastructure with k passive monitors, and depicts the arrival of loss vector L_i for the i^{th} monitor.

5.3.3. Passive Monitoring on the Microgrid Testbed

The data used to train and evaluate the anomaly detection performed by the passive monitoring is collected from the microgrid co-simulation testbed. This involves establishing a closed loop between I/O ports on the OPAL-RT DRTS that has IEC 61850 SV frames passing through the OpenStack emulated network and having a VM sniff the frames and save them to trace files. First, the MUs to monitor are modified within the HYPERSIM microgrid simulation model to transmit IEC 61850 SV through a specific I/O port on the DRTS. That port is connected to a physical Cisco switch that assigns it a VLAN and passes it through the VLAN trunk to the OpenStack network. The VLAN assigned to the I/O port used by the simulated MUs causes the IEC 61850 SV stream to be forwarded to an OpenStack subnet to which the sniffer VM is connected. The sniffer uses a custom Python script to read the IEC 61850 SV frames and save them to a trace file. The trace file contains the streams of multiple MUs, and so is split into smaller trace files according to MAC address in order to have all the frames from each MU be saved to its own trace file. After reading the frames, the sniffer performs its second role of being a network bridge. It forwards the frames through a network interface connected to a subnet that is separate from the one used to sniff them. This subnet connects to a VLAN that allows the frames to be forwarded back to the OPAL-RT simulator through an I/O port configured as the receiver for the IEC 61850 SV frames, closing the loop. The received frames are read by the protection relays in the HYPERSIM model that correspond to the MUs whose measurements are being recorded.

Not every line and DER is equally sensitive to microgrid faults. For example, the power dynamics of the ESS 2 DER line do not change much in the event of a line fault in the simulation. By contrast, the ESS 3 line power dynamics are much more sensitive to faults. This is shown in Figure 5.13, which plots how a simulated line-to-ground fault on the shunt compensation line impacts

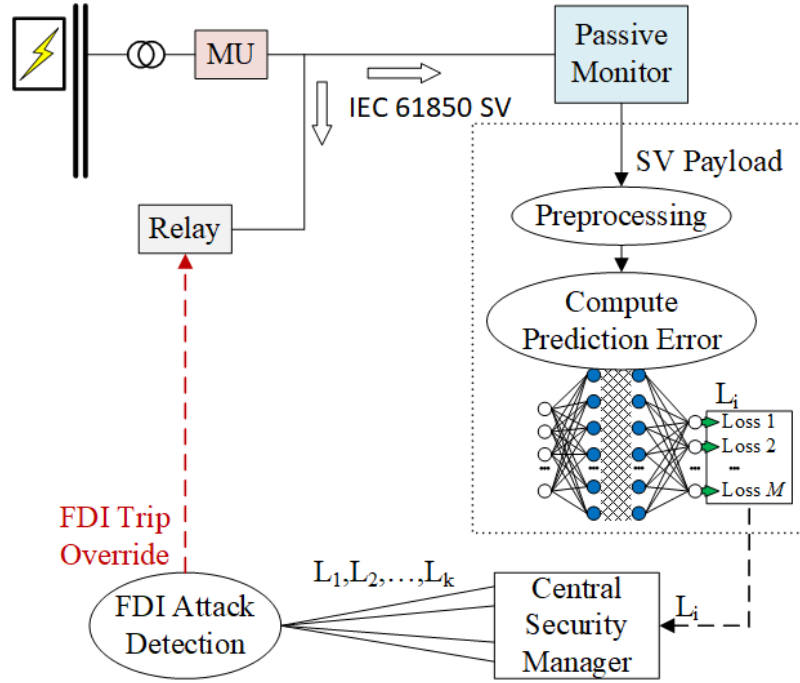
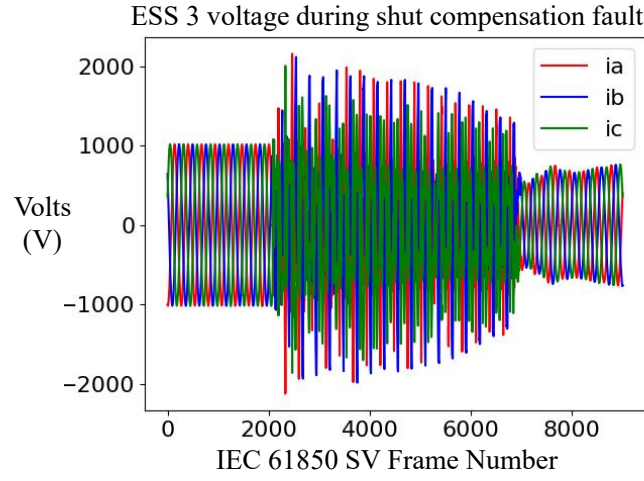


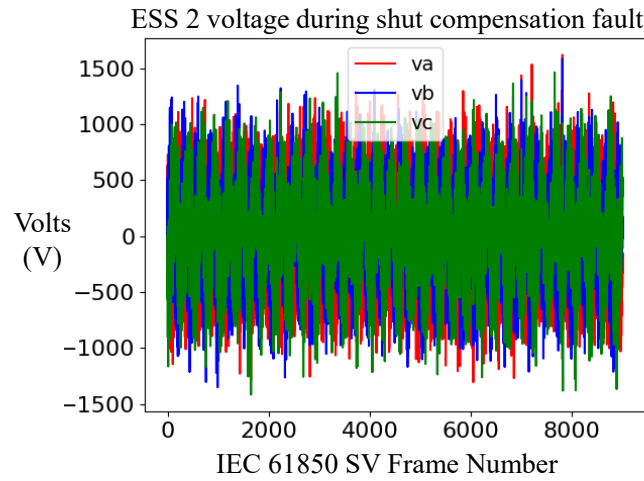
Figure 5.12: Design of the FDI attack detection and response system

the voltage on the ESS 3 and ESS 2 lines at the low voltage side of their transformers. The plots span a range of 9,600 IEC 61850 SV frames, which corresponds to a time range of two seconds. As can be seen for ESS 3 in Figure 5.13-a, the fault occurs between the 2000th and 7000th frame. Yet, no discernible effects are seen at ESS 2 in Figure 5.13-b during that time. The anomaly detection for power measurement data will be most accurate when there is a clear distinction between the fault and non-fault behavior on the monitored lines. ESS 3 is thus a better choice for passive monitor deployment than ESS 2.

An empirical approach is used to find the best DERs and load lines on which to deploy passive monitors. This approach involves adding fault locations to many different buses on the HYPERSIM microgrid model. The faults at these different locations are triggered one at a time, with their impacts on the MU measurements being observed. After simulating over a dozen faults across the microgrid model, it is observed that the MUs at the CHP, ESS 3, Load 2, and Load 3 lines are the most sensitive to faults. As such, these lines are chosen as the locations where passive monitors are deployed. Figure 5.14 shows the placement of passive monitors at these four locations relative to the rest of the modeled microgrid. Figure 5.15 demonstrates the sensitivity of the chosen passive



(a)



(b)

Figure 5.13: Effect of shunt compensation fault at ESS 3 and ESS 2.

monitor locations by plotting how the reported voltage at those four locations changes during a line-to-ground fault near the PCC. Specifically, Figure 5.15-a, Figure 5.15-b, Figure 5.15-c, and Figure 5.15-d show the three-phase voltage during the fault at the CHP, ESS 3, Load 2, and Load 3, respectively. The plots cover the impact across 4,800 IEC 61850 SV frames produced in one second and have a fault commence at around the 1400th frame. Note that the impact of the fault as observed at Load 2 and Load 3 is nearly identical due to them both being connected directly to the high voltage bus of the microgrid having the same power consumption (1 MW).

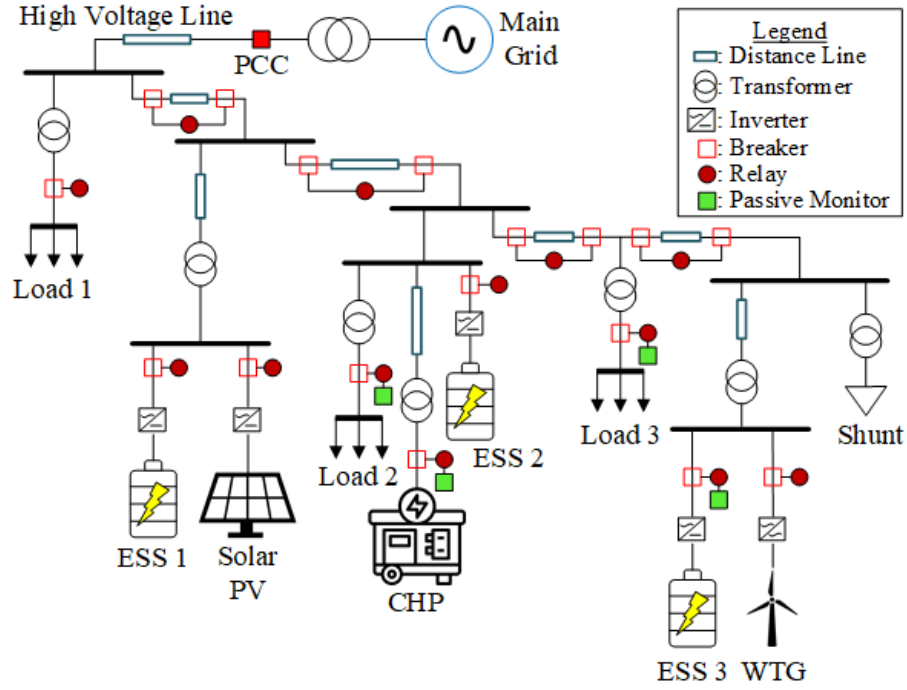


Figure 5.14: Placement of the passive monitors and relays in the microgrid.

5.3.4. Data Model

The instantaneous current and voltage values follow a sinusoidal (sine) waveform. The oscillatory waveforms for instantaneous AC current and voltage make their values not ideal as machine learning features intended for anomaly classification. The proposed passive monitoring solution preprocesses the instantaneous current and voltage curves to infer power quality measurements that remain largely stable under normal operation of the microgrid and thus are better suited as machine learning features. A custom Python script is used to implement a power quality feature extractor that reads the raw current and voltage values and produces a snapshot of desired power quality features. These features include the following:

- The Root Mean Square (RMS) of the voltage phases
- The sum of the voltage phases
- The active power

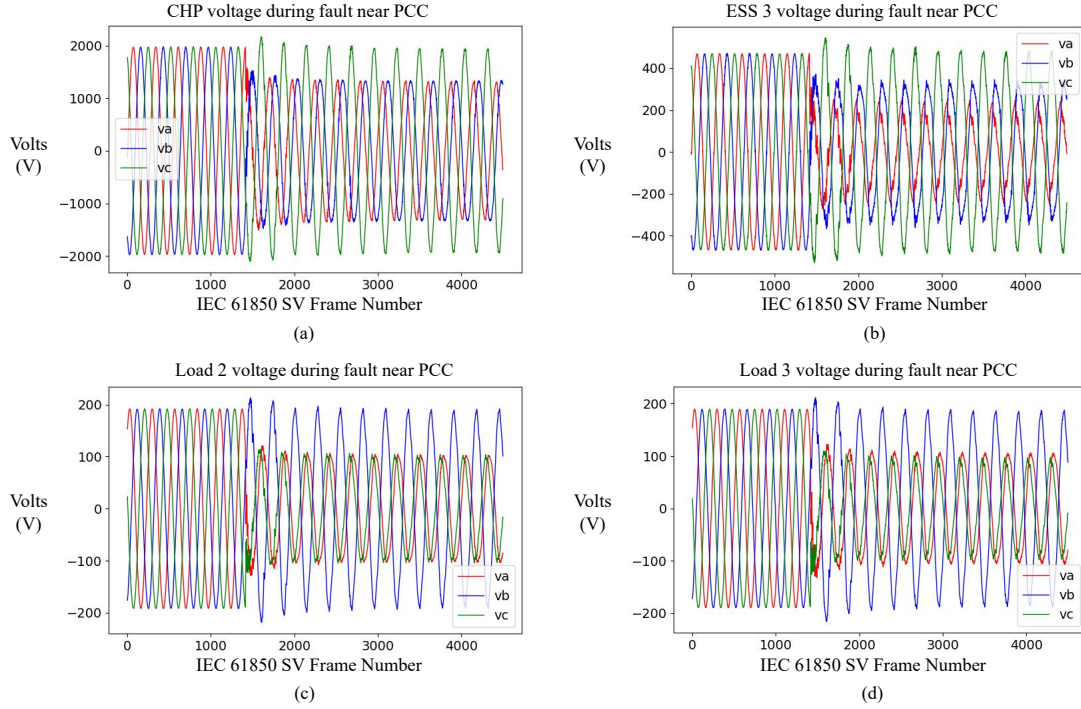


Figure 5.15: Effect of a fault near the PCC at various microgrid locations.

- The frequency of the voltage phases
- The phase angle difference between each pair of voltage phases
- The symmetrical components of the voltage phases (zero, positive, negative)

The RMS voltage, voltage sum, and active power can be calculated with just the latest sample of instantaneous voltage and current readings. However, the estimation of the other features requires that the extractor maintain a short-term memory of previous values and waveform characteristics. Measuring the voltage frequency is done by measuring the time taken between the last two zero voltage crossings that occurred on a voltage phase, which represents the half cycle time $T/2$ for that phase. The frequency is then computed as $f = 1/T$. The frequency measurements are updated every time any of the three phases has a zero voltage crossing. Zero crossings caused by jitter and harmonics are accounted for by ignoring any frequency estimations that exceed 200 Hz, which is much higher than the frequency that can be reasonably reached while also being low enough to be exceeded by the observed jitter on the testbed. The zero crossings for current and voltage phases are

also when their respective amplitudes are updated, with the amplitude set as the highest magnitude measured for that phase between the zero crossings. The last known amplitude of the voltage phase is used to estimate its present phase angle as per (4.3) and (4.4) previously given in Section 4.1.1.1. The phase angles are then compared against each other to find the phase angle difference between each pair of phases. Under normal conditions, the differences should all be close to 120° .

The symmetrical components are an alternative representation of three-phase vectors in the complex plane that are designed to quantify the amount of imbalance between the three phases [113]. There are three symmetrical components to calculate: (i) the zero component, (ii) the positive component, and (iii) the negative component. The zero component is a single vector, while the positive and negative components both represent a triad of vectors of equal magnitude that are spaced exactly 120° apart. The difference between the positive and negative components is that the positive component rotates counterclockwise on the complex plane in accordance with the ABC phase sequence, while the negative component rotates clockwise in accordance with the ACB sequence. The sum of the three components yields the original complex plane vector representation of the three phases. Under ideal three-phase conditions, the positive component has magnitude equal to the voltage magnitude while the magnitudes of the negative and zero components are near zero. Higher magnitudes for the negative and zero components indicate larger imbalances between the three phases of the original signal.

The zero (Z_c), positive (P_c), and negative (N_c) components of the voltage can be computed from instantaneous voltage phasors V_a , V_b , and V_c via linear transformation that applies the phasor rotation operators $\alpha = 1.0 \angle 120^\circ$ and $\alpha^2 = 1.0 \angle 240^\circ$ on them. To apply the rotation on the voltage phasors, both the rotation operators and voltage phasors are expressed as the sum of their real and imaginary parts. These sums are then multiplied together to get the rotated phasor. The rotation operators are converted to real and imaginary components as follows, with j denoting the imaginary part ($j^2 = -1$):

$$\alpha = \cos(120^\circ) + j * \sin(120^\circ) = -0.5000 + 0.866025j \quad (5.1)$$

$$\alpha^2 = \cos(240^\circ) + j * \sin(240^\circ) = -0.5000 - 0.866025j \quad (5.2)$$

The symmetrical components of the voltage are computed by multiplying the voltage phasors with the rotation operators α and α^2 as per the following matrix equation [113]:

$$\begin{bmatrix} Z_c \\ P_c \\ N_c \end{bmatrix} = 1/3 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (5.3)$$

After performing the needed calculations on the IEC 61850 SV payloads, the extractor generates a snapshot of the desired power quality features corresponding to a particular timestamp. When run on an Intel i7-10750H processor with GeForce RTX 2060 and a 2x2.6 GHz processor, the extractor is able to perform the feature extraction calculation an order of magnitude faster than the time between SV frames, meaning that the extractor runs fast enough to create snapshots for each SV frame. The sampling rate can, however, be reduced to suit the performance capacity of the devices available to a given system.

As with the active monitoring, the snapshots generated through passive monitoring are arranged by increasing timestamp to create a sliding window of time series inputs. The time series input is passed through a DNN to make a prediction on the incoming power quality measurements. The prediction is compared with the measurement features of the actual snapshot and the prediction errors for each feature are compiled into loss vector L . The loss vectors are sent to the central security manager as soon as they are ready. However, the passive monitors do not give any labels to the snapshots. Rather, it is the role of the central security manager to determine if the prediction errors of the passive monitors, when considered collectively, indicate the presence of an FDI attack.

5.3.5. Prediction Model Training for Power Quality Prediction Error

The passive monitors use LSTM models to predict the values of the next power quality snapshot. These models are trained on a dataset containing a total of 141,000 IEC 61850 SV samples collected

from the CHP, ESS 3, Load 2, and Load 3 lines that are simulated in the co-simulation testbed for the islanded microgrid. A separate LSTM model is trained for each of the lines, to be used by the passive monitor responsible for monitoring the line. The dataset contains samples collected from a range of acceptable operating conditions with respect to the microgrid voltage frequency. The acceptable frequencies in the dataset range from 59.7 Hz to 61.0 Hz, which are within the acceptable bounds specified in IEEE 1547 for the islanded microgrid. The dataset is split into a training set and a validation set. The training set contains 2/3 of the total samples while the validation set contains the remaining samples. Because the use case of mounting an FDI response within 0.16 seconds is time sensitive, the proposed LSTM models, along with the tested MLP models, use no more than one hidden layer. Considering that the MLP model with one hidden layer does not meet the definition of DNN, the LSTM and MLP models used on passive monitoring data are henceforth referred to collectively as ANNs. As with the models used for NSM anomaly detection, the hidden layer for the passive monitoring anomaly detection models have twice as many nodes as there are input features. The models use Linear activation for their output layers. The ANN models are trained for 100 epochs. As for their other hyperparameters, the ANN models have their learning rate set to 0.001, their decay set to 0.0001, and apply a hidden layer dropout of 0.2. A sliding window size $N = 5$ is used for the time series input of power quality snapshots.

5.3.6. Anomaly Classification

Upon receiving a snapshot prediction loss vector, the central security manager processes the vector in order to assign it a normal (o), mildly anomalous (m), or highly anomalous (h) classification. This classification is based on whether any of the snapshot features exceed their corresponding anomaly thresholds. The anomaly threshold for the combination of a given passive monitor i and feature j is a function of the maximum absolute error mae_{ij} found during the validation phase of the trained ANNs.

For loss vector L_i of passive monitor i with values $[l_{i1}, l_{i2}, \dots, l_{iM}]$ corresponding to M features, the classification c_i of L_i is determined using the following equation:

$$c_i = \begin{cases} h & \text{if } \exists j \in [1, M] \cap \mathbb{N} \mid l_{ij} \geq 2 \times mae_{ij} \\ m & \text{if } \exists j \in [1, M] \cap \mathbb{N} \mid l_{ij} \geq mae_{ij} \text{ and } \forall j \in [1, M] \cap \mathbb{N}, l_{ij} < 2 \times mae_{ij} \\ o & \text{otherwise} \end{cases} \quad (5.4)$$

The central security manager organizes the loss vectors into collections according to arrival time. Each collection contains one vector from each passive monitor. The classifications of the vectors in the collection are counted. Based on the counts for each anomaly class relative to each other, the central manager decides whether or not to label the whole collection as an FDI attack. The collection is labeled as a fault spoofing FDI attack if the following two conditions are true:

- (1) There is at least one vector classified as h ; and
- (2) There are more o classifications than h or m classifications combined.

If the combined number of h or m classifications exceeds the number of o classifications, then the collection is treated as presenting legitimate fault conditions. The existence of the m classification allows for monitors to be sensitive to smaller fluctuations that may occur at locations that are distant from a legitimate fault, while at the same time not treating the fluctuations as a fault if they appear when the rest of the microgrid is operating normally i.e, there are no highly anomalous (h) readings anywhere on the microgrid.

The success of this detection approach is reliant on the assumption that an FDI attack becomes much more difficult with each additional passive monitor location that needs to be compromised by the threat actor. The approach breaks down if the threat actor compromises at least half of the passive monitor locations, regardless of whether the actor modifies the IEC 61850 SV payloads or the prediction losses reported by the passive monitor. To note is that the FRT response that follows from a fault spoofing FDI can cause legitimate fluctuations in the monitored power lines that may lead to their prediction losses being classified as anomalous. Therefore, it is important that the central manager be capable of discerning a fault spoofing FDI attack near the very beginning of its execution. Otherwise, it may mistake the impact of the FRT on the non-attacked monitor readings

as indicators of the observed fault conditions being genuine.

5.3.7. Simulated Fault Spoofing FDI Attack Scenario

This thesis research considers the scenario where a threat actor has compromised the MU at ESS 3 and is capable of reading its measurements and modifying its output according to its observations. The threat actor is assumed to have observed a real line-to-ground fault, originating from the PCC, with the compromised MU. Based on the real fault behavior, the attacker crafts an extrapolated sequence of instantaneous current and voltage measurements that the ESS 3 protection relay will interpret as a fault and trip the line CB after some time. In accordance with the minimum fault clearing time specified in IEEE 1547, the relay will wait at least 0.16 seconds before sending a trip command to the CB in the event that the fault does not clear within that time.

Figure 5.16 demonstrates the impact of a fault spoofing FDI attack from the MU at ESS 3. At 0.7s, the compromised MU replays the PCC fault conditions to the protection relay, as shown in Figure 5.16-a. During the fault clearing window, the ESS 3 controller will respond by initiating FRT measures, injecting reactive power into the microgrid. The effect of the FRT response is presented in Figure 5.16-b and Figure 5.16-c corresponding to the MU measurements at the CHP and ESS 3, respectively. Shortly after the fault clearing window has passed, the ESS 3 protection relay trips the line to disconnect it from the microgrid. This results in the complete loss of electric current coming from ESS 3, as seen in Figure 5.16-d.

To simulate the FDI attack on the testbed, the OpenStack bridge VM script that forwards the IEC 61850 SV frames from the MUs to the relays is modified to intercept and modify the outgoing ESS 3 SV frames to contain fault readings in their payload. The attack injects false measurements at a time where no discontinuity between the legitimate measurements and the false measurements can be observed. To ensure no injection discontinuity, the attack script has its injection payload begin with a voltage phase that has just risen above the zero crossing point (phase angle ≈ 0). It then waits until the legitimate measurements pass the zero crossing on that phase to begin the injection of the modified payloads.

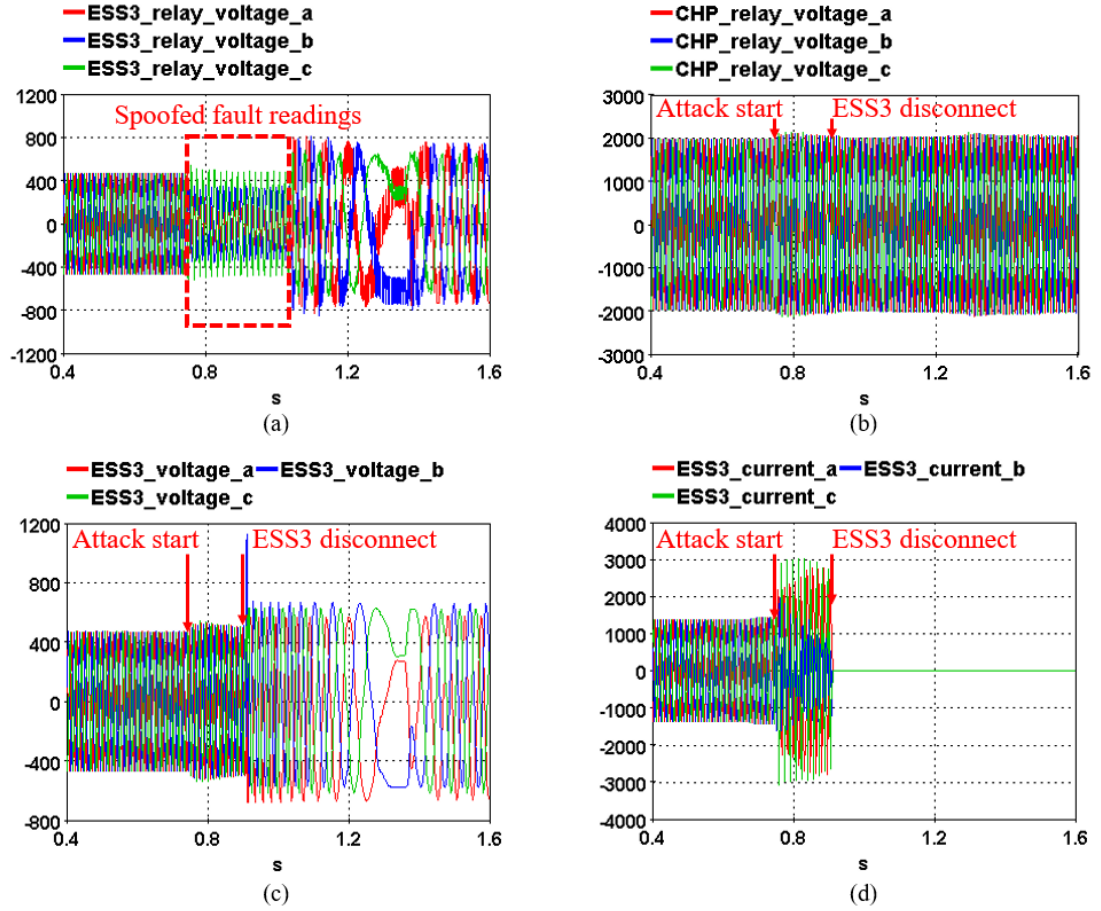


Figure 5.16: Impact of FDI attack on microgrid power line.

5.3.8. Evaluation of FDI Attack Detection

While the LSTM model is effective at detecting fault spoofing FDI attacks, it incurs more overhead than competing machine learning methods. A testing dataset containing fault spoofing FDI attacks is generated using the microgrid co-simulation testbed to evaluate the LSTM model against competing models, notably the MLP. The testing dataset contains 13,236 samples collected during normal operation and 240 samples collected while simulating a fault spoofing attack from the MU at ESS 3. The LSTM and MLP are also compared with supervised learning models, namely the Decision Tree and the SVM. Practically speaking, supervised learning models will struggle to be effective for fault spoofing use cases due to the many ways that a fault can impact the power system according to its location and which phases are affected. The ANNs are trained on only normal operation data, which is much more readily accessible. Even so, the supervised learning models

are included in the evaluation to evaluate their detection success under the ideal circumstances of being trained on the attack they are being tested on and to take note of their overall execution time. A separate training set is prepared for the supervised learning models. The supervised learning dataset adapts the training dataset used for the ANNs by adding 16,460 labeled samples for legitimate faults to the dataset. The legitimate fault samples include line-to-ground and line-to-line faults whose location and affected phase match samples found in the testing dataset.

Table 5.7 gives the precision, recall, F1 score, True Negative Rate (TNR), and average sample processing time for the proposed LSTM model and competing models on the testing dataset. Note that TNR refers to the proportion of negatives that are correctly labeled as negative. The tests are all performed on a i7-10750H processor with GeForce RTX 2060 and a 2x2.6 Ghz processor. The Decision Tree is trained to make decisions according to Gini impurity. It is also tested both with and without Principal Component Analysis (PCA) being applied for dimensionality reduction. The SVM is configured with $C = 1$ and $\text{Gamma} = 1/(M + \sigma^2)$, where M is the number of features and σ^2 is the dataset variance.

The result show that the LSTM has a substantially higher precision and especially recall than the MLP. However, its required computation time is over ten times larger than the MLP. Speaking of computation time, all the tested models take more time to compute than the time between IEC 61850 SV frames, with the exception of the non-PCA decision tree. This means that the proposed FDI attack detection approach described in this chapter should be designed to not have the prediction model evaluate every IEC 61850 SV sample. The LSTM model is found to have the best precision and best TNR, but the SVM has the best recall and F1 score. Overall, the results motivate the need for more work to be done to improve the detection accuracy and the execution time of the model proposed for FDI detection using power quality data.

Table 5.7: FDI attack detection results for different machine learning models

Model	Precision	Recall	F1 Score	TNR	Prediction Time (s)
LSTM 1-Layer	0.9868	0.9250	0.9549	0.9998	0.005200
MLP 1-Layer	0.9328	0.4754	0.6298	0.9993	0.000650
Decision Tree	0.9425	0.924	0.9332	0.9989	0.000060
Decision Tree with PCA	0.9560	0.9960	0.9756	0.9991	0.000540
SVM	0.9808	1.000	0.9903	0.9996	0.001150

Chapter Six

Threat Hunting

Threat hunting refers to the collection and correlation of system alerts and artifacts to gather context surrounding an ongoing attack [25]. These threat hunting activities are done proactively with the intent of detecting the presence of an attack before it causes significant impact. Threat hunting aims to produce attack hypotheses that accurately describe the extent of an ongoing attack campaign and the threat actor responsible for the attack, and in particular those perpetrated by Advanced Persistent Threats (APTs). It does so by leveraging recorded telemetry, system knowledge, and available Cyber Threat Intelligence (CTI). By correctly correlating the alerts triggered by the same attack campaign and the underlying threat actor, the system security operators gain valuable context that helps uncover the full scope of the system compromise, inform mitigation strategies and anticipate subsequent Tactics, Techniques, and Procedures (TTPs) and attack targets.

Though threat hunting is an ever-growing topic within enterprise networks, there are gaps when it comes to designing a threat hunting approach that is adapted for ICS networks such as the microgrid. Compared to enterprise networks, ICS networks have a much wider range of device types with their own functional roles. These devices also use OT protocols not typically found in enterprise systems. In the case of the microgrid, an unchecked APT can cause adverse impacts on the microgrid power system, such as line disconnections and power instability. Such consequences need to be properly taken into account when prioritizing threat hunting efforts. There is also a notable lack of automation within existing threat hunting operations when it comes to associating an attack hypothesis with the most likely APT behind the TTPs that make up the hypothesis. For the most part,

the association of suspicious activities to APTs is done manually by security operators [25]. When considering that threat hunting activities are being performed in an online fashion to expose ongoing attacks, integrating automation into the process can enable faster identification of the intruding APT, thus allowing for a response actions to be taken sooner.

This chapter proposes a novel threat hunting automation approach for threat hunting in the context of the microgrid. It refers to CTI and a kill chain model to associate security alerts with MITRE ATT&CK TTPs. These associations are made by utilizing a vector embedding model derived from a Knowledge Graph (KG) built from the CTI. The approach then correlates the alerts into attack hypotheses. Each hypothesis is assigned a severity related to the extent of the security threat it poses. Other activities of the proposed approach include filtering redundant hypotheses and identifying the APT that is responsible for the suspicious activity described in a given hypothesis. The research surrounding the proposed approach is supported by an APT attack model that is used to generate a dataset of TTP sequences on the microgrid. Although presented as a microgrid security solution, the proposed approach can be readily adapted to other ICS environments.

6.1. Microgrid Security Alerts

The automated threat hunting approach proposed in this thesis analyzes suspicious telemetry with consideration of the microgrid network and CTI for ICS systems. Before diving into the details of the threat hunting approach, it is important to understand the nature of the observable security alerts that are used as inputs to the proposed threat hunting approach. In the context of this thesis, security alerts encapsulate the telemetry that is analyzed for indications of suspicious activity. Security alerts can be generated by commercial solutions for IDS and Security Information and Event Management (SIEM), or through proprietary monitoring tools. This thesis assumes the presence of a dedicated security manager within the microgrid control centre, henceforth referred to as the threat manager, that is dedicated to correlating security alerts and engaging in threat hunting within the system.

6.1.1. Alert Characteristics

The suspicious activity may be reported as an event, or may have been detected by reading log data. In either case, an alert is generated to summarize the suspicious activity for use in threat hunting. The alert may originate from a device operating system, a network interface of a device, or a communication channel between two network interfaces. Each security alert contains the following information fields regarding the suspicious activity:

- **LogTimestamp:** Unix timestamp of when it occurred
- **ReceiveTimestamp:** Unix timestamp of when it was reported
- **LocationType:** whether its origin is a device, interface, or communication channel
- **LocationName:** name of the origin device/interface/channel
- **LocationID:** unique identifier of the origin device/interface/channel
- **HostID:** (for interfaces) unique identifier of the host device for the interface
- **PurdueLevel:** Purdue level of the device/interface/channel
- **AlertType:** either “event” or “log”, depending on the reporting method
- **ApplicationList:** list of applications involved (e.g., SSH)
- **DataComponent:** MITRE ATT&CK data component to which it is mapped
- **AlertID:** unique identifier of the alert itself

This information is used to guide threat hunting efforts and associate the alert with TTPs. The **DataComponent** field deserves special mention here. This field is used as a bridge between the system telemetry and the MITRE ATT&CK framework. It describes, at a high level, the means through which the suspicious activity is detected. The mapping between alert source and ATT&CK data component is assumed to have been manually established by the system security designer when integrating security monitoring solutions into the microgrid. In the case of a deployed ICS solution, for example, an ATT&CK data component would be defined for each possible alert it can generate.

6.1.2. Alerts from Active and Passive Monitoring

The active and passive monitoring platforms proposed in this thesis are among the detection sources that are expected to generate security alerts to use as input for the threat hunting process. For the hybrid rules-based and anomaly-based detection scheme for microgrid NSM data, anomalies within the NSM features would trigger a security alert and assign it to the appropriate data component. Table 6.1 presents the proposed mapping between the affected MIB object and the data component to assign to it. It may be appropriate to generate multiple alerts for some violated rules or anomalies in order to associate multiple data components with the single anomaly. As for anomalies in the passive monitoring of the power quality data, the $\langle$ Manipulation of View $\rangle$ technique can be assigned to any alerts generated by the passive monitoring scheme.

6.2. Overview of the Threat Hunting Approach

The security alerts that are collected by the threat manager initiate an ordered set of analysis steps. These steps result in sets of MITRE ATT&CK TTPs being assigned as potential causes of the alerts. The steps also include an update to the pool of active attack hypotheses under consideration when new alerts arrive. The proposed threat hunting approach has the following steps, as summarized in Figure 6.1:

- Group alerts that likely correspond to the same suspicious action by considering their network and temporal proximity;
- Match alerts groups to TTPs by leveraging the vector embedding of a KG constructed from CTI on ICS;
- Correlate alert groups into attack hypotheses according to their sequencing in the Unified Kill Chain [114] adapted for ICS; and
- Validate attack hypotheses with reference to severity criteria.

Table 6.1: MIB objects collected for microgrid active monitoring

Data Component	Anomalous MIB objects
⟨Network Traffic Flow⟩	tCInterPDUTime, tCPDURTT, tCRxCnt, tCRxPdu, tCTxPdu, tcpInSegs, tcpOutSegs
⟨Network Traffic Content⟩	tCRxCritical, tCRxUnsolicitedReq, tCTxCritical, tCTxUnsolicitedReq
⟨Masquerading⟩	tcpActiveOpens
⟨Denial of Service⟩	tCOutErrCnt, tCInErrCnt, tcpRetransSegs, tcpOutRsts, tcpInErrs

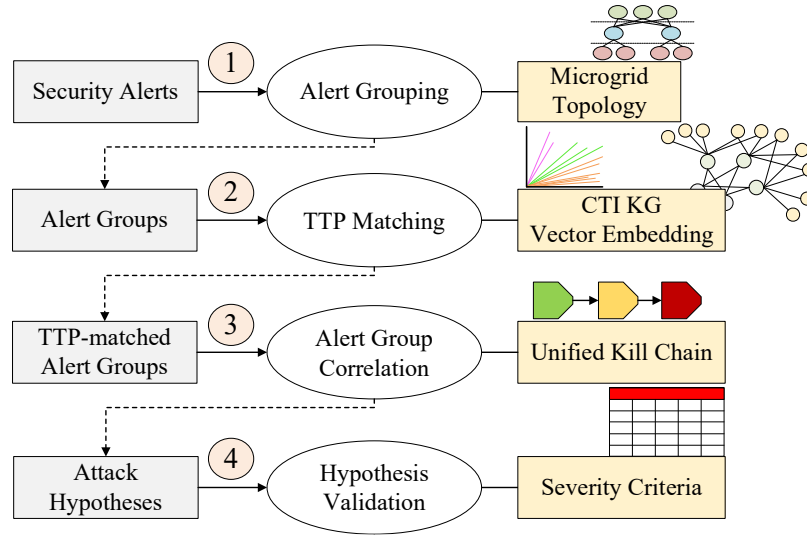


Figure 6.1: Summary of the threat hunting methodology.

6.3. Security Alert Grouping

A security alert analyzed in isolation can potentially provide valuable insights into which TTPs might have been performed by an APT. However, the accuracy of the association between alerts and TTPs can be greatly improved by enriching the context surrounding the analyzed alerts. One way to do this would be to have multiple alerts from different sources providing information pertaining to the same activity. For example, a computer login from an unusual user at an unexpected time can be noticed by both a passive network monitor and a host logging service, which together provide complementary contextual information surrounding the login. With this in mind, security alerts that are likely to be indicators of the same instance of suspicious activity are placed into a security alert

group. It is the security alert group as a whole that is analyzed and associated with TTPs. That is, all the alerts in the group become associated with the same set of TTPs as the outcome of the TTP matching process described later in this chapter. The resulting security alert groups are later correlated together in order to construct attack hypotheses.

Each batch of security alerts observed by the threat manager is placed into new or existing alert groups, as appropriate. This grouping follows an iterative process, where security alerts corresponding to activity that occurred in a relatively short time frame and which involve the same asset or assets are placed into the same group. This process is described in Algorithm 1. For an alert to be added to the group, there must be at least one other alert in the group that is both adjacent to it and whose *LogTimestamp* is within ΔT_{max} of the alert being considered. For the purposes of alert grouping, alerts are adjacent if they involve assets that are directly connected to one another, e.g., a network interface and a communication link connected to that interface. The value of ΔT_{max} is set to 5 seconds in order to be reasonably certain that alert groups falling outside that time window are indeed the result of separate system activities. However, this value is subject to change according to the needs of the particular system.

The security alerts corresponding to a particular activity in the network are not necessarily generated at the same time. Specifically, they may have a *ReceiveTimestamp* that is relatively far apart from its *LogTimestamp*. This occurs when the alert is generated as a result of investigations performed by threat hunting tools in response to previous alerts. Such alerts are treated as a new batch on which Algorithm 1 is applied to potentially add them to existing security alert groups. Any alerts added to a group that is already a part of a hypothesis will initiate a re-evaluation of the hypothesis with the newly added alerts taken into account.

An APT that has infiltrated an ICS will perform multiple attack steps in the pursuit of their ultimate objective. A defining characteristic of APTs is their patience when it comes to progressing through their attack steps. It is realistic to expect the APT attack campaign to last weeks or even months in order to adequately study the system network topology and traffic, as well as to reduce the chance of being detected. It is thus important that records of security alerts are maintained for a sufficient period of time in order to be considered for correlation with future alerts that result from the activities of the same APT attack campaign. The limit on how long a security alert remains

Algorithm 1 Alert grouping algorithm

Input: Security alerts $A = \{(l_1, t_1), (l_2, t_2), \dots, (l_n, t_n)\}$ **Output:** Alert groups G

```
 $G \leftarrow \emptyset$ 
for  $a = (l, t) \in A$  do
  for  $g \in G$  do
    if  $\exists a_g = (l_g, t_g) \in g \mid l$  adjacent  $l_g$  and  $|t - t_g| < \Delta t_{max}$  then
       $g \leftarrow g \cup \{a\}$ 
      set  $\langle a \text{ grouped} \rangle$ 
    end if
  end for
  if  $\langle a \text{ grouped} \rangle$  not set then
     $g^a \leftarrow \{a\}$ 
     $G \leftarrow G \cup \{g^a\}$ 
  end if
end for
```

in active memory for use in threat hunting before being archived or discarded is up to the policy set by the system management. Security alerts that remain in consideration for a longer period of time are more likely to be correctly correlated with all future activity belonging to the same attack campaign. This comes with the tradeoff of requiring more storage memory to keep active records of these alerts, as well as more computational resources needed to compare new security alerts with the larger number of alerts in active memory.

6.4. CTI Knowledge Graph

The proposed threat hunting approach makes use of CTI on ICS attacks for several threat hunting tasks. This includes the initial matching of security alerts to TTPs, the correlation of alert groups into attack hypothesis, and the validation of attack hypotheses. These tasks refer to a KG built from available CTI in order to facilitate their automation.

6.4.1. Construction of the KG

The KG used in this thesis contains CTI on entities of interest for cyber threat hunting. These entities include detection sources, TTPs, APTs, malware, and ICS assets, as described by the MITRE

ATT&CK framework. The purpose of the KG is to act as the core reference for finding which entities are most related to a given set of input entities. Specifically, the use case of interest is finding the TTPs that have the best correspondence with a set of one or more data components, with the data components originating from the alerts belonging to a security alert group. The KG represents each unique entity as a node and associations between two entities are represented by an edge that connects their respective nodes. The use of TTPs as a CTI source gives the proposed threat hunting approach the advantage of basing its analysis on generalized knowledge that can be applied to different microgrid systems regardless of the specific device models and firmware they have deployed. This is in contrast to threat detection approaches that use datasets containing specific alert attributes for training purposes.

The entities and associations used to build the KG are primarily sourced from the version of MITRE ATT&CK CTI the public repository released in November of 2023 by the MITRE Corporation [115]. This repository contains a collection of all the entities specified in the MITRE ATT&CK framework, dividing the entities into separate collections for the Enterprise, ICS, and mobile matrix domains. Furthermore, the collections also contain a set of relationships that each correspond to an association between two entities in the ATT&CK CTI. Within the repository, each entity and each relationship is described within its own JSON file using the STIX 2.0 format [116]. Though the repository of CTI for ICS is sizable, it cannot be said to be truly extensive. Notably, some APTs labeled as ICS threat actors by MITRE are also included in the enterprise matrix repository, where they have relationships to techniques that are, strangely, not included in the ICS matrix repository. A total of 33 relationships omitted from the ICS repository are present in the Enterprise repository are added to the set of relationships used to build the KG. With these corrections, a total of 1458 relationships are found between MITRE ATT&CK ICS entities.

The KG derived from the MITRE ATT&CK CTI to define its nodes and edges is a bipartite graph. This means that nodes in the graph have a class label, and no two nodes within the same class have a relationship with each other. The KG constructed from the MITRE ATT&CK entities and relationships has the following node classes: (i) Data Source, (ii) Data Component, (iii) Tactic, (iv) Technique, (v) APT, (vi) Malware, and (vii) Asset.

As shown previously in Figure 2.1, the technique entities have relationships with the entities of

all the other classes, with the exception of the data source class. Each data component has a relationship to exactly one data source, though a data source can have multiple data components. Together, the types of relationships found between these entity classes in the MITRE ATT&CK framework form the CTI ontology. Additional CTI sources that are distinct from MITRE ATT&CK can be incorporated into the KG provided that the entities they describe can be mapped to this ontology. One such CTI source is the APT campaign database published by Tizio et al. [117]. This database is a public repository that describes associations between TTPs and threat actors, namely APTs and malware. These associations are sourced from a collection of real security incident reports, which are all included in the repository for transparency. Within the APT campaign database are associations between APTs and ATT&CK techniques that are not found in the MITRE ATT&CK repository. Specifically, a total of 51 associations not found in the ATT&CK repository are found in the APT campaign database. These associations are added to the set of relationships used to build the KG. It is worth noting that the APT campaign database does not include TTPs or APTs that are not already defined in the MITRE ATT&CK framework. It is, however, not a universal requirement for added CTI sources to strictly contain the TTPs and threat actors already included in ATT&CK. Novel instances of the ontology entities can be added freely. For example, an organization may add associations between TTPs and a novel APT that has emerged and appears to exclusively target one's own system network.

Regarding the KG implementation, the attributes of the KG nodes and edges are first parsed and saved into Comma-Separated Value (CSV) files. For the nodes corresponding to a CTI entity, there are only two saved attributes: (i) the entity name, and (ii) the entity class. As for the edges, the only information of interest is the names of the two nodes connected by the edge. The CSV files describing the nodes and edges are then imported into a local Neo4j database [118], where the nodes and edges between entities can be visualized through the Neo4j query language. The APT campaign database of Tizio et al. [117] contains Python scripts that leverage the Neo4j module to automate the import process from CSV file sources. For the purpose of implementing the KG used in this thesis, these original scripts are adapted to import the node and edge attributes of interest from the source CSV files. After the KG has been uploaded to Neo4j, the nodes and edges are exported to a file in JSON string format using the Neo4j web interface. Finally, a custom Python

script is used to load the exported JSON strings, which together describe the full KG, into memory. Once in memory, the KG can readily be referenced by the scripts implementing the processes that comprise the proposed threat hunting approach. The Python objects representing the KG nodes and edges are also exported to Python pickle files, which are read on subsequent executions of the KG implementation script to speed up data loading times.

The cyber threat landscape is constantly evolving to employ novel exploits and evasive techniques. In turn, the body of CTI undergoes regular updates to reflect new behaviors that have been observed from APT and malware. Updates to the TTPs defined in the MITRE ATT&CK framework are infrequent compared to changes in malware hashes, malicious IP addresses, and domain names used by attackers. This is reflected in the Pyramid of Pain [23], with MITRE ATT&CK TTPs falling into the network/host artifact level. Still, the MITRE ATT&CK Matrix has received updates over time to reflect new knowledge of APT behavior. For example, the set of TTPs associated with the Sandworm Team APT expanded in light of the 2016 attack on the Ukrainian power system to which they are attributed. Notably, the TTPs they used during the 2016 Ukraine attack differed substantially from the 2015 Ukraine attack occurring one year prior that is also attributed to them. It is important to consider the logistics of how threat hunting solutions that leverage the MITRE ATT&CK framework adapt when the MITRE CTI is updated. The KG resulting from the methodology described in this chapter is expected to be available within a local database accessible through the microgrid control centre network by the threat manager. It is the responsibility of the security operations team to update this local database periodically with new CTI. The proposed update process for the KG can be broken down into the following steps: (i) import the newest version of the MITRE ATT&CK CTI repository, (ii) parse the contents of the updated repository into CSV files, (iii) reset the Neo4j database storing the KG, (iv) run the Neo4j import scripts to rebuild the KG in Neo4j with the new updates, (v) re-export the contents of the Neo4j database to JSON string files, and (vi) reload the exported contents into memory within Python. The KG implementation described here needs to be restarted each time a new version of the KG needs to be loaded. However, a live implementation of the proposed threat hunting approach may aim to minimize loss of availability when applying CTI updates. If so, the live implementation can instead be designed to read the updated KG directly from the Neo4j server. Such a design might benefit from having a

separate transitional Neo4j database that hosts the updated KG so that the KG can be rebuilt without tearing down the previous version.

6.5. Matching Alerts to TTPs

Recall that each security alert is associated with a data component and one or more assets according to its DataComponent and HostID fields, respectively. From these fields, the security alert is directly associated with up to one data component node in the CTI KG and with one or more asset nodes. From these nodes, associations between the alert and possible TTPs, specifically MITRE ATT&CK techniques and tactics, can be established. For example, security alerts generated by deep packet inspection tools can be mapped to the ⟨Network Traffic Content⟩ data component node. This node has edges to technique nodes relating to reconnaissance, malicious payload transfer, and unexpected commands. If the alert occurred on the link between the maintenance workstation and application server, then the asset nodes would be ⟨Workstation⟩ and ⟨Application Server⟩.

Rather than attempting to correlate security alerts to TTPs purely in isolation from other alerts, the security alert groups created through the method described in Section 6.3 are considered and matched to TTPs collectively. This serves to reduce the ambiguity that would emerge when attempting to use a single alert to infer the TTPs performed on a system.

6.5.1. Converting KG Nodes to Vector Embedding Space

To measure the similarity of nodes within the CTI KG, a vector embedding model for the entire KG is constructed in which the similarity of two KG nodes can be quantified. Through this model, each node in the KG is uniquely mapped to a vector of the same name in the common N -dimensional vector embedding space produced by the model. The vector embedding space is used to find the set of MITRE ATT&CK techniques and tactics that are most closely associated with the input combination of data components corresponding to a given security alert group.

The KG is converted to vector embedding space using the Node2vec algorithm [119]. Node2vec can be described as a variant of the Word2Vec embedding algorithm [120], originally intended for Natural Language Processing (NLP), that has been adapted for graphs. Essentially, the Node2vec

algorithm performs random walks on the KG to generate a dataset of sequences composed of the names of the nodes. The Word2vec algorithm is then applied on the dataset of node sequences. The only KG node features used by the algorithm are the name of the node (e.g., $\langle \text{Network Traffic Flow} \rangle$) and the edges it has to other nodes. Generating a dataset composed of random walks is a systematic way of uncovering associations and patterns in the edge paths of the knowledge graph that may not be immediately apparent upon human inspection. As parameters, the vector embedding space generated in this thesis through Node2vec has a dimensionality of 20, a random walk length of 16, and generates 100 random walks in total. The vector embedding space is implemented in Python and leverages the `node2vec` Python module.

Prior to transforming the ATT&CK KG with Node2vec, graph weights are assigned to the KG edges to emphasize the connections between nodes that have a relatively low number of associations to other nodes. For example, the $\langle \text{Network Traffic Content} \rangle$ data component is directly connected to many ATT&CK technique nodes, and thus it is hard to make solid conclusions about what specific attack techniques triggered a security alert with that data component. However, the $\langle \text{Firmware Modification} \rangle$ data component is only directly connected to attack techniques related to rootkits and modifications to system or module firmware. The calculated edge weights quantify the precision of each edge association by choosing an edge weighting algorithm that yields a lower edge weight between nodes with a higher number of connecting paths. That is to say, the edges of nodes with more neighbors are de-emphasized. To achieve this, the complement of the Semantic Connectivity Score (SCS) [121] is used to calculate the weights between each edge connecting a pair of nodes in the ATT&CK KG. The use of SCS is chosen here because it accounts for the length of the connected paths while being suitable for cases like the KG where the nodes themselves do not have weights. Given nodes n_1 and n_2 connected by an edge, the SCS and its complement SCS' for a given edge are calculated as follows:

$$SCS(n_1, n_2) = 1 - \frac{1}{1 + \sum_{l=1}^{\tau} \beta^l path^l(n_1, n_2)} \quad (6.1)$$

$$SCS'(n_1, n_2) = 1 - SCS(n_1, n_2)$$

Here, $path^l(n_1, n_2)$ is the number of connected paths of length l between n_1 and n_2 , τ is the

maximum path length considered, and β is the damping factor that determines how much the impact of a path on the SCS is reduced as path size l increases. The SCS equation uses $\tau = 3$ and $\beta = 0.5$ as parameter values. Preliminary experimental results show that this weighting approach ultimately leads to higher APT identification accuracy at the attack hypothesis validation stage compared to the vector embedding space generated from unweighted KG edges.

Cosine similarity is calculated between vectors in the embedding space to identify the techniques and tactics exhibiting the highest similarity to a set of data components. Cosine similarity is a measure of the closeness between two vectors. It has seen widespread use for the purposes of determining the relatedness between the vector embedding representations of two concepts due to its efficiency and effectiveness [122]. The total similarity score TSS_i for node i in the KG to the ATT&CK data component set $D = \{d_1, d_2, \dots, d_n\}$ of size n in the security alert group, affecting the set of ATT&CK ICS assets A , can be expressed as follows:

$$TSS_i = \sum_{j=1}^n \text{cosim}(v_i, v_{dj}) \quad (6.2)$$

Here, $\text{cosim}(v_i, v_{dj})$ is the cosine similarity between vector embedding v_i for node i and vector embedding v_{dj} for the node corresponding to the j th data component in the alert group. Figure 6.2 summarizes the steps taken to calculate TSS from the data components in D within the KG. The total similarity score for a given technique or tactic is equal to the sum of cosine similarities between it and all data components in the group. To help reduce false positives, techniques and tactics in the node similarity ranking are filtered out if they do not have a direct connection to all data components in D . In addition, techniques that do not target any assets in A in the KG are filtered out along with their corresponding tactics. The TTP matching process results in the security alert group being matched to the techniques and tactics with the highest total cosine similarity to each data component in that group. The techniques and tactics are ordered by decreasing total similarity score, and then the top K techniques and top K tactics are chosen as the best ones. Through experimentation, it is found that setting $K = 4$ yields suitable results for TTP matching. Regarding the implementation of TSS , a Python code base is used to apply *node2vec* on the KG and to calculate cosine similarity within the resulting KG vector embedding space. It defines a k_v parameter and constrains the

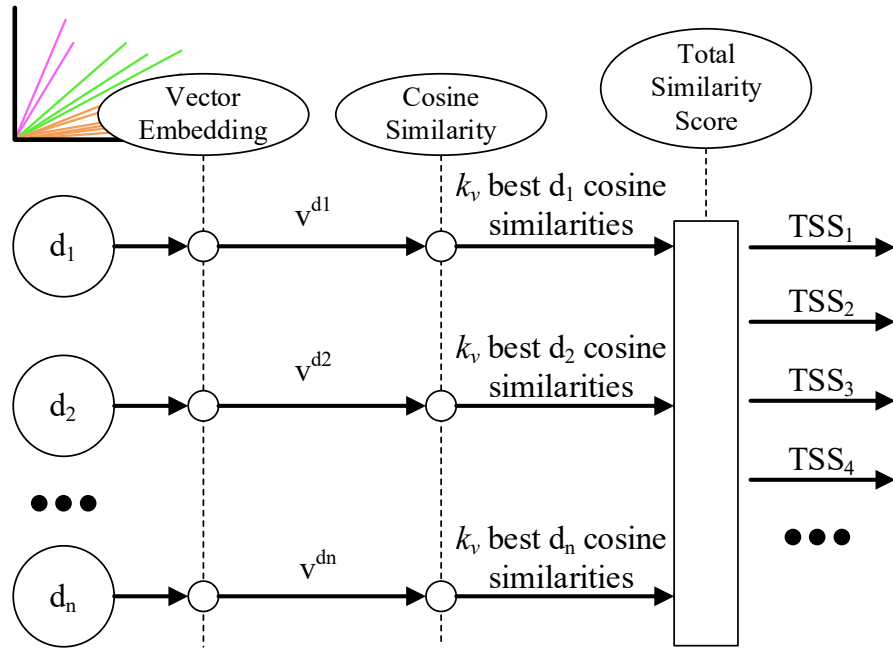


Figure 6.2: Steps taken to find total similarity score from data components.

techniques and tactics considered for a data component to the top k_v closest ones according to cosine similarity. The union of the nodes corresponding to the k_v techniques and tactics across the data components in D are treated as the set of candidates for which to compute TSS . For performance reasons, k_v is set to 10.

6.5.2. Success Rate of Vector Embedding

The proposed TTP matching method becomes increasingly accurate at finding the most relevant ATT&CK tactics and techniques as the number of distinct data components among the alerts in the alert group increases. This is explained by larger sets of data components having fewer techniques and tactics shared in common compared to groups with fewer data components. This increase in accuracy is demonstrated empirically by applying the proposed TTP method, based on the KG vector embedding, on unique combinations of D and A . Additionally, the effectiveness of the KG vector embedding for TTP matching is compared to a naïve method of TTP matching. The naïve method associates the data components and assets of an alert group to techniques and tactics solely by referring to the KG. This competing method, referred to here as the “Direct KG reference”

method, finds all techniques having a direct relationship with the KG data component nodes and asset nodes in the alert group. It is assumed that some data components corresponding to a technique used by an attacker may not be detected by an alert. Thus, the direct KG reference method does not exclude techniques that, in addition to the data components in the alert group, have a direct relationship to KG data components not found in the alert group.

The precision and recall of both the vector embedding and direct KG reference method when matching unique combinations of $D + A$ to techniques are given in Table 6.2, sorting by the size of D . When computing the precision and recall, a true positive is a technique that has a direct relationship with exclusively the data components in D that also has a relationship with an asset in A . Essentially, the precision gives insight on how much ambiguity remains on average when matching alert groups to techniques. It is worth noting that the direct KG reference method, by definition, cannot exclude the techniques that meet the true positive condition and thus has no false negatives. The results show that the vector embedding methods outperforms the direct KG reference method when it comes to precision, but comes with the risk of false negatives.

6.6. Attack Hypothesis Construction

An attack hypothesis represents a suspected attack by an APT. It is composed of security alert groups, each of which contains one or more alerts and one or more MITRE ATT&CK ICS assets, as shown in Figure 6.3. The alerts within an attack hypothesis, along with their corresponding TTP matches, are analyzed collectively to determine how similar they are to APT behavior patterns described in CTI and how severe the underlying threat might be. When a new security alert group is formed, it initializes a new active attack hypothesis that contains it. The new alert group is also checked against every active attack hypothesis to determine if the group is sufficiently correlated to one or more groups in the active hypothesis. If the alert group is correlated with alerts in the existing hypothesis, a new hypothesis is created by copying the existing hypothesis and then adding the new alert group to the copy. As such, all hypotheses with more than one alert group can be considered to be a child of a parent hypothesis, rather than being dynamic hypotheses that grow in size over time.

Table 6.2: Technique precision and recall for data component and asset combinations

Size of D	Unique Combinations of $D + A$	Average (and Total) Number of Techniques associated with $D + A$ in KG	Precision With Direct KG Reference	Precision With Vector Embedding (top $K = 4$)	Recall With Vector Embedding (Top $K = 4$)
1	42	1.3571 (57)	0.1090	0.3170	0.7600
2	45	1.6888 (76)	0.1641	0.2900	0.6178
3	35	1.0572 (37)	0.3895	0.5469	0.9487
4	79	1.1392 (90)	0.6667	0.7563	1.0
5	21	1.4762 (31)	0.8857	0.9394	1.0
6	10	1.0 (10)	1.0	1.0	1.0
7	16	1.0 (16)	1.0	1.0	1.0
8	12	1.0 (12)	1.0	1.0	1.0

6.6.1. Unified Kill Chain

An APT engages in different attack tactics over the course of their intrusion campaign on a target system, with the different tactics corresponding to different stages of the attack. Each stage may be separated by a large gap of inactivity that may make it difficult to link the different stages of their attack together based on simple temporal proximity rules. However, by matching alerts with tactics, it becomes possible to correlate alerts together based on whether they present a realistic and expected sequence of tactic, in addition to the source of the alerts in the system network.

The alert correlation method adopted for this thesis models the sequence of ICS ATT&CK tactics representing different attack stages according to the Unified Kill Chain (UKC) model [114]. The UKC is an attack model that innovates on both the MITRE ATT&CK framework and the Cyber Kill Chain® [82] originally developed by Lockheed Martin. It incorporates the ATT&CK tactics, which have no strict ordering within the ATT&CK framework, into an ordered sequence of tactics. This integrates the ATT&CK tactics into phases in a way that is comparable to the cyber kill chain. At the same time, the UKC assumes that an attacker may skip attack phases, contrary to the cyber kill chain that treats its phases as being obligatory stages of an advanced attack.

The UKC divides the tactics into three goal cycles: *In*, *Through*, and *Out*. As the name suggests, the *In* cycle contains the tactics that an attacker uses to infiltrate a system and establish a foothold. The *Through* cycle includes the tactics that an attacker uses to move laterally through

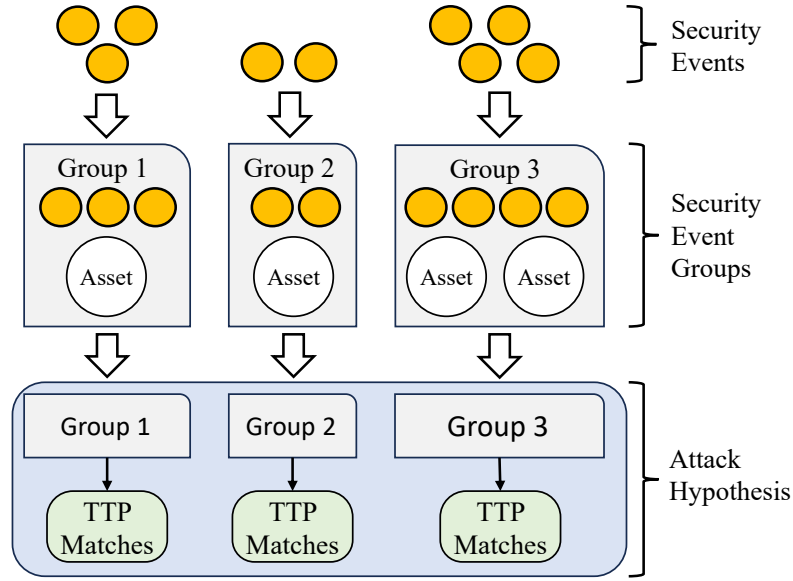


Figure 6.3: Composition of attack hypotheses from grouped security alerts.

the system to compromise the asset that allows the attacker to accomplish their end goal, referred to here as the “end goal asset”. The *Through* cycle can be repeated multiple times as needed to pass through intermediary assets between the initial entry point and the end goal asset. The *Out* cycle includes the tactics involved in harming the operations of the system, as well as collecting any system-specific operational data necessary to execute the final stages of the attack effectively.

We have made some minor adaptations to the original UKC so that it describes the sequence of MITRE ATT&CK ICS tactics rather than the tactics belonging to the ATT&CK Enterprise matrix. These adaptations result in the ordered tactic sequence shown in Figure 6.4, with the original UKC on the left and the modified version used for this thesis on the right. An illustrative example of UKC order distance is given in Figure 6.5. As shown, the UKC order distance between “Persistence” and “Discovery” is three, while the distance between “Persistence” and “Lateral Movement” is six.

6.6.2. Security Alert Group Correlation

Once a security alert group is matched to ATT&CK TTPs as described in Section 6.5, it is checked against each active attack hypothesis to determine if it meets the conditions to be combined with the hypothesis to create a child hypothesis. An alert group that meets the conditions is

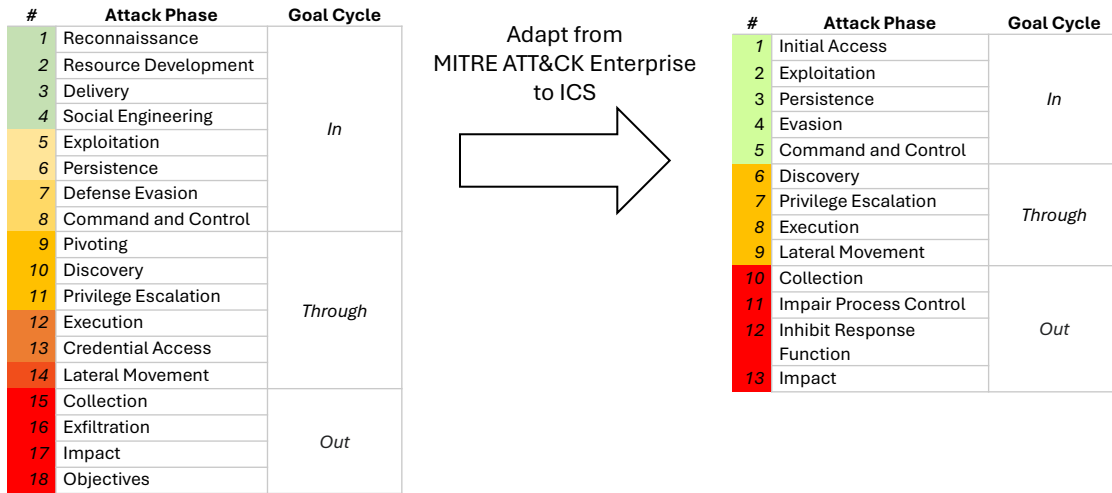


Figure 6.4: Adaptation of the Unified Kill Chain tactics from Enterprise to ICS.

considered to be “correlated” to the hypothesis. The aim is to have the alert group be correlated to a hypothesis in cases where it corresponds to a previously unobserved stage of the hypothesized attack campaign. To be correlated to a given hypothesis, the alert group must meet the following two conditions:

- (1) The alert group originates from an asset within the neighborhood of the hypothesis; and
- (2) The alert group a) includes an alert that involves the same data component as an alert in the hypothesis, b) is matched to a tactic that is within UKC range from a tactic matched to the most recent alert group already in the hypothesis, or c) is matched to a *Through* cycle tactic and originates from an asset that directly neighbors the hypothesis.

Condition (1) checks that the alert group corresponds to assets that are close enough to the hypothesis within the network to be feasibly reachable by an APT that has compromised the assets covered in the hypothesis. What constitutes being within the neighborhood of an attack hypothesis is controlled by the “network hop distance” parameter. Condition (2a) suggests that the APT may be reusing a previously successful attack technique, while Condition (2b) suggests that the APT has advanced to the next stage of their attack. Whether a tactic is within UKC range of another is controlled by the “UKC order distance limit” parameter. Meanwhile, Condition (2c) accounts for lateral movement done by the APT to reach either another intermediary asset or the end goal asset.

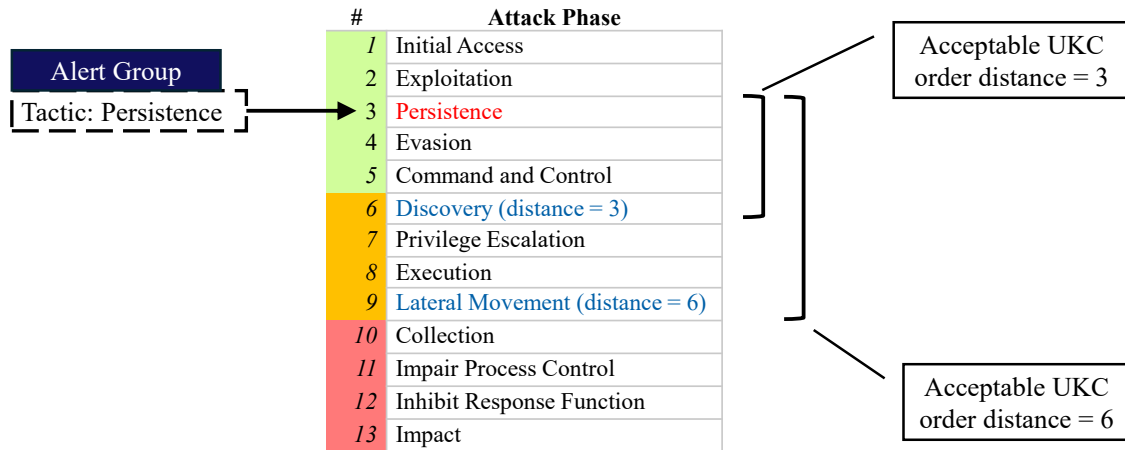


Figure 6.5: Example of UKC order distance from the Persistence tactic.

6.6.3. Considerations for Real-World Deployment

The proposed threat hunting approach helps reduce false positives by finding TTPs to which multiple types of security alerts converge, rather than analyzing alerts in isolation. However, the approach requires mapping between ATT&CK data components and security-related data reported to the SIEM by a Network Intrusion Detection System (NIDS) or host-based monitoring tools. Alerts generated within the microgrid system network can reasonably be mapped to ATT&CK data components if formatted in a structured way. One way to establish a clear alert structure would be to collect them through the SNMP or Syslog protocols, whose use is specified within the IEC 62351 suite of standards [48] for power system security. It should be acknowledged that alerting tools deployed in the microgrid may struggle to keep up with increasingly sophisticated and stealthy threat actors, such as those with living-off-the-land capabilities, which would reduce the pool of security alerts with which to base threat hunting decisions.

Regarding scalability, a challenge faced by the proposed approach is that it must deal with a larger volume of alerts when used within a larger system. Larger systems naturally generate more alerts, and thus will have more active attack hypotheses at any one time. Also, the number of active hypotheses is expected to increase in systems with a higher amount of connectivity between assets even if the number of individual assets is otherwise the same. This increase would be due to the overall larger number of attack hypothesis within the neighborhood of an alert group. Adapting to

a larger target system requires more storage space for hypotheses and possibly a shorter maximum time range within which a hypotheses can reside in the active pool.

6.7. APT Attack Sequence Generation

There is a gap in publicly available data that corresponds to the sequential arrangement of TTPs and lateral movement paths that APTs use to escalate privileges and move through an ICS environment such as the microgrid. To address the gap in diverse TTP sequences from APTs, we devise a methodology to synthetically generate sequences of MITRE ATT&CK tactics on a series of connected microgrid assets. The perpetration of each sequence is designated to an APT and their actions are based on the TTPs attributed to them in the CTI KG. The APT traverses the microgrid network from an initial access point to a final target in pursuit of its end goal. The sequences include one or more ATT&CK techniques that correspond to each performed tactic and to the APT. They also include a random set of ATT&CK data components for each technique, where the data components in the set are associated with the detection of the technique. During tests, each data component associated with the technique raises an alert that is sent to the threat manager, with the data component and the origin asset (targeted by the technique) being recorded in the alert fields. It should be noted that the scope of the APTs considered in this thesis is limited to those found in the MITRE ATT&CK ICS framework.

6.7.1. Attack Sequence Generation Methodology

The steps of the methodology to generate an attack sequence on the microgrid are as follows:

- (1) Select the perpetrator APT for the sequence;
- (2) Select an initial asset on which the APT performs the *Initial Access* tactic;
- (3) Select the end goal asset on which the APT performs its final tactic;
- (4) Select a lateral movement path from the initial asset to the end goal asset;
- (5) Select the UKC *In* and *Through* cycle tactics that the APT performs on the initial asset;

- (6) Select the UKC *Through* cycle tactics that the APT performs on each intermediary asset along the attack path between the initial and end goal asset; and
- (7) Select the UKC *Out* cycle tactics that the APT performs on the end goal asset.

The attack sequence that results from following these steps is structured as shown in Figure 6.6. In the methodology used in this research, “performing” a tactics means to use between one and three techniques, chosen at random, that have direct associations in the KG with the tactic, the perpetrator APT, and the targeted ICS assets at that step of the sequence. For each of those techniques, one to four data components are randomly chosen as detectors of the technique usage, with the chosen data components all having a direct association with the technique in the KG. Should a sequence include a technique with no associated data components, the technique usage is assumed to be undetectable. In accordance with the UKC assumption that not all tactics are mandatory, the *⟨Evasion⟩* and *⟨Privilege Escalation⟩* tactics are treated as optional. They have a 50% chance of being skipped for a given attack sequence. Regarding Step (4) of the methodology, the selected lateral movement path is not necessarily a shortest path between the initial and end goal assets. Some portion of the attack sequences in the dataset include a detour in their lateral movement path where the APT moves to an asset that is equidistant to the end goal asset, in terms of network hop distance, compared to the prior asset in the sequence. All transitions from one asset to another in the attack path are accompanied by the APT performing the *⟨Lateral Movement⟩* tactic, and all such transitions are between assets that share a network link in the microgrid system topology, i.e., they are adjacent.

The dataset resulting from the attack sequence generation methodology is split such that approximately 60% of the sequences belong to a tuning set, while the remaining sequences fall into a validation set. The proportion of each APT acting as the perpetrator across the attack sequences in the tuning and validation sets is roughly equal to that found in the full dataset. There are 163 sequences in the tuning set and 114 in the validation set, totaling 227 sequences in the original dataset. These sequences are split between eight APTs that use at least seven ATT&CK techniques in the KG. These APTs include APT33, Dragonfly, FIN6, FIN7, Lazarus Group, OilRig, Sandworm Team, and TEMP.Veles. The number of ATT&CK techniques used in the generated attack sequences range from 3 to 28, with a mean of 9.91 and a median of 8.

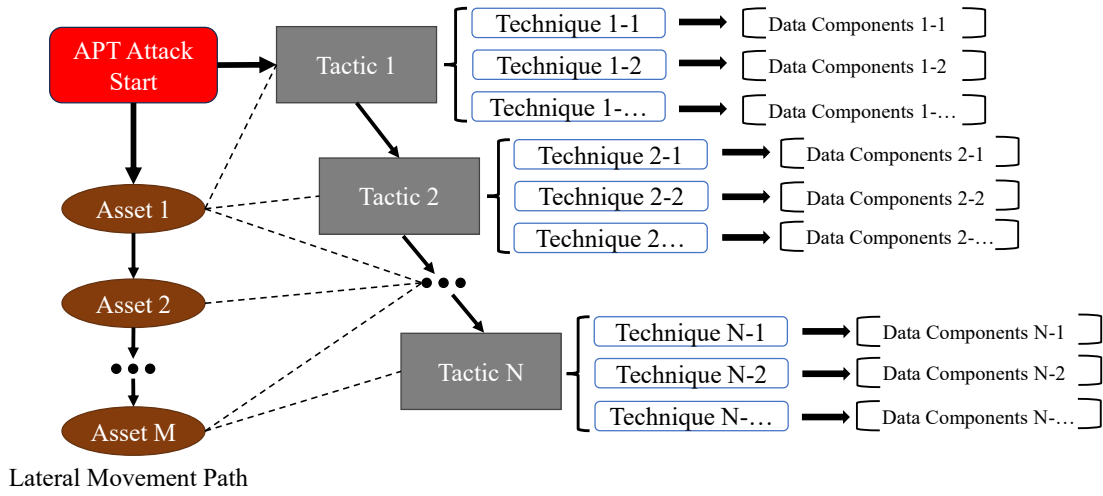


Figure 6.6: Structure of the generated APT attack sequences.

The sequences are simulated by generating security alert groups from the information contained in the data component sets chosen for each technique used in the attack sequence, then updating the active hypothesis pool each time a new alert group is generated. The sequences are stored in a JSON file containing a list of attack sequences. Each attack sequence is described as a list of ATT&CK tactics, which are ordered according to when each tactic is performed in the sequence. Each tactic also contains a list of the ATT&CK techniques it uses, again ordered according to when the techniques are used. Each technique is associated with a target asset and a list of data components that are observed by defenders when the technique is used. The sequences are executed by loading them into Python and then generating the security alert groups corresponding to each technique one at a time. The proposed threat hunting process, also implemented in Python, then updates the active attack hypothesis pool after every new security alert group. The update adds the hypotheses to the active pool that result from correlating the new alert group to existing hypotheses. The proposed approach uses the microgrid reference model described in Section 4.2.3, accessible programmatically within Python, to determine if the security alerts meet the attack hypothesis correlation conditions.

6.7.2. Usage of the Tuning and Validation Sets

The tuning set is used to determine the expected baselines for “technique coverage percent” TC_T corresponding to APT $t \in T$, where T is the set of all APTs under consideration as potential

attack perpetrators. The term “tuning” is used instead of “training” to distinguish the usage of the tuning set from the way that a training set is typically used in machine learning contexts for adjusting model weights. The technique coverage percent for a given APT and attack hypothesis is a measure of how many ATT&CK techniques matched to the alerts in the hypothesis are used by the APT compared to the total set of techniques used by that APT according to the KG derived from CTI. It can be expressed as follows:

$$TC_t = \frac{N_t^{hyp}}{N_t^{KG}} \quad (6.3)$$

Here, N_t^{hyp} is the number of techniques used by APT t matched to alerts in the hypothesis and N_t^{KG} is the number of techniques used by APT t according to the CTI KG.

To elaborate on the baselines, the tuning set is used to calculate the mean μ_t^{TC} and standard deviation σ_t^{TC} for the value of TC_t corresponding to APT t across all the attack hypotheses constructed from the attack sequences in the tuning set. The mean and standard deviation are used to identify the APTs that are most likely to be the perpetrator of the attack campaign being described by the hypothesis.

The validation set is used to calibrate the thresholds of the severity feature values used to compute the severity score of an attack hypothesis. Hypotheses with a sufficiently high severity score, as determined by the total severity score threshold, undergo the validation process to determine if they are to be flagged for review by security operators while identifying the perpetrator APT. The validation set is also used to determine the effectiveness of the APT identification process applied to hypotheses of that cross the total severity score threshold.

6.8. APT Identification Method

A hypothesis that has passed the validation process is attributed to a perpetrator APT, technique coverage percent TC_t , mean μ_t^{TC} , and standard deviation σ_t^{TC} for each APT. The μ_t^{TC} and σ_t^{TC} are computed from the attack sequence tuning set. The APT identified as the most likely perpetrator of a given hypothesis is the APT whose TC_t has the highest normalized distance D_t^n from its mean, which is expressed as:

$$D_t^n = \frac{TC_t - \mu_t^{TC}}{\sigma_t^{TC}} \quad (6.4)$$

The APT with the highest normalized distance may change as more alerts are added to a hypothesis. This means that the APT designated as the perpetrator of a hypothesis is dependent on the strictness of the conditions that result in a hypothesis passing the validation process, with more lenient conditions resulting in earlier validation. The conditions that determine whether or not a given hypothesis passes the validation process are based on features related to the severity of the hypothesized attack, which are described in the next section.

6.9. Attack Hypothesis Severity

Every attack hypothesis has a severity score. This score is computed based on information contained in the correlated alerts that comprise the hypothesis. A hypothesis with a higher severity score is presumed to represent an attack that has progressed further along its attack kill chain compared to one with a lower score.

The computation of severity score proposed in this thesis takes into consideration the topology and essential functions of the microgrid. The computed severity score is aligned with the danger that the suspected attack poses on the microgrid system. Specifically, the severity score is computed according to five attack hypothesis features:

- (1) The number of correlated security alerts;
- (2) The full range of Purdue levels associated with the affected assets;
- (3) The weighted sum of microgrid operational functions dependent on the affected assets;
- (4) The suspected ATT&CK tactic that is highest in the UKC order; and
- (5) The highest normalized distance for the technique coverage percent.

The above features are chosen due to them being a potential indicator of how many attack actions have already been taken by the attacker and by extension how close the attacker is to reaching

their final objective. The microgrid operational functions, along with the assets involved in performing them and their relative weight for maintaining safe microgrid operation, are presented in Table 6.3. Each microgrid function dependent on at least one affected asset in the attack hypothesis has its weight added to the total weight of affected microgrid functions. The microgrid functions and relative weights are chosen based on expert knowledge attained from consultation with experts in microgrid modeling and operation. Notably, the frequency control, managed by the Microgrid Central Controller (MGCC), is given the highest weight. As previously demonstrated in Chapter 5, attacks that violate the integrity of the data exchanged for frequency control, whether from the Merging Unit (MU) or the supervisory controller, can result in the microgrid operating at unsafe frequencies and in the complete de-energization of the microgrid. The load control and load shedding operations are also important for maintaining stability under changing power demand conditions and thus are given a relatively high weight. Protection function responses, which are directly dependent on relay configurations, are expected to be engaged only during rare events, such as line faults. As such, relay reconfiguration is given a relatively lower weight to reflect the time afforded to security operators to uncover attacks modifying their configuration before they are needed for protection responses. Similarly, a relatively low weight is assigned to the management of ESS SoC due to the time available to resolve attacks against the ESS management function being large compared to the time available to respond to frequency and load control before adverse effects are experienced. In addition, the failure cases for the management of ESS SoC, namely the ESS being full when it is to absorb power or empty when it is to discharge, is not expected to have as dramatic an impact as a relay failing to isolate a faulted line. For this reason, ESS management is given an even lower weight than relay reconfiguration.

The severity features are each given a “severity level”. The possible levels are *LOW* severity, *MEDIUM* severity, and *HIGH* severity. These severity levels directly correspond to a numeric value. The *LOW* severity level has a numeric value of 0. It indicates that the feature does not give solid evidence of an attack presence. The *MEDIUM* and *HIGH* severity levels have the value of 1 and 3, respectively. A *MEDIUM* severity suggests that the feature gives considerable evidence to suggest that an attack is being perpetrated. The *HIGH* severity level is used to indicate a much stronger suspicion of an attack relative to *MEDIUM* severity. The total severity score of a given

Table 6.3: Weights and assets assigned to microgrid functions

Function Name	Weight	Assets Involved in Function
Frequency Control	10	PCC MU, PCC Supervisory Controller, MGCC ESS Supervisory Controller, ESS Controller
Load Control	6	Operator Workstation, Telecontrol Server, Load Supervisory Controller, Load Controller
Load Shedding	6	PCC MU, PCC Supervisory Controller, MGCC Load Supervisory Controller, Load Controller
Frequency Protection	3	PCC, PCC MU, PCC Supervisory Controller, MGCC, DER and Load Supervisory Controllers, Circuit Breakers
Over/under Voltage Protection	3	MUs, Relays, Circuit Breakers
Over/under Current Protection	3	MUs, Relays, Circuit Breakers
Relay Reconfiguration	3	Operator Workstation, Telecontrol Server, Supervisory Controllers, Relays
Manage ESS State of Charge	1	ESS controller, ESS Supervisory Controller, MGCC

attack hypothesis is equal to the weighted sum of the severity feature values. The total severity score is used as a means to filter out the hypotheses that present lower risk for the sake of highlighting the ones that warrant more immediate attention. Specifically, hypotheses whose severity score cross a configured threshold undergo the APT verification step of the validation process before being attributed to an APT and marked as “urgent”.

The weights and severity level ranges for the severity features are given in Table 6.4. The normalized distance for technique coverage is given a higher weight than the other features because, in testing, it is found to be the most effective at differentiating true positives from hypotheses containing noisy alerts. To further reduce false positives, a rule is imposed to filter hypothesis with a *LOW* severity for the highest technique coverage percent normalized distance feature (D_{max}^n). Admittedly, this may make it more difficult to validate novel attacks that do not match any known APT behaviors. However, the other severity features are not dependent on having prior knowledge of particular APT profiles, thereby allowing even novel APTs to be detected once they progress far enough into their kill chain.

The specific value threshold parameters for each severity level across the hypothesis severity

Table 6.4: Severity levels and weights for hypothesis feature value ranges

Feature	Weight	LOW	MEDIUM	HIGH
Number of correlated security alert groups	1	0 - 4	5 - 6	7+
Purdue level range affected	1	0	0.5 - 2.0	2.5+
Total weight of microgrid functions affected	1	0 - 2	3 - 7	8+
ATT&CK Tactic with highest UKC order	1	0 - 2	3 - 5	6+
Highest normalized distance D_{max}^n among APTs	1.5	$D_{max}^n < -0.5$	$-0.5 \leq D_{max}^n < 0.25$	$0.25 \leq D_{max}^n$

features are chosen based on the results of a grid search strategy applied on the parameters. This involves using the attack sequence validation dataset to observe how the feature values for an attack hypothesis increases as the attack progresses and as more correlated alerts are added to the hypothesis. For a given attack sequence i in the validation set, the value of severity feature f is determined for the hypothesis as it reaches the end of each UKC cycle, namely the *In*, *Through*, and *Out* cycles. In particular, the hypothesis formed after the *Out* cycle is completed corresponds to the end of the attack sequence, where the APT has reached their end goal. The severity values of feature f after attack sequence i has completed the first, second, and third UKC cycle are denoted as $SEV_{f,i}^1$, $SEV_{f,i}^2$, and $SEV_{f,i}^3$, respectively. A grid search approach is used to find the combination of lower bound thresholds for reaching *MEDIUM* severity and *HIGH* severity that maximizes the total difference in severity level between each UKC cycle across all attack sequences in the validation set. In this way, the threshold parameters to choose that lead to the most gradual increase in severity level as the attacks progress are determined, as opposed to having severity levels spike near the start or end of the attack. The smoothness of the increase in severity level for feature f is measured by a smoothness metric denoted as SM_f . Computing this metric first requires the computation of $SAVG_f^{2-1}$ and $SAVG_f^{3-2}$ for a given combination of *MEDIUM* and *HIGH* lower bound thresholds, where $SAVG_f^{2-1}$ is the average difference between the first and second UKC cycles across N validation set attack sequences and $SAVG_f^{3-2}$ is the average difference between

the second and third UKC cycles. They are computed as follows:

$$\begin{aligned} SAVG_f^{2-1} &= \frac{1}{N} \sum_{i=1}^N SEV_{f,i}^2 - SEV_{f,i}^1 \\ SAVG_f^{3-2} &= \frac{1}{N} \sum_{i=1}^N SEV_{f,i}^3 - SEV_{f,i}^2 \end{aligned} \quad (6.5)$$

To note is that the severity feature values of an attack hypothesis never decreases when an alert is added to it, and so taking the absolute value of the difference is unnecessary. Once $SAVG_f^{2-1}$ and $SAVG_f^{3-2}$ are computed, the SM_f metric is then calculated for feature f as follows:

$$SM_f = SAVG_f^{2-1} \times SAVG_f^{3-2} \quad (6.6)$$

The lower bound severity level thresholds that result in the highest value of SM_f are chosen for severity feature f . This method for finding the severity level thresholds is repeated for all severity features, with the computation of the smoothness metric for a given feature being independent of the other features.

6.10. Managing the Number of Hypotheses with MCDM

An important threat hunting challenge to address is the scalability of the threat hunting solution when it comes to maintaining the size of the active hypothesis pool. If steps are not taken to regularly remove hypotheses from the active pool, the number of hypotheses in the active pool to grow exponentially as the number of generated security alerts increases. This is because the number of neighboring hypotheses in the pool will rise, which in turn results in more correlations and an increasing number of new hypotheses being created with each successive security alert. For this reason, the proposed threat hunting approach engages in filtering of the active hypothesis pool. This filtering is based on Multi-Criteria Decision Making (MCDM), which computes relative scores for a given set of options depending on how well those options align with a predefined set of criteria.

The main motivating factor leading to the selection of filtering based on MCDM over other automated approaches is the desire to have a filtering process that can be tuned according to the

security priorities of system security operators. In MCDM, the priorities can be expressed as criteria with numeric weights relative to the other criteria. This helps ensure that the hypotheses that are preserved in the active hypothesis pool are well aligned with the security concerns of operators. In the case where incorporating particular criteria for comparison is deemed unnecessary, then alternative filtering methods that randomly select a subset of hypotheses or that select sets of similar hypotheses determined through clustering methods can be used instead.

There are multiple MCDM methods that have been proposed. They can be described as falling into the following categories: (i) Multi-Attribute Utility methods (MAUT), (ii) outranking methods, (iii) compromise methods, and (iv) other MCDM methods [123]. The MAUT methods give quantitative scores for each entry among a set of alternatives, with higher scores corresponding to higher alignment with the criteria. Meanwhile, the outranking methods are designed to find the best course of action among a set of actions. By contrast, the compromise methods compare options against an idealized benchmark. The MCDM filtering used in the proposed threat hunting approach uses MAUT over methods from the other categories due to its use case of choosing among system state descriptors (the attack hypotheses) rather than courses of action and due to there being no idealized attack hypothesis for comparison purposes. The severity features used to calculate the severity score of each attack hypothesis, previously listed in Table 6.4, are also chosen as criteria for the MCDM filtering of hypotheses, with the exception that the number of correlated security alert groups is omitted as a criteria due to the other criteria being correlated with it. Within the MAUT category are the Analytic Hierarchy Process (AHP) and Analytic Network Process (ANP) methods. In AHP, the criteria are assumed to be independent of one another. The ANP method is a more generalized version of AHP that accounts for dependencies between criteria. The criteria chosen for MCDM pertaining to attack hypotheses are assumed to be independent, and thus AHP is chosen as the MCDM method to utilize for hypothesis filtering in order to avoid unneeded complexity.

To give a brief description, the AHP method gives a means to quantify the relative priority score for each option, referred to as an “alternative” in the context of AHP, in a list of alternatives based on how well they meet a set of fixed criteria. The relative importance of one criterion over another is specified in pairwise fashion. The relative weights of each criterion are used to form a pairwise comparison matrix of rows and columns, with a given matrix entry corresponding to how much

more importance is placed on the row criteria over the column criteria. A fixed priority vector is then calculated from the pairwise comparison matrix by taking the sums of the different rows and then normalizing them so that the priority vector values add up to one (1.0). The normalized priority vector values reflect the overall importance of each criterion. The pairwise comparison matrix for attack hypotheses and its corresponding priority vector are given in Table 6.5.

To compute the scores for all alternatives within an input set of alternatives, a criteria matrix is computed for each criterion with dimensions $D \times D$, where D is the number of alternatives. The criteria matrix values for a given criteria corresponding to a severity feature is determined according to the difference in severity levels for the feature as follows:

- *MEDIUM* 3 : 1 *LOW*
- *HIGH* 3 : 1 *MEDIUM*
- *HIGH* 5 : 1 *LOW*

Each criteria matrix also has its own priority vector, computed in the same way as the priority vector for the pairwise comparison matrix. The final score for each alternative is then found by multiplying each of its criteria matrix priority vector entries with the corresponding value in the priority vector of the pairwise comparison matrix, then taking of the sum of those products. An illustrative example of AHP is given in Figure 6.7 for three hypotheses HYP_A , HYP_B , and HYP_C . In this example, the criteria matrix for technique coverage percent criteria (C_1) is derived from HYP_A , HYP_B , and HYP_C having severity levels *LOW*, *MEDIUM*, and *HIGH*, respectively. The other criteria matrices are formed from different combinations of severity levels for their respective feature. The example shows that, with the priority vector resulting from the pairwise comparison matrix in Table 6.5 and with the given criteria matrices, HYP_C has the highest AHP score.

The threat manager is responsible for applying the proposed MCDM filtering to the active hypothesis pool. It does so after it has finished updating the active hypothesis pool with new hypotheses constructed from correlations between existing hypotheses and the latest security alert group formed from the latest batch of security alerts. The filtering process starts by separating the active hypotheses into clusters according to the number of security alerts groups they have in common. To form the clusters, the attack hypotheses in the active pool with four or more alert groups are set as

Table 6.5: Pairwise comparison table of criteria weights for the AHP

Criteria	C_1	C_2	C_3	C_4	Priority Vector
Technique coverage percent (C_1)	1	0.3333	0.2	0.2	0.1050
Highest UKC order tactic (C_2)	3	1	0.1428	0.1428	0.0508
Purdue level range affected (C_3)	5	7	1	3	0.5294
Microgrid functions affected (C_4)	5	7	0.3333	1	0.3148

C1	Hyp _A	Hyp _B	Hyp _C	Priority Vector
Hyp _A	1	0.3333	0.2000	0.1031
Hyp _B	3	1	0.3333	0.2915
Hyp _C	5	3	1	0.6054

C2	Hyp _A	Hyp _B	Hyp _C	Priority Vector
Hyp _A	1	5	5	0.6574
Hyp _B	0.2000	1	0.3333	0.0916
Hyp _C	0.2000	3	1	0.2510

C3	Hyp _A	Hyp _B	Hyp _C	Priority Vector
Hyp _A	1	1	0.3333	0.2000
Hyp _B	1	1	0.3333	0.2000
Hyp _C	3	3	1	0.6000

C4	Hyp _A	Hyp _B	Hyp _C	Priority Vector
Hyp _A	1	0.2000	0.2000	0.0910
Hyp _B	5	1	1	0.4545
Hyp _C	5	1	1	0.4545

Pairwise Comparison	0.1050	0.0508	0.5294	0.3148
Priority Vector				



Hyp _A	0.1031	0.6574	0.2000	0.0910	=	0.1787
------------------	--------	--------	--------	--------	---	--------



Hyp _B	0.2915	0.0916	0.2000	0.4545	=	0.2842
------------------	--------	--------	--------	--------	---	--------



Hyp _C	0.6054	0.2510	0.6000	0.4545	=	0.5370
------------------	--------	--------	--------	--------	---	--------

Figure 6.7: Illustrative example of AHP for three attack hypotheses.

candidates for the clustering and subsequent MCDM filtering. From there, the clustering process begins. This involves selecting an attack hypothesis at random from among the candidates, then placing it into a cluster alongside all candidate hypotheses that are within the neighborhood hop distance and that share at least 75% of the same security alert groups. The clustering is repeated iteratively for all unclustered candidates until all candidates belong to a cluster. This process is summarized in Algorithm 2, with $InNeighborhood(a_i, a_j)$ being true only if attack hypotheses a_i and a_j are within the neighborhood hop distance and with $SimilarGroups(a_i, a_j)$ being true only if the 75% or more shared alert groups condition is met.

Once the candidate hypotheses are clustered, the AHP method is applied separately to each cluster, where each hypothesis in the cluster is set as an alternative for the AHP method. All hypotheses that have the highest AHP score within their cluster, or tied for highest, are kept in the active pool, while the rest are filtered out of the pool, i.e., are discarded. As a safeguard for large numbers of hypotheses within a cluster being tied for the highest score, the maximum number of hypotheses

Algorithm 2 Hypothesis clustering algorithm

Input: Candidate attack hypothesis pool $A = \{a_1, a_2, \dots, a_n\}$

Output: Attack hypothesis clusters C_A

```
 $C_A \leftarrow \emptyset$ 
while  $A \neq \emptyset$  do
  Select  $a_i \in A$ 
   $C \leftarrow \{a_i\}$ 
  for  $a_j \in A, a_j \neq a_i$  do
    if  $InNeighborhood(a_i, a_j)$  and  $SimilarGroups(a_i, a_j)$  then
       $C \leftarrow C \cup \{a_j\}$ 
    end if
  end for
   $C_A \leftarrow C_A \cup C$ 
   $A \leftarrow A \setminus C$ 
end while
```

that are retained in a cluster is set to 20.

The impact of the MCDM filtering described here is evaluated on the synthetically generated attack sequence validation dataset, as well as on a separate dataset of alert groups with randomly chosen data components and affected assets. The set of alert groups generated from randomly chosen data components is effectively treated as a dataset of noisy alerts for the purposes of this thesis, i.e., a dataset of false positives. The number of unique data components corresponding to the noisy alerts added to a single alert group ranges from one to three. The impact of MCDM is measured by observing the number of alert group sequences that result in the active hypothesis pool exceeding a given size limit. The percentage of alert group sequences that result in the active hypothesis pool crossing size limit $N \in \{100, 500, 1000, 2000, 3000, 4000, 6000, 8000\}$ is given in Table 6.6. This includes the attack sequences with and without MCDM filtering, as well as the sequences of random alerts with and without filtering. It also compares the results when setting the acceptable UKC order distance for alert correlation to three and when setting it to six. The table shows that the hypothesis pool created from legitimate attacks undergo more filtering than the pool created from random alerts. This is because the alerts generated from real attacks are more localized to particular sections of the microgrid network. As a result, the MCDM filtering produces fewer candidate clusters for alerts caused by legitimate attacks, leading to more hypotheses being discarded. It is also worth noting that the number of total hypotheses in the active pool

Table 6.6: Impact of MCDM filtering on the size of the active hypothesis pool

Type of Sequence	Percentage (%) of Sequences that Cross Hypothesis Pool Size Limit N							
	100	500	1,000	2,000	3,000	4,000	6,000	8,000
Acceptable UKC Order Distance = 3								
APT Attack, No MCDM	100	65.52	56.90	45.98	39.66	36.21	31.61	28.16
APT Attack, MCDM	86.21	35.63	17.82	5.17	4.02	0	0	0
25 Random Alert Groups, No MCDM	100	100	100	90.00	80.00	80.00	80.00	75.00
25 Random Alert Groups, MCDM	100	93.75	81.25	56.25	56.25	37.50	18.75	18.75
Acceptable UKC Order Distance = 6								
APT Attack, No MCDM	100	68.97	56.90	47.13	40.23	39.66	32.18	29.31
APT Attack, MCDM	88.51	37.36	20.11	6.32	4.02	1.72	0	0
25 Random Alert Groups, No MCDM	100	100	100	90.00	90.00	85.00	85.00	85.00
25 Random Alert Groups, MCDM	100	93.75	93.75	81.25	68.75	62.50	50.00	43.75

increases slightly when the acceptable UKC order distance is increased. For example, some attacks that undergo MCDM filtering still result in over 4,000 hypotheses when the acceptable UKC order distance is six, but never do so when the order distance is three. The impact of raising the UKC order distance is more pronounced when looking at the sequences of random alert groups. Figure 6.8 gives a plot of the number of active hypotheses as a function of the number of alert groups in the hypothesis and the acceptable UKC order distance depending on whether MCDM is used.

6.11. Managing Noisy Alerts with the APT Verification Step

Despite the best efforts to increase the accuracy of detection tools within the microgrid, there are still bound to be false positives when it comes to the alerts that reach the threat manager. It is therefore important to account for the presence of false positives when validating attack hypotheses.

6.11.1. APT Verification Step

In the proposed threat hunting approach, attack hypotheses undergo an APT verification step once they cross the configured severity score threshold. This validation step evaluates whether is sufficient evidence to link the hypothesis to a particular APT. To do so, it first determines the top two APTs for the hypothesis according to the normalized distance D_t^n for technique coverage

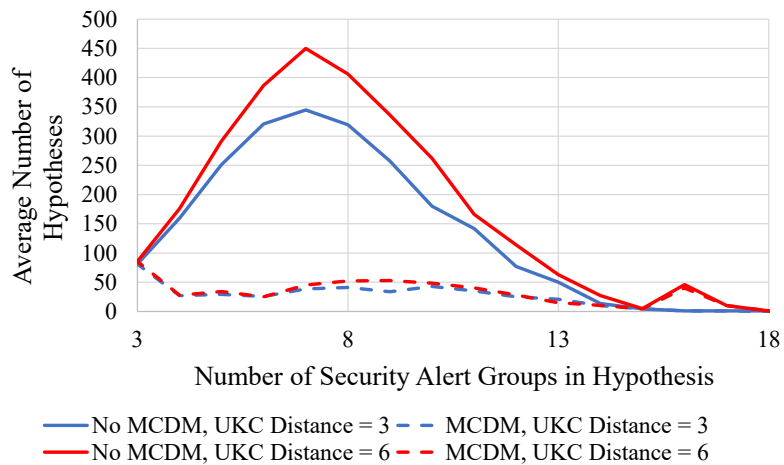


Figure 6.8: Number of attack hypotheses with and without MCDM.

percent. Each security alert group in the hypothesis is checked against these APTs. Any alert group with no associations between their technique matches and any of the APTs is masked. A “reviewed severity score” is computed to represent the severity score after excluding the masked alerts from the hypothesis. If the reviewed severity score still exceeds the configured severity score threshold, then the hypothesis passes the validation check and is marked as “urgent”, warranting review by security operators. However, if the reviewed severity score falls below the threshold, then the hypothesis remains in the active pool, where it can be correlated to future alerts. All masking of alert groups is undone in this case, returning the hypothesis to its original state prior to the validation check.

6.11.2. Considerations for Novel Attacks

There is concern about how the APT verification step proposed in this thesis may handle the detection of novel threat actors. In particular, there is the potential for a novel threat actor to follow a TTP pattern that is dissimilar to those used by any APT found within the body of CTI. In theory, an attacker could use techniques that would result in security alerts that are highly likely to be masked during the APT verification step in a way that consistently results in the reviewed severity score falling below the severity threshold. The threat of an APT employing novel combinations of techniques that may be misinterpreted as false positives cannot be entirely eliminated by the proposed threat hunting strategy. However, the proposed approach can be adapted to mitigate the risk

that such an attack reaches its end goal before being detected and resolved. One way to address the detection of novel attacks would be to have a hard severity score limit that lies above the one triggering the APT verification. The hard limit would force the “urgent” label on any hypotheses whose severity score exceeds it, regardless of what the reviewed severity score calculation discussed above. This ensures that novel attacks can progress only to a limited extent before the attack hypotheses constructed from observed system activity reach a sufficiently high severity score and are subsequently brought to the attention of security operators. Recall that most of the severity features used to calculate the severity score are independent of the CTI on ICS APTs and thus will contribute to a higher severity score as a novel attack progresses.

6.12. Experimental Evaluation

The effectiveness of the proposed threat hunting approach is assessed through experimental evaluation. This assessment includes the ability to correctly correlate alerts resulting from the actions of the same attack campaign. It also includes the accuracy of the APT identification process for hypotheses at different stages of the underlying attack and under different amounts of system noise and attack obfuscation.

The synthetically generated attack sequences and random noise sequences are stored in JSON files and loaded into Python to be simulated on the microgrid system model presented in Section 4.2.3. The results presented in this section are derived from the outputs of the Python implementation of the proposed threat hunting approach after simulating the progression of the attack and random noise sequences through the microgrid system model.

6.12.1. Security Alert Group Correlation Accuracy

Recall from Section 6.6.2 that the proposed threat hunting approach correlates a security alert group to an existing attack hypothesis in its neighborhood if (i) there is an identical match between the data components of an alert in the group and in the hypothesis, (ii) if there is close alignment of the tactics they are matched to according to the UKC tactic sequence, or (iii) if there is evidence of lateral movement. The relevance of the correlation conditions is evaluated by calculating the

True Positive Rate (TPR) for correlating an alert group corresponding to an attack step from the synthetically generated validation set to an attack hypothesis. This involves simulating each attack sequence in the validation set one attack step at a time to generate a series of security alert groups and then finding the total percentage of attack steps whose alert groups are correlated to any of the hypotheses constructed as the attack progresses. Table 6.7 shows the TPR for correlated security alerts when using different values of acceptable UKC tactic order distances, as well as the TPR when the identical data component condition is ignored. The False Negative Rate (FNR) can be derived by calculating the complement of the TPR. As previously shown in Table 6.6, increasing the acceptable UKC order distance comes at the cost of performance due to increasing the total number of successful correlations, which directly results in more attack hypotheses to consider. When using only the identical data-component condition, the TPR achieved is 65.79%, highlighting the contribution of UKC-based sequencing.

6.12.2. APT Identification Accuracy

The effectiveness of the APT identification method described in Section 6.8 is evaluated by finding its precision and recall for matching a given attack hypotheses to the perpetrator APT from which the microgrid attack sequence that generated the hypothesis is sourced. The identification method yields a true positive if the true perpetrator APT is among the top two APTs according to normalized distance D^n for that hypothesis. Using the top two APTs increases the true positives but requires operators to consider the possibility of at least the top two APT being the perpetrator rather than just one when deciding on the appropriate response action. If the true perpetrator is not within the top two APTs, then the hypothesis is marked as a false positive for the APT with the highest D^n . The precision and recall for attributing hypotheses to the correct perpetrator APT using this evaluation scheme is given in Table 6.8. The results are collected for all the ICS APTs in scope across different values set for the severity threshold, denoted as *SEV*. Specifically, it covers the results when *SEV* is set to 4, 6, and 8. To collect the attack hypotheses used for testing, the attack sequences in the validation set are progressed up until the point where at least one hypothesis has a reviewed severity score above *SEV*. Once this point is reached, the APT identification is tested on all hypotheses generated over the course of the simulated attack sequence that have the highest

Table 6.7: Correctly correlated security alerts groups

Correlation Method	TPR for UKC Order Distance				
	2	3	4	5	6
All correlation conditions	0.9374	0.9455	0.9455	0.9465	1.0000
Correlation with no data component match condition	0.7888	0.9024	0.9067	0.9486	0.9807

Table 6.8: APT identification success rate

APT	$SEV = 4.0$		$SEV = 6.0$		$SEV = 8.0$	
	Precision	Recall	Precision	Recall	Precision	Recall
APT33	1.0000	1.0000	1.0000	0.7895	0.7188	1.0000
Dragonfly	1.0000	0.1071	1.0000	0.1852	1.0000	0.1500
FIN6	0.3902	1.0000	0.4189	1.0000	0.5052	1.0000
FIN7	0.7973	1.0000	0.7375	1.0000	0.0769	1.0000
Lazarus Group	1.0000	1.0000	1.0000	1.0000	1.0000	0.8714
OilRig	1.0000	0.2131	1.0000	0.4808	1.0000	0.5000
Sandworm Team	0.5102	1.0000	0.6279	1.0000	0.6200	1.0000
TEMP.Veles	0.7042	0.7600	0.9077	0.8136	0.8500	0.8824
Overall	0.7436	0.6552	0.8080	0.7624	0.7815	0.7205

number of security alert groups and whose reviewed severity score is above SEV . Note that higher values for SEV lead to attack sequences progressing further into their kill chain overall before a hypothesis resulting from the attack is validated.

The results of the APT identification precision and recall show that the ICS APTs do not all show the same accuracy trends as SEV increases. Some APTs, such as APT33 and FIN7, are more easily distinguished from other APTs during the early stages of the attack, as evidenced by the higher precision and recall for $SEV = 4$ compared to the higher SEV values. Meanwhile, APTs such as OilRig and FIN6 are easier to distinguish from other APTs as SEV increases, which suggests that the set of techniques they tend to use in the later stages of their kill chain are dissimilar to those used by their peers.

Also of interest is the impact of noisy alerts, which corresponds to purely benign actions that nevertheless raise an alert, on the success rate of the APT identification, as well as the impact of attack activities that go undetected. To measure the effect of noise on precision and recall, a noise parameter Z is introduced into the simulation of the attack sequences in the validation set. The

value of Z refers to the number of random alert groups created within the attack simulation every time an alert group resulting from the attack sequence is produced. This results in a signal to noise ratio of $1:Z$. Also, a masking parameter U is introduced to control the number of attack steps in the sequence that go undetected. The value of U refers to the number of attack steps whose alerts are manually masked during the attack sequence simulation. The attack steps chosen for alert masking are decided at random. This results in the removal of U alert groups to correlate to attack hypotheses that would otherwise be observed by the threat manager.

The F1-score for different Z and U parameter settings, including the baseline of $Z = 0$ and $U = 0$, is given in Figure 6.9-a across a range of severity threshold values (SEV) tested on the validation attack sequence dataset. It can be seen that both the presence of noise and the incidence of undetected attack activity reduce the success rate of the proposed APT identification approach. The validation process aims to control the extent in which noisy alerts and undetected attack activity affects the APT identification step. However, random noise in the microgrid system reduces the likelihood that an attack campaign launched by an APT is never validated. Figure 6.9-b shows the percentage of attack sequences that result in at least one attack hypothesis passing the validation process for different SEV values. Predictably, the introduction of noise increases the incidence of validated hypotheses across the attack sequences in the validation dataset, while removing alerts decreases the overall success rate. This means that, in a system with noisy alerts, it is less likely that an attack campaign will go completely unnoticed, even if the chance of attributing it to the correct APT is reduced. Considering the F1 score and validation success percentage together, having $SEV = 7.0$ is the most suitable for maintaining apt identification accuracy in the face of noise and missed alerts without compromising too much on validation success rate. However, having $SEV = 6$ should be acknowledged as producing significantly better APT identification results under the ideal case of having no noisy alerts or undetected attack steps. Choosing a higher value for SEV also has the benefit of reducing the incidence of hypothesis containing solely noisy alerts from passing the validation process.

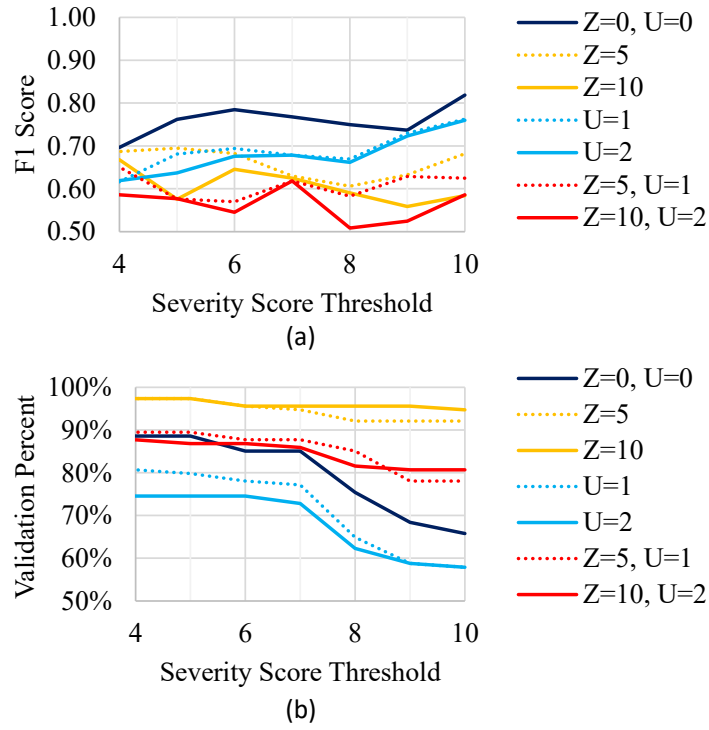


Figure 6.9: APT identification results for different severity thresholds.

6.12.3. Hypothesis Validation Statistics

As previously noted, the severity threshold (SEV) affects how far an attacker progresses through their kill chain in the microgrid before a hypothesis exposing their attack is validated. The impact of SEV on the UKC phase at which a given attack produces a validated hypothesis is tested using the attack sequences in the validation dataset. Each step in the attack sequence directly corresponds to a specific ATT&CK tactic, and thus the validation of a batch of hypotheses resulting from a new security alert group can be mapped to that tactic. Figure 6.10 presents the distribution of the furthest tactic reached by the attack sequences along the UKC before the emergence of an attack hypothesis whose reviewed severity score exceeds the SEV values of 4, 6, and 8. As shown, most attack sequences are validated while the APT is in the UKC *Through* cycle. The sequences that manage to reach the $\langle$ Impact $\rangle$ tactic before producing a validated hypothesis are considered to have reached their goal. They are thus treated as not having been validated and having effectively gone

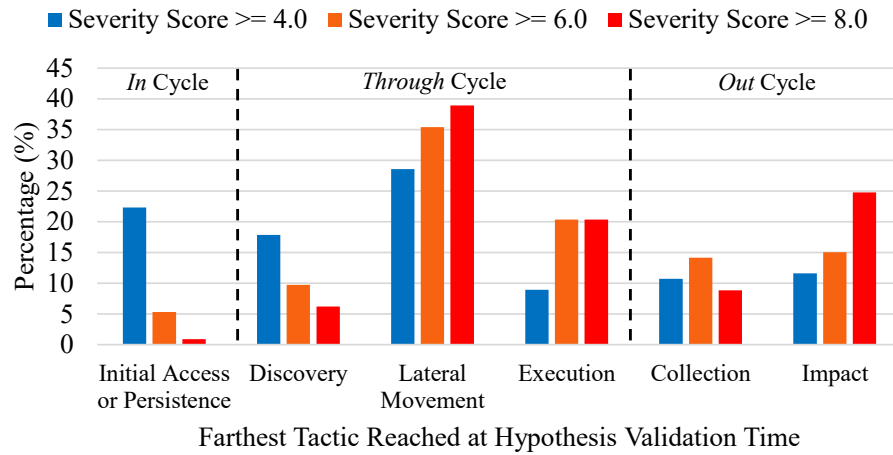


Figure 6.10: Tactic reached before hypothesis is validated.

undetected. The effect of increasing SEV on the farthest tactic reached before validation is also evident. Approximately 22% of attack sequences are detected at the $\langle$ Initial Access $\rangle$ or $\langle$ Persistence $\rangle$ ATT&CK tactic for $SEV = 4$, while only about 1% are detected at that stage for $SEV = 8$. Conversely, the number of sequences that reach the $\langle$ Impact $\rangle$ tactics jumps from about 12% to 25% between $SEV = 4$ and $SEV = 8$. Ultimately, the SEV threshold value should be chosen with consideration of the risk that an attack goes undetected. At the same time, it should consider the overhead of having more false alarms when choosing lower SEV values.

Chapter Seven

Attack Step Prediction

The ability to detect a malicious intrusion is a fundamental capability that is needed to ensure the security of a given system. The two preceding chapters related to microgrid active and passive monitoring (Chapter 5) and microgrid threat hunting (Chapter 6) are aligned with this need. Naturally, the activities of a security framework that acts to protect the system should not stop at the detection of malicious activity. An appropriate and effective response is also necessary to neutralize the threat being posed by the malicious entity.

Security operators typically endeavor to isolate and contain an active threat on the network based on the security alerts and observables that have already been made apparent to them. However, responding solely to confirmed information does not necessarily constitute a thorough inspection of the system environment. An APT already detected by security operators may have advanced deeper into the network than what is revealed by the time a response is initiated. In such a case, it is possible that an APT succeeds in compromising the critical system elements needed to reach their end goal. To prevent this outcome, security operators must not only remove the presence of the APT in assets that have been directly implicated in the attack but also ensure that the surrounding assets are not compromised. In particular, it is crucial that security response efforts are prioritized to the key system assets put at risk by the apparent attack.

This chapter presents a risk-based methodology with which to direct and prioritize the security response efforts that may be warranted by a validated hypothesis. Building on the attack hypotheses constructed and validated according to the threat hunting approach presented in the prior chapter,

a methodology to predict the attack steps on the microgrid that would follow from a given attack hypothesis is devised. This methodology generates predictions corresponding to the highest risk for several possible assumptions on the attack strategy. The risk presented by a given attack hypothesis considers the extent of the damage to the microgrid that would result from attacker actions on various microgrid assets. At the same time, the risk accounts for the feasibility and likelihood of such actions. In effect, the evaluation of the impact magnitude combined with its likelihood of occurrence constitutes a risk assessment [91], [92], [93]. The risk assessment involved in the proposed methodology is performed based on information taken from the microgrid system topology, the CTI on ICS threat actors, and the attack hypothesis.

The intended outcome of the proposed methodology is to provide a means to make practical decisions on how to respond to an active threat on the microgrid even in the absence of real data on the frequency with which threat groups progress to a given asset on the microgrid network. It weighs the possibilities of the more immediate steps that a detected attack campaign may realize and provides quantitative values expressing the relative risk they pose on the microgrid. The methodology also does not rely on exploitation probabilities assigned to low-level vulnerabilities potentially present in particular system assets.

The methodology assigns relative scores to possible ways an APT may progress given a specific attack hypothesis. Of particular interest is the prediction of the next asset that the APT will target, i.e., the subsequent asset in the attack kill chain. Knowing where an APT is likely to strike next, or has already struck without being noticed, helps steer attack response efforts. The identification of the subsequent asset is supported by calculating numeric risk values for all potential next targets. Risk values are computed for each pairing between a subsequent asset and an end goal asset. In this context, a subsequent asset refers to a host asset that is suspected to be directly reachable by the APT within the microgrid network. It can thus be the next target for lateral movement on the part of the APT. The end goal asset is the host asset reached at the final stage of the APT attack campaign on which the APT acts to negatively impact microgrid power operations. Risk values express the risk associated with the attack paths extending from a particular state of an ongoing attack campaign. They are intended for use by security operators to help predict the intentions of the APT, where possibilities with higher risk values require more attention.

We formulate a risk value definition based on the product of components that relate to the likelihood and impact of an attack path. Specifically, it is a product of three components: (i) the likelihood of the subsequent asset being the next lateral movement target, (ii) the path exploitability of the shortest path between the subsequent asset and an end goal asset, and (iii) the impact associated with the compromise of the end goal asset in terms of damage to the microgrid. The likelihood component is calculated using a Bayesian Network constructed from the microgrid attack sequence dataset, described in Section 6.7. The path exploitability relates to the likelihood that an APT will successfully exploit the assets along the path to gain the privileges needed to perform a certain attack impacting the microgrid power system. The APT laterally across the assets on the path in pursuit of reaching its end goal. Its calculation is based on the microgrid network model topology previously described in Section 4.2.1. The impacts associated with possible end goal assets are determined based on results from simulated attacks launched on the microgrid co-simulation testbed, which are presented later in this chapter. The breakdown of the likelihood, path exploitability, and impact components is given in Figure 7.1.

An important feature to incorporate into the proposed attack step prediction methodology is the flexibility to adjust the weight given to the subsequent asset likelihood, the path exploitability, and the impact when calculating the risk value. The methodology proposed in this chapter is a preliminary work intended to provide a strong starting point for system owners to deploy a risk-driven response plan for their own microgrid system networks and possibly other ICS networks as well. Different operators may identify different priorities for the proposed risk components as part of their overall response strategy and philosophy. For example, a security team that believes APTs predominantly follow the path of least resistance, even at the expense of lower reward, may place a high weight on path exploitability when computing risk. However, an organization that believes an APT will pursue the worst-case scenario for the utility may place high weight on the impact component. Operators may also want to consider multiple predictions that each account for different assumptions about APT behaviour. The proposed methodology incorporates parameters that adjust the weight of each risk value component to allow for this flexibility.

Separate sections of this chapter are devoted to the components of the attack step prediction methodology. The topics covered by these sections are as follows:

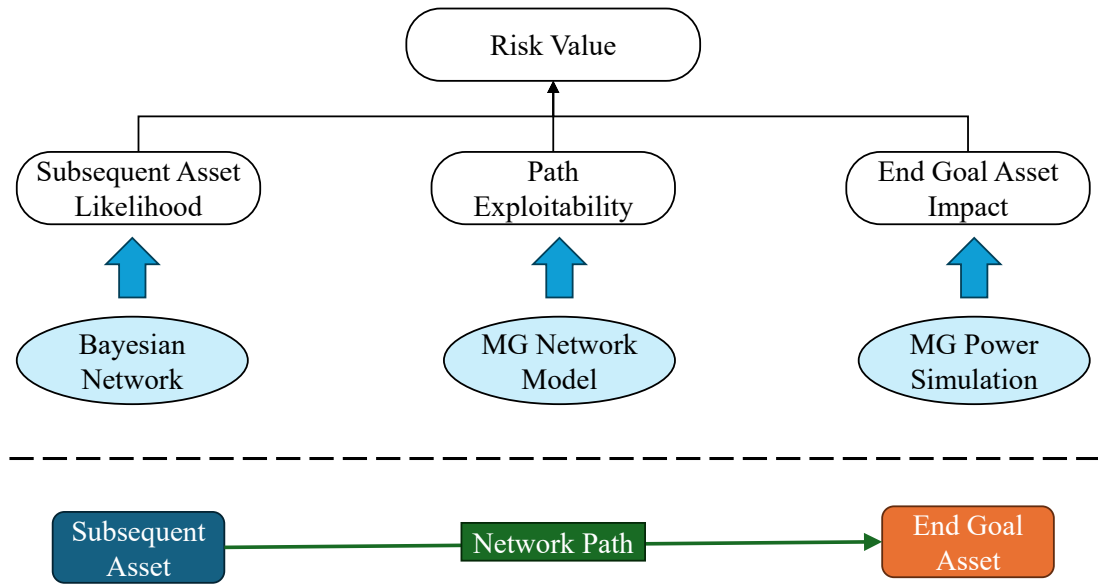


Figure 7.1: Overview of the components involved in calculating risk value.

- Elaborating a method to compute likelihood of subsequent attack steps that follow from an attack hypothesis based on a Bayesian network derived from CTI;
- Determining the path exploitability based on the path distance along the microgrid graph model and the vulnerability of the assets along the path;
- Associating various asset compromises with an impact by assessing results obtained from the microgrid power system simulation model under impact severity criteria; and
- Computing the risk values for the subsequent assets identified during attack step prediction.

7.1. Likelihood of Subsequent Attack Steps According to CTI

In the interest of evaluating what actions an APT might employ after a given stage of their kill chain, the APT behaviour patterns described in available CTI are a valuable source to consider. This section proposes a method to gain insights on how ICS APT groups behave in the microgrid according to available ICS CTI. Specifically, it uses the attack sequence dataset, previously presented in Section 6.7, that contains Tactic, Technique, or Procedure (TTP) sequences corresponding to ICS

APTs. The proposed method assesses relative attack step likelihood based on a discrete Bayesian network built from the attack sequence dataset. The dataset sequences are expanded to include sub-sequences that have not reached the final objective of the original attack sequence. Given an attack hypothesis describing the current stage of an APT attack campaign, the Bayesian network enables the assignment of probabilities to possible future attack steps. The proposed risk assessment incorporates these probabilities into the risk value computations that surround the analysis of the attack hypothesis.

7.1.1. Bayesian Network

A Bayesian network is a directed acyclic graph containing nodes that correspond to a variable, or state, whose value is expressed as a probability distribution [124]. A key feature of the Bayesian network is that the probability of a variable having a particular value is influenced by the values of the nodes with edges directed to the node containing the variable in question. Put another way, node variables in the Bayesian network are affected by nodes found upstream in the network. For example, $p_B(Y)$ is the probability that node B is Y , and has conditional probability $p_B(y|A = Y)$ given that node A is Y , with A having a directed edge whose head connects to B . Nodes can contain continuous variables expressed as probability functions or contain discrete variables expressed as a probability table of mutually exclusive values [124], [125].

A Bayesian network is used to model the probabilities of different possible feature values pertaining to the attack step that follows from a given attack hypothesis. These features are represented as discrete variable nodes in the Bayesian network, with each feature of interest corresponding to one node. The Bayesian network is transparent in that the statistical reasoning for the value of a node can be traced directly to its observed frequency in the dataset under specific known conditions. It also offers the flexibility of being able to provide predictions for different combinations of unknown inputs. For example, it can make a prediction on the next attack step even if the underlying APT is treated as an unknown. The Bayesian network can even work backwards to determine the probability distribution of an upstream node variable given the values of downstream nodes. For example, if node D is influenced by upstream node U , the priority $p_U(Y|D = Y)$ can be calculated when there is evidence that D equals Y . As for datasets, the Bayesian network can be effective even

with smaller sample sizes compared to other machine learning models like neural networks.

7.1.2. Preparing the Attack Sequence Dataset

Various existing works have explored the use of Bayesian networks to support security risk assessment for information systems [126], [127], [128]. They make use of historical data and known system exploits to determine security risks associated with past events and make decisions based on them. The Bayesian network proposed in this thesis distinguishes itself from existing works by centering on the observables and risks of a particular attack campaign. The attack campaign is expressed as a sequence of TTPs that make up the values of the observed features and the features to predict through the Bayesian network. These features include the APT group, the TTPs used by the threat actor, and the set of assets affected by the attack.

The attack sequence dataset used to build the prediction model for attack steps is the same one that was generated following the methodology described in Section 6.7. However, the dataset is not split into a tuning set and a validation set when fitting data into the Bayesian network. Instead, the attack sequence dataset undergoes a separate processing procedure entirely to create the samples used for the Bayesian network.

A preprocessing step is first applied on the attack sequences to produce subsequences that correspond to partially completed attacks. The intent is to prepare a dataset that enables prediction of the features of the attack step following a truncated view of the complete sequence. An attack sequence A_{sq} can be expressed as a list of ordered attack steps $A_{sq} = [ats_1, ats_2, \dots, ats_i, \dots, ats_K]$, where ats_i is the i th attack step and K is the total number of attack steps. Prior to feature value extraction, the preprocessing step introduces a truncation parameter t to break down the sequences into “truncation sets” of four components: (i) the observed subsequence $A_{sq}^O(t)$, (ii) the remaining subsequence $A_{sq}^R(t)$, (iii) the last observed attack step $lt(t)$, and (iv) the prediction target $pt(t)$. These components are defined as follows:

$$\begin{aligned}
A_{sq}^O(t) &= [ats_1, ats_2, \dots, ats_{t-1}] \\
A_{sq}^R(t) &= [ats_t, ats_{t+1}, \dots, ats_K] \\
lt(t) &= ats_{t-1} \\
pt(t) &= ats_t
\end{aligned} \tag{7.1}$$

A visualization of the attack sequence segments that are taken as truncation set components is given in Figure 7.2. To note is that ats_t is shared between $pt(t)$ and the first element of $A_{sq}^R(t)$. Similarly, ats_{t-1} is shared by $lt(t)$ and the last element of $A_{sq}^O(t)$. For each sequence, a separate truncation set is created for each integer value of $t \in [3, K - 1] \cap \mathbb{N}$, where K is the length of the full sequence. For example, if $K = 9$, then there will be six truncation sets across $t \in \{3, 4, 5, 6, 7, 8\}$. Each of these sets becomes a data point that is included in the dataset used to build the Bayesian network. The minimum value of $t = 3$ for the truncation parameter is chosen because the longest attack sequences for the APT33 and FIN7 threat groups only contain four attack steps. Thus, $t = 3$ is the largest minimum value that can be chosen for the truncation parameter that still allows APT33 and FIN7 to be included in the Bayesian network dataset. This choice of minimum value for the truncation parameter means that all attack sequences with length $K < 4$ are excluded from the Bayesian network dataset. In total, 1100 truncation sets are created from the attack sequence dataset. The Bayesian network is fitted to these truncation sets, treating them as the sample data.

Recall that each attack sequence is designated to a perpetrator APT. Also, each of its attack steps are associated with a MITRE ATT&CK tactic, a technique, an ICS asset, and one or more detection data components. From this information, a set of features is extracted from each truncation set for the purposes of building the Bayesian network, with each feature corresponding to a separate network node. The features extracted from the truncation sets are described in Table 7.1, categorized as either observed features or prediction features. The observed variables are treated as independent of each other and are used as inputs from which to compute the conditional probability of the prediction features. This is not to say that there is no correlation between the observed variables. Rather, they are treated as independent to reflect that they are all sourced from separate

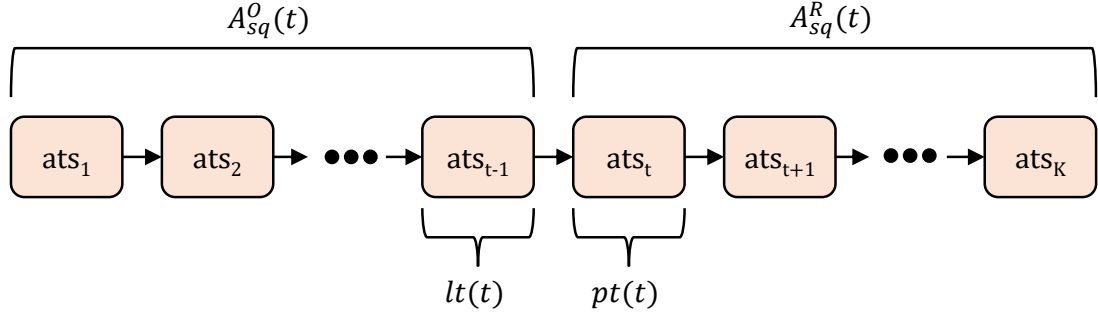


Figure 7.2: Dividing of an attack sequence into a truncation set.

pieces of information available within a given attack hypothesis. This is in contrast to the prediction features that are inferred from the observed features. The $Asset_i$ feature represents a series of binary features for all assets A_i within the set of 10 ICS asset types present in the microgrid model, with $i \in [1, 10] \cap \mathbb{N}$. $Asset_i$ is set to 1 if A_i is affected by an attack step in $A_{sq}^O(t)$ and is otherwise set to 0. As a reminder, the ICS assets present in the microgrid model are Application Server, Programmable Logic Controller (PLC), Workstation, Supervisory Control Server, Control Server, Intelligent Electronic Device (IED), Safety Controller, Virtual Private Network (VPN) Server, Data Historian, and Field I/O. Like $Asset_i$, the $NextTech_j$ feature refers to a series of binary features that indicate whether ATT&CK technique $Tech_j$ is affected by $pt(t)$. A total of 15 different techniques are used across all $pt(t)$ found in the Bayesian network dataset, and thus $j \in [1, 15] \cap \mathbb{N}$. The other features are referred to as “enumerable” to indicate that they have a fixed set of possible values, specifically character string values, that they can exhibit.

7.1.3. Bayesian Network Structure

The structure of the proposed Bayesian network is depicted in Figure 7.3. It is constructed using the attack sequence features previously described in Table 7.1, resulting in a graph with a depth of three. All the observed features have a directed edge to the $NextTactic$ and $NextAsset$ prediction features. This means that assigning a value to the observed variables according to available evidence directly impacts the probability distribution of these target features. This is justified given how the tactic and microgrid asset involved in the next attack step is closely related to the progress an APT has already made. This progress is measured in part by the microgrid assets already affected and the

Table 7.1: Features extracted for the Bayesian network

Category	Feature Name	Feature Type	Description
Observed Features	APT	Enumerable	The perpetrator APT designated to the attack sequence
	$LastTactic$	Enumerable	The ATT&CK tactic used in $lt(t)$
	$LastAsset$	Enumerable	The ATT&CK ICS asset affected by $lt(t)$
	$Asset_{i \in [1,10] \cap \mathbb{N}}$	Binary	Whether ATT&CK ICS asset A_i is affected by an attack step in $A_{sq}^O(t)$
Prediction Target Features	$NextTactic$	Enumerable	The ATT&CK tactic used in $pt(t)$
	$NextAsset$	Enumerable	The earliest ATT&CK ICS asset to appear in $A_{sq}^R(t)$ that differs from the one affected by $lt(t)$ (if no such asset exists, it is set to the asset affected by $lt(t)$)
	$NextTech_{j \in [1,15] \cap \mathbb{N}}$	Binary	Whether ATT&CK technique $Tech_j$ is used for $pt(t)$

stage the APT has reached in the attack kill chain, modeled using the Unified Kill Chain (UKC). In turn, the $NextTactic$ and $NextAsset$ features have a directed edge to the $NextTech_j$ target feature. These dependencies are justified by the direct association that the techniques have to the tactics and assets within the MITRE ATT&CK framework. The Bayesian network presented in this thesis is implemented in Python and leverages the `pgmpy` module [129]. To note is that $NextTactic$, $NextAsset$, and $NextTech_j$ are the next attack elements relative to the given hypothesis. The values of these features may become the part of the set of observed features ($LastTactic$, $LastAsset$, and $Asset_i$) for the hypothesis that includes the alerts that were generated after the attack performed the next step in their attack campaign.

Following its construction, the Bayesian network is fitted to the truncation sets created from the attack sequence dataset. The fitting process automatically computes the “prior” and “posterior” probability distributions for each feature. The former refers to the probability distribution in the absence of any observations related to an active hypothesis, while the latter refers to the distribution that reflects the conditional probabilities that follows from one or more observations [124].

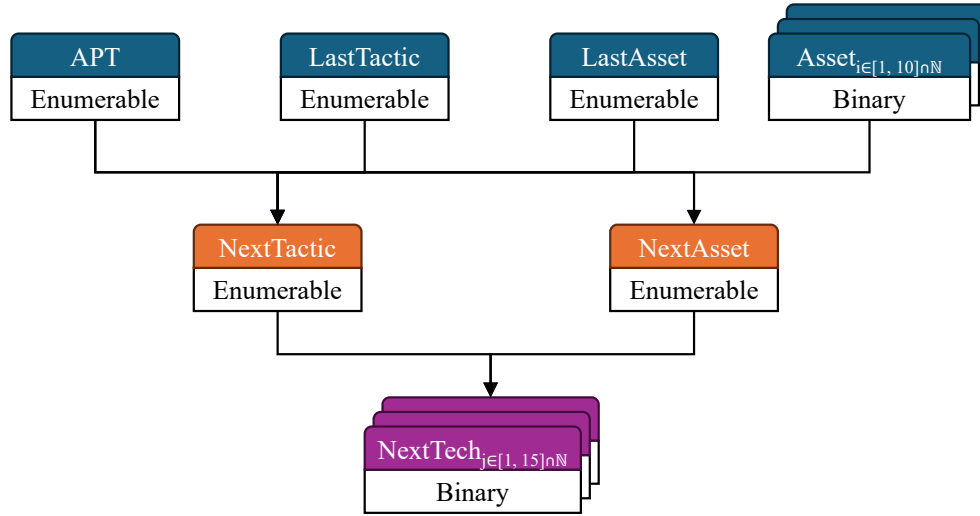


Figure 7.3: Connectivity of attack features within the Bayesian Network.

7.1.4. Measuring Information Gain from Attack Sequence Evidence

In essence, the fitted Bayesian network compiles statistics surrounding the microgrid attack sequence samples, generated based on ICS CTI, at different stages of their kill chain. The compiled data is used to analyze the attack sequence dataset to help identify patterns of behaviour for different sets of known parameters. For example, it may be the case that multiple different APTs tend to use the same TTPs after reaching a particular ICS asset, or perhaps they tend to move laterally to the same asset after having performed a particular tactic. In this way, the Bayesian network offers flexibility for the input parameters and desired outputs.

Beyond indicating the likelihood of specific outcomes, a variable's probability distribution also expresses the degree of certainty associated with its value. For a binary variable, a 50-50 probability of being 1 or 0 indicates the highest level of uncertainty for that variable. Conversely, a 95-5 probability split for a binary variable indicates high confidence that a variable will equal the more likely value. With access to the full probability distribution (where all probabilities add up to 1), the confidence towards that variable can be quantified through the calculation of the information entropy present in the distribution. Given the probability $p_f(v)$ that discrete variable f has value $v \in V$ and the natural logarithm $\ln(p_f(v))$ of that probability, where V is the set of all possible values for f , the entropy $H(f)$ for f is calculated as follows [130]:

$$H(f) = - \sum_{v \in V} p_f(v) \times \ln(p_f(v)) \quad (7.2)$$

Using (7.2), the entropy of a binary variable having a 50-50 split is calculated as ~ 0.6931 , while a 95-5 split yields an entropy of ~ 0.1985 . The lower the entropy, the lower the uncertainty surrounding the value of the variable. In this context, lower uncertainty equates to higher confidence.

The calculation of information entropy can be used to quantify the change in confidence towards the microgrid attack step prediction features as the input parameters surrounding the observed features are modified. Table 7.2 presents an example of how a given probability distribution for the *NextAsset* feature can be used to calculate its entropy, with the given distribution yielding an entropy of 2.1503. By setting the variables of the observed features according to evidence uncovered from observing the active attack sequence, the entropy of the probability distribution for the next targeted asset can be reduced. The entropy reduction on the prediction features is measured for various combinations of observed features having their values assigned. The most salient features result in the largest entropy reductions when their values are assigned. The entropy changes to *NextTactic*, *NextAsset*, and *NextTech_j* that arise from these combinations are given in Table 7.3. The resulting entropy values show that knowing the last tactic used in the attack (*LastTactic*) significantly reduces the uncertainty regarding the next tactic and next technique that will be used. Knowledge on the last asset affected by the attack (*LastAsset*) and on the full set of affected assets (*Asset_i*) also leads to pronounced entropy reduction for the next targeted asset. Having the *APT* feature be known significantly decreases the entropy of the prediction features as well, particularly for predicting the next tactic. This suggests that knowing the perpetrator APT can greatly improve the accuracy of the attack step predictions. It also underscores the importance of accurate APT identification within a microgrid threat hunting framework.

Security operators may choose to tailor their response efforts in accordance to the probability distribution of incoming attack actions provided by the Bayesian network, with targeted responses in cases of high confidence. However, the risk factors that follow after the attacker has taken the predicted action are also an important factor in how best to prioritize security response. These factors are all incorporated into the risk values of the paths extending from a hypothesis. The

Table 7.2: Prior probabilities of the next asset to be targeted

Value of $NextAsset$	Prior Probability
Workstation	0.1675
Data Historian	0.1179
Supervisory Control Server	0.1641
Safety Controller	0.0564
Field I/O	0.0547
Control Server	0.1624
Programmable Logic Controller (PLC)	0.1470
Application Server	0.0291
Intelligent Electronic Device (IED)	0.0513
Virtual Private Network (VPN) Server	0.0496
Total Entropy	2.1503

Table 7.3: Entropy of prediction features with varying amounts of evidence

Known Observed Feature Variables	Entropy		
	$NextTactic$	$NextAsset$	$NextTech_j$
None (prior probabilities)	1.5970	1.9492	0.3080
$LastTactic$	1.0742	1.8492	0.2514
$LastAsset$	1.4224	1.3501	0.2939
$Asset_i$	1.2807	1.0225	0.2653
$LastTactic + LastAsset$	0.8324	1.1480	0.2096
$LastTactic + Asset_i$	0.7570	0.9199	0.2025
$LastAsset + Asset_i$	1.2094	0.8816	0.2462
$LastTactic + LastAsset + Asset_i$	0.6841	0.7871	0.1837
$LastTactic + APT$	0.3421	1.5801	0.1455
$LastAsset + APT$	0.8981	0.9044	0.2033
$Asset_i + APT$	0.8204	0.6673	0.1855
$LastTactic + LastAsset + APT$	0.1514	0.7965	0.1058
$LastTactic + Asset_i + APT$	0.1340	0.6151	0.0997
$LastAsset + Asset_i + APT$	0.8090	0.6197	0.1814
$LastTactic + LastAsset + Asset_i + APT$	0.1269	0.5704	0.0963

intended usage of the proposed risk assessment methodology is to make decisions based on the risk values rather than solely on the values and entropy of the Bayesian probability distribution.

7.1.5. Use of Bayesian Network for Risk Assessment

Using the observed feature values that correspond to a validated attack hypothesis as input, the Bayesian network provides a probability distribution for the next asset that the APT will target for lateral movement and exploitation. To demonstrate the use of the Bayesian network, let us sample a truncation set from an example sequence in the attack sequence dataset. The sampled truncation set corresponds to a partial attack sequence attributed to Sandworm Team that has executed an exploit on Operator Workstation 1 (a $\langle \text{Workstation} \rangle$). It has the following values for the observed feature variables, with the *used* value indicating that the binary variable is equal to 1:

- *APT* : Sandworm Team
- *LastTactic* : $\langle \text{Execution} \rangle$
- *LastAsset* : $\langle \text{Workstation} \rangle$
- *Asset*: $\langle \text{Workstation} \rangle$: *used*

Given these known values as inputs, the Bayesian network yields the following probability distributions for $p(x = y)$, where x is a prediction target feature and y is a possible value for x :

- $p(\text{NextTactic} = \langle \text{Lateral Movement} \rangle) = 1.0$
- $p(\text{NextAsset} = \langle \text{Control Server} \rangle) = 0.4445$
 $p(\text{NextAsset} = \langle \text{Data Historian} \rangle) = 0.2222$
 $p(\text{NextAsset} = \langle \text{Application Server} \rangle) = 0.2222$
 $p(\text{NextAsset} = \langle \text{Virtual Private Network (VPN) Server} \rangle) = 0.1111$
- $p(\text{NextTech}:\langle \text{Valid Accounts} \rangle = \text{used}) = 0.9428$
- $p(\text{NextTech}:\langle \text{Remote Services} \rangle = \text{used}) = 0.3559$
- $p(\text{NextTech}:\langle \text{Lateral Tool Transfer} \rangle = \text{used}) = 0.1043$
- $p(\text{NextTech}:\langle \text{Exploitation of Remote Services} \rangle = \text{used}) = 0.0653$

For the observed and prediction features listed above, all omitted features or enumerable values have a probability of 0%.

In this example, the prediction variables indicate that the attacker is planning to use the ⟨Lateral Movement⟩ tactic next and has a 44.45% chance of moving to a control server from the workstation. It has a 22.22% chance of moving to a database server, a 22.22% chance of moving to an application server, and 11.11% chance of moving to a VPN server. It is also very likely to use the ⟨Valid Accounts⟩ technique. It is also likely to use a combination of complementary techniques from among ⟨Remote Services⟩, ⟨Lateral Tool Transfer⟩, and ⟨Exploitation of Remote Services⟩. These results suggest that, according to the ICS CTI fitted to the Bayesian network, Sandworm Team is most likely to target the control server at this stage of their kill chain. If the *APT* variable is to be omitted, the probabilities of moving laterally to the ⟨Control Server⟩, the ⟨Data Historian⟩, the ⟨Application Server⟩, and the ⟨Virtual Private Network ⟨VPN Server⟩ at this stage become 52.54%, 20.34%, 15.25%, and 11.86%, respectively. This means that the probability of Sandworm Team moving directly to the control server from a workstation is below average for the tested APTs.

The *NextAsset* prediction feature is of particular interest as it factors into the likelihood component of the risk value calculation. This component reflects the likelihood that an APT will move laterally to reach that asset based on their behaviour profile or the behaviours of APTs in general, depending on the amount of trusted information available. The assets with a non-zero probability of being the *NextAsset* value are treated as the candidates for being the subsequent asset of the attack campaign. This set of candidate assets, corresponding to non-zero probabilities for *NextAsset* with truncation parameter t , is labeled as NA_t for future reference. Security operators may adjust the weight placed on the difference in targeting probability between subsequent asset candidates according to the context they consider and the mitigation strategy they adopt. The likelihood contribution $C^L(A)$ for subsequent asset A is defined as follows:

$$C^L(A) = (p(\text{NextAsset} = A))^{W^L} \quad (7.3)$$

Here, W^L is the weight parameter chosen for the likelihood contribution. It is a positive real value that controls the importance placed on subsequent assets having a higher probability of being

the value of the *NextAsset* feature in the Bayesian network. Noting that W^L is used as an exponent and that $p(\text{NextAsset}) \in [0, 1]$, it follows that W^L values between 0 and 1 de-emphasize the difference between the probability of each possibility, with $C^L(A)$ approaching 1 for all subsequent assets as W^L approaches 0. If W^L equals 1, then the likelihood contribution for a subsequent asset is equal to its probability of being targeted next according to the Bayesian Network. Values greater than 1 for the weight exacerbate the difference in probabilities in the Bayesian distribution. Taking the previous example where $p(\text{NextAsset} = \langle \text{Control Server} \rangle)$ is 0.4445 and $p(\text{NextAsset} = \langle \text{Database Server} \rangle)$ is 0.2222, having W^L equal 2 results in likelihood contributions of 0.1976 and 0.0494, respectively. The ratio between the contributions for targeting a $\langle \text{Control Server} \rangle$ and targeting a $\langle \text{Database Server} \rangle$ rises from about 2:1 to about 4:1 when W^L is increased from 1 to 2.

The overall posterior probability of each possible value of *NextAsset*, across all sets of truncations in the Bayesian network dataset and with all observed feature values being known, is given in Table 7.4. The posterior probabilities are also isolated for each APT in Table 7.5. It can be seen that attacks by APT33 and FIN7 are localized to engineering workstations. Dragonfly, FIN6, and Oil-Rig are also more specific in the assets they target. These APTs contrast Lazarus Group, Sandworm Team, and TEMP.Veles, whose targeted assets have a wider spread. By isolating the probabilities according to APT, it can be seen that, for most of the APTs, the control servers (including supervisory control servers) are often the next targeted asset.

7.2. Path Exploitability

When anticipating the actions of APTs aiming to cause harmful physical impacts on the microgrid, it is important to not only consider the microgrid assets that are most critical for maintaining safe operation of the physical system, but also how easily an active APT may be able to exploit those assets given their current set of privileges. The path exploitability represents the ease with which an APT may reach and compromise a particular asset in the microgrid network from a given starting point (the subsequent asset). The path exploitability is a numeric value with which the difficulty of traversing along different attack paths, from the perspective of the attacker, can be compared on a relative scale. The microgrid system model described in Section 4.2.1 is used here for the purpose

Table 7.4: Overall next target posterior probability for all ATT&CK ICS assets

Value of $NextAsset$	Posterior Probability
Application Server	0.0409
Control Server	0.1900
Data Historian	0.0900
Field I/O	0.0300
Intelligent Electronic Device (IED)	0.0309
Programmable Logic Controller (PLC)	0.0836
Safety Controller	0.0291
Supervisory Control Server	0.1373
Virtual Private Network (VPN) Server	0.0382
Workstation	0.3300

of demonstrating the calculation of path exploitability.

Calculating the path exploitability involves the computation of two subcomponents: the path exploit resistance and the path distance factor. The former accounts for the ease of compromising the assets along the path, while the latter accounts for the number of network hops between the assets at the start and end of the path.

7.2.1. Exploit Resistance

An APT may forego attempting to inflict the highest amount of damage to the system in favor of attempting an attack that is more likely to succeed. Attack success likelihood considers the system security measures that are in place and the capabilities of security operators with respect to detection and response. For example, it may be easier to uncover and exploit a vulnerability within a particular field I/O device than it is to exploit a server machine with a recent operating system that regularly receives security update patches. As such, the risk assessment methodology proposed in this thesis considers how ease of asset exploitation can be factored into the risk calculations.

The microgrid system model considered in this thesis does not specify any device models, brands, or operating systems for its assets. It does, however, map each asset to an ICS asset defined in the MITRE ATT&CK framework. It can be presumed that the ICS asset types defined by MITRE differ in how easy it is for an attacker to exploit them for the purposes of continuing their cyber kill chain. It can also be presumed that an attacker can reuse exploits on assets having an ICS asset type that they have already successfully exploited. With these assumptions, the assets

Table 7.5: APT next target posterior probability for all ATT&CK ICS assets

Value of $NextAsset$	Posterior Probability			
	APT33	Dragonfly	FIN6	FIN7
Application Server	0	0.0146	0	0
Control Server	0	0.1460	0.2727	0
Data Historian	0	0.0146	0.2061	0
Field I/O	0	0	0	0
IED	0	0	0	0
PLC	0	0	0.1757	0
Safety Controller	0	0	0	0
Supervisory Control Server	0	0.0438	0.1758	0
VPN Server	0	0	0	0
Workstation	1	0.7810	0.1697	1

Value of $NextAsset$	Posterior Probability			
	Lazarus Group	OilRig	Sandworm Team	TEMP.Veles
Application Server	0.0739	0	0.0602	0.0633
Control Server	0.2309	0.2376	0.1365	0.1582
Data Historian	0.1616	0.1492	0.0482	0.0506
Field I/O	0.0698	0	0.0642	0.0317
IED	0.0551	0	0.0723	0.0443
PLC	0.0698	0.1657	0.0643	0.0316
Safety Controller	0.0698	0	0.0562	0.038
Supervisory Control Server	0.1295	0.1160	0.1446	0.2089
VPN Server	0.1163	0	0.0643	0.0633
Workstation	0.0233	0.3315	0.2892	0.3101

in the microgrid system model can be given a relative “exploit resistance value” according to the ATT&CK ICS asset to which they are mapped, with higher values indicating that it is more difficult for an attacker to compromise the asset. At this stage of research, the exploit resistance values are treated as parameters that can be adjusted to account for different security postures and levels of security hardening. It is expected that system designers for a particular ICS network will assign exploit resistance values to their various assets that best match the specifics of their deployment.

The exploit resistance value for an asset A_i is denoted as $ResV(A_i)$ and is set to a positive real number, i.e., a number greater than zero. While the output of $ResV(A_i)$ can be made dynamic based on the exploits already believed to have been performed by a suspected APT, the heterogeneity of the assets in the modeled microgrid system at each Purdue level affords the $ResV(A_i)$ to be kept static at this stage of research. The total exploit resistance $Res(P)$ of path $P : A_S \rightarrow A_E$ between

a given starting asset A_S and end goal asset A_E is calculated as the sum of all exploit resistance values corresponding to the assets $\{A_1^P, A_2^P, \dots, A_k^P\}$ along the path, where k represents the number of unique ICS assets on the path. Mathematically, $Res(P)$ is expressed as follows:

$$Res(P) = \sum_{i=1}^k ResV(A_i^P) \quad (7.4)$$

To designate the exploit resistance values of each asset, this thesis takes the approach of defining resistance profiles that correspond to different assumptions and/or policies. In this thesis, three mutually exclusive resistance profiles to choose from are considered:

- *Profile A*: all assets have an exploit resistance value of 1.0
- *Profile B*: the exploit resistance value of an asset is proportional to its Purdue level.
- *Profile C*: the exploit resistance value of an asset is highly proportional to its Purdue level.

Profile A corresponds to the default assumption of treating every asset as equally vulnerable to exploitation. Though perhaps unrealistic, it provides a risk model where path exploitability is solely a function of the path distance. *Profile B* and *Profile C* follow from the assumption that the boundaries to external entities, i.e., the Remote Access Servers (RASs), will have the most stringent security protections, and that the security measures implemented and/or the computational resources available for security features diminish as the devices lie closer to the microgrid field lines. This is not a wholly unreasonable assumption given the prevalence of legacy equipment in the lower microgrid Purdue levels and the trust placed towards operators working for the microgrid system owner. The exploit resistance value $ResV(A_i)$ for asset A_i at Purdue level PL_{A_i} is calculated for the three profiles using the following linear equations:

$$\begin{aligned} \text{Profile A : } ResV(A_i) &= 1.0 \\ \text{Profile B : } ResV(A_i) &= 0.65 + 0.10 \times PL_{A_i} \\ \text{Profile C : } ResV(A_i) &= 0.30 + 0.20 \times PL_{A_i} \end{aligned} \quad (7.5)$$

Table 7.6 provides the exploit resistance values of the assets when using *Profile B* or *Profile C*. The values for *Profile A* are omitted from the table since they are all the same. The table also lists

Table 7.6: Exploit resistance values for Profile B and Profile C

Microgrid Asset	Purdue Level	Profile B Resistance Value	Profile C Resistance Value
Remote Access Server	3.5	1.00	1.00
Operator Workstation	3	0.95	0.90
Maintenance Workstation	3	0.95	0.90
Database Server	3	0.95	0.90
Application Server	3	0.95	0.90
Telecontrol Server	3	0.95	0.90
Microgrid Central Controller (MGCC)	3	0.95	0.90
Supervisory Controller	2	0.85	0.70
Protection Relay	1	0.75	0.50
Point of Common Coupling (PCC)	1	0.75	0.50
Controller (ESS, Load)	1	0.75	0.50
Merging Unit (MU)	0	0.65	0.30
Circuit Breaker (CB)	0	0.65	0.30

the Purdue level of the assets.

7.2.2. Path Distance Factor

The proximity of an APT to the various microgrid assets affects the risk that the APT poses. The risk assessment methodology proposed in this thesis measures the proximity of two assets using the network hop distance for the shortest path between the assets. Thus, the shortest path P between a subsequent asset candidate and a given end goal asset is used for the calculation of path risk for those assets, with the APT using the end goal asset directly to impact the microgrid power system. For reference, the network hop distance between the various microgrid assets is given in Table 7.7. Note that the hop distance includes transitions between the host network interfaces and the host itself, with it taking three hops to transit from one host operating system to another. The hop distance of a path is zero in the case where the subsequent asset is also the end goal asset. Also, the ESS/Load controllers, the CBs, and the MUs have a distance to their local supervisory controller and relay that differs from the distance to those belonging to different Distributed Energy Resources (DERs) or loads.

For the purposes of this thesis, it is assumed that an attack path becomes more difficult to fully exploit as the number of required lateral movement actions to complete the path increases.

Table 7.7: Network hop distance between various microgrid assets of interest

Microgrid Asset	Telecontrol Server and MGCC	Supervisory Controllers (local, non-local)	Relays (local, non-local)	PCC
Operator (OP) Workstation	3	6	9	9
Database Server	4	6	6	9
Application Server	4	6	6	9
Maintenance (MT) Workstation	6	3	3	6
Remote Access Server (OP)	6	9	12	12
Remote Access Server (MT)	9	6	6	9
ESS/Load Controller	6	3, 7	4, 7	10
DER/Load Circuit Breaker	6	3, 7	3, 7	10
DER/Load Merging Unit	6	3, 7	3, 7	10
PCC Merging Unit	6	3, 7	7	4

Following this assumption, a path distance factor $Pf(h_P)$ for network hop distance h_P of path P is defined as follows:

$$Pf(h_P) = (K_{Pf})^{\alpha \times h_P} \quad (7.6)$$

A higher path distance factor indicates that it is more challenging for the attacker to successfully exploit all the assets along the path, all else being equal. It is modeled as an exponential function having parameters $K_{Pf} > 1.0$ and $\alpha > 0$ as real values corresponding to the base and exponential coefficient, respectively. The constraints on these parameters ensure that the function strictly increases as hop distance h_P increases. This assumes that the complexity of having to compromise a series of assets, without being thwarted by security operators, rises aggressively as the number of required network hops increases. The path distance factors yielded by example combinations of K_{Pf} and α across network hop distances [1, 12] are given in Table 7.8 as per (7.6). The upper limit of 12 represents the largest distance found between any two assets within the microgrid model under consideration. The results for hop distance h_P equal to zero are omitted since the result for the zero input always equals one (1). In the interest of demonstrating candidate asset risk values whose difference in orders of magnitude are reasonably constrained, the parameters for K_{Pf} and α are set to 1.5 and 0.75, respectively. With this choice of parameters, the range of possible path distance factors within the considered microgrid system is about 38.5.

Table 7.8: Path distance factor computed for different parameter combinations

Hop Distance	$K_{Pf} = 1.1$			$K_{Pf} = 1.3$		
	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
1	1.0488	1.0741	1.1000	1.1402	1.2175	1.3000
2	1.1000	1.1537	1.2100	1.3000	1.4822	1.6900
3	1.1537	1.2392	1.3310	1.4822	1.8046	2.1970
4	1.2100	1.3310	1.4641	1.6900	2.1970	2.8561
5	1.2691	1.4296	1.6105	1.9269	2.6748	3.7129
6	1.3310	1.5356	1.7716	2.1970	3.2565	4.8268
7	1.3960	1.6493	1.9487	2.5050	3.9646	6.2749
8	1.4641	1.7716	2.1436	2.8561	4.8268	8.1573
9	1.5356	1.9028	2.3579	3.2565	5.8765	10.6045
10	1.6105	2.0438	2.5937	3.7129	7.1544	13.7858
11	1.6891	2.1953	2.8531	4.2334	8.7103	17.9216
12	1.7716	2.3579	3.1384	4.8268	10.6045	23.2981

Hop Distance	$K_{Pf} = 1.5$			$K_{Pf} = 1.7$		
	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
1	1.2247	1.3554	1.5000	1.3038	1.4888	1.7000
2	1.5000	1.8371	2.2500	1.7000	2.2165	2.8900
3	1.8371	2.4900	3.3750	2.2165	3.3000	4.9130
4	2.2500	3.3750	5.0625	2.8900	4.9130	8.3521
5	2.7557	4.5745	7.5938	3.7681	7.3145	14.1986
6	3.3750	6.2003	11.3906	4.9130	10.8898	24.1376
7	4.1335	8.4039	17.0859	6.4058	16.2128	41.0339
8	5.0625	11.3906	25.6289	8.3521	24.1376	69.7576
9	6.2003	15.4389	38.4434	10.8898	35.9360	118.5879
10	7.5938	20.9259	57.6650	14.1986	53.5016	201.5994
11	9.3004	28.3630	86.4976	18.5127	79.6533	342.7190
12	11.3906	38.4434	129.7463	24.1376	118.5879	582.6222

7.2.3. Computing Path Exploitability

The proposed attack step prediction methodology determines the overall path exploitability $E(P, h_P)$ of a given path P with h_P network hops, using the total exploit resistance $Res(P)$ (7.4) and the path distance factor $Pf(h_P)$ (7.6) as follows:

$$E(P, h_P) = \frac{Res^{min}}{Res(P) \times Pf(h_P)} \quad (7.7)$$

Here, Res^{min} is the lowest exploit resistance value among the ones assigned to assets in the system. It acts as a normalizing coefficient that ensures the value of $E(P, h_P)$ lies between 0 and 1.

Path exploitability $E(P, h_P)$ is equal to 1 in the case where the subsequent asset is also the end goal asset while at the same time having the lowest exploit resistance among assets in the system. The path distance factor $Pf(h_P)$ already has a minimum value of 1, and thus does not need to be factored into the normalization when constraining path exploitability to fall between 0 and 1.

7.2.4. Use of Path Exploitability for Risk Assessment

The path exploitability $E(P, h_P)$ is incorporated into the risk value calculation. Like with the likelihood contribution derived from weighing the probability of the subsequent asset candidate, a weight parameter is introduced to control the extent in which path exploitability contributes to the overall risk value. The path exploitability contribution $C^E(P, h_P)$ of a path P with h_P network hops is calculated as follows:

$$C^E(P, h_P) = (E(P, h_P))^{W^E} \quad (7.8)$$

The W^E parameter, used as an exponent, is the weight applied to the path exploitability. It is set to a positive real value. Just like the likelihood weight W^L , increasing W^E exacerbates the difference between two path exploitability values. Higher W^E values will have $C^E(P, h_P)$ favor paths that have higher path exploitability. Lower W^E values lead to $C^E(P, h_P)$ having less influence on the risk calculations. Notably, $C^E(P, h_P)$ approaches 1 as W^E approaches zero, regardless of the path exploitability.

7.3. Impact Evaluation

There are multiple assets in the microgrid that influence the operation of the microgrid power system network and its power quality. Field devices, IEDs, and control servers are among the assets whose compromise permits an attacker to negatively impact the microgrid with little delay. The extent of the damage resulting from asset compromises is not equal across the various microgrid assets. Some may permit an attacker to destabilize the microgrid voltage and frequency, while others may allow the attacker to deny electricity to a portion of consumers. Others still may put the microgrid at risk of a complete blackout when attacked by a knowledgeable threat actor.

This section elaborates on the impact component of the risk value calculation. To that end, it explores the potential impacts that the microgrid system under consideration, detailed in Chapter 4, could experience. The microgrid co-simulation testbed is used to simulate the impacts of several harmful operations on microgrid power quality. These harmful operations are mapped to assets in the microgrid whose compromise can enable the execution of the associated operation. The assets are associated with a numeric impact value based on the extent of the damage resulting from the harmful operations tested on each asset. A semi-quantitative scoring method is used to determine the impact values. This scoring method is derived from the impact analysis methodology followed by the National Electric Sector Cybersecurity Organization Resource (NESCOR) [131] for failure scenario assessment.

7.3.1. Impact Criteria

The microgrid co-simulation testbed developed in this thesis is capable of providing insight into some of the damage that can be expected from an attacker compromising a given asset. The nature of the impacts potentially realized by an attack is scored using a set of criteria. There are many ways in which an attack can harm the microgrid and the energy consumers relying on it. Some of these include monetary losses, damage to reputation, degradation of energy delivery, and even human injury. However, this thesis does not speculate on all types of harm. Its scope is limited to the damages that are more directly visible within the context of the microgrid co-simulation testbed.

NESCOR has defined potential failure scenarios for power systems and developed a risk-based ranking method to prioritize them. Their efforts are described in a report [132] published by the Electric Power Research Institute (EPRI). As part of their risk analysis, NESCOR establishes 15 impact criteria with which to give a total impact for the failure scenario under consideration. The values assigned to each criterion can be 0, 1, 3, or 9. A value of 0 corresponds to no impacts relevant to the criteria, while a value of 9 indicates a very high impact for the criteria. This thesis uses a subset of the NESCOR impact criteria in order to score the impacts observed when simulating attacks on the microgrid co-simulation platform that originate from a given asset. Table 7.9 shows the impact criteria from among those established by NESCOR that correspond to impact criteria considered in this thesis. Specifically, the NESCOR criteria that inspire the impacts considered

Table 7.9: NESCOR impact criteria used for proposed microgrid risk assessment

Impact Criteria	Is Used
System Scale	✓
Public safety concern	
Workforce safety concern	
Ecological concern	
Financial impact of compromise on utility	
Restoration costs	✓
Negative impact on generation capacity	✓
Negative impact on the energy market	
Negative impact on the bulk transmission system	✓
Negative impact on customer service	
Negative impact on billing functions	
Destroys goodwill toward utility	
Immediate economic damage	
Long term damage	
Loss of privacy for stakeholders	

in this thesis are 1) the scale of the affected system, 2) restoration costs, 3) impact on generation capacity, and 4) impact on bulk transmission. Regarding bulk transmission, the impact analysis used here considers impact on the microgrid distribution system rather than the bulk transmission system. This is because the microgrid is being tested in islanded mode and thus does not directly impact the bulk transmission system of the main grid. The microgrid main bus is rated at 25 kV, which falls well below what can be considered a transmission line and into the range of distribution [133]. Exploration of additional impact criteria can be explore in potential future work. The scores to set for the criteria when assessing a simulated microgrid attack are based on a relative scale between the least severe and most severe damages associated with the criteria.

7.3.2. Impact Scoring for Asset Compromise

An impact score is determined for various microgrid assets based on the extent of the damage that may be realized from a threat actor who has compromised the asset. The extent of the damages

caused by an attack is gauged from the simulated power dynamics observed during the simulation of the microgrid power model described in Section 4.1.1. Specifically, the changes to the simulated power dynamics are observed through the HYPERSIM software using the ScopeView feature.

The impact score $IS(s)$ for a given attack scenario s is found by considering how it scores on the individual impact criteria. For each criterion i , an impact criteria value $ICV_i^s \in \{0, 1, 3, 9\}$ for the scenario is assigned for that criterion based on the impacts observed from simulating the attack on the microgrid model. The impact score is the total of the impact criteria values across N criteria. This is formalized as follows:

$$IS(s) = \sum_{i=1}^N ICV_i^s \quad (7.9)$$

The scoring system applied to the chosen impact criteria is given in Table 7.10. The system scale criteria consider the number of loads that are negatively affected by the attack on the microgrid. The system scale criteria range only goes up to 3 rather than 9. This is because the scope of the microgrid model simulation is currently constrained to the islanded microgrid. Thus, the scale of the damage does not extend to the rest of the smart grid as far as power system safety is concerned. The quantification of restoration costs is simplified to the number of DERs that may require maintenance or even replacement due to being exposed to highly unsafe line conditions, such as massive fluctuations in current or voltage. Negative impacts on generation capacity are quantified as a percentage of loss in total capacity, as is done by NESCOR. As previously stated in Section 4.1.1, the generation capacity of the CHP, wind farm, and solar PV array are 10 MW, 6.6 MW, and 1.5 MW, respectively. ESS 1 has a capacity of 1 MW, while ESS 2 and ESS 3 each have a generation capacity of 2 MW. This gives a total generation capacity of 23.1 MW, relative to which the percentage of generation loss is measured. As for the impact on the microgrid distribution system, the impact score considers the degree of power quality degradation experienced by the high voltage bus of the microgrid, i.e., the main bus. The worst-case scenario for microgrid distribution would be a total blackout resulting in all power delivery being cut off.

The impact score of a given asset is equal to the highest impact score from among the set of tested attack scenarios that can be realized from the compromise of the asset. Given an attack

Table 7.10: Ranges of conditions for each impact criteria value

#	Impact Criteria	Score			
		0	1	3	9
1	System Scale	No loads affected	One load affected	Two or more loads affected	N/A
2	Restoration costs (equipment damage)	No DER damage	Possible damage to one DER	Possible damage to two DERs	Possible damage to three or more DERs
3	Negative impact on generation capacity	No visible loss	<10% loss	<20% loss	>20% loss
4	Negative impact on the microgrid distribution system	No effect on main bus	Minor disturbance on main bus	Instability on main bus	Total loss of power delivery (blackout)

scenario set $\{s_1, s_2, s_3, \dots\}$ that can be launched with the compromise of asset A , the impact score $IS(A)$ is defined, through the use of (7.9) to calculate $IS(S)$, as follows:

$$IS(A) = \max(IS(S)), S \in \{s_1, s_2, s_3, \dots\} \quad (7.10)$$

Furthermore, the impact scores can be normalized to fall within a range of $[0, 1]$ by dividing the impact score $IS(A)$ by the largest possible impact score IS^{max} . Tallying the highest possible impact values ICV_i^{max} for the chosen impact criteria yields a maximum impact score $IS^{max} = 30$. The normalized impact score $IS^N(A)$ is formalized as follows:

$$IS^N(A) = \frac{IS(A)}{IS^{max}} \quad (7.11)$$

The attack scenarios simulated on the microgrid model that are used to determine asset impact scores include the following:

- Tripping of a DER line CB to disconnect the line
- Tripping of a load feeder line CB to disconnect the line
- Setting the power setpoint of one ESS to the maximum value
- Setting the power setpoint of one ESS to the minimum value

- Setting the power setpoint of all ESSs to the maximum value
- Setting the power setpoint of all ESSs to the minimum value
- Reducing the reported microgrid frequency to a very low value

The tripping of a power line of a particular DER or load is presumed to be possible through the compromise of either the relay or the supervisory controller responsible for that DER or load. In the case of a compromised relay, the injection of an IEC 61850 GOOSE trip command can be used to trip the CB. The relay is also assumed to be able to send GOOSE trip commands at the behest of the supervisory controller managing the line, and thus a compromised supervisory controller can also be used to launch a line disconnection attack. The line disconnection attacks are also possible from the telecontrol server, but with the difference being that a compromised telecontrol server can repeat the attack for all the microgrid lines and can thus disconnect multiple lines at once. The setting of a single ESS to the maximum or minimum setpoint value is assumed to be realizable through the compromise of the supervisory controller responsible for the control of the ESS. The setpoint being at the maximum value causes the ESS to inject power into the microgrid at full capacity, while setting it to the minimum value causes the ESS to absorb power from the microgrid to the best of its ability. Setting multiple ESSs to the maximum or minimum power setpoint is treated as a possibility that arises from the compromise of the MGCC, which is normally responsible for performing microgrid frequency control via the adjustment of the ESS setpoints. A compromise on the PCC would presumably allow for a modification attack on the microgrid frequency measurements reported to the MGCC. Notably, attacks on the integrity of the MGCC messages used for frequency control are also explored in Section 5.2.5 as part of the evaluation of the active monitoring platform proposed in Chapter 5 of this thesis. The assets associated with at least one simulated microgrid attack when compromised are listed in Table 7.11.

The attacks simulations conducted for this impact analysis are not exhaustive. Notably, the simulated attacks produce effects that are immediately apparent to microgrid operators. That is, they do not make any attempts to be stealthy so as to go unnoticed. An analysis that includes stealthy attacks could factor in the chance of being noticed as an additional impact criteria.

Table 7.11: Assets associated with the microgrid attacks being simulated

Simulated Microgrid Attack	Compromised Assets
Tripping of a DER line	Relay Circuit Breaker Supervisory Controller (DER) Telecontrol Server
Tripping of a load feeder line	Relay Circuit Breaker Supervisory Controller (Load) Telecontrol Server
Setting the power setpoint of one ESS to the maximum value	ESS Controller Supervisory Controller (ESS) MGCC
Setting the power setpoint of one ESS to the minimum value	ESS Controller Supervisory Controller (ESS) MGCC
Setting the power setpoint of all ESSs to the maximum value	MGCC
Setting the power setpoint of all ESSs to the minimum value	MGCC

7.3.2.1 Simulated Attack Example: Disconnecting the Wind Farm

To help demonstrate the process used to determine asset impact score, the impact of a CB tripping attack on Wind Turbine Generators (WTGs) is presented. A forced trip of the wind farm CB during normal operation of the modeled microgrid causes notable disturbances to the power quality. The wind farm ceases to deliver any power to the microgrid once its CB is tripped by the attacker. The impact of the attack on the simulated power system dynamics is observed using ScopeView on HYPERSIM. Figure 7.4 shows the ScopeView output for microgrid frequency that occurs after the wind farm CB is tripped at around the 0.3s mark. The frequency undergoes a sudden drop, followed by an oscillation of decreasing magnitude in an attempt to stabilize the grid. Helping to explain this behaviour are the active power measurements for the other DERs within the simulation. Figure 7.5 shows the power output of the solar PV array, wind farm, and CHP. Figure 7.6 shows the power output of the ESSs, as well as the power being delivered to the loads. The measurements show a loss of 5 MW at the wind farm, which is followed by the CHP and ESSs increasing their power output to make up for the loss of generation. Another notable event is the massive spike in power output seen

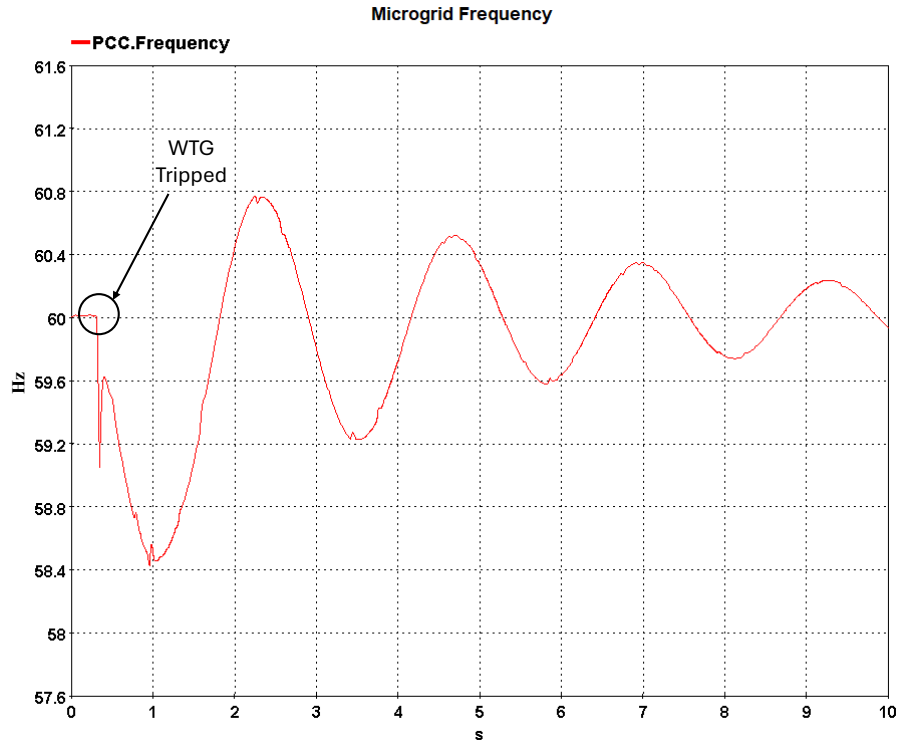


Figure 7.4: Impact on microgrid frequency due to WTG tripping attack.

momentarily at the solar PV array at around the 1s mark. As an additional complication, the line conditions immediately following the tripping of the wind farm lead to the ESS 3 relays sending a trip command to disconnect the ESS 3 CB as a protection response. This tripping response renders ESS 3 unable to participate in frequency control and power balancing. Figure 7.6 also shows the effect that this attack has on the loads. The loads experience temporary instability, followed by some oscillations in power delivery whose period roughly matches with the frequency oscillation.

Scoring the impact of the tripping attack on the wind farm according to Table 7.10, the score for system scale is 3 given that all the loads are affected. The score for restoration costs is 1 in light of the potentially unsafe power spike at the solar PV array. The score for negative impact on generation capacity is 9, given that the disconnection of the wind farm corresponds to the full loss of its 6.6 MW generation capacity while the remaining DERs have a capacity totaling 16.5. Finally, the score for negative impact on the microgrid distribution system is 3 due to the sizable frequency oscillations experienced by the microgrid. In total, the impact score for the tripping attack of the wind farm is 16.

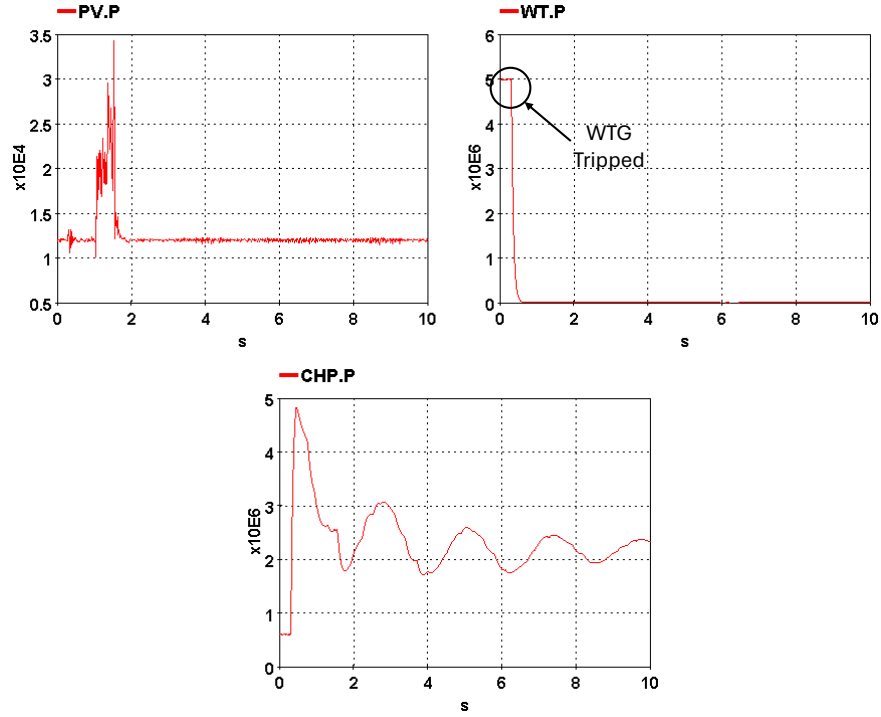


Figure 7.5: Impact on PV, WTG, and CHP due to WTG tripping attack.

7.3.2.2 Impact Scores Across All Microgrid Assets

The impact score $IS(A)$ of a given asset A is equal to the highest impact score among attacks that directly involve the asset. The impact scores for all microgrid assets directly involved in one or more attacks simulated on the islanded microgrid are given in Table 7.12 along with their normalized value. The numbered criteria correspond to the numbers given to the criteria listed previously in Table 7.10. Notably, the ESS supervisory controllers, the telecontrol server, and the MGCC are involved in more than one of the simulated attacks. For future reference, the set of assets listed in the table, having a strictly positive impact score, is labeled as ISA .

According to the microgrid power system simulation, the disconnection of the CHP, which can be launched from the CHP equipment or the telecontrol server, results in not only the highest impact among the tested attacks, but the highest possible impact rating according to the impact criteria being used. The microgrid frequency and voltage become completely unstable with the loss of the CHP synchronous generator, which provides most of the microgrid generation capacity. The

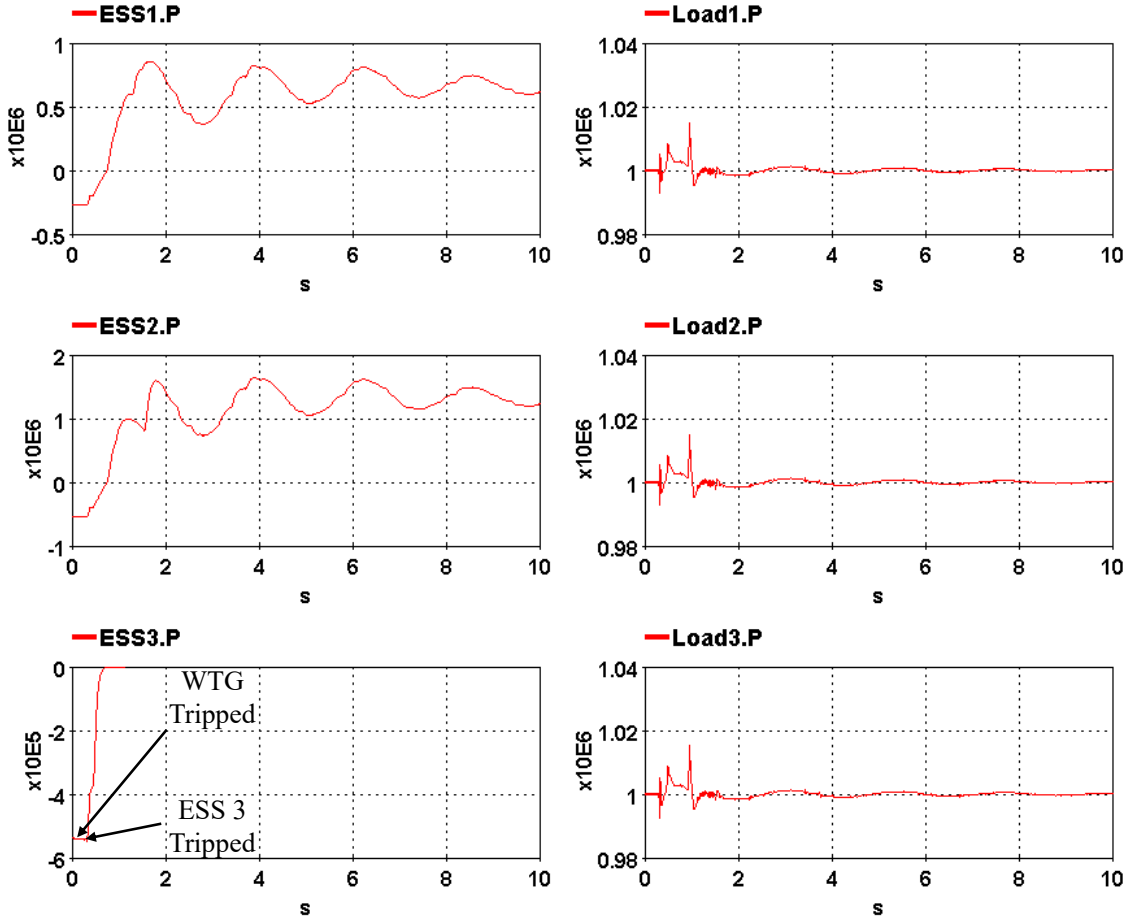


Figure 7.6: Impact on ESSs and loads due to WTG tripping attack.

loss of the CHP causes the microgrid to experience a blackout as the DERs are all disconnected by the protection relays after a short period of time. Consequently, all the microgrid loads are deprived of the power that the microgrid would normally generate. The next highest impact comes from setting all the ESSs to the maximum setpoint, which causes a sharp rise in the microgrid frequency to a level that requires a full disconnection of the microgrid. This attack can be realized directly from the MGCC or by manipulating the PCC frequency measurements as described in Attack I in Section 5.2.5 and whose impact is shown in Figure 5.8. It is also observed that the three ESSs do not cause the same impacts when disconnected or when their setpoints are manipulated. The disconnection of ESS 3 causes some disturbance to the loads that are not seen from the disconnection of ESS 1 or ESS 2, while setting the ESS 2 setpoint to the maximum value produces a greater disturbance compared to the manipulation of ESS 1 and ESS 3 setpoints.

Table 7.12: Impact scores assigned to assets directly associated with attacks

Microgrid Asset A	Criteria #				Impact	Normalized
	1	2	3	4	$IS(A)$	$IS^N(A)$
Telecontrol Server	3	9	9	9	30	1.0
MGCC	3	0	9	9	21	0.7000
ESS 1: Relay / CB	0	0	1	1	2	0.0667
ESS 2: Relay / CB	0	0	1	1	2	0.0667
ESS 3: Relay / CB	3	0	1	1	5	0.1667
ESS 1: Supervisory Controller	3	0	0	1	4	0.1333
ESS 2: Supervisory Controller	3	0	0	3	6	0.2000
ESS 3: Supervisory Controller	3	0	1	1	5	0.1667
ESS 1: (Setpoint) Controller	3	0	0	1	4	0.1333
ESS 2: (Setpoint) Controller	3	0	0	3	6	0.2000
ESS 3: (Setpoint) Controller	3	0	0	1	4	0.1333
Load 1: Relay / CB / Supervisory Controller	1	0	0	1	2	0.0667
Load 2: Relay / CB / Supervisory Controller	1	0	0	1	2	0.0667
Load 3: Relay / CB / Supervisory Controller	1	0	0	1	2	0.0667
CHP: Relay / CB / Supervisory Controller	3	9	9	9	30	1.0
Wind Farm: Relay / CB / Supervisory Controller	3	1	9	3	16	0.5333
PCC: Supervisory Controller / PCC	3	0	9	9	21	0.7000

7.3.3. Use of Impact Score for Risk Assessment

The impact score for an asset is incorporated into the risk value calculation alongside the subsequent asset likelihood and the path exploitability. Specifically, the impact contribution $C^{IS}(A_E)$ computation uses the normalized impact score $IS^N(A_E)$. The impact score, in turn, is computed on the end goal asset A_E that is part of the exploit path $P : A_S \rightarrow A_E$. Making use of a distinct weight parameter W^{IS} for impact, the impact contribution is calculated as follows:

$$C^{IS}(A_E) = (IS^N(A_E))^{W^{IS}} \quad (7.12)$$

As with W^L and W^E , the W^{IS} parameter is a real number greater than zero. It controls the degree to which differences in impact score affect the impact contribution. Higher values for W^{IS} lead to higher risk values being assigned for final assets with higher impact score, and lower W^{IS} values downplay the influence of impact on risk value calculation relative to the subsequent asset likelihood and path exploitability.

7.4. Measuring Risk

The attack step prediction methodology compares the risks posed by the various possible options an APT has when continuing their attack campaign. To quantify these risks on a relative scale, a risk value is computed for different attack paths extending from a given attack hypothesis.

An attack hypothesis has numerous combinations of subsequent asset and end goal asset, with each combination having a corresponding shortest path across the microgrid network. Given the subsequent asset candidates in NA_t at truncation t and the positive impact score assets in ISA , the subsequent asset $A_S \in NA_t$, end goal asset $A_E \in ISA$, and shortest exploit path P with h_P hops between A_S and A_E constitute an attack path tuple $AP = \langle A_S, P : A_S \rightarrow A_E, A_E \rangle$. Figure 7.7 summarizes the relationship between the observed attack steps $A_{sq}^O(t) = [ats_1, ats_2, \dots, ats_{t-1}]$, the subsequent asset predictions in NA_t with probability $p(A_S)$, and the end goal assets in ISA .

A risk value is computed for each attack path, with the risk value being a function of the contributions calculated for subsequent asset likelihood $C^L(A_S)$ (7.3), path exploitability $C^E(P, h_P)$ (7.8), and impact score $C^{IS}(A_E)$ (7.12). The contributions for subsequent asset likelihood, path exploitability, and impact score are shorthanded to C^L , C^E , and C^{IS} , respectively, for the remainder of this section. The risk value $Risk(AP) \in [0, 1]$ for attack path AP is computed as the product of the attack path contributions. The risk value computation is formalized as follows:

$$Risk(AP) = C^L \times C^E \times C^{IS} \quad (7.13)$$

To help in summarizing the trends seen across the risk values, an aggregate risk value is associated with each truncation set extracted from the attack sequence dataset. For this thesis, the aggregation simply assigns the risk value of the truncation set to be the maximum risk value found across all attack paths that branch from the truncation set. The attack campaigns considered in this thesis progress from the most recently compromised asset and do not have attack paths that intersect with themselves. Thus, attack paths that would pass through assets already noted in the asset compromise path of the truncation set are discarded.

The risk values are proportional relative to the risk attributed to the path. For example, if the

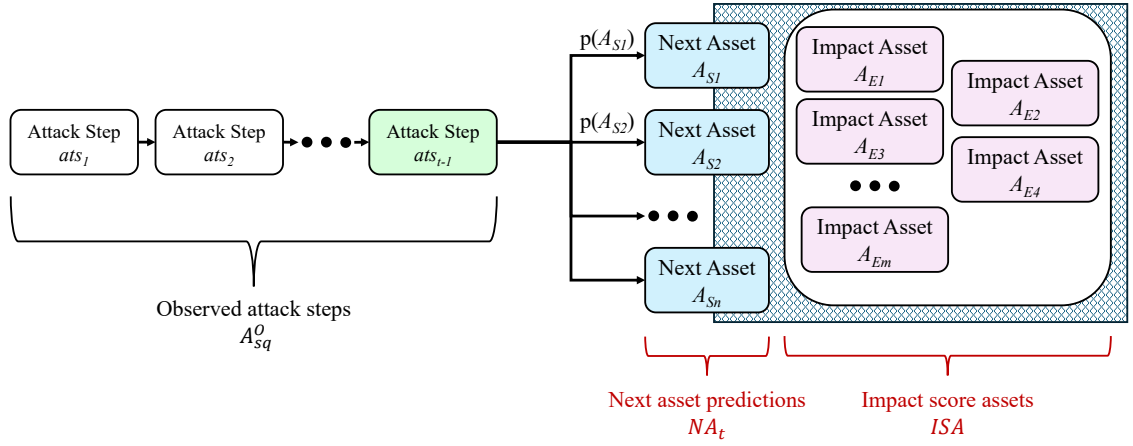


Figure 7.7: Relationship between truncation set elements and predictions.

value for $Risk(AP_1)$ is twice as large as that of $Risk(AP_2)$, then the risk associated with attack path AP_1 is twice as large as AP_2 . Figure 7.8 shows the risk value histograms for exploit resistance *Profile A*, *Profile B*, and *Profile C* defined in (7.5). The average risk values for *Profile A*, *Profile B*, and *Profile C* are 0.1938, 0.1466, and 0.0848, respectively. As can be seen, the risk values of the truncation sets do not appear to follow a normal distribution (bell curve). Rather, there is a high proportion of low risk samples that are mixed with a portion of relatively high-risk samples.

Using the risk value, the attack paths extending from an attack hypothesis can be compared against one another for various combinations of weight parameters W^L , W^E , and W^{IS} . Moreover, the risk values pertaining to the attack hypotheses under consideration, computed using the same weight parameters, can directly be compared to one another in order to assess the relative risk presented by the hypothesis. This is valid when the subsequent asset likelihoods for the compared attack hypotheses are determined using the same dataset. This validity also relies on having the normalization of path exploitability and impact score be based on the static configuration of the microgrid network rather than on the individual hypothesis features. Such is the case in the proposed methodology. Beyond comparing the risk values computed for the truncation sets against each other, it is also possible to assess how well the risk values align with the correct prediction of the attacker intent. More specifically, one can measure the risk values of paths that lead to the end goal of the attack associated with the truncation set, then determine how high those values are compared to the risk values calculated for all possible paths that extend from the truncation set.

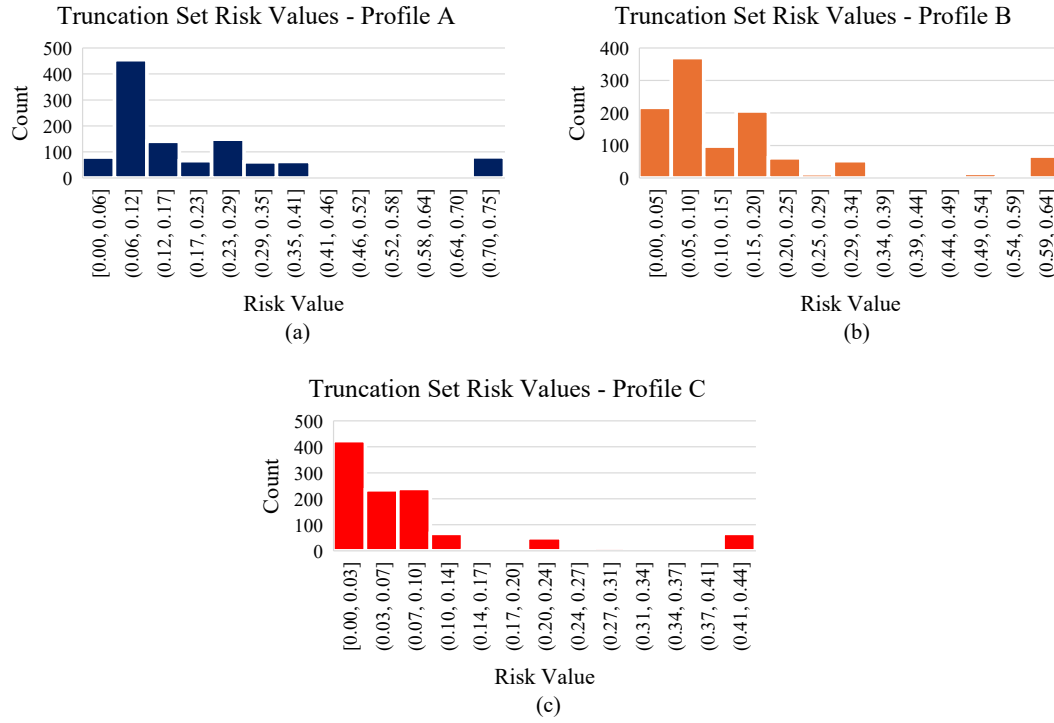


Figure 7.8: Risk value histograms for (a) Profile A (b) Profile B (c) Profile C.

7.4.1. Example of Risk Value Calculation

To provide more clarity, the risk value calculation is demonstrated on the previously used example of the truncation set that has Sandworm Team perform the $\langle \text{Execution} \rangle$ tactic on an operator $\langle \text{Workstation} \rangle$. This truncation set has the following probabilities for the $NextAsset$ variable, i.e., the next lateral movement target:

- $p(NextAsset = \langle \text{Control Server} \rangle) = 0.4445$
- $p(NextAsset = \langle \text{Data Historian} \rangle) = 0.2222$
- $p(NextAsset = \langle \text{Application Server} \rangle) = 0.2222$
- $p(NextAsset = \langle \text{Virtual Private Network (VPN) Server} \rangle) = 0.1111$

There are actually two $\langle \text{Control Server} \rangle$ assets adjacent to the operator workstation. These are the telecontrol server and the MGCC. Both are attributed to a subsequent asset likelihood of

0.4445. Which of these two assets will give an overall higher risk value depends on their respective path exploitability and impact scores of the paths that use them as a starting point. Choosing *Profile B* and a value of 1.0 for all the weight parameters (W^L , W^E , W^{IS}), the maximum risk value for the truncation set is $Risk(AP_{max}) = 0.3041$. The attack path AP_{max} with the highest risk corresponds to a direct lateral movement from the workstation to the telecontrol server that is followed by a CHP disconnection attack launched from the telecontrol server. The $Risk(AP_{max})$ value is found by multiplying the contributions of the telecontrol server, which has subsequent asset likelihood contribution $C^L = 0.4444$, path exploitability contribution $C^E = 0.6842$, and impact contribution $C^{IS} = 1.0$. Elaborating on the path exploitability specifically, the Res^{min} , $Res(P)$, and $Pf(hp)$ values used in (7.7) are 0.65, 0.95, and 1.0, respectively. Among all the possible attack paths considered for this truncation set, one of the paths with the lowest risk, AP_{min} , begins at the Remote Access Servers of the Operations Zone and ends at the relay of a load. This path has the asset sequence [Operations Remote Access Server, Operator Workstation 2, Database Server, Maintenance Workstation, Load 1 Relay]. It has $Res(P) = 4.6$ and $Pf(hp) = 38.4434$ (see Table 7.8) in *Profile B* which, with $Res^{min} = 0.65$, gives $C^E = 0.003675$. Multiplying this C^E value with $C^{IS} = 0.0667$ and $C^L = 0.1111$ yields $Risk(AP_{min}) = 2.723 * 10^{-5}$. These results are several orders of magnitude lower than the ones found for AP_{max} . The large ratio in risk value between AP_{min} and AP_{max} is sensible given the convoluted nature of the former compared to the latter from the attacker perspective.

7.4.2. Risk Value Statistics Across All APTs and Microgrid Assets

Of interest is the impact that changing the computation parameters has on the risk values calculated across the attack sequence dataset targeting the microgrid model. Understanding the influence of the parameters on the computed risk helps guide system operators towards having results that best match their priorities if they were to adopt the proposed methodology. Table 7.13 provides the averages across the maximum risk values calculated for the truncation sets using (7.13), aggregated according to the ICS APTs (*APT*) found in the truncation set. It includes values for various combinations of values for the W^L , W^E , and W^{IS} parameters tied to C^L , C^E , and C^{IS} , respectively. Meanwhile, Table 7.14 provides the average risk values when aggregating by the last asset to be

affected (*LastAsset*). These tables also include the overall average of the risk values across all the truncation set samples extracted from the attack sequence dataset. Additionally, they provide the computed risk values for the three exploit resistance profiles defined in this thesis, namely *Profile A*, *Profile B*, and *Profile C*.

Table 7.13: Average risk values for APT

ICS APT	Risk Value						
	$W^L=1.0$	$W^L=2.0$	$W^L=1.0$	$W^L=1.0$	$W^L=0.5$	$W^L=1.0$	$W^L=1.0$
	$W^E=1.0$	$W^E=1.0$	$W^E=2.0$	$W^E=1.0$	$W^E=1.0$	$W^E=0.5$	$W^E=1.0$
	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=2.0$	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=0.5$
<i>Profile A</i>							
APT33	0.2008	0.2008	0.0403	0.2008	0.2008	0.4481	0.2008
Dragonfly	0.3033	0.3013	0.1862	0.2889	0.3049	0.4853	0.3140
FIN6	0.5484	0.4358	0.5221	0.5062	0.6399	0.5947	0.5772
FIN7	0.2008	0.2008	0.0403	0.2008	0.2008	0.4481	0.2008
Lazarus Group	0.4403	0.2860	0.4266	0.4241	0.5684	0.4720	0.4602
OilRig	0.4309	0.3098	0.4130	0.3948	0.5762	0.4912	0.4544
Sandworm Team	0.3628	0.2400	0.3358	0.3475	0.4976	0.4405	0.3815
TEMP.Veles	0.3975	0.2827	0.3635	0.3797	0.5098	0.4699	0.4123
Full Average	0.4068	0.3008	0.3672	0.3841	0.5118	0.4870	0.4259
<i>Profile B</i>							
APT33	0.1374	0.1374	0.0189	0.1374	0.1374	0.3707	0.1374
Dragonfly	0.2109	0.2094	0.0892	0.2008	0.2120	0.4052	0.2186
FIN6	0.4114	0.3319	0.2985	0.3754	0.4762	0.5133	0.4361
FIN7	0.1374	0.1374	0.0189	0.1374	0.1374	0.3707	0.1374
Lazarus Group	0.3261	0.2150	0.2351	0.3118	0.4181	0.4068	0.3444
OilRig	0.3277	0.2414	0.2401	0.2967	0.4287	0.4273	0.3479
Sandworm Team	0.2721	0.1815	0.1879	0.2585	0.3672	0.3815	0.2894
TEMP.Veles	0.2937	0.2082	0.1990	0.2793	0.3755	0.4035	0.3065
Full Average	0.3021	0.2252	0.2047	0.2829	0.3766	0.4186	0.3188
<i>Profile C</i>							
APT33	0.0669	0.0669	0.0045	0.0669	0.0669	0.2587	0.0669
Dragonfly	0.1051	0.1043	0.0235	0.0999	0.1056	0.2864	0.1114
FIN6	0.2319	0.1914	0.1013	0.2075	0.2649	0.3818	0.2495
FIN7	0.0669	0.0669	0.0045	0.0669	0.0669	0.2587	0.0669
Lazarus Group	0.1802	0.1210	0.0752	0.1695	0.2295	0.3019	0.1954
OilRig	0.1885	0.1440	0.0834	0.1672	0.2387	0.3212	0.2024
Sandworm Team	0.1528	0.1029	0.0616	0.1426	0.2021	0.2845	0.1672
TEMP.Veles	0.1602	0.1126	0.0621	0.1512	0.2046	0.2967	0.1705
Full Average	0.1674	0.1263	0.0669	0.1543	0.2061	0.3091	0.1803

Looking at Table 7.13, it can be seen that FIN6 consistently presents the highest risk compared

Table 7.14: Average risk values for ATT&CK assets

ATT&CK Asset	Risk Value						
	$W^L=1.0$	$W^L=2.0$	$W^L=1.0$	$W^L=1.0$	$W^L=0.5$	$W^L=1.0$	$W^L=1.0$
	$W^E=1.0$	$W^E=1.0$	$W^E=2.0$	$W^E=1.0$	$W^E=1.0$	$W^E=0.5$	$W^E=1.0$
	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=2.0$	$W^{IS}=1.0$	$W^{IS}=1.0$	$W^{IS}=0.5$
<i>Profile A</i>							
Application Server	0.1889	0.1469	0.1215	0.1889	0.3212	0.3782	0.1889
Control Server	0.5594	0.4623	0.5459	0.5145	0.6414	0.5834	0.5925
Data Historian	0.1865	0.1436	0.1246	0.1855	0.3171	0.3707	0.1871
Field I/O	0.2230	0.2230	0.1523	0.1561	0.2230	0.3555	0.2665
IED	0.1223	0.0993	0.0470	0.1163	0.1389	0.2904	0.1928
PLC	0.7000	0.7000	0.7000	0.4900	0.7000	0.7000	0.8367
Safety Controller	0.1212	0.1149	0.0625	0.1064	0.1264	0.2894	0.2421
VPN Server	0.1718	0.1681	0.0338	0.1718	0.1738	0.3927	0.1718
Workstation	0.4120	0.2433	0.3735	0.4097	0.5542	0.4739	0.4135
Full Average	0.4068	0.3008	0.3672	0.3841	0.5118	0.4870	0.4259
<i>Profile B</i>							
Application Server	0.1414	0.1111	0.0585	0.1414	0.2258	0.3288	0.1414
Control Server	0.4315	0.3559	0.3287	0.3929	0.4954	0.5113	0.4606
Data Historian	0.1363	0.1062	0.0592	0.1357	0.2202	0.3189	0.1368
Field I/O	0.1841	0.1841	0.0925	0.1289	0.1841	0.3301	0.2200
IED	0.1006	0.0826	0.0361	0.0937	0.1135	0.2615	0.1669
PLC	0.5829	0.5829	0.4870	0.4080	0.5829	0.6385	0.6967
Safety Controller	0.0968	0.0923	0.0511	0.0799	0.1005	0.2548	0.2171
VPN Server	0.1175	0.1150	0.0158	0.1175	0.1189	0.3247	0.1175
Workstation	0.2888	0.1702	0.1849	0.2873	0.3890	0.3961	0.2899
Full Average	0.3021	0.2252	0.2047	0.2829	0.3766	0.4186	0.3188
<i>Profile C</i>							
Application Server	0.0774	0.0616	0.0145	0.0774	0.1142	0.2447	0.0774
Control Server	0.2517	0.2069	0.1184	0.2250	0.2897	0.3879	0.2746
Data Historian	0.0726	0.0572	0.0144	0.0723	0.1097	0.2341	0.0728
Field I/O	0.1222	0.1222	0.0326	0.0856	0.1222	0.2783	0.1461
IED	0.0662	0.0558	0.0228	0.0579	0.0738	0.2066	0.1245
PLC	0.3800	0.3800	0.2109	0.2660	0.3800	0.5142	0.4542
Safety Controller	0.0671	0.0643	0.0394	0.0460	0.0694	0.1978	0.1803
VPN Server	0.0572	0.0560	0.0038	0.0572	0.0579	0.2266	0.0572
Workstation	0.1460	0.0857	0.0485	0.1453	0.1971	0.2806	0.1465
Full Average	0.1674	0.1263	0.0669	0.1543	0.2061	0.3091	0.1803

to the other APTs. This can be explained by how often FIN6 tends to target the control servers compared to the others. By contrast, FIN7 and APT33 have the lowest risk values across the board. This can be explained by how these two APTs limit their scope to the workstations rather than moving

laterally across the microgrid. Increasing the W^L weight for likelihood ($W^L = 2.0$) increases their risk relative to the average. The same is true for Dragonfly. The opposite trend is observed when W^L is reduced to 0.5. Some other interesting trends emerge from these results. For example, the risk values for OilRig, relative to the other APTs, rises when increasing W^E so as to emphasize path exploitability. This suggests that OilRig leans towards following a strategy that leads to the easiest exploit paths. The trends seen for the APTs risk values are rather consistent across the three exploit resistance profiles.

As for insights surrounding the microgrid assets that can be gained from Table 7.14, attacks that have reached a PLC present the highest risk for the most part. This is explainable by the immediate proximity of the PLCs to their respective supervisory controllers, which have a high impact score associated with them. The PLC risk values are, however, overtaken by those of the control server once W^{IS} is set to 2.0. The control servers have a very high impact score associated with them, and so emphasizing impact score by extension reduces the significance of their higher exploit resistance compared to field devices like the PLC. As for the assets with relatively low risk values, the IED and safety controller generally present the lowest risk due to being localized to one microgrid DER. However, increasing the weight for path exploitability, W^E , drastically reduces the risk values associated with attacks that have only reached the VPN server. This is due to the VPN having a noticeably large hop distance to assets associated with high impacts while also being surrounded by the assets with the highest exploit resistance.

7.4.3. Alignment of Risk Value with Attacker Intent

While the risk values help to compare the danger posed by various attack possibilities, there is also potential to leverage them for predicting the final target of an APT during an ongoing attack campaign. Recall that an attack sequence is described as an ordered list of attack steps $A_{sq} = [ats_1, ats_2, \dots, ats_i, \dots, ats_K]$. The truncation set used for calculating the risk values is defined using truncation parameter t that yields $A_{sq}^O(t) = [ats_1, ats_2, \dots, ats_{t-1}]$ and $lt(t) = ats_{t-1}$. The risk value calculation as described thus far has pertained to assigning a risk for the different possible prediction targets $pt(t) = ats_t$. However, it is also possible to look further ahead and consider the attack paths that lead to the final target of the attack, which is exploited at the final attack step ats_k .

Having a method to predict the ultimate intent of an APT can be instrumental towards security responses that intercept an attack before it can cause major harm to microgrid operations.

Within the context of this thesis, the risk values are said to be aligned with attacker intent if the risk values for attack paths associated with the intent are among the highest of all potential paths. Here, the attacker intent corresponds to the compromise of the end goal asset through which they will execute the actions that meet their final objective. That is, the compromise of end goal asset A_{ats_k} is associated with the final attack step ats_K . The alignment of the risk value with the attacker intent is measured by finding the percentile of the highest risk value among paths that extend from the observed subsequence ending at ats_{t-1} to A_{ats_k} compared to all non-zero-risk paths that extend from the subsequence. The method used to determine the highest risk value $Risk^{Int}$ associated with the intent is formalized as follows:

$$Risk^{Int} = \max(Risk(A_S, P, A_E)), \quad A_E = A_{ats_k} \quad (7.14)$$

Figure 7.9 shows the average percentile for $Risk^{Int}$ across all the truncation sets in the attack sequence dataset depending on the exploit resistance profile and weight parameter combination used. These results act as a gauge for how accurate the risk values are as predictors of the APT actions that will follow from a given state. It can be seen that, even in the worst case, the risk value for paths that lead to the intended final target of the APT is, on average, greater than about 80% of the risk values across all paths extending from the same truncation set. Interestingly, the lowest percentiles are observed when increasing W^{IS} so as to emphasize impact score. Conversely, the highest percentiles are seen when emphasis on impact score is reduced by decreasing W^{IS} . This suggests that the magnitude of the negative impact on the microgrid is a weakest predictor for attacker intent than the likelihood derived from CTI and the path exploitability. Also of note is that *Profile C* results in higher overall percentiles for the risk values. This may be reflective of how microgrid assets at lower Purdue levels being treated as more vulnerable to exploitation is an assumption that is closer to reality.

Figure 7.10 shows the average percentile for $Risk^{Int}$ when aggregating by APT. It can be seen that the proposed methodology for quantifying risk is most effective at predicting the end goal of

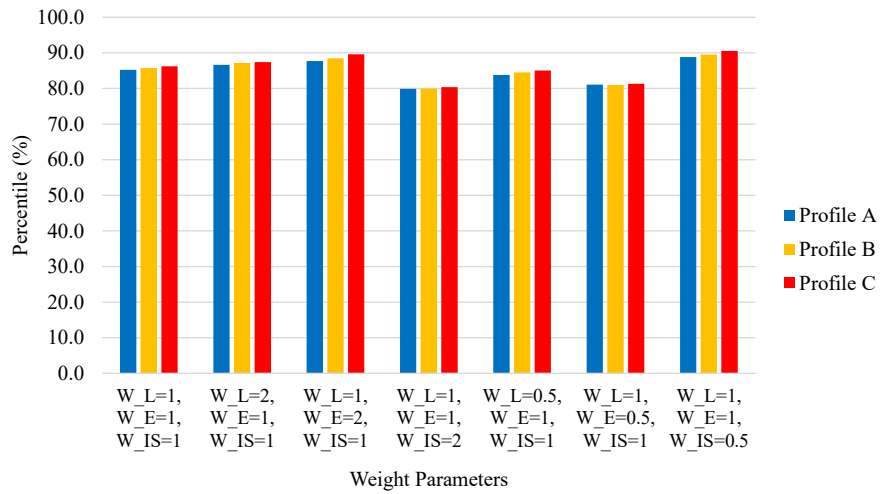


Figure 7.9: Average percentile of end goal asset risk values for different weights.

FIN6 and OilRig. Notably, the percentiles for APT33, FIN7, and Dragonfly are omitted. Within the attack sequence dataset, these APTs exclusively have end goal assets that are not directly associated with an attack on the microgrid power system. That is, their end goal assets are not in the set of positive impact score assets *ISA*. With no impact score associated with them, the paths leading to their intended end goal assets have no risk value associated with them. This highlights the current gap in the impact score assessment that forms part of the proposed attack prediction methodology. The assessment has yet to consider the magnitude of cyber-related impacts on the microgrid network such as those that result from data exfiltration or industrial espionage. There is room to expand the impact assessment so as to have a relative impact associated with the compromise of assets that do not have an apparent effect on the microgrid power system.

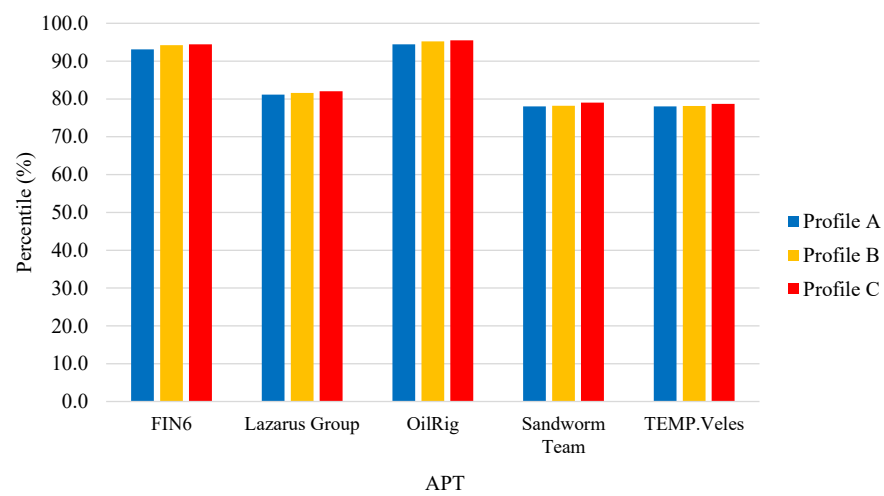


Figure 7.10: Average percentile of end goal asset risk values for different APTs.

Chapter Eight

Conclusion and Future Work

The microgrid is a valuable energy generation and distribution system within the smart grid, but whose reliance on digital technology for protection and control makes it vulnerable to cyber attacks that can adversely affect its ability to safely provide electric power. We propose an automated framework for microgrid security monitoring and threat hunting to improve the security posture of the microgrid. The framework enables microgrid security operators to better detect system anomalies from telemetry data, correlate alerts to help identify APTs attack campaigns while leveraging CTI, and assesses the risks and intentions associated with an ongoing attack campaign. The effectiveness of the proposed framework is demonstrated using a microgrid co-simulation testbed for assessing impacts on the microgrid power system, as well as a model of the microgrid system topology ranging from the microgrid control center to the DER field devices.

Firstly, we presented background information on the microgrid and a literature review on relevant smart grid security topics. This is followed by details surrounding the microgrid co-simulation testbed and the modeling of the microgrid system topology. We then presented research that uses machine learning models to detect anomalies within system telemetry, trained solely on normal operation data. These models are used to detect attacks on microgrid telecontrol messages using IEC 62351-7:2017 NSM data collected through active polling. They are also used to identify FDI attacks on physical power measurements that attempt to spoof a line fault. Next, we proposed a threat hunting approach that uses public CTI on ICS attacks to form attack hypothesis subjected to a validation process. Finally, we present a method for anticipating the attack activity that may follow

from observed attack hypotheses by associating possible attack paths with a semi-quantitative risk value to use for relative comparison of the possible paths.

8.1. Summary of Contributions

We presented several research contributions. The first is an evaluation of various machine learning models and configurations for detecting anomalies within IEC 62351-7:2017 NSM active monitoring data that result from attacks on IEC-104 telecontrol messages. The second contribution is the proposal for a passive monitoring scheme that uses distributed monitors, whose prediction errors are aggregated by a central manager, to detect FDI attacks on physical measurements in real-time. The effectiveness of the passive and active monitoring data is evaluated using data collected from the microgrid co-simulation testbed under normal operation and under simulated microgrid attacks. The LSTM deep learning model is found to be particularly effective at identifying anomalies associated with attacks on the monitored microgrid telemetry.

The third contribution is the proposal of a novel threat hunting approach. The approach leverages a knowledge graph derived from CTI on ICS attacks to match alerts with TTPs, correlate them into attack hypotheses, validate the hypotheses according to a severity score, and associate validated hypotheses with APT groups. To evaluate the proposed microgrid threat hunting approach, a dataset of attack sequences acting on the modeled microgrid system is generated, with each sequence being aligned with the behaviours of a particular APT known through CTI. In particular, the effectiveness of the proposed APT identification method is evaluated and is found to be very effective at identifying certain APTs while struggling to detect others. It is also capable of accommodating random noise and undetected attack events without severe loss of accuracy. Notably, the proposed threat hunting approach generally detects attacks in its evaluation dataset that are within the *Through* phase of the UKC.

The final contribution is a methodology for calculating risks associated with attack paths extending from an attack hypothesis. This methodology is proposed for the purpose of supporting the prediction of the attack actions to follow and to help steer response efforts according to security priorities. The attack sequence dataset used to evaluate the threat hunting approach is also used to

train the Bayesian network used for attack step prediction. In addition, the dataset is used to evaluate the level of correspondence between attack path risk values extending from partial attack sequences and the intended final target of the attack sequence. The evaluation reveals that the calculated risks for the attack paths are well aligned with the intended attack paths found in the evaluation dataset.

Together, our contributions support a layered microgrid monitoring framework that detects intrusions within the microgrid operations network, identifies the possible TTPs and threat actors involved in detected cyberattacks, and determines the risks posed by ongoing attacks. The elements of the proposed framework each offer insights and methodologies that can be adapted by microgrid owners, as well as owners of other ICS networks, to bolster the effectiveness of their security monitoring and protection solutions.

There are, however, notable limitations with the proposed framework. It does not currently incorporate human-in-the-loop processes in its design. Rather, it assumes that human operators will act on the output of the framework. Details on how particular network artifacts or IoCs can be mapped to threat actors and TTPs for threat hunting is also very limited, opting instead to associate more generalized alert data to TTPs. As for the proposed prediction, it has yet to account for attack paths that include superfluous lateral movements intended to obfuscate the end goal of the attacker.

8.2. Insights

The research surrounding each of the presented contributions reveals important insights to consider in the context of microgrid cybersecurity. These insights highlight some of the strengths and limiting factors of the proposed framework for microgrid security monitoring and threat hunting.

While the LSTM is very effective at identifying anomalies in microgrid telemetry, it is more computationally expensive than some of the more lightweight MLP and classifier models. However, the ANN models like LSTM and MLP have a major advantage over classifier models for microgrid anomaly detection in that they do not need attack data in their training set to be effective. This is essential in the case of the microgrid, where sample attack data is generally scarce. As for scalability concerns, care is needed when increasing the scope of the active NSM monitoring platform given that collection delays increase as the number of monitored MIBs and NSM agents increases. Scaling

of the platform may require the adoption of an NSM manager hierarchy to limit the computational burden on any one manager.

Threat hunting faces the challenge of needing to manage the size of the active hypothesis pool in order to maintain scalability. The proposed MCDM filtering method is found to be effective at managing the hypothesis pool size, but the effectiveness of other possible filtering methods cannot be ruled out. As for the success rate of APT identification, the accuracy of the proposed identification method varies depending on the configured hypothesis severity threshold and the attacking APT. This demonstrates the differences across the APTs, with some of them possibly requiring a highly adaptable threat hunting approach to accurately differentiate them from other APTs at different stages of their attack kill chain. The choice of severity threshold also impacts how far an attacker will progress through their attack kill chain while still evading detection. There is a trade-off between how early an attack hypothesis corresponding to an attack campaign is detected and the false positive rate for hypotheses generated purely from random noise.

The proposed attack prediction methodology explores how different weight values placed on the risk calculation factors affect the alignment of the risk with APT intent. In particular, adding to the weight of the impact contribution to the risk value results in lower predictive alignment. This suggests that APTs, according to available CTI, are expected to follow microgrid attack paths that are easier to exploit rather than attempt to maximize the damage they will cause.

8.3. Future Research

When exploring possibilities for future research, there are various aspects of the research we presented that can be extended and iterated upon. Regarding the proposed anomaly detection approach, there is room to explore the effectiveness and efficiency of the transformer deep learning model in particular. This would require that the continuous values of the monitored microgrid telemetry be mapped to the semantic embedding space of the transformer model before they can be used as prediction inputs. As for the monitored data itself, the set of NSM data monitored over SNMP can be extended to include MIBs for other data sources, such as the computation resources being consumed by the various microgrid assets and statistics related to the usage of particular applications such as

Secure Shell (SSH). In turn, the alerts generated from anomalies surrounding these other sources can be mapped to MITRE ATT&CK data components or even to TTPs directly in order to be less reliant on the KG vector embedding model used to associate alerts with TTPs.

Regarding the threat model, there is room to elaborate a method for generating novel behaviour profiles that can be associated with hypothetical threat groups having no CTI, and then proposing a method for distinguishing between known APTs and novel ones. Regarding the attack prediction methodology, a more accurate assessment on the exploitability of each microgrid asset can be done by enumerating the particular vulnerabilities and their corresponding TTP associated with the real computing devices behind these assets and assigning their level of vulnerability based on an aggregate of scores tied to their individual vulnerabilities.

The impact assessment of the microgrid model can also be expanded to consider a wider range of attacks. In particular, there are attacks whose impacts are not observed on the microgrid power network but which nonetheless negatively impact the utility, such as data theft and ransomware attacks. Including such attacks into the risk assessment would allow for impact scores to be assigned to a wider range of microgrid assets and in doing so provide better risk assessments on APTs like APT33 that largely target the ICS cyber network. The range of attack scenarios explored in this research can be expanded as well. Notably, attacks that are effective while the microgrid is in grid-connection mode can be assessed, as can attacks that target the process that has the microgrid transition from islanded mode to grid-connected mode. However, the extent in which such attacks can be observed to affect the main grid within the co-simulation testbed is limited due to modeling the main grid as simply a voltage source.

References

- [1] M. A. Judge, A. Khan, et al., “Overview of smart grid implementation: Frameworks, impact, performance and challenges,” *Journal of Energy Storage*, Feb. 2022.
- [2] S. Ishaq, I. Khan, et al., “A review on recent developments in control and optimization of micro grids,” *Energy Reports*, Mar. 2022.
- [3] I. Patrao, E. Figueres, et al., “Microgrid architectures for low voltage distributed generation,” *Renewable and Sustainable Energy Reviews*, vol. 43, pp. 415–424, Mar. 2015.
- [4] N. Hussain, M. Nasir, et al., “Recent developments and challenges on AC microgrids fault detection and protection systems – a review,” *Energies*, May 2020.
- [5] Microgrid Knowledge, *Microgrid cybersecurity: Protecting and building the grid of the future*, S&C Electric Company, Jan 2019. Available: <https://www.sandc.com/globalassets/sac-electric/documents/public---documents/sales-manual-library---external-view/technical-paper-100-t116.pdf?dt=638446507764693160>, Accessed: Jan. 2025.
- [6] Advanced Grid Research, *Voices of experience: Microgrids for resiliency*, US Department of Energy, Nov. 2020.
- [7] L. Mariam, M. Basu, and M. F. Conlon, “Microgrid: Architecture, policy and future trends,” *Renewable and Sustainable Energy Reviews*, vol. 64, pp. 477–489, Oct. 2016.
- [8] S. H. Rouhani, C.-L. Su, et al., “Cyber resilience in renewable microgrids: A review of standards, challenges, and solutions,” *Energy*, vol. 309, Nov. 2024.

- [9] ICS-CERT, *Cyber-Attack Against Ukrainian Critical Infrastructure*, Cybersecurity and Infrastructure Security Agency, Jul. 2021. Available: <https://us-cert.cisa.gov/ics/alerts/IR-ALERT-H-16-056-01>, Accessed: Jan 2025.
- [10] A. Greenberg, 'Crash Override': *The malware that took down a power grid*, Wired, Jun. 2017. Available: <https://www.wired.com/story/crash-override-malware/>, Accessed: Jan 2025.
- [11] K. Proska, J. Wolfram, et al., *Sandworm disrupts power in Ukraine using a novel attack against operational technology*, Mandiant, Nov 2023. Available: <https://cloud.google.com/blog/topics/threat-intelligence/sandworm-disrupts-power-ukraine-operational-technology/>, Accessed: Jan 2025.
- [12] Abhijnan Rej, *China Targeted India's Power Grid, New Report Says*, The Diplomat, Mar. 2021. Available: <https://thediplomat.com/2021/03/china-targeted-indias-power-grid-new-report-says/>, Accessed: Jan. 2025.
- [13] M. Karanfil, "Network and system management for the security monitoring of microgrids using IEC 62351-7," Master's Thesis, Concordia University, Apr. 2019. Available: <https://spectrum.library.concordia.ca/id/eprint/985404/>, Accessed: Dec. 2025.
- [14] OPAL-RT Technologies, *HYPERSIM: The power system simulator of tomorrow*, 2025. Available: <https://www.opal-rt.com/systems-hypersim/>, Accessed: Dec. 2025.
- [15] Openstack, *The most widely deployed open source cloud software in the world*, An OpenInfra Foundation Project, 2025. Available: <https://www.openstack.org/>, Accessed: Dec. 2025.
- [16] M. Imran, H. U. R. Siddiqui, et al., "A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems," *Computers & Security*, vol. 134, p. 103 445, 2023.

- [17] Y. Bansal and R. Sodhi, “Microgrid fault detection methods: Reviews, issues and future trends,” in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2018, pp. 401–406.
- [18] C. Yuan, K. Lai, et al., “Multilayered protection strategy for developing community microgrids in village distribution systems,” *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 495–503, 2017.
- [19] IEC Technical Committee 57, *Power systems management and associated information exchange - data and communications security — part 7: Network and system management (nsm) data object models*, Jul. 2017.
- [20] IEC Standard 60870, *IEC Telecontrol Equipment and Systems-Part 5-104: Transmission Protocols—Network Access for IEC 60870-5-101 Using Standard Transport Profiles*, 2006.
- [21] F. K. Kaiser, U. Dardik, et al., “Attack hypotheses generation based on threat intelligence knowledge graph,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, 2023.
- [22] S. M. Milajerdi, B. Eshete, et al., “POIROT: Aligning attack behavior with kernel audit records for cyber threat hunting,” *2019 ACM SIGSAC Conf. on Computer and Commun. Security*, pp. 1795–1812, Nov. 2019.
- [23] D. Bianco, *The pyramid of pain*, Jan. 2014. Available: <https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>, Accessed: Mar. 2025.
- [24] I. Bücher, *How to address alert fatigue in the process industry without turning off alerts*, Precognize: A Samson Company, Dec. 2022. Available: <https://www.precog.co/blog/alert-fatigue-in-the-process-industry/>, Accessed: Mar. 2025.
- [25] B. Nour, M. Pourzandi, and M. Debbabi, “A survey on threat hunting in enterprise networks,” *IEEE Communication Surveys and Tutorials*, vol. 25, no. 4, pp. 2299–2324, Aug. 2023.
- [26] The MITRE Corporation, *MITRE ATT&CK*, 2025. Available: <https://attack.mitre.org/>, Accessed: Dec. 2025.

- [27] G. Di Tizio, M. Armellini, and F. Massacci, “Software updates strategies: A quantitative evaluation against advanced persistent threats,” *IEEE Transactions on Software Engineering*, May 2022.
- [28] M. Karanfil, D. E. Rebbah, et al., “Security monitoring of the microgrid using iec 62351-7 network and system management,” in *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Jul. 2022, pp. 1–5.
- [29] M. Karanfil, D. E. Rebbah, et al., “Detection of microgrid cyberattacks using network and system management,” *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2390–2405, May 2023.
- [30] M. Karanfil, E.-N. S. Youssef, et al., “Real-time protection against microgrid false data injection attacks using passive monitoring,” in *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 2023, pp. 1–5.
- [31] Y. Parag and M. Ainspan, “Sustainable microgrids: Economic, environmental and social costs and benefits of microgrid deployment,” *Energy for Sustainable Development*, vol. 52, pp. 72–81, Oct. 2019.
- [32] Securing Energy Infrastructure Engineering Task Force, *Reference architectures as a means of influencing electric energy operational technology/industrial control system security outcomes*, Cybersec., Energy Sec., and Emergency Response, US. Dep. of Energy, Mar. 2022.
- [33] D. Garton, “Purdue model framework for industrial control systems & cybersecurity segmentation,” National Petroleum Council Study on Oil and Natural Gas Transportation Infrastructure, Topic Paper, Nov. 2019.
- [34] S. Safdar, B. Hamdaoui, et al., “A survey on communication infrastructure for micro-grids,” in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul. 2013, pp. 545–550.
- [35] A. S. Mohamed, M. F. M. Arani, et al., “False data injection attacks against synchronization systems in microgrids,” *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4471–4483, Sep. 2021.

- [36] S. C. Vegunta, M. J. Higginson, et al., “Ac microgrid protection system design challenges—a practical experience,” *Energies*, vol. 14, no. 7, Apr. 2021.
- [37] OPC Foundation, *What is opc?* 2025. Available: <https://opcfoundation.org/>, Accessed: Mar. 2025.
- [38] S. Sen and V. Kumar, “Microgrid control: A comprehensive survey,” *Annual Reviews in Control*, vol. 45, pp. 118–151, Jun. 2018.
- [39] K. Sirviö, K. Kauhaniemi, et al., “Functional analysis of the microgrid concept applied to case studies of the sundom smart grid,” *Energies*, Aug. 2020.
- [40] IEC Technical Committee 57, *Communication networks and systems for power utility automation – part 9-2: Specific communication service mapping (scsm) – sampled values over ISO/IEC 8802-3*, Sep. 2011.
- [41] IEC Technical Committee 57, *Communication networks and systems for power utility automation – part 7-2: Basic information and communication structure – abstract communication service interface (ACSI)*, Aug. 2010.
- [42] IEEE Standards Coordinating Committee 21, *IEEE std 1547-2018:IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces*, Feb. 2018.
- [43] L. Meegahapola, I. U. Nutkani, and B. McGrath, “Investigation of fault ride-through capability of ac/dc hybrid microgrids during ac network faults,” in *2017 IEEE Power & Energy Society General Meeting*, Jul. 2017, pp. 1–5.
- [44] P. WG, *IEEE standard for electric power systems communications — distributed network protocol (DNP3)*, Oct. 2012.
- [45] C.-Y. Lin and S. Nadjm-Tehrani, “Understanding iec-60870-5-104 traffic patterns in scada networks,” in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, May 2018, pp. 51–60.

- [46] Y. Yang, K. McLaughlin, et al., “Stateful intrusion detection for iec 60870-5-104 scada security,” in *2014 IEEE Power and Energy Society General Meeting Conference & Exposition*, 2014.
- [47] Q. S. Qassim, N. Jamil, et al., “Simulating command injection attacks on iec 60870-5-104 protocol in scada system,” *International Journal of Engineering & Technology*, 2018.
- [48] IEC Technical Committee 57, *Power systems management and associated information exchange - data and communications security — part 1: Communication network and system security - introduction to security issues*, May 2007.
- [49] J. Case, M. Fedor, et al., *RFC-1157: A simple network management protocol (SNMP)*, May 1990.
- [50] J. Case, K. McCloghrie, et al., *RFC-3418: Management information base (mib) for the simple network management protocol (SNMP)*, Dec. 2002.
- [51] W. Stallings, *SNMP, SNMPV2, SNMPv3, and RMON 1 and 2*, 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998, ISBN: 0201485346.
- [52] P. Gao, F. Shao, et al., “Enabling efficient cyber threat hunting with cyber threat intelligence,” in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr. 2021, pp. 193–204.
- [53] H. Albasheer, M. Md Siraj, et al., “Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: A survey,” *Sensors*, vol. 22, no. 4, Feb. 2022.
- [54] M. Stewart, *The actual difference between statistics and machine learning*, Medium, 2019. Available: <https://medium.com/data-science/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3>, Accessed: Oct 2025.
- [55] University of North Dakota, *Machine learning vs. statistics: What’s the best approach?* University of North Dakota, Dec. 2025. Available: <https://und.edu/blog/machine-learning-vs-statistics.html>, Accessed: Dec 2025.

- [56] G. Rigatos, D. Serpanos, and N. Zervos, "Detection of attacks against power grid sensors using kalman filter and statistical decision making," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7641–7648, Dec. 2017.
- [57] K. Manandhar, X. Cao, et al., "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, Sep. 2014.
- [58] C. Ruben, S. Dhulipala, et al., "Hybrid data-driven physics model-based framework for enhanced cyber-physical smart grid security," *IET Smart Grid*, vol. 3, no. 4, pp. 445–453, Apr. 2020.
- [59] M. Lei and M. Mohammadi, "Hybrid machine learning based energy policy and management in the renewable-based microgrids considering hybrid electric vehicle charging demand," *International Journal of Electrical Power & Energy Systems*, vol. 128, p. 106702, Jun. 2021.
- [60] O. A. Beg, L. V. Nguyen, et al., "Signal temporal logic-based attack detection in DC microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3585–3595, Jul. 2019.
- [61] J. Zhang, S. Sahoo, et al., "Mitigating concurrent false data injection attacks in cooperative dc microgrids," *IEEE Transactions on Power Electronics*, vol. 36, no. 8, pp. 9637–9647, Jan. 2021.
- [62] Y. Zhang, V. V. G. Krishnan, et al., "Cyber physical security analytics for transactive energy systems," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 931–941, 2020.
- [63] I. Siniosoglou, P. Radoglou-Grammatikis, et al., "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1137–1151, Jun. 2021.
- [64] A. Presekal, A. Ştefanov, et al., "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Transactions on Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sep. 2023.

- [65] A. Kavousi-Fard, W. Su, and T. Jin, “A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 650–658, Jun. 2021.
- [66] J. Inoue, Y. Yamagata, et al., “Anomaly detection for a water treatment system using unsupervised machine learning,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Dec. 2017, pp. 1058–1065.
- [67] S. Ahmed, Y. Lee, et al., “Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest,” *IEEE Transactions Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, Oct. 2019.
- [68] A. Parizad and C. J. Hatziadoniu, “Cyber-attack detection using principal component analysis and noisy clustering algorithms: A collaborative machine learning-based framework,” *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4848–4861, Nov. 2022.
- [69] E. Hodo, S. Grebeniuk, et al., “Anomaly detection for simulated iec-60870-5-104 traffic,” in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, Association for Computing Machinery, Aug. 2017, pp. 1–7.
- [70] M. Ma, A. Lahmadi, and I. Chrisment, “Detecting a stealthy attack in distributed control for microgrids using machine learning algorithms,” in *IEEE Conference on Industrial Cyber-physical Systems (ICPS)*, vol. 1, Jun. 2020, pp. 143–148.
- [71] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA workbench - online appendix for data mining: Practical machine learning tools and techniques*, 2016. Available: https://ml.cms.waikato.ac.nz/weka/Witten_et_al_2016_appendix.pdf, Accessed: Oct. 2025.
- [72] Y. Xu, Z. Yin, et al., “Midas: Safeguarding iot devices against malware via real-time behavior auditing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4373–4384, 2022.
- [73] A. Nadeem, S. Verwer, et al., “Alert-driven attack graph generation using s-pdfa,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 731–746, 2022.

- [74] M. Asiri, N. Saxena, et al., “Understanding indicators of compromise against cyber-attacks in industrial control systems: A security perspective,” *ACM Trans. Cyber-Phys. Syst.*, vol. 7, no. 2, Apr. 2023.
- [75] S. Wang, Z. Wang, et al., “Threatrace: Detecting and tracing host-based threats in node level through provenance graph learning,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3972–3987, Sep. 2022.
- [76] Q. Wang, W. U. Hassan, et al., “Threatrace: Detecting and tracing host-based threats in node level through provenance graph learning,” *Network and Distributed Systems Security (NDSS) Symposium*, Feb. 2020.
- [77] K. Liu, F. Wang, et al., “A review of knowledge graph application scenarios in cyber security,” *arXiv:2204.04769 [cs.CR]*, Apr. 2022.
- [78] Z. Li, J. Zeng, et al., “Attackg: Constructing technique knowledge graph from cyber threat intelligence reports,” in *Computer Security – ESORICS 2022*, 2022, pp. 589–609.
- [79] B. Nour, M. Pourzandi, et al., “AUTOMA: Automated generation of attack hypotheses and their variants for threat hunting using knowledge discovery,” *IEEE Trans. Netw. Service Manag.*, Mar. 2024.
- [80] B. Abdeen, E. Al-Shaer, et al., “SMET: Semantic mapping of CVE to ATT&CK and its application to cybersecurity,” in *Data and Applications Security and Privacy XXXVII*, Jul. 2023, pp. 243–260.
- [81] S. Choi, J.-H. Yun, and B.-G. Min, “Probabilistic attack sequence generation and execution based on MITRE ATT&CK for ICS datasets,” in *Proceedings of the 14th Cyber Security Experimentation and Test Workshop*, Aug. 2021, pp. 41–48.
- [82] Lockheed Martin, *Gaining the advantage: Applying cyber kill chain® methodology to network defense*, 2015.
- [83] CVE Program, *CVE® Program mission*, 2025. Available: <https://www.cve.org/>, Accessed: Dec. 2025.

- [84] D. K. Marzagão, T. D. Huynh, et al., “Provenance graph kernel,” *Journal OF LaTeX class files*, vol. 14, no. 8, Sep. 2021.
- [85] B. Zhang, X. Liu, et al., “Hidden markov model based cyberattack prediction in power systems,” *IEEE Transactions on Smart Grid*, Oct. 2024.
- [86] Y. Chen, J. Hong, and C.-C. Liu, “Modeling of intrusion and defense for assessment of cyber security at power substations,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2541–2552, Jul. 2018.
- [87] M. Villarreal-Vasquez, G. Modelo-Howard, et al., “Hunting for insider threats using lstm-based anomaly detection,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 451–462, Feb. 2023.
- [88] S. M. Khalil, H. Bahsi, et al., “Threat modeling of cyber-physical systems - a case study of a microgrid system,” *Computers & Security*, vol. 124, Jan. 2023.
- [89] E. R. Ling and M. Ekstedt, “A threat modeling language for generating attack graphs of substation automation systems,” *International Journal of Critical Infrastructure Protection*, vol. 41, Jul. 2023.
- [90] J. Katina, T. Pléta, et al., “Industrial control systems (ICS) cyber prediction model,” *Entrepreneurship and Sustainability Center*, vol. 5, no. 1, pp. 86–96, Dec. 2023.
- [91] L. Langer, F. Skopik, et al., “From old to new: Assessing cybersecurity risks for an evolving smart grid,” *Computers & Security*, vol. 62, pp. 165–176, Sep. 2016.
- [92] K. Yan, X. Liu, et al., “A cyber-physical power system risk assessment model against cyberattacks,” *IEEE Systems Journal*, vol. 17, no. 2, pp. 2018–2028, Jun. 2023.
- [93] S. Mishra, T. Kwasnik, and K. Anderson, “Microgrid resilience: A holistic and context-aware resilience metric,” *Energy Systems*, vol. 14, pp. 1081–1105, Nov. 2023.
- [94] IEC Technical Committee 57, “Communication networks and systems for power utility automation - part 1: Introduction and overview,” International Electrotechnical Commission, Technical Report, Mar. 2013.

- [95] X. Diao, Y. Zhao, et al., “Dynamic probabilistic risk assessment for electric grid cybersecurity,” *Reliability Engineering & System Safety*, vol. 241, p. 109 699, Jan. 2024.
- [96] Y. Du, H. Tu, et al., “Development of a controller hardware-in-the-loop platform for microgrid distributed control applications,” in *2018 IEEE Electronic Power Grid (eGrid)*, Nov. 2018, pp. 1–6.
- [97] Distribution Resources Integration Working Group, *IEEE standard for the specification of microgrid controllers*, Dec. 2017.
- [98] Y. Ma, X. Hu, et al., “Real-time control and operation for a flexible microgrid with dynamic boundary,” in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2018, pp. 5158–5163.
- [99] MathWorks, *Matlab: Math. graphics. programming*. 2025. Available: <https://www.mathworks.com/products/matlab.html>, Accessed: Dec 2025.
- [100] W. Danilczyk, Y. Sun, and H. He, “Angel: An intelligent digital twin framework for microgrid security,” in *2019 North American Power Symposium*, Oct. 2019, pp. 1–6.
- [101] I. K. Kiran and J. A. Laxmi, “Shunt versus series compensation in the improvement of power system performance,” *International Journal of Applied Engineering Research*, vol. 2, no. 1, 2011.
- [102] M. Ahmed, L. Meegahapola, et al., “Stability and control aspects of microgrid architectures—a comprehensive review,” *IEEE Access*, vol. 8, pp. 144 730–144 766, Aug. 2020.
- [103] MZ Automation, *Open source libraries for iec 61850 and iec 60870-5-104*, 2025. Available: <https://libiec61850.com/libiec61850-now-on-github/>. Accessed: Dec. 2025.
- [104] Microchip Technology Inc., *PI controller v4.2 user guide*, 2023.
- [105] P. Denholm, T. Mai, et al., “Inertia and the power grid: A guide without the spin,” National Renewable Energy Laboratory, Technical Report, May 2020.
- [106] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

- [107] C. Xu, J. Shen, et al., “An intrusion detection system using a deep neural network with gated recurrent units,” *IEEE Access*, vol. 6, pp. 48 697–48 707, Aug. 2018.
- [108] NetSNMP, *NetSNMP*, Jun. 2024. Available: <http://www.net-snmp.org/>, Accessed: Mar. 2025.
- [109] SQLite Consortium, *What is SQLite?* Feb. 2025. Available: <https://www.sqlite.org/index.html>, Accessed: Mar. 2025.
- [110] Elastic, *Elasticsearch*, 2025. Available: <https://www.elastic.co/elasticsearch>, Accessed: Dec. 2025.
- [111] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, Jul. 2009.
- [112] OSS Nokalva Inc., *Asn.1 made simple - what is asn.1?* 2025. Available: <https://www.oss.com/asn1/resources/asn1-made-simple/introduction.html>, Accessed: Mar. 2025.
- [113] C. I. Ciontea, “The use of symmetrical components in electrical protection,” in *2019 72nd Conference for Protective Relay Engineers (CPRE)*, 2019, pp. 1–8.
- [114] P. Pols, *The unified kill chain: Raising resilience against advanced cyber attacks through threat modeling*, Feb. 2023. Available: <https://www.unifiedkillchain.com/>, Accessed: Oct. 2025.
- [115] MITRE, *Cyber threat intelligence repository expressed in STIX 2.0*, 2025. Available: <https://github.com/mitre/cti>, Accessed: Oct. 2025.
- [116] OASIS Cyber Threat Intelligence, *Introduction to stix*, 2024. Available: <https://oasis-open.github.io/cti-documentation/stix/intro>, Accessed: Oct. 2025.
- [117] G. D. Tizio, M. Armellini, and F. Massacci, “Software updates strategies: A quantitative evaluation against advanced persistent threats,” *IEEE Trans. Softw. Eng.*, May 2022.
- [118] Neo4j, *Neo4j: Graph database and analytics*, 2025. Available: <https://neo4j.com/>, Accessed: Oct. 2025.

- [119] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [120] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, Jan. 2017.
- [121] R. Manrique and O. Marino, "Knowledge graph-based weighting strategies for a scholarly paper recommendation scenario," *Workshop on Knowledge-aware and Conversational Recommender Systems (KaRS)*, vol. 7, pp. 1–4, 2018.
- [122] G. Mongaras, T. Dohm, and E. C. Larson, "Cottention: Linear transformers with cosine attention," *arXiv: Computer Science - Machine Learning*, Sep. 2024.
- [123] D. Maček, I. Magdalenić, and N. B. Ređep, "A systematic literature review on the application of multicriteria decision making methods for information security risk assessment," *International Journal of Safety and Security Engineering*, vol. 10, no. 2, pp. 161–174, Apr. 2020.
- [124] J. L. Puga, M. Krzywinski, and N. Altman, "Bayesian networks," *Nature Methods*, vol. 12, pp. 799–800, Sep. 2015.
- [125] Bayes Server, *Bayesian networks - an introduction*, 2025. Available: <https://www.bayesserver.com/docs/introduction/bayesian-networks/>, Accessed: Oct. 2025.
- [126] M. Khosravi-Farmad and A. Ghaemi-Bafghi, "Bayesian decision network-based security risk management framework," *Network and Systems Management*, vol. 28, pp. 1794–1819, Aug. 2020.
- [127] K. Huang, C. Zhou, et al., "Application of bayesian network to data-driven cyber-security risk assessment in SCADA networks," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Dec. 2017, pp. 1–6.

- [128] J. Wang, K. Fan, et al., “A method for information security risk assessment based on the dynamic bayesian network,” in *2016 International Conference on Networking and Network Applications (NaNA)*, Sep. 2016, pp. 279–283.
- [129] A. Ankan and J. Textor, “Pgmpy: A python toolkit for bayesian networks,” *Journal of Machine Learning Research*, vol. 25, no. 265, pp. 1–8, 2024.
- [130] R. M. Gray, *Entropy and Information Theory*, 2nd. New York, NY, USA: Springer, 2011, ISBN: 978-1-4419-7970-4.
- [131] EPRI, *NESCOR*, 2024. Available: <https://smartgrid.epri.com/nescor.aspx>, Accessed: Oct. 2024.
- [132] NESCOR, *Electric sector failure scenarios and impact analyses – version 3.0*, Electric Power Research Institute (EPRI), Dec. 2015. Available: <https://smartgrid.epri.com/doc/nescor%20failure%20scenarios%20v3%2012-11-15.pdf>, Accessed: Oct. 2025.
- [133] IAC Electricals, *Difference between transmission and distribution lines*, 2025. Available: <https://www.iacelectricals.com/difference-between-transmission-and-distribution-lines/>, Accessed; Oct. 2025.