A Multi-Agent Defense Methodology with Machine Learning against Cyberattacks on Distribution Systems

Jennifer Appiah-Kubi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Chen-Ching Liu, Chair
Paul K. Ampadu
Ming Jin
Homero G. M. Escobar
Ali Mehrizi-Sani

June 21, 2022
Blacksburg, Virginia

Keywords: Distribution Automation, Anomaly Detection, Direct Switching Attack, Intrusion Prevention, Multi-agent System, Consensus Protocol, Machine Learning, Reinforcement Learning, Bi-level Optimization, Attacker-Defender Model, Unbalanced Multi-Phase System, Supply Chain Attack, Petri Net, Distribution Network Configuration

A Multi-Agent Defense Methodology with Machine Learning against Cyberattacks on Distribution Systems

Jennifer Appiah-Kubi

(ABSTRACT)

The introduction of communication technology into the electric power grid has made the grid more reliable. Power system operators gain visibility over the power system and are able to resolve operational issues remotely via Supervisory Control And Data Acquisition (SCADA) technology. This reduces outage periods. Nonetheless, the remote-control capability has rendered the power grid vulnerable to cyberattacks. In December 2015, over 200,000 people in Ukraine became victims of the first publicly-reported cyberattack on the power grid. Consequently, cyber-physical security research for the power system as a critical infrastructure is in critical need.

Research on cybersecurity for power grids has produced a diverse literature; the multi-faceted nature of the grid makes it vulnerable to different types of cyberattacks, such as direct power grid, supply chain and ransom attacks. The attacks may also target different levels of grid operation, such as the transmission system, distribution system, microgrids, and generation. As these levels are characterized by varying operational constraints, the literature may be categorized not only according to the type of attack it targets, but also according to the level of power system operation under consideration. It is noteworthy that cybersecurity research for the transmission system dominates the literature, although the distribution system is noted to have a larger attack surface.

For the distribution system, a notable attack type is the so-called direct switching attack, in which an attacker aims to disrupt power supply by compromising switching devices that

connect equipment such as generators, and power grid lines. To maximize the damage, this attack tends to be coordinated as the attacker optimally selects the nodes and switches to attack. This decision-making process is often a bi- or tri-level optimization problem which models the interaction between the attacker and the power system defender. It is necessary to detect attacks and establish coordination/correlation among them. Determining coordination is a necessary step to predict the targets of an attack before attack completion, and aids in the mitigation strategy that ensues.

While the literature has addressed the direct switching attack on the distribution system in different ways, there are also shortcomings. These include: (i) techniques to establish coordination among attacks are centralized, making them prone to single-point failures; (ii) techniques to establish coordination among attacks leverage only power system models, ignoring the influence of communication network vulnerabilities and load criticality in the decisions of the attacker; (iii) attacker-defender optimization models assume specific knowledge of the attacker resources and constraints by the defender, a strong unrealistic assumption that reduces their usability; (iv) and, mitigation strategies tend to be static and one-sided, being implemented only at the physical level, or at the communication network level.

In light of this, this dissertation culminates in major contributions concerning real-time decentralized correlation of detected direct switching attacks and hybrid mitigation for electric power distribution systems. Concerning this, four novel contributions are presented: (i) a framework for decentralized correlation of attacks and mitigation; (ii) an attacker-defender optimization model that accounts for power system laws, load criticality, and cyber vulnerabilities in the decision-making process of the attacker; (iii) a real-time learning-based mechanism for determining correlation among detected attacks and predicting attack targets, and which does not assume knowledge of the attacker's resources and constraints by

the power system defender; (iv) a hybrid mitigation strategy optimized in real-time based on information learned from detected attacks, and which combines both physical level and communication network level mitigation.

Since the execution of intrusion detection systems and mechanisms such as the ones proposed in this dissertation may deter attackers from directly attacking the power grid, attackers may perform a supply chain cyberattack to yield the same results. Although, supply chain cyberattacks have been acknowledged as potentially far-reaching, and compliance directives put forward for this, the detection of supply chain cyberattacks is in a nascent stage. Consequently, this dissertation also proposes a novel method for detecting supply chain cyberattacks. To the best of the knowledge of the author, this work is the first preliminary work on supply chain cyberattack detection.

A Multi-Agent Defense Methodology with Machine Learning against Cyberattacks on Distribution Systems

Jennifer Appiah-Kubi

(GENERAL AUDIENCE ABSTRACT)

The electric power grid is the network that transports electricity from generation to consumers, such as homes and factories. The power grid today is highly remote-monitored and controlled. Should there be a fault on the grid, the human operator, often remotely located, may only need to resolve it by sending a control signal to telemetry points, called nodes, via a communication network. This significantly reduces outage periods and improves the reliability of the grid. Nonetheless, the high level connectivity also exposes the grid to cyberattacks. The cyber connectivity between the power grid and the human operator, like all communication networks, is vulnerable to cyberattacks that may allow attackers to gain control of the power grid. If and when successful, wide-spread and extended outages, equipment damage, etc. may ensue. Indeed, in December 2015, over 200,000 people in Ukraine became victims to the first publicly reported cyberattack on a power grid. As a critical infrastructure, cybersecurity for the power grid is, therefore, in critical need.

Research on cybersecurity for power grids has produced a diverse literature; the multi-faceted nature of the grid makes it vulnerable to different types of cyberattacks, such as direct power grid, supply chain and ransom attacks. Notable is the so-called direct switching attack, in which an attacker aims to compromise the power grid communication network in order to toggle switches that connect equipment such as generators, and power grid lines. The aim is to disrupt electricity service. To maximize the damage, this attack tends to be coordinated; the attacker optimally selects several grid elements to attack. Thus, it is necessary to both detect attacks and establish coordination among them. Determining coordination is

a necessary step to predict the targets of an attack before attack completion. This aids the power grid owner to intercept and mitigate attacks. While the literature has addressed the direct switching attack in different ways, there are also shortcomings. Three outstanding ones are: (i) techniques to determine coordination among attacks and predict attack targets are centralized, making them prone to single-point failures; (ii) techniques to establish coordination among attacks leverage only power system physical laws, ignoring the influence of communication network vulnerabilities in the decisions of the attacker; (iii) and, studies on the interaction between the attacker and the defender (i.e., power grid owner) assume specific knowledge of the attacker resources and constraints by the defender, a strong unrealistic assumption that reduces their usability.

This research project addresses several of the shortcomings in the literature, particularly the aforementioned. The work focuses on the electric distribution system, which is the power grid that connects directly to consumers. Indeed, this choice is ideal, as the distribution system has a larger attack surface than other parts of the grid, and is characterized by computing devices with more constrained computational capability. Thus, adaptability to simple computing devices is a priority. The contributions of this dissertation provide leverage to the power grid owner to intercept and mitigate attacks in a resilient manner. The original contributions of the work are: (i) a novel realistic model that shows the decision making process of the attacker and their interactions with the defender; (ii) a novel decentralized mechanism for predicting the targets of coordinated cyberattacks on the electric distribution grid in real-time and which is guided by the attack model, (iii) and a novel hybrid optimized mitigation strategy that provides security to the power grid at both the communication network level and the physical power grid level.

Since the power grid is constructed with smart equipment from various vendors, attack-

ers may launch effective attacks by compromising the devices deployed in the power grid through a compromised supply chain. By nature, such an attack is evasive to traditional intrusion detection systems and algorithms such as the aforementioned. Therefore, this work also provides a new method to defend the grid against supply chain attacks, resulting in a mechanism for its detection in a critical power system communication device.

Dedication

To God, and to my parents, who nurtured me into what I am.

Acknowledgments

I am thankful to God for the many blessings He gave me and the light He shed on my path. In particular I have found the advice, guidance and support of Prof. Chen-Ching Liu, my academic advisor, invaluable, a blessing I am truly thankful for. It is said that pleasant words are a honeycomb, sweet to the soul, and medicine to the bones. Indeed, his words of encouragement could not be an over-emphasis of this saying. Over the past four years, his friendliness, extensive and profound knowledge, and timely corrections have prodded me to go beyond what I once thought was my limit, to think outside of the box, and has nurtured me intellectually, professionally and personally. I owe him my deepest and most heartfelt appreciation.

In addition, I would like to express my sincere gratitude to my committee members, all of whom provided constructive feedback about my work. In particular, I would like to appreciate the corrections of Drs. Ali-Mehrizi Sani and Ming Jin, the valuable mentorship of Dr. Paul Ampadu, and the guidance of Dr. Homero G. Murzi, who also helped to prepare me for work in academia. I would also like to thank Dr. Vassilis Kekatos who served as a committee member for my qualifying exam, providing constructive feedback, and Dr. Virgilio Centeno who served on my masters' committee. Furthermore, I truly appreciate the support and friendship of Dr. Jerry John Kponyo and Dr. Emmanuel Frimpong from Kwame Nkrumah University of Science and Technology, Ghana. Their belief in me spurred me on in my Ph.D. journey and encouraged me to press on to the desired end.

I have found the friendship and companionship of my lab mates very fulfilling. To Ruoxi

Zhu, Nitasha Sahani, Lung-An Lee, Chensen Qi, Dr. Juan Carlos Bedoya, and Dr. Manish Singh, whom I bothered every now then, and who shared my burdens with me, thank you! My appreciation also goes Akshay Jain, Fahad Alsaeidi, and Baza Somda Rodriguez, whose insightful questions during my presentations made me think out of the box, and to Sangeetha Rajasekeran and Vivek Aditya for sharing the first blurry years of graduate school with me. I also thank Ike Dimobi, Mana Jalali, Dr. Suchishmita Biswas, and Dr. Sina Taheri, for helping to make my years at the Power and Energy Center pleasant. Moreover, I truly appreciate the assistance of Victoria Deal and Lisa Burns, our administrative assistants, who went above and beyond to ensure my time at the Power and Energy Center was comfortable and memorable. Also, I would like to express my heartfelt appreciation to Renee Cloyd, Dr. Treymane Waller, the New Horizon Graduate Scholars (NHGS) cohort, and the Center for Enhancement of Engineering Diversity (CEED) ambassadors for the friendship, mentorship and opportunities they gave me to learn, develop professionally, and serve our beloved community.

My deepest gratitude goes to Douglas Asante, his wife, Christiana Appiah, and their boys, Yaw and PJ, whose friendship and love created the home I needed in Blacksburg. I cannot over-emphasize how massively appreciative I am of them. I thank Alfred Agbekudzi who has been a true friend these four years. To Dr. Ange Kakpo, Nicole Nunoo, Festus Anima, and the rest of the African graduate students whom I shared many pleasant memories with, thank you for your friendship. My roommates – Rohini Banik, Sabreen Hamad, Parul Manocha – have been exceptional and I owe them my appreciation.

My deep heartfelt gratitude goes to my dearest friend and sister, Crystal Haun, whose sisterly affection and care for me was sweet and encouraging. I also truly appreciate all the amazing Christian brothers and sisters from Christ Church, Radford, who met my need for

companionship and opened their homes to me: Pastor Anthony Mathenia, McKenzie West and her family, Sue and Jeff Berkeley, Mike and Bethany Atha, Jeneca and Noel. A special thanks goes to Alex Latham, Abi and Maddie for being my friends and for the numerous times they gave me a ride to church. I also appreciate the friendship of John Bayelma Konlan, Dr. Obed Worlanyo Abotsi, and George Obeng-Akrofi, my friends who shared my Ph.D. journey with me in their own universities and offered the friendly presence necessary to spur me on. I truly am appreciative of Mr. Eric Obeng, his wife Lydia T. Obeng, and their children, Norbert, Neris, and Natalie, for being there for me all these years, for opening their home to me and recognizing me when I was once a stranger.

My journey could not have been completed as beautifully as it has been without the constant support and prayer of my family. The rally of my sister Millicent, and my brothers, Eric, Francis, Mike and Andrew was deeply encouraging. My parents, Mr. Francis Appiah-Kubi and Mrs. Felicia Agyekum, were a towering support for me over the long distance. They understood little about my work, yet they boasted of me proudly. My dearest uncle, Charles Agyekum, and his wife Afua Boafo, and their girls, Adwoa and Adwoa, supported me in ways hardly expressible in words. To them, I owe my sincerest appreciation. Thank you to my cousins, aunts, uncles, and grandparents, who constantly prayed for and with me and had nothing but good wishes in their hearts for me. Again, I would like to express my deepest gratitude to my godparents, Mr. and Mrs. Yankey for caring deeply for and about me. I am very thankful for the love of Gregory Annan, the man of my dreams, for staying by my side, for understanding me and sharing my burdens.

Finally, I would like to acknowledge the financial support from the National Science Foundation (award no. 1837359), SIEMENS Corporation, Commonwealth Cyber Initiative, the Bradley Department of Electrical and Computer Engineering, and the Power and Energy

Center of Virginia Tech.

Publications from this Dissertation

Journal Papers

J. Appiah-Kubi and C. -C. Liu, "Decentralized Intrusion Prevention (DIP) Against Co-Ordinated Cyberattacks on Distribution Automation Systems", *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 389–402, 2020.

L. –A. Lee, C. -C. Liu, J. Wang, **J. Appiah-Kubi**, K. P. Schneider, F. K. Tuffner, and D. T. Ton. "Critical Values of Cyber Parameters in a Dynamic Microgrid System", *IET Generation, Transmission & Distribution*, vol.16, no. 1, pp. 99–109, 2022.

(Under review) C. -C. Liu, D. Boroyevich, I. Cvetkovic, A. K. Jain, N. Sahani, L. –A. Lee, **J. Appiah-Kubi**, K. P. Schneider, F. K. Tuffner, D. Ton, "Microgrid Building Blocks: Concept and Feasibility", *IEEE Transactions on Sustainable Energy*, May 2022.

(Under review) **J. Appiah-Kubi**, and C. –C. Liu, "Cyberattack Correlation and Mitigation for Distribution Systems via Machine Learning", *IEEE Open Access Journal of Power and Energy*, June 2022.

Conference Papers

J. Appiah-Kubi and C. -C. Liu, "Decentralized Correlation and Mitigation of Cyberattacks on Distribution Systems", *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 2021, pp. 1–6.

J. Appiah-Kubi, C. –C. Liu, R. Zhu, M. Otto, J. Vempati, "Detection of Supply Chain Cyberattacks for Power Grids", *CIGRE 2022 Kyoto Symposium*, Japan, 2022.

Book

C. -C. Liu, J. C. Bedoya, N. Sahani, A. Stefanov, **J. Appiah-Kubi**, C. C Sun, J. Y. Lee, and R. Zhu, "Cyber–Physical System Security of Distribution Systems", *Foundations and Trends in Electric Energy Systems 4*, no. 4, pp. 346–410, 2021.

Contents

List of Figures

List of Tables

Nomenclature

$\mathcal{S}$	Set of source nodes
$\mathcal{D}$	Set of demand nodes
$\mathcal{N}$	Set of nodes in distribution network
$\mathcal{E}$	Set of lines
$\mathcal{D}_s$	Set of demand nodes whose load are connected through remote-control switches
$\mathcal{E}_s$	Set of lines with remote-control switches
$\mathcal{D}_{sp}$	Set of demand remote-control switches whose agents have shut down remote control capability
$\mathcal{E}_{sp}$	Set of line remote-control switches whose agents have shut down remote control capability
$\overline{V}$	Upper voltage limit
$\underline{V}$	Lower voltage limit
q_i	Attack quality of demand node i
κ_i	Criticality of load at demand node i
Φ_i	Set of phases at node i
Φ_{ij}	Set of line phases between node i and node j
r^{ij}	Resistance matrix for lines between node i and node j

x^{ij} Reactance matrix for lines between node i and node j

R Attacker's monetary reward per kW disrupted power

B Attacker's budget

C_F Fixed cost of attack

$C_{e_t}^{ij}$ Attacker's variable cost to toggle remote-control switch of line (i, j)

$C_{e_k}^{ij}$ Attacker's variable cost to perform denial of service attack on remote-control switch of line (i, j)

C_d^i Attacker's variable cost to toggle remote-control switch connecting load at demand node i

e_{ij}^t Binary variable indicating that remote-control switch of line (i, j) is to be toggled

e_{ij}^k Binary variable indicating that remote-control switch of line (i, j) is to be kept in the current state

z_i Binary variable indicating that load at demand node i is to be disconnected

s_{ij} Binary variable indicating the state of remote-control switch on line (i, j)

s_{ij}^c Binary variable indicating current state of remote-control switch on line (i, j)

y_{ij} Binary variable set by operator to indicate the state of remote-control switch on line (i, j) which is not selected for attack

w_{ij} Binary variable indicating power flow direction on line (i, j)

F_{ij} Power flow limit for line (i, j)

P_i^ϕ Active power injected into phase ϕ of source node i, or the load served at phase ϕ of demand node i following attack

P_{ij}^ϕ Active power flow on phase ϕ of line (i, j) following attack

P_{ic}^ϕ Active demand at phase ϕ of node i prior to attack

Q_i^ϕ Reactive power injected into phase ϕ of source node i, or the reactive load served at phase ϕ of demand node i following attack

Q_{ij}^ϕ Reactive power flow on phase ϕ of line (i, j) following attack

Q_{ic}^ϕ Reactive demand at phase ϕ of node i prior to attack

θ_i^ϕ Angle of load at phase ϕ of node i

U_i^ϕ Square of voltage magnitude at phase ϕ of node i

δ Scaling parameter used in re-configuring the distribution network

W Sliding window of time within which anomaly is monitored

T Time period within which communication level mitigation is enforced

c_t Default NIDS threshold for monitoring attack type c

c_n New NIDS threshold for monitoring attack type c following attack

c_r Maximum recorded normal event related to attack c

ρ Attack likelihood index

$\mathcal{N}_L$ Set of unique communication network types in the set of received alerts

$\mathcal{N}_N$ Set of unique communication network types of all agents in the distribution system

$\mathcal{F}_L$	Set of unique firmware types in the set of received alerts
$\mathcal{F}_N$	Set of unique firmware types of all agents in the distribution system
u^n	Communication network type of an agent
u^f	Firmware type of an agent
K	Number of alerts to activate physical mitigation
C	Set of likely attacks
ω_c	Weight assigned to an attack c
ν	Attack potential
n	Percentage reduction in thresholds
t	Maximum time allowed between two attacks, after which all nodes enter protective mode
$\mathcal{G}$	Graph model of distribution network
$\mathcal{V}$	Set of nodes of graph $\mathcal{G}$
E	Set of edges of graph $\mathcal{G}$
k	An arbitrary graph state in agent communication
μ	Weight assigned to one of three attack correlation factors
m	Number of nodes in graph $\mathcal{G}$
$\mathbf{A}(k)$	Adjacency matrix in state k of graph $\mathcal{G}$
$\Delta(k)$	Degree matrix in state k of graph $\mathcal{G}$

$p^{[i]}(k)$ Vector in the memory of agent i for storing the values of its neighbors in the kth state

$\sigma_i(k)$ Message sent by agent i in the kth state of the third phase

D Diameter of the network graph

$n_c(k)$ Number of communicating agents in state k

$g_i(k)$ ID of the selected value of agent i in the kth state of the third phase

$\mathbf{v}^{[i]}(k)$ Vector in the memory of agent i for storing the IDs of its neighbors in the kth state

Chapter 1

Introduction

By the integration of information and communications technology (ICT), the electric power grid has evolved into a more automated and remotely controllable system. This has allowed for Supervisory Control And Data Acquisition (SCADA), which results in a more observable and controllable grid with increased reliability. Nevertheless, the remote control capability has made the power grid vulnerable to cyberattacks. This is especially the case considering that operational technology (OT) networks, once isolated, are now connected to general-purpose networks [1]. If and when successful, a cyberattack on the power grid may result in cascading failures, extended and/or far-reaching outages, and equipment damage, among others.

In December 2015, over 200,000 people in Ukraine suffered a power outage that lasted about six hours [2], [3]. The attackers executed malware through phishing emails to obtain VPN credentials. From this attack, remote control actions were launched through the operations center computers. Denial of Service (DoS) attacks jammed phone reports of the outage to the call center. Furthermore, the data destruction software, KillDisk, was used to erase the reboot software in workstations, causing a delay in power system restoration. Observations made from the Ukraine attack include: (i) the hackers were knowledgeable about the operation of the targeted grid; (ii) they were able to manipulate the cyber-power system from the Distribution System Operator (DSO) operations center; (iii) and, the hackers had knowledge of critical control and operation devices. The in-depth information was obtained

by penetrating the SCADA system and staying undetected for at least six months. After observing for six months, the hackers gained sufficient knowledge about the operation and critical information of the power system. With the information garnered, the hacker(s) conducted an attack through the SCADA system to operate circuit breakers in the substations, causing a major power outage.

As demonstrated by the real-world cyberattack, it is critical to fully understand the vulnerabilities of the cyber-power system in order to develop the capabilities for detecting cyber intrusions and take timely mitigation actions. Although cyber intrusions can be launched by compromising control center computers, damages could also be caused by man-in-the-middle attacks on the communication system between the control center and field devices. Therefore, defense of the power grid and its communication system is a critical issue for power systems. This is even more critical for the electric distribution system, which is widely acknowledged to have a wider attack surface and is characterized by numerous computationally constrained intelligent devices.

By nature, the power grid, including distribution systems, is vulnerable to various forms of cyberattacks [4], [5] such as false data injection attacks [6], [7], and load altering attacks [8]–[10]. These attacks are threats to stability and control of the target power grid. However, they must be covertly and stealthily launched, making them difficult to execute. Another attack type that may well effect dire consequences on the power grid is the control signal attack, including the direct switching attack and pricing attacks [11]. In control signal attacks, the attacker aims to gain direct control over the physical device, and the attacks are often not covert [4].

While researchers have focused on direct power grid attacks, such as the aforementioned, major attack incidents in recent times have exposed the criticality of cyber security for supply chains. For critical infrastructures such as the power grid, supply chain cyber security

is essential for reliability and resilience of the cyber-physical system.

Supply chain attacks can be launched on a target through a compromised product, service or connection to and/or from a supplier. All levels of today's power system (e.g., transmission, distribution, etc.) make use of equipment from various vendors for advanced automation and control. Both software and hardware components may need to be updated periodically, or even replaced. This may lead to benign and/or expected behavioral changes of equipment. However, the change may also be effected through a compromised supply chain of malicious nature. The recent SolarWinds attack [12] on the US government and private business computer systems is a severe lesson to learn from.

Due to their severity, direct switching attacks and supply chain attacks are the focus subjects of this dissertation, whose work presents major novel contributions toward their detection and mitigation.

1.1 Direct Switching Attacks

By direct switching attacks, switches and circuit breakers connecting power system equipment such as lines, load, and generators are toggled. The attacks tend to be coordinated as multiple elements in the grid need to be attacked to achieve the objective of the attacker on the radial distribution network. In [13], a set of decentralized algorithms are presented to detect man-in-the-middle attacks on a distribution system. The algorithms aim to prevent direct switching of circuit breakers and tampering with relay settings that could lead to voltage violations and inconsistent protections settings. The concept of attack target prediction is explored in [14]. Here, attack templates are used to pre-compute substation correlation sets for attacks. When an attack is detected at a substation, the closest fitting set is selected and protected. It is noted that the technique in [14] uses a centralized architecture.

The direct switching attack is also studied using interdiction models of the interactions between the attacker and system operator in a bi- or tri-level optimization problem [5]. The tri-level optimization problem is known as the defender-attacker-defender (DAD) model whereas the bi-level model is an attacker-defender (AD) model. In [15] a DAD model is presented that includes the defender's planning stage hardening decisions, the attacker's coordinated decisions to maximize damage, and the defender's attack response such as distribution network reconfiguration (DNR) and optimal DG islanding. The DAD model in [16] considers an attacker whose coordinated attack is both cyber and physical in nature and who selects the optimal time to launch attacks. The model is formulated over a 24-hour time horizon. In [17], an AD model is presented in which the defender's constraints include AC Optimal Power Flow (OPF) equations. Reference [18] proposes a zero-sum stochastic game approach toward modeling the relationship between the attacker and the defender.

1.1.1 Drawbacks of the Literature on Direct Switching Attacks

While the aforementioned studies have established their capabilities, there are also potential drawbacks:

1. In both AD and DAD models, the attacker is assumed constrained by the number of lines/nodes they are able to attack. The assumption in DAD models especially is that a protected line/node cannot be attacked. Clearly, the assumptions may not be valid.

2. By modeling the actions of defender and attacker as a single tri-level optimization problem, interdiction studies assume knowledge of the attacker's constraints by the defender. However, in practice, the true optimal defender action may be significantly different from what is found by the multi-level optimization problem. A preferred approach is for the defender to take optimal actions, having *learned* the motives of the attacker.

3. While an optimal mitigation strategy is enforced in response to an optimally launched attack, the mitigation strategy is implemented *after* attacks to restore load. Nevertheless, a preferred approach is to dynamically curtail the ability of the attacker *before* attack completion.

4. Mitigation strategies are either enforced at the communication level alone or the physical level alone. Mitigation on both levels is a holistic solution that provides stronger resilience.

1.1.2 Contributions to Direct Switching Attack Modeling and Mitigation

The major contributions towards modeling of direct switching attacks and their mitigation are as follows:

1. A novel more-realistic bi-level optimization model is presented that models the interaction of the attacker with the power system defender.

2. Guided by the attack model, a novel machine-learning-based technique for learning the motives of the attacker, and consequently, for predicting the targets of an attack in real-time is also proposed. The technique assumes no knowledge of attacker information by the power system defender.

3. A hybrid mitigation strategy that combines both physical level and communication network level mitigation is proposed. Both levels of the strategy are optimized in real-time based on information learned from detected attacks. The physical level mitigation performs contingency analysis and distribution network reconfiguration (DNR). To the best of the author's knowledge, this is the first combination of both physical-

and communication-level mitigation that also leverages the learned behavior of the attacker. Thus, compared to the existing literature, the ability of the attacker to complete attacks is curtailed.

1.2 Coordinated Cyberattacks

In a coordinated cyberattack, an attacker may use several attack strategies to attack one target, or may attack several parts of one system, or both. The literature on studies pertaining to coordinated cyberattacks is diverse. Reference [19] explores the coordination of physical attacks and cyberattacks on the grid. The authors formulate a bi-level model for coordinating the two types of attacks which minimizes the attack cost according to a predefined budget, while seeking to maximize the reward.

Reference [14] uses attack templates to formulate correlation indices for attacks. The correlation index is a set of substations that are likely to be attacked based on certain observed attack patterns. The use of optimal power flow to determine the correlation index allows operators to schedule appropriate system reconfiguration and/or load shedding schemes in advance. In [20] the correlation index generator formulated in [14] is combined with an event manager, a correlation knowledge database and a response manager, to detect, correlate and respond to attacks.

Reference [21] proposes to use Flexible AC Transmission System (FACTS) devices to periodically perturb the reactance of certain lines in the network. Thus, an attack constructed with outdated reactances can be detected by the bad data detector (BDD). In [22], the authors propose a method to detect and correlate attacks, using data collected from IDSs installed at different substations. The technique measures correlation according to patterns of abnormal behavior, criticality of substations and the geographical correlation. This mechanism provides correlation of attacks, after they happen. Therefore, it does not aid in target

prediction.

1.2.1 Drawbacks of the Literature on Coordinated Cyberattack Detection and Mitigation

1. Determination of the targets of an attack hinges on power system models. This is unrealistic as the decisions of attackers may well depend on cyber vulnerabilities in the communication infrastructure, and on the criticality of load.

2. Establishing coordination of attack and the enforcement of mitigation is centralized, which can be prone to single point failures. A decentralized architecture is preferable.

3. In addition, the complete delegation of attack response strategies to the human operator is undesirable. In certain scenarios, the human operator may be unable to implement specific defense strategies. An example is when an attacker implements DoS so that the network is unreachable to the operator. The integration of an automated and intelligent mitigation strategy that augments the efforts of the human operator is preferable.

1.2.2 Contributions to Coordinated Cyberattack Detection and Mitigation

This dissertation develops the following major contributions towards attack correlation, and subsequently, towards predicting the targets of a coordinated cyberattack:

1. A novel switching attack problem is formulated. The attack model accounts for communication network vulnerabilities and the criticality of load. By modeling the attacker's constraint in terms of monetary costs, the model significantly improves the practicality

relative to those in the literature. Thus, this highlights the impact of planning stage security mechanisms implemented a priori. Furthermore, the model shows what type of attack is to be conducted.

2. A novel real-time decentralized mechanism for establishing coordination of attacks and predicting the targets of an attack is proposed. The coordination/prediction is performed without knowledge of the attacker's parameters, and the technique is less prone to single point failures.

1.3 Supply Chain Cyberattacks

The impact of cyberattacks on supply chains has been investigated from a business perspective [23], and the North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) 013 [24] standard outlines compliance checks that ensure and promote a trustworthy supply chain. While these are useful for different purposes, research on supply chain cyberattack *detection*, especially for a highly automated power grid, is under-developed. The state-of-the-art of cyber security measures for power grids comprises firewalls, cryptographic techniques, etc. Unfortunately, should a piece of equipment or service be compromised through the supply chain, its internal malicious processes may be unobservable by such external security systems. An anomaly detection method that observes the internal processes of the equipment is in need.

In the last three decades, process mining has been explored and applied in several respects. The core of process mining is to discover the underlying process of a system given its event log. It is also applied in conformance checking [25] to detect when a series of events deviates from an expected process. In this vein, process mining is useful for anomaly detection. In [26], process mining is applied to model the normal process of an industrial control system. The model is then applied in periodic anomaly detection. The approach is neither real-time,

nor automated.

1.3.1 Contributions on Supply Chain Attack Detection

Regarding supply chain attack detection, this dissertation presents a novel model-based method to detect supply chain attacks in a power system in real time. As a typical substation or distribution node has numerous components that may have been purchased from different vendors, the supply chain detection problem can be concentrated on one component at a time. For this study, the remote terminal unit (RTU) is the object of focus. This may include feeder RTUs in a distribution system.

The contributions are as follows:

1. The normal behavior of a RTU is learned using process mining, and modeled as a Petri Net. To the best of the knowledge of the author, this is the first application of process mining as a preliminary stage in the detection of supply chain attacks.

2. Following this, a set of algorithms are developed that leverage the mathematical support for Petri Nets to detect various forms of supply chain attacks in real time.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents a framework that allows for decentralized direct switching attack correlation. The prescriptions of the framework are used to develop a novel attack correlation technique that leverages consensus among the agents of a multi-agent system. This is presented in Chapter 3. Next, Chapter 4 presents an algorithm that follows the prescriptions of the framework to implement attack correlation. This novel algorithm is guided by a mathematical model of attacker behavior and introduces machine learning to improve on target prediction. The algorithm includes physical

level and communication level mitigation strategies. In Chapter 5, major contributions toward supply chain attack detection are presented. Finally, the dissertation is concluded in Chapter 6, and recommendations for future work are put forward.

Chapter 2

A Framework for Decentralized Attack Correlation and Mitigation on Distribution Systems[1]

As direct switching attacks tend to be coordinated, it is necessary to establish correlation among detected attacks. The electric power distribution system comprises several nodes connected by power lines to one another and to load. The network formed is typically radial. To ensure its security, a holistic approach is preferred in which the network is considered as a unit. This necessitates a means by which the entire network is monitored. In the event that suspicious activity is detected, the network is required to mitigate the impact of the attack as a unit. This can be done more effectively if the trajectory of the attack is known and predicted in real-time, as it allows the power system defender to predict the targets of the attack. Subsequently, a correlation and prediction operation is required, whose output informs the recommended mitigation action. Therefore, the proposed cybersecurity structure is made up of four key components: real-time monitoring, network-wide communication, attack target prediction, and mitigation. A security agent, i.e., an autonomous software program, is installed at each remote-control switch on a computing device that also serves

[1]

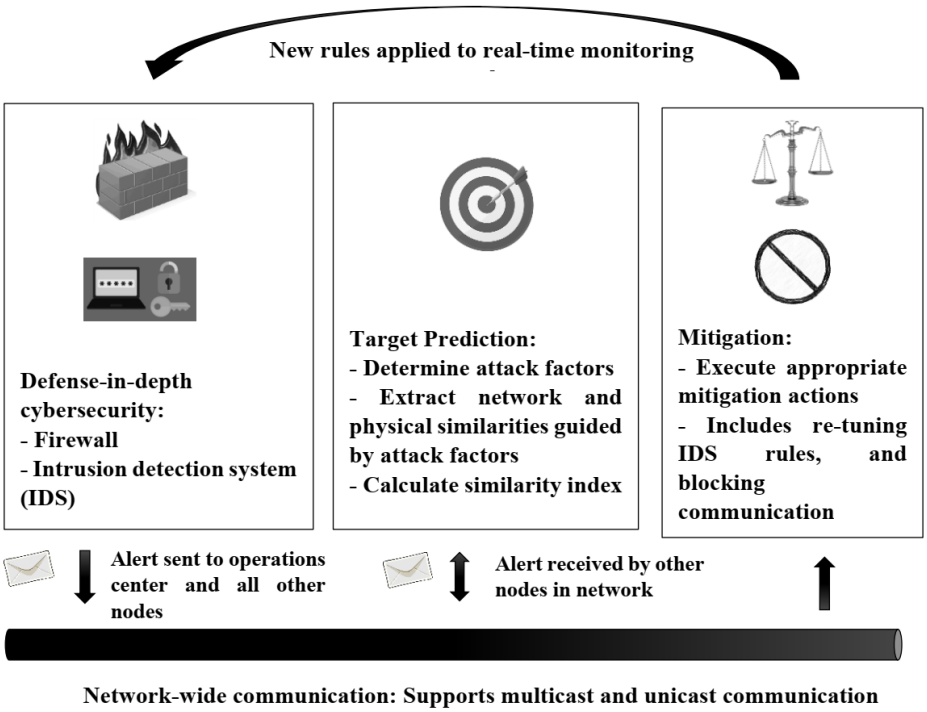

Figure 2.1: The proposed framework for decentralized correlation and mitigation of cyber-attacks

as a remote terminal unit (RTU) to implement a model of the proposed framework. An illustration of the framework is shown in Fig. 2.1.

2.1 Real-time Monitoring

Real-time monitoring is a first and critical step to detecting and mitigating attacks. It involves the deployment of cybersecurity tools and algorithms at distribution nodes. They may include firewalls, anomaly/intrusion detection systems (ADSs and IDSs), and access control lists. The detection system deployed may be network-based [27], or host-based [28]. It may also incorporate deep packet inspection to ensure that both incoming and outgoing packets adhere to certain physical laws [27].

While a defense-in-depth approach is preferred, real-time monitoring tools must conform to the limitations imposed by existing infrastructure. For instance, transmission system sub-

stations are typically housed in structures well set up with physical security layers. They are also equipped with devices that have relatively more computational and storage capabilities than is typically found in devices at the distribution system level. On the other hand, distribution system nodes may simply be pole-mounted setups, with simple intelligent devices for remote telemetry. Thus, equipment at the transmission level is generally more physically secure and capable of implementing more sophisticated security algorithms.

Ideally, real-time monitoring is required at each remote-control node in the network in order to ensure maximum protection network-wide. However, in large power networks especially, this is potentially costly. Thus, exploration of the concept of security zoning, as expounded in the ISA- 62443 standards [29], is encouraged. In security zoning, grid assets, such as nodes, are clustered into security zones and each cluster assigned to a set of cybersecurity tools. It is worth mentioning that the assigned set of security tools need not differ from one another in terms of the level of security provided, as is recommended in the literature, e.g., [30]. In the proposed framework, security agents communicate with others in the network. Therefore, a weakly protected zone or node, once compromised, may serve as the attacker's gateway into the grid.

2.2 Network-wide Communication

In the proposed framework, network-wide communication is required. This allows the distribution network to be abreast of the security issues within itself, thus, allowing suspicious behavior observed at one node to be known by other nodes. Therefore, a communication network that allows for node-to-node interactions (at least multicast communication) is required. Such a requirement is most feasible in wireless networks, such as cellular networks. Especially as development of 5G technology progresses, this option will take advantage of higher data rates and better quality of service [31]. A wired communication network may

still be implemented. Nonetheless, it is more practical to follow a radial topology, as does the power network, or a star topology. In the case of a star topology, the operations center may act as mediator between all nodes. The star topology is known to suffer from single point of failures. Consequently, wireless technology is preferred.

Once an attack is detected, the agent alerts the utility's control center and all other security agents in the distribution network. The structure and form of the alert must be standardized across the network. For the proposed cybersecurity architecture, the alert is not only required to contain information about the attack, such as the timestamp and type, but also contain information about the alerting node. This includes the size of load being supplied by the node, the criticality of that load, and the software executed by the agent deployed. This data is helpful in providing an indication of which nodes are potential targets of the attacker.

2.3 Target Prediction

Once an alert has been received, agents predict, with some certainty, the extent to which they believe they would be a target for an attacker. This prediction is made by first identifying parameters relevant in determining an attacker's choice of target nodes. Guided by these parameters, the agent compares its own network and power system data to that extracted from the received alert. The similarity between the agent's data and the information obtained from the alert is then quantified. The derived quantity becomes the certainty of prediction.

This method of target prediction is expressly different from existing ones in the literature [14], [22]. As opposed to a central operator collecting information from all nodes and determining correlation from a pre-calculated set, this method is distributed, and correlation is determined in real-time when an alert is received. It also combines both cyber and power system factors in determining the motive of an attacker, as compared to existing literature in which only power system factors (especially from power flow analysis) are used [14].

2.4 Mitigation

The ultimate goal of the proposed framework is to protect power system assets before, during and after the occurrence of a suspected attack. Therefore, following predictions made in the previous step, the agent re-tunes network cybersecurity measures as necessary, sometimes warranting stricter security controls. In an extreme event, an agent may turn off all communication functionalities if it predicts with near absolute certainty that it will be a target for an attacker. However, inasmuch as mitigation may necessitate stricter enforcement of cybersecurity measures, it should not be overly restrictive to authorized users. In essence, the level of mitigation measures activated should be commensurate with the level of threat perceived from the previous step. Thus, this framework encourages dynamic mitigation that adapts to accommodate changes in perceived threat levels. This implies that machine learning techniques, especially reinforcement learning, should be explored to encourage learning of optimal mitigation strategies. Nevertheless, in spite of the approach taken, the computational requirements should be within acceptable levels for the equipment deployed in the electric distribution network. Physical mitigation, such as a reconfiguration of the distribution system, may also be executed.

Chapter 3

Decentralized Intrusion Prevention (DIP) against Direct Switching Cyberattacks on Distribution Systems[1]

The fast increasing connectivity of the grid implies that the distribution grid today, or smart grid, is more vulnerable. Thus, research into intrusion/anomaly detection systems at the distribution level is in critical need. This study presents a novel approach toward intrusion prevention, using a multi-agent system, at the distribution system level. It is a prototype developed according to the prescriptions of the framework presented in Chapter 2.

3.1 Multi-Agent System (MAS) for Intrusion Detection

For maximum damage, intruders may attack multiple nodes in the distribution system. First, as an attack model, assume that the motive of the attacker is to disrupt power supply to

[1]©2020 IEEE, Reprinted, with permission, from J. Appiah-Kubi and C. -C. Liu, "Decentralized Intrusion Prevention (DIP) Against Co-Ordinated Cyberattacks on Distribution Automation Systems", *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 389–402, 2020, doi: 10.1109/OAJPE.2020.3029805.

an area or load of their choice. Assume also, that no social engineering practices are used, so that mainly man-in-the-middle (MitM) attacks are exploited. The set of possible attacks includes replay, denial of service, password hacks, and packet modification/falsification.

An agent is an autonomous software that is able to accept inputs from its environment, process it and take actions based on the outcome. An agent may communicate with another agent for a given purpose. Suppose that a distribution network has an agent installed at each node. The agent, implemented in Volttron [32], is installed on a computing device integrated with remote terminal unit (RTU) functions. In this application, communication-wise, agents are connected as the nodes are electrically. That is, if node A is connected to node B through a distribution line, then in the inter-agent communication structure, node A interacts directly with node B. This serves as the initial communication structure of the agents. Also, each agent has three modules: a network-based intrusion detection system (NIDS) module, a prediction module, and a social module.

Again, all agents are assumed to be aware of their own node parameters such as the communication protocol being run (e.g. DNP3, Modbus, etc.), criticality of its load, its neighbors, and software being run by communication devices deployed at the node. The node data is essential for prediction of whether an attack is coordinated, and the targets of the attack.

The proposed Decentralized Intrusion Prevention (DIP), is a four-phase algorithm, depicted in Fig. 3.1. The phases of the algorithm are explained next.

3.1.1 First Phase

In the first phase, the NIDS module of the agent monitors the local network. The NIDS may be based on any protocol, as the application demands. However, the NIDS implemented in this research is based on DNP3, specifically DNP3 with Secure Authentication v5 (SAv5), as the communication protocol between the operation center and the node. The development

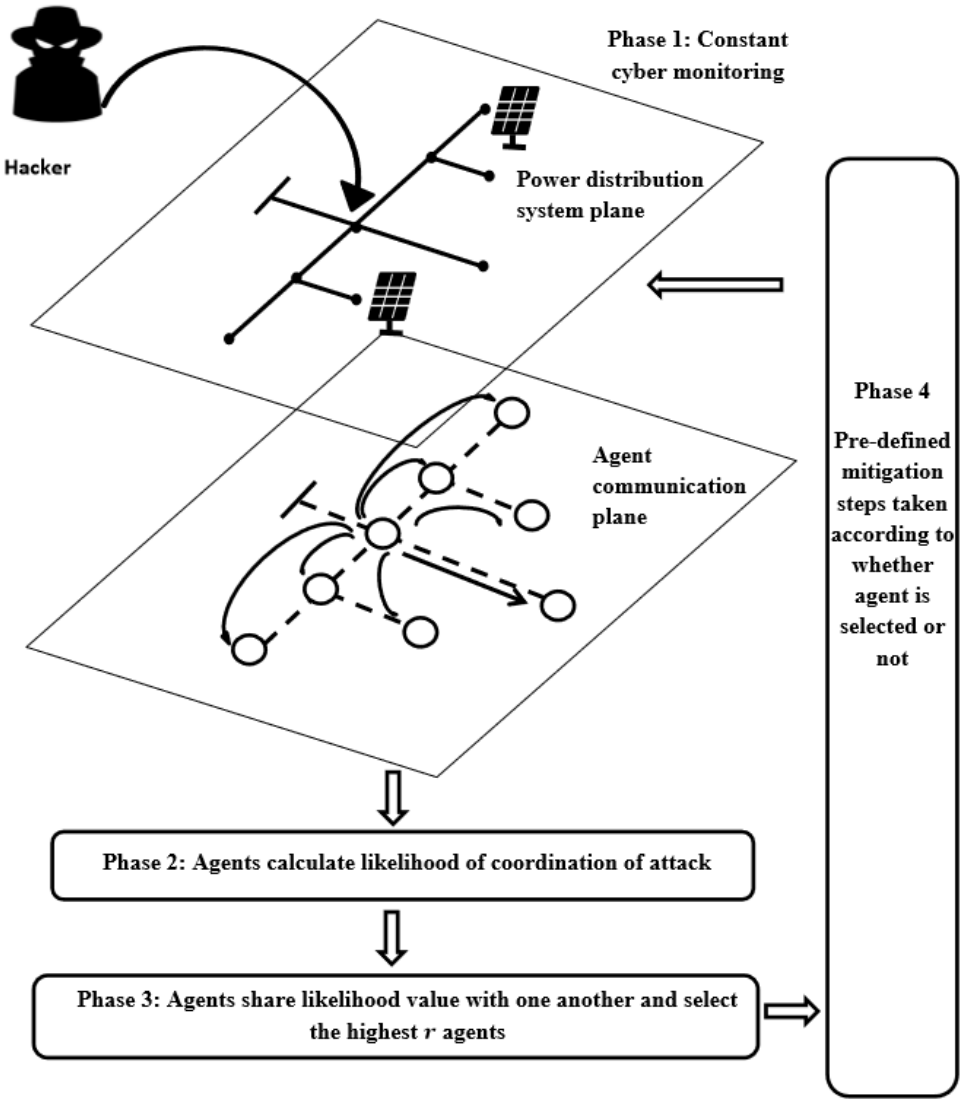

Figure 3.1: An illustration of the proposed Decentralized Intrusion Prevention (DIP) algorithm

of a secure key distribution mechanism for the protocol is assumed to be available. The algorithm for the NIDS is developed to monitor for the following set of attacks from the intruder:

- Flooding (Denial of Service (DoS)): In order to ensure that the RTU at the node is not flooded, the time difference between the arrival of packets is monitored. An alert is triggered when the time interval is lower than a pre-determined threshold value and observation is made d_t consecutive times.

- Packet falsification/modification: In DNP3 SAv5, critical functionalities such as write, delete, and operate require the receiver to challenge the sender. The sender produces a unique tag through a hashing function, which is sent to the receiver. At the receiver, the same hashing operations are performed and the results compared to the tag received. If they are equal, the identity of the sender is verified and the command is implemented. If the result at the node is not the same as the tag received, the NIDS triggers an alert.

- Replay: In DNP3 SAv5, each critical packet has a challenge sequence number. The sequence number is checked to ensure that already received numbers are not repeated. In the proposed NIDS, an alert is raised when the challenge sequence number of a received packet is smaller than or equal to the latest recorded at the node.

- Brute-force password hack: By successfully logging in, the adversary may be able to install/delete applications, and/or run commands. The login credentials are assumed unavailable to the attacker. Hence, in the proposed NIDS, p_t consecutive failed password attempts to log in is flagged.

The NIDS is able to accurately detect configured attacks. For instance, consider a complicated attack where the adversary copies the unique tag of a packet in transit and attaches

this to a fabricated packet. The NIDS at the node is able to detect this. Indeed, at the node, the hashing operation is performed over the entire received packet, and the results compared to the unique tag. This attack succeeds only when the attacker is aware of the correct hash key. In this chapter, sharing of all cryptographic keys has been assumed secure and confidential.

All agents continually monitor their own cyberspace in the first phase, using the proposed NIDS. A transition is made into the next phase if and when an intrusion is detected. Suppose an intrusion is detected at one of the nodes in phase 1. The node is required to broadcast an alert to the operations center and to all agents in the network. The alert contains the following: time stamp of the detected intrusion, ID of the reporting agent, suspected attack type, other descriptive data such as the protocol being run.

The broadcast alert is encapsulated in a TCP/IP packet. A hash digest is also added to the message to provide authentication. It is assumed that the key for this hashing operation is updated when session keys for the communication between nodes and the operation center are updated. Therefore, for securely exchanged and updated keys, the attacker is unable to falsify or modify an alert. Having received the alert, each recipient agent enters the second phase.

3.1.2 Second Phase

Each recipient agent needs to predict with some certainty whether an attacker will target its node. This is done by measuring its correlation with an alerting node (agent). The proposed correlation model uses three factors: attack patterns, criticality of load at the node, and software correlation. An attacker may be attempting different attack techniques, such as replaying an already captured packet at one node, while modifying the same at another node. Thus, the observation of attack patterns is a good indicator of correlation. In addition, nodes

that serve critical loads are by nature attractive to attackers. Subsequently, in a coordinated attack, criticality of load at a node is an indicator of correlation. Lastly, an attacker may leverage vulnerabilities that may be present in firmware and other software implemented by the intelligent device. In this case, targeted agents may not be correlated according to the load type of their nodes or attack patterns, but according to the software they run. In this chapter, these three factors are used to measure correlation, and consequently used to predict the likelihood that an attacker will target a recipient node. The measure of correlation is obtained by calculating three indices to quantify each of the three factors: intrusion potential, criticality index, and software and operational vulnerability index.

(i) Intrusion potential (ν): This is the likelihood that some observed pattern evolves into an intrusion. The received alert is compared to current security logs from the past T units of time. T is a user-defined variable. Let there be a set of attacks $\mathcal{C} = \{d, f, l, p\}$ where d, f, l, p represent flooding (DoS), packet falsification, replay, and attempted password login, respectively. For an attack $c \in \mathcal{C}$, let c_t be a threshold value and c_r be its maximum recorded normal occurrence in the security logs at the recipient node. Also, let ω_c be the weight assigned to that attack. The weight is chosen according to the following rules:

   (a) if the attack was not reported and its occurrence in security logs at the recieving station is below half of its threshold, $\omega_c = 1$,

   (b) if the attack was not reported but its occurrence in security logs at the receiving station is greater than or equal to half of its threshold, $\omega_c = 2$,

   (c) for a reported attack, if its occurrence in security logs at the recipient node is below half of its threshold, $\omega_c = 2$, and

   (d) for a reported attack, if its occurrence in security logs is greater than or equal to

Table 3.1: Criticality indices for different nodes.

Node type	Criticality index (κ)
Node supplying highly critical load (e.g. hospitals, critical infrastructures such as water supply)	0.9
Node supplying critical load (e.g.important industry and commercial load)	0.6
Node supplying non-critical load (e.g. homes)	0.3

half of its threshold, $\omega_c = 3$.

The intrusion potential is then given by:

$$\nu = \frac{\sum_{c \in \mathcal{C}} \omega_c \left(\frac{c_r}{c_t}\right)}{\sum_{c \in \mathcal{C}} \omega_c} \tag{3.1}$$

Equation (3.1) is a weighted average of ratios which correlates log patterns to a reported attack. In order to capture the possible use of different attack techniques at a time, (3.1) considers log patterns for all types of attacks in the attack set. For example, an attacker may send a command to open the load switch at a node through replay or a falsified packet, and immediately follow up with a flooding attack in order to prevent the node from sending status information to the operations center.

(ii) Criticality index (κ): In the algorithm, criticality indices are set according to Table 3.1.

(iii) Software and operational vulnerability index (ψ): The software vulnerability index measures the extent to which the affected device of the reporting node is related to that of the recipient node. The inclusion of this index is to capture similar software and/or operational vulnerabilities that an attacker could be leveraging. Set $\psi = 1$ if both devices are of the same manufacturer and use the same software versions, $\psi = 0.66$ if they are of the same manufacturer but use different software versions, $\psi = 0.33$ if

they are of the same manufacturer but different software, and $\psi = 0$ if they are not related.

Note that a node may be correlated to an alerting node in one or more of the three factors. An accurate decision is made when all factors are considered. A weighted sum of the indices takes into account all three factors and captures the overall correlation between the recipient node and an alerting node. Subsequently, having found the three indices, an empirical judgement is made as follows:

$$\rho = \mu_\kappa \kappa + \mu_\nu \nu + \mu_\psi \psi \tag{3.2}$$

$$\mu_\kappa + \mu_\nu + \mu_\psi = 1 \tag{3.3}$$

$$\mu_\kappa, \mu_\nu, \mu_\psi \geq 0 \tag{3.4}$$

The symbols μ_κ, μ_ν and μ_ψ are non-negative weights assigned to the different indices. They are normalized by equation (3.3). The value ρ is a prediction made by the recipient agent regarding the extent to which it believes it will be a target for an attacker. Clearly, ρ is affected by the values of κ, ψ and ν which are assigned based on how the features of the receiving node compare with those of the alerting node. It is also affected by the weights μ_κ, μ_ν and μ_ψ, and it is necessary to choose the weights in order to maximize the accuracy of this prediction. An experiment on tuning the weights is provided in the simulations presented in this chapter.

3.1.3 Third Phase

In this phase, agents share their judgment values, ρ, with their neighbors, encapsulating them in TCP/IP packets. This phase is implemented by the social module of the agent. A neighbor of a node is any node which has a direct communication link to it. Having received their neighbors' values, each agent selects the highest r of them, together with the IDs of the

agents whose values are selected. The selected list is shared again, and the process repeated. At each sharing stage, the current selected list is compared to the previously sent list. If the two are equal, an agent does not re-share as this will be repetitive. By doing so, network resources are reserved for nodes that have new data to share. The process is repeated until each agent has the same list of agent IDs and their corresponding values, which are indeed the highest in the network.

The proposed consensus protocol, the link drop max consensus protocol, belongs to the family of max consensus algorithms [33]–[35], in which the maximum initial value in the network is sought. However, the proposed protocol differs in the check for repeated information.

3.1.4 Fourth Phase

In this phase, agents take specific communication network level mitigation actions based on the outcome of phase 3. If an agent's value is selected, it enters a protective mode for a time period T. The agent that broadcasts the alert also enters protective mode. In protective mode, all remote control functionalities are disabled. This allows the operation center time to attend to the security issues that arise. On the other hand, if an agent is not selected, all NIDS thresholds (d_t, p_t, etc.) are set to $n\%$ of their initial values. Should another alert be broadcast within t units of time from a previously received one, all agents enter the protective mode. T, t and n are user-defined variables.

It should be noted that when a node undergoes an attack, say a DoS attack, and broadcasts this, it enters protective mode after the first round of the algorithm. Thus, before T elapses, the node cannot be attacked again since it is isolated from further communication. A node in protective mode continues to function electrically and may send periodic status information to the operations center; only outgoing traffic from the node is allowed. This is different from an explicit ingress filtering scheme implemented to mitigate attacks, as discussed in

[36]. In such schemes, packets from only the offending address are dropped. While this method may serve to selectively block an attacker's traffic so that the operator is still able to communicate with the node, a skillful attacker who sends packets with varying spoofed addresses may escape the check.

3.2 The Link Drop Max Consensus Protocol

The link drop max consensus protocol allows a group of distributed agents, each with some value, to share and select the maximum value amongst them without repeating already shared information. In this section, the terms "algorithm" and "protocol" refers to the link drop max consensus protocol. It is assumed that there is no data loss in the network.

Premise: Nodes in the distribution grid are not always communicating but only initiate a conversation when a decision is to be made. They are made aware of a decision to be made when one of the nodes informs them to that effect. The communication topology starts with an initial structure that changes as the conversation progresses. The initial structure is identical to the topology of the distribution system. It is also assumed that there are no islands. In the initial network, every pair of connected nodes has a two-way link between them.

From these, the initial model of the distribution network is a strongly connected digraph $\mathcal{G}$ with a set of vertices $\mathcal{V}$ and a set of extraverted edges E. The vertices represent the nodes while edges represent the active communication links between the nodes. Hence, 'vertex', 'agent' and 'node' are used interchangeably, as are 'edge' and 'link'.

Thus, $\mathcal{G} = (\mathcal{V}, E)$, where $\mathcal{V}$ is a non-empty set consisting of m nodes. There is no loop in the graph, i.e., there is no single edge that connects a vertex to itself.

The graph possesses an adjacency matrix $\mathbf{A}$, which is a matrix with elements $a_{ij} = \{0, 1\}$.

An element a_{ij} is 1 if node i communicates to node j, and 0 if it does not. It should be noted that $a_{ij} = 1$ necessarily means $a_{ji} = 1$ for the initial network, but is not guaranteed for all time. The degree matrix of the graph is the diagonal matrix whose diagonal elements are the total number of edges attached to a node. The neighborhood of a node $i \in \mathcal{V}$ is given as $\mathcal{N}_i := \{j : a_{ij} = 1\}$.

By nature, the state of the graph for max consensus protocols does not evolve according to a set of linear dynamic equations as is the case for the average consensus protocol (ACP) [37]. Rather, the proposed algorithm follows a sequence of discrete logical steps. Assume that nodes in the grid are selecting the max value among all the judgement values calculated (in the second phase) by the nodes in the grid.

1. At the start of a conversation, each agent i in the distribution network has an initial value $\rho_i(0)$ and an initial ID $g_i(0)$. The initial ID is the ID configured for the node. To standardize notation, let ρ be the vector of attack likelihood indices in the graph, and ρ_i be the attack likelihood index of the ith agent. The initial state vector of the graph is $\rho(0)$, and $\mathbf{A}(0) = \mathbf{A}^\top(0) \in \mathbb{R}^{m \times m}$.

2. Each agent i also possesses a vector $\mathbf{p}^{[i]}(k) \in \mathbb{R}^{|\mathcal{N}_i|}$ in memory which stores the values of its neighbors, and a vector $\mathbf{v}^{[i]}(k) \in \mathbb{R}^{|\mathcal{N}_i|}$ which stores their IDs. At the start of the conversation, $p_j^{[i]}(0) = -\infty$ and $v_j^{[i]}(0) = g_j(0)$ for $j \in \mathcal{N}_i$.

3. The information shared by agent i in event k is a concatenation of the agent's selected value and its ID, denoted by $\sigma_i(k) = \{\rho_i(k)|g_i(k)\}$.

4. Agent i shares $\sigma_i(k)$ with its neighbors if $a_{ij}(k) = 1$, for $j \in \mathcal{N}_i$.

5. For $i \in \mathcal{V}$, $\rho_i(k+1) = \max\{\|\mathbf{p}^{[i]}(k)\|_\infty, \rho_i(k)\}$. That is, the next state of node i is the maximum of all received values, including its own. The agent ID of the selected value

is also stored as $g_i(k+1)$. Thus, $\sigma_i(k+1) = \{\rho_i(k+1)|g_i(k+1)\}$.

6. Following the above steps, the adjacency matrix is updated as follows. For all $i \in \mathcal{V}$, $j \in \mathcal{N}_i$,

$$a_{ij}(k+1) = \begin{cases} 1 & \text{for } \rho_i(k+1) - \rho_i(k) > 0 \\ 0 & \text{otherwise} \end{cases}$$

7. The process is repeated from step 4 until convergence is attained.

Step 6 is essential in order to avoid repetition and subsequent information accumulation. Since the graph starts with a two-way link between every connected pair of nodes, the dropping of links by a node only makes those links one-way. The node, therefore, may still receive information from its neighbors but does not communicate back unless the update rule indicates so. Consequently, an agent may re-establish already dropped links.

The proof of convergence of the algorithm, as well as its speed, is presented next. It is shown that the speed of convergence is bounded above by the diameter, D, of the network graph.

3.2.1 Convergence of the Link Drop Max Consensus Algorithm

The conditions for convergence are:

1. A consensus is reached: for any initial state of the graph $\rho(0)$ there exists a value $\rho_b \in \{\rho(0)\}$ where $\rho_b = \max\{\rho(0)\}$, possessed by agent with initial ID g_b such that $\lim_{k\to\infty} \rho_i(k) = \rho_b$ and $\lim_{k\to\infty} g_i(k) = g_b$ for all $i \in \mathcal{V}$.

2. Agents are no longer communicating. That is, $E = \emptyset$.

In the sub-sections that follow, convergence of the algorithm has been shown, first for the case where the maximum value is being chosen and for when r highest values are being chosen (such as ranking of the 3 highest judgement values).

The Case for One Value

Assume that there is a distribution grid in which each node i has calculated some unique initial value $\rho_i(0)$. This is the first graph state $\rho(0)$. Now let the maximum value be ρ_b and the agent that has ρ_b be agent b with initial ID g_b. At the beginning of each cycle, an agent may drop the link between itself and its neighbors. The probability for this is dependent on the outcome of the update rule of the adjacency matrix from the previous cycle. Let the probability that the ith node will drop links in the kth cycle be $p_i(k)$. Then, at the end of the first cycle for agent b, $p_b(2) = 1$ since $\rho_b > \rho_i$ for all $i = 1, ..., b-1, b+1, ..., m$. Since $\rho_b(1) - \rho_b(0) = 0$, $a_{bj}(2) = 0$ for all $j \in \mathcal{N}_b$ in the second cycle.

In the first cycle, all neighbors of agent b received ρ_b. Thus, in the second cycle, $\rho_j(2) = \rho_b$, $p_j(2) = 0$ and $a_{jt}(2) = 1$ for all $j \in \mathcal{N}_b$ and all $t \in \mathcal{N}_j$. The neighbors of agent b re-share ρ_b with their own neighbors. In the third cycle, $\rho_j(3) = \rho_b$, $p_j(3) = 1$ and $a_{jt}(3) = 0$ for all $j \in \mathcal{N}_b$ and all $t \in \mathcal{N}_j$.

Thus as $k \to \infty$,

$$\rho_i(k) = \rho_b \quad \forall \quad i = 1, ..., m \tag{3.5}$$

$$p_i(k) = 1 \quad \forall \quad i = 1, ..., m \tag{3.6}$$

$$a_{ij}(k) = 0 \quad \forall \quad i, j = 1, ..., m, i \neq j \tag{3.7}$$

$$\Delta(k) = \mathbf{0} \tag{3.8}$$

$$\mathbf{A}(k) = \mathbf{0} \tag{3.9}$$

The Laplacian matrix is therefore:

$$\mathbf{L}(k) = \Delta(k) - \mathbf{A}(k) = \mathbf{0} \tag{3.10}$$

The Laplacian is now a zero matrix. Hence, it has 0 eigenvalues with multiplicity m. Also, $E = \emptyset$. There is no more active communication between any two agents. Thus, the criteria for convergence have been achieved.

The Case for r Values

In addition to the assumption of unique values in the previous case, it is assumed, without loss of generality, that $0 \leq \rho_i(0) \leq 1$ for all $i = 1, ..., m$. Suppose that agents are selecting the highest r values. Hence, as is done in phase 3, in each message sent by a node, there is a list of the best r values and their agent IDs. In this case, convergence is proven by observing the following:

(i) The system behaves like r graphs superimposed on each other. Agents in each graph find the maximum value in their graph.

(ii) The r graphs are produced from the same initial graph at the start of the communication.

(iii) In the first graph, agents vote for only the maximum value ρ_{b1}. In the second graph, agents vote for the maximum value ρ_{b2} with ρ_{b1} set to 0. In the third, agents vote for the maximum value ρ_{b3} with $\rho_{b2} = 0$ and $\rho_{b1} = 0$, and so on.

(iv) Each graph has a different sequence of adjacency matrices as the agent conversation

progresses.

(v) Nevertheless, in each graph, only the best value is being voted for and is similar to the case for one value. It follows that as $k \to \infty$, $\mathbf{L}_j(k) = \mathbf{0}$ for all $j = 1, ..., r$.

Considering that the initial graph is strongly connected, the "spread" of ρ_b from agent b to all agents to achieve convergence can be viewed as a progression on a radial static directed acyclic graph (DAG), with agent b as the root. Thus, despite the temporal nature of the graph, time analysis pertaining to static DAGs can be applied. Consequently, the speed of convergence, i.e., the number of iterations required for attaining convergence, assuming synchronism, has been shown in [38] to be bounded above by D, where D is the diameter of the graph. Thus, the time complexity is $O(D)$. In the asynchronous case, this is $O(BD)$, where B is a measure of the asynchronism [35].

It is also noteworthy that in real implementations of the algorithm, there could be $n > r$ agents with the same maximum value. Since the agent ID is stored with the selected value, it follows that until there is an explicit rule to govern how to select, there may not be convergence. This is because, while agents agree on what the maximum value is, they may not converge on the ID of the agent with this value.

3.2.2 Advantage of the Link Drop Max Consensus Protocol

Collisions are stochastic events that may occur in a given communication network. However, repeated sharing of redundant information between neighbors unnecessarily exposes the link to collisions. This is especially true for heavily constrained networks. When a collision occurs on the link between neighbor agents, a retransmission is required, which potentially increases the time required to achieve consensus.

The likelihood of collision occurring is dependent on the probability that a link is used

at the same time by neighbors. Therefore, it is reduced when the probability of a neighbor communicating decreases. This is the unique feature of the link drop max consensus protocol, when compared to other max consensus protocols, e.g., [33]–[35]; it avoids the sharing of redundant information.

In the link drop max consensus protocol, the probability that an agent has the max value, and subsequently will not communicate in the next state is $\frac{1}{n_c(k)}$ where $n_c(k)$ is the number of communicating nodes in the kth state. Considering that there is convergence, it follows that $n_c(k) \to 1$ as $k \to D$. Let the probability that a neighbor communicates in the kth state be $p_c(k)$. In the original max consensus protocol, $p_c = 1$ for all time during agent communication. However, in the link drop max consensus protocol, $p_c = 1$ only in the first state. In subsequent, states $p_c(k) = 1 - \frac{1}{n_c(k)}$. Hence, $p_c \to 0$ as $k \to D$.

For simplicity of illustration, assume a circular graph with $V = 13$ nodes, implementing the link drop max consensus protocol. All agents communicate in the first state; $p_c = 1$. For the second and third states, $p_c = 1 - \frac{1}{V}$ and $p_c = 1 - \frac{1}{V-1}$ respectively. For subsequent states $k = 3, ..., D$ in the circular graph, $n_c(k) = V - 1 - \sum_{k=3}^{D} 2(k-1)$. Therefore, $p_c(k)$ is given by:

$$p_c = 1 - \frac{1}{V - 1 - \sum_{k=3}^{D} 2(k-1)} \tag{3.11}$$

Figure 3.2 shows a plot of p_c against state k for a circular graph with 13 nodes. From this, it is shown that the link drop max consensus protocol reduces the probability of agent communication over time, and therefore reduces the likelihood of collisions over time.

3.3 Setup for Validation

The IEEE 13-Node Feeder (shown in Fig. 3.3) is used for testing and validation. In the setup, nodes 671, 675, 680 and 692 serve highly critical load; nodes 632, 633, 645 and 646

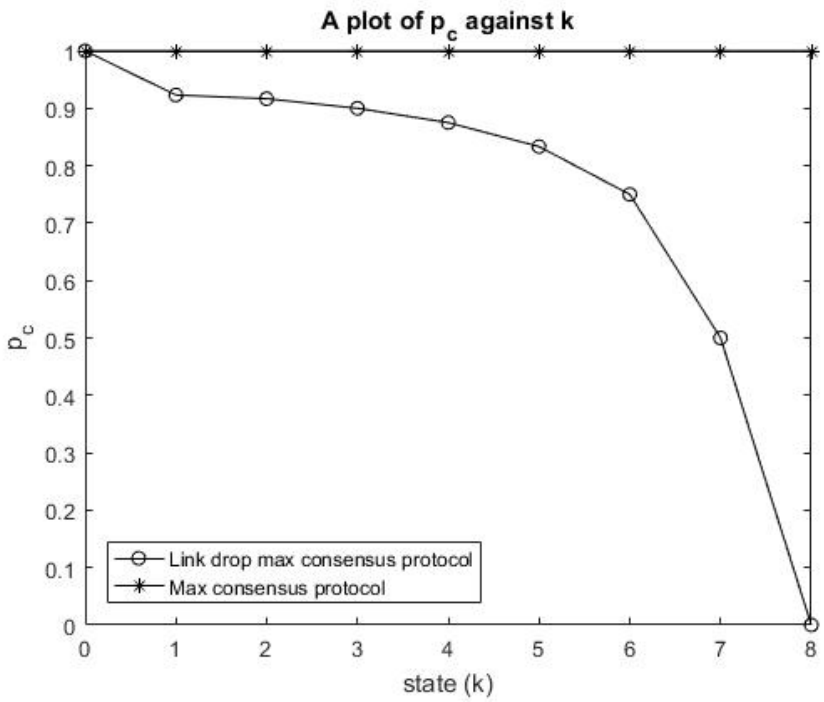

Figure 3.2: Comparing variation in p_c with state in the link drop max consensus protocol and the generic max consensus protocol for a 13-node circular graph

serve critical load; and nodes 611, 634, 652, 670 and 684 serve non-critical load.

Agents are developed using Volttron, a Unix-based Pacific Northwest National Lab (PNNL) open-source agent-based platform [32]. Agents are developed on a virtual Linux-Mint system. The agents form part of the cyber model of the grid and are initially connected to communicate with one another as the nodes are connected electrically.

Scapy, an open source packet manipulation tool [39], is used. Originally, Scapy has no DNP3 library. A DNP3 library with SAv5 functionality is built to extend its capabilities for this test. An attack simulation platform is also developed using the added DNP3 library of Scapy to implement the different attacks enlisted in this chapter, completing the cyber model.

The power system model is built using Power Factory DIgSILENT, an industry level power system simulation tool. DIgSILENT is executed on Windows 10 operating system. Matrikon OPC server is used to connect the cyber and power system models in a real time environment.

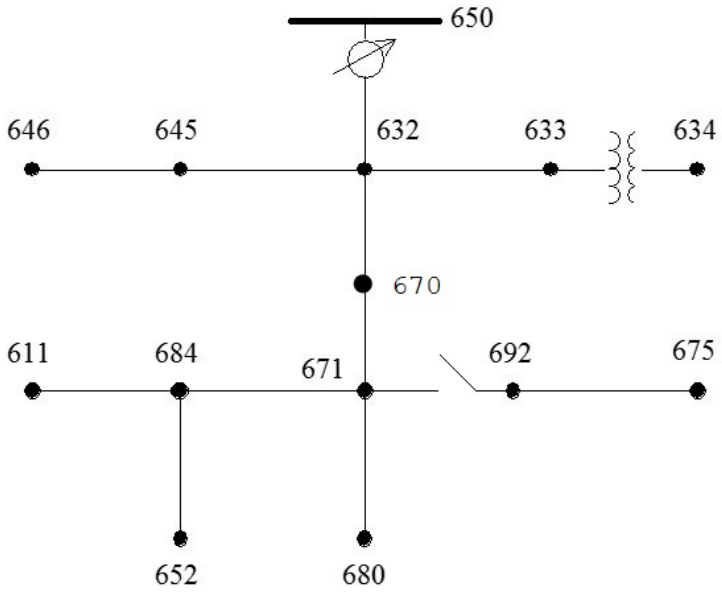

Figure 3.3: IEEE 13-Node Test Feeder

The interaction among these components is shown in Fig. 3.4.

3.4 Simulations and Results

Two studies are performed in the simulation. In the first study, some properties of DIP are investigated, while in the second, the performance of DIP under coordinated attacks is assessed.

3.4.1 Study 1: Investigating Some Properties of DIP

The efficiency of DIP in detecting attacks is dependent on that of the NIDS implemented in the first phase. Nevertheless, the accuracy of predicting the correlation of attacks is dependent on the relational weights used in finding the judgement value in the second phase. In surveyed papers on coordinated cyberattacks, authors do not explicitly assess the accuracy of suggested techniques for determining correlation. In this study, such an investigation is

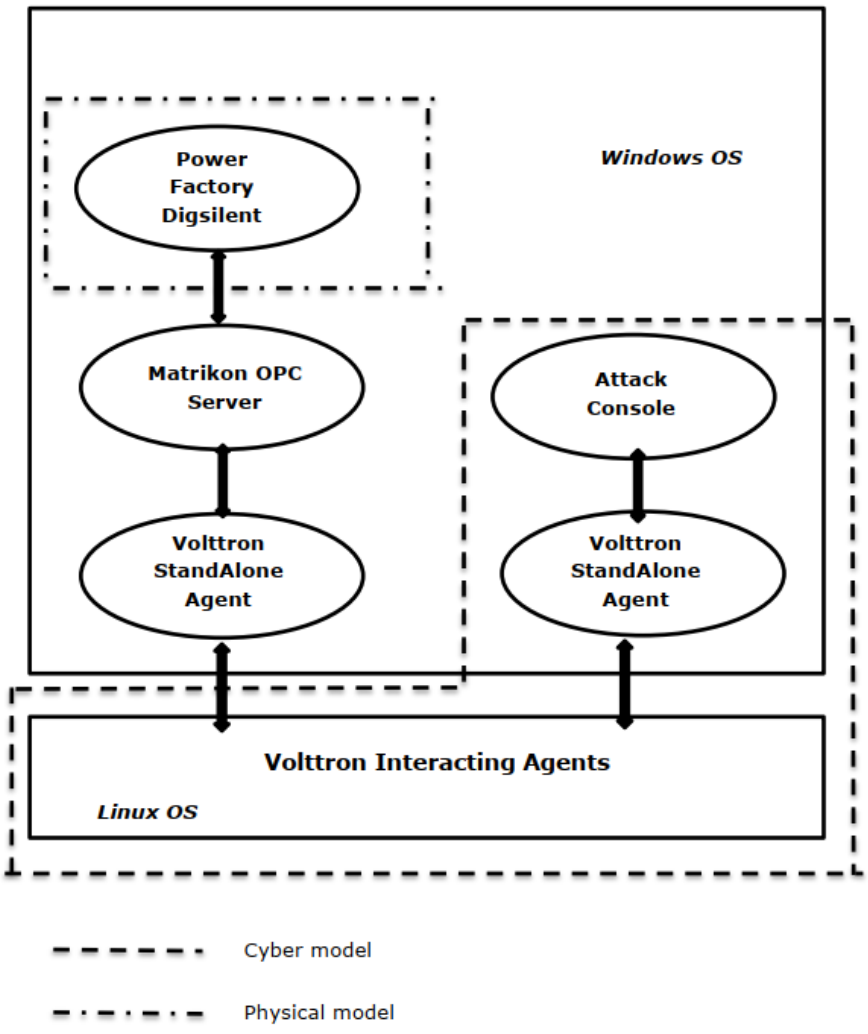

Figure 3.4: A cyber-power system simulation setup

conducted.

Efficiency of NIDS Implemented in the First Phase

The performance of the NIDS in the first phase is measured by the false positive and false negative ratios. A smaller ratio is desirable. The false positive ratio (FPR) is the proportion of normal packets that are misclassified. For the implemented NIDS, the FPR may be determined from the probability that an authorized user enters the wrong credentials p_t times, the probability that the operation center sends replayed packets or packets with incorrect hashes, and the probability that the operational center sends a cluster of packets greater than what is used in detecting flooding.

The false negative ratio (FNR) is the proportion of abnormal packets that are misclassified. For the proposed NIDS of phase 1, the FNR may be determined by comparing how many alerts are generated with how many malicious packets are sent. To measure the FNR, the NIDS is installed on the slowest computer available, which is an Intel Core i3 computer. The plot in Fig. 3.5 shows the FNR for when 2000 packets and 5000 packets are sent at varying rates.

The highest FNR, 45%, occurs when 5,000 packets are sent at a rate of 10,000 packets per second (pps). The FNR appears to increase with increasing number of packets sent. It is also observed that when 5,000 packets are sent at 7,000 pps, the FNR is close to 45% and higher than that obtained at 8,000 pps and 9,000 pps respectively. This indicates that the relationship between the rate of flooding and FNR is not exactly linear. When implemented on a real RTU, the FNR may be higher.

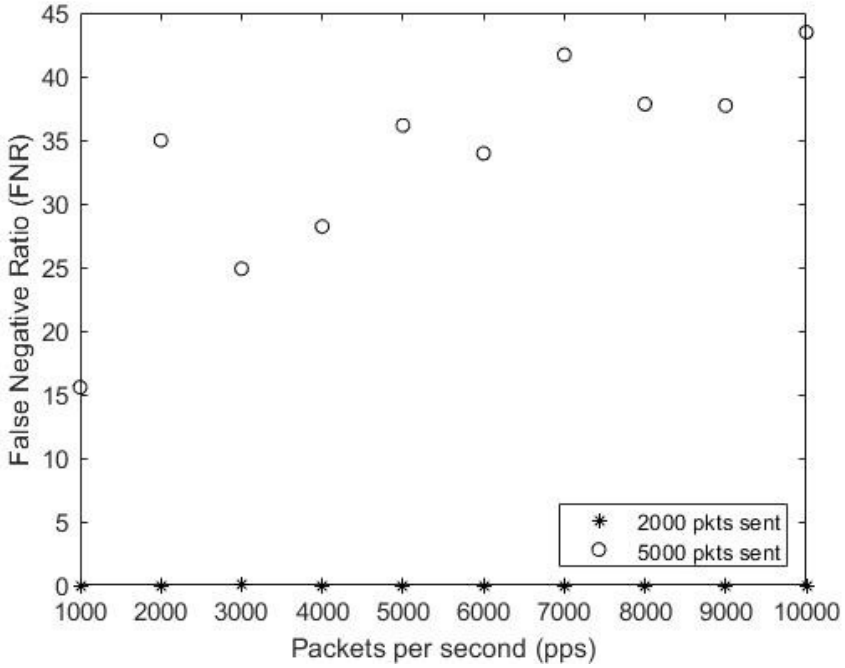

Figure 3.5: FNR with varying packet rates for different total packets sent

Effects of Relational Weights on the Prediction Success Rate of DIP

In the second phase, DIP predicts the motive and/or leverage of the attacker, and therefore predicts their target nodes. The accuracy of this prediction is termed the prediction success rate (PSR) of DIP. The PSR is the proportion of targeted nodes that enter protective mode after the first detection of intrusion. For instance, if the attacker is aiming for critical nodes, and two, instead of all four, entered protective mode at the end of the fourth phase, then the PSR is 50%. The PSR is dependent on the judgement value ρ. As long as an agent has one of the highest three ρ values, it is guaranteed to enter protective mode. It is therefore desired that at the end of phase 2, nodes whose agent will be targeted by the attacker have the highest judgement values.

The judgement value ρ is in turn determined by the probabilistic indices κ, ν, ψ and the preset weights $\mu_\kappa, \mu_\psi, \mu_\nu$. As indicated in an earlier section, the values assigned to the indices

depend on the correlation between the features of the alerting node and those of the recipient node. The PSR is therefore dependent on: (i) the correlation of features among nodes, and (ii) the preset weights.

Consider two networks, A and B, adapted from the IEEE 13-Node Test Feeder. Let all the critical nodes of network A run the same fictitious software, while those of network B run different software. Assuming an attacker targets the critical nodes, then for the same set of weights and the same attack potential, the critical nodes of A are more likely to enter protective mode than those of network B. This is because even though all critical nodes will have the same $\kappa = 0.9$, those of network A will also have $\psi = 1$, while those of network B will have different values of $\psi < 1$. Thus, the PSR of network A is higher than that of network B, for the same weights. In order to increase the PSR of network B, the weights must be changed. Therefore, for every network, the preset weights need to be well chosen in order to achieve its maximum PSR.

Algorithm 1 details a tuning procedure for choosing the weights. Let nodes 671, 675, 680, and 692 of the IEEE 13-Node Test Feeder be highly critical nodes, and let them all run a fictitious software ABC OS9 1.0.1. Using Algorithm 1, a weight μ is varied from 0.1 to 0.9 while keeping the other weights at $0.5(1-\mu)$ each. For each variation of the weight, $M = 100$ attacks are performed. The resulting PSR is recorded and a curve is plotted, as indicated in step 10 of the algorithm. This is repeated for each of the weights. Fig. 3.6 shows the curves obtained, and the common PSR that preserves equation (3.3) is approximately 85%. At the end of the tuning algorithm the weights are chosen as $\{\mu_\kappa, \mu_\psi, \mu_\nu\} = \{0.1, 0.282, 0.618\}$ and the resulting PSR is 86%.

Assume a second network adapted from the IEEE 13-Node Test feeder. The features of the nodes are set according to Table 3.2. Using the same algorithm, the weights are determined as $\{\mu_\kappa, \mu_\psi, \mu_\nu\} = \{0.406977, 0.2825, 0.310523\}$. The PSR recorded is 61.33%. To illustrate

Table 3.2: Distribution of features in network B

Features	Nodes
Nodes supplying highly critical load	680, 692, 671, 675
Nodes implementing fictitious software ABC OS9 1.0.1	611, 634, 671, 652
Random nodes to be attacked	632, 645, 652, 692

the strengths of DIP, even with a network of low PSR, the node features in Table 3.2 and the weights found for this configuration are used for the second study.

Algorithm 1 A tuning algorithm for determining the relational weights

Let $W = \{\mu_\kappa, \mu_\nu, \mu_\psi\}$
LOOP Process
for each $\mu \in W$ **do**
  $\mu = 0.1$
  while $\mu \leq 0.9$ **do**
    Set $\mu_n = 0.5(1 - \mu)$ for all $\mu_n \in W \setminus \mu$
    Perform M attacks corresponding to μ
    Record the average PSR
    $\mu = \mu + 0.1$
  end while
  Plot a curve of average success rate against μ
end for
Determine the common PSR that preserves equation (3.3)
LOOP Process
for each $\{\mu_1, \mu_2\} \in W$ **do**
  Determine the intersection of their curves with the minimum success rate line
  Set $\mu_3 = 1 - \mu_2 - \mu_1$ for $\mu_3 \in W \setminus \{\mu_1, \mu_2\}$
  Perform $\mathcal{N}$ attacks with weights $\{\mu_1, \mu_2, \mu_3\}$ and record the average success rate
end for
Select $\{\mu_\kappa, \mu_\nu, \mu_\psi\}^*$ which gives the highest PSR

3.4.2 Study 2: Assessing the Performance of DIP

In this section, a base case in which there is DAS without inter-node communication, as is the case in DIP, is first implemented. Next, DIP is simulated under a sequential coordinated attack, and under a concurrent coordinated attack. Altogether, three scenarios are simulated.

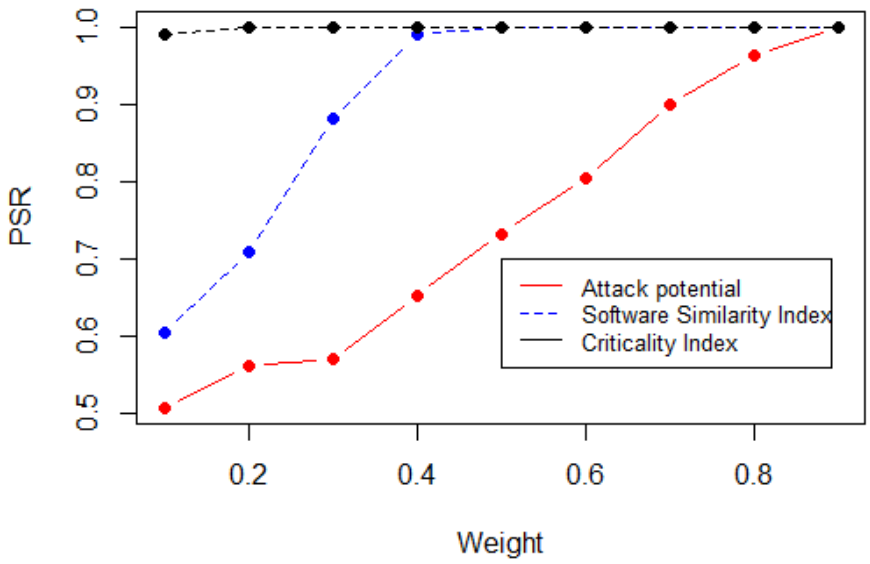

Figure 3.6: Plots of PSR against varying weights, as obtained from Algorithm 1

In all simulations, the following thresholds are used: $d_t = 200$, $f_t = 1$, $l_t = 1$, and $p_t = 5$. The replay and packet modification thresholds are chosen with the assumption that the operation center does not send badly crafted packets. The password threshold is chosen according to standard industry practice. The flooding threshold is chosen such that it is above the average cluster sent by the operations center. Also a flooding alert will be triggered much earlier before malicious packets are missed, according to FNR patterns.

Here in this study, the entire network, with all agents, is implemented on an Intel core i7 computer. Also, agents are set to select the agents with the maximum 3 values in addition to the alerting node. Since the simulation is performed on a single computer, the communication between agents is considered synchronous. The diameter, D, of the test feeder is 6. Thus, using a time unit of 0.5s according to observed processing speed of the computer, it is determined that the upper bound to convergence is 3s. In the event that an alert is detected, all agents not in protective mode reduce their thresholds to $n = 50\%$, rounded up to the

nearest whole number.

The adversary has already performed reconnaissance on the grid and has a good knowledge of which nodes to attack. Their motive is to disrupt power supply to the highly critical load.

Scenario 1: DAS without Inter-node Communication

In the base case, a NIDS with functions described in the first phase, is installed at each node. However, there is no inter-node communication. The NIDS, nevertheless, alerts the control center of suspicious activities. At 06:57:43,341, the adversary begins attacks at node 692 since this node serves a critical load. Five password login attempts are made. The NIDS at the node flags this and blocks the attacker. The attacker then attempts to log in to nodes 671, 675, 680, 645 and 646 in succession. Finally, they are able to log in to node 632 on the third count (recorded at 07:03:34,231), and execute a command to disconnect the load. Even though there are NIDSs at each of the nodes in the network, the attacker is still able to launch an attack due to the large attack surface. Thus, it is observed that the installation of non-interacting NIDSs does not provide maximum cybersecurity.

Scenario 2: DAS with Node Communication using DIP under Sequential Coordinated Cyberattack

In the second scenario, DIP is implemented. Table 3.3 shows the sequence of events and results in this scenario. It is observed, in the first event, that after the first attack is detected at 7:06:27,668, the attacked node (node 692) broadcasts an alert. All agents come to a consensus approximately 3s after the alert is received at the nodes. Nodes 692, 671, 680 and 646 enter protective mode. Node 692 is unresponsive to further attacks conducted in event 4. Having found three highly critical nodes in protective mode, the attacker further attempts to log in to node 675. However, after 3 unsuccessful attempts (due to a reduction

Table 3.3: Sequence of events in scenario 2

Event	Attack	Time Stamp	Result	Time Stamp
1	Replay attack launched on node 692	7:06:27,521	Attack detected	7:06:27,668
2	Agent at node 692 broadcasts alert	7:06:27,764	Agents calculate ρ and enter into conversation	-
3	Agent reach consensus	7:06:30,993	Agents 646, 671, 680, 692 in protective mode. All others have reduced thresholds	7:06:30,993
4	Attacker attempts to log in to node 692	7:07:24,693	No login console available, node in protective mode	-
5	Attacker attempts to log in to node 675	7:08:35,874	After three unsuccessful attempts, attacker is blocked	7:09:06,314
6	Alert is broadcast	7:09:06,400	All agents in protective mode	7:09:06,771

of thresholds at the end of event 3), an alert is broadcast. At 7:09:06,771, all agents enter protective mode, an average of 371ms after the second alert is broadcast. Compared to the first scenario, DIP significantly reduces the effective attack surface of the distribution system, and the performance is validated.

Scenario 3: DAS with Node Communication using DIP under Concurrent Coordinated Attack

In this scenario, the system implemented in scenario 2 is used. The adversary attacks two nodes at the same time. The attacker has already captured a packet meant to open a switch at node 675 and intends to replay this. Meanwhile, there is also an attempt to log in to node 671 using a password hack. At 15:26:43,176, node 671 detects the consistent password attempts and flags this. An alert is sent to all other nodes and to the control center. Agents start calculating their judgement values. However, at 15:26:43,264, less than 100ms later,

node 675 receives a replayed packet and flags this immediately. It broadcasts an alert to all agents and to the control center. Nodes begin to abort inter-node communication at 15:26:43,331. At 15:26:43,695 all nodes are in protective mode.

3.5 Comparison with other Work

In this section, the proposed DIP for DASs is compared with related work in the literature.

3.5.1 Comparison with Deep Packet Inspection

First, the NIDS implemented in DIP (henceforth referred to as NIDS-DIP) is compared with an NIDS similar to that provided in [27] (a REFerence method henceforth referred to as NIDS-REF). NIDS-REF, among others, performs deep packet inspection to ensure that packets adhere to accepted operational procedures. To achieve an objective comparison, NIDS-REF is developed according to the DNP3 SAv5 protocol, even though the original implementation in [27] is based on Modbus. The following attacks are executed: flooding, packet modification, replay, password hack, invalid command, and breach of operational procedure.

The detection results show that NIDS-DIP is able to detect and correctly alert on flooding, packet modification, replay, and password hack. This is expected as it has been configured to monitor for these attacks. However, for invalid command and breach of operational procedure attacks, NIDS-DIP detects the presence of attacks but alerts these under a different category. This is due to the use of integrity and authentication checks in DNP3 SAv5. An attacker's packet that contains invalid commands or breaches operational procedure also fails integrity and/or authentication check.

NIDS-REF accurately detects and alerts on flooding, invalid command, and breach of op-

erational procedure attacks. Password hacks are undetected since it is not configured to monitor for such attacks. NIDS-REF is not configured to monitor for packet modification and replay, hence, it does not alert on these. However, due to the use of DNP3 SAv5, a packet that fails authentication and/or integrity checks is dropped and an error message logged.

NIDS-DIP performs intrusion detection using network features of a received packet; it does not perform deep packet inspection. While this makes it easier to implement and lighter to install, it is unable to detect insider attacks as well as attacks from hackers with insider details.

3.5.2 Comparison with Centralized Correlation

Secondly, the decentralized correlation technique of DIP is compared to the centralized correlation method proposed in [22] (a REFerence method henceforth referred to as CENTRAL-REF). CENTRAL-REF originally correlates a location index, critical index and an abnormal behavior index in an iterative matrix multiplication technique. Once a steady state correlation vector has been obtained, the maximum value is selected as the correlation index. Moreover, the technique correlates already detected attacks in several substations to establish whether they are correlated or not. However, to achieve an objective comparison, the method has been adapted to predict the targets of an attack. This is done by applying the technique to correlate the three indices found for each node.

The tests are performed on the network with node features shown in Table 3.2. This is the same network used in studies in subsection 3.4.1. Table 3.4 summarizes the outcome.

The offline computation time required in DIP is to determine the values of the relational weights μ_κ, μ_ν and μ_ν. In CENTRAL-REF, no such precomputation is required. The online attack correlation time of DIP is also expectedly longer than that of CENTRAL-REF. This

Table 3.4: Comparing DIP with CENTRAL-REF

	Offline Computation Time (hrs)	Online Attack Correlation Time (s)	Prediction Success Rate (PSR) (%)
DIP	4	3.325	61.33
CENTRAL-REF	0	0.48	41.33

is primarily due to the time complexity of the consensus algorithm employed in DIP, which is absent in centralized systems such as CENTRAL-REF. The PSR of 61.33% for DIP is same as that which is obtained under subsection 3.4.1. Nonetheless, the PSR of DIP is found to be higher than that of CENTRAL-REF. This is due to the fact that CENTRAL-REF applies equal weights to the indices. Thus, for the same network features, CENTRAL-REF tends to have a lower PSR than DIP.

Chapter 4

Cyberattack Correlation and Mitigation for Distribution Systems via Machine Learning[1]

Cyber-physical system security for electric distribution systems is critical. In direct switching attacks, often coordinated, attackers seek to toggle remote-control switches in the distribution network. Due to the typically radial operation, certain configurations may lead to outages and/or voltage violations. Existing optimization methods that model the interactions between the attacker and the power system operator (defender) make unrealistic assumptions that reduce the usability of the techniques. Furthermore, the trend with coordinated cyberattack detection has been the use of centralized mechanisms, correlating data from dispersed security systems. This can be prone to single point failures. While the method from Chapter 3 is decentralized, the mitigation strategy may be over-protective. In certain cases where the human operator requires critical access to nodes, a dire situation may be exacerbated when nodes are in protective mode. In this study, novel mathematical models are presented for the attacker and the defender. The model does not assume knowledge of the attacker's parameters by the defender. Instead, a machine learning (ML) technique implemented by a multi-agent system establishes coordination in the detection of attacks in

[1] *(Under review)* J. Appiah-Kubi, and C. –C. Liu, "Cyberattack Correlation and Mitigation for Distribution Systems via Machine Learning", *IEEE Open Access Journal of Power and Energy*, June 2022.

a decentralized manner, predicting the targets of the attacker. Furthermore, agents learn optimal mitigation of the communication level through Q-learning. The learned attacker motive is also used by the defender to determine a new configuration of the distribution network.

4.1 System Model

4.1.1 Electric Distribution System Model

Electric distribution systems typically comprise feeders and their laterals. The load and feeder nodes are connected through switching and protection devices such as fuses, reclosers, and manual as well as remote-control switches. Distribution systems are typically operated in a radial configuration, so that each node is connected to only one feeder at a time.

The operations center monitors the distribution network through a Supervisory Control and Data Acquisition (SCADA) system. The setup comprises a SCADA master at the operations center, and (feeder) remote terminal units (RTUs) that interface the remote-control switches. As far as SCADA and cyber issues are concerned, only remote-control switches are of interest.

Power flow equations characterize the voltage, current and power flow, and operation should be conducted within nodal voltage limits. Let $\overline{V}$ and $\underline{V}$ be the upper and lower voltage limits respectively. In a later section, the power system model is presented and used in optimal reconfiguration.

4.1.2 Attack Model

It is assumed that, in a distribution network, the goal of the attacker is to create islands, disrupting power supply to the load. This implies that the attacker may reconfigure the network to serve this purpose.

While there are varying attack techniques, in this chapter, it is assumed that the attacker does not have authentic user credentials; they must decipher this. The attacker is, therefore, an external attacker who executes man-in-the-middle (MitM) attacks. These include replay, denial of service (DoS), packet modification/falsification, and password hacks. Consequently, only a limited information is available to the attacker. It is assumed that they are aware of only the topology of the network, and the maximum size and composition of loads. Replay, packet falsification, and password hacks are useful for toggling switches and may be combined with DoS to increase the likelihood of success.

The attractive switches for the attacker are those that connect significant portions of critical load. This is measured by the 'attack quality', or simply, the 'quality' of node i, q_i, which is given by (4.1).

$$q_i = \kappa_i \sum_{\phi \in \Phi_i} {P_i}^{\phi} \tag{4.1}$$

In (4.1), κ_i is the criticality of load at node i. The criticality index for highly critical load (such as hospital load) is set at 2.0, while that for critical load (such as industry load) is set at 1.5. The criticality index for non-critical load (e.g., residential load) is set at 1. The criticality of the ith node, κ_i, is the weighted sum of criticality of its load components. For instance, if the load is composed of 20% non-critical load, 30% critical load, and 50% highly critical load, then the criticality is 1.65. For a switch that connects a line (i.e., disconnect switch), the attack quality is the maximum load quality lost when the switch is disconnected.

Moreover, some nodes may require fewer resources to attack than others. This may be the case if, for instance, a vulnerability is already found in firmware tools installed on the RTU deployed at the node. The attacker seeks to find the set of switches to toggle and the set of switches whose status must remain in the current state to maximize the total disrupted power. Since it is assumed that the actions of the attacker are not concealed from the

operator, the attacker must maximize their reward subject to the operator maximizing the load served in the ensuing attack. Consequently, the attacker's behavior is modeled by a bi-level optimization problem as follows.

$$\max R\Bigg[\sum_{i\in\mathcal{D},\phi\in\Phi_i}\kappa_i({P_{ic}}^{\phi}-P_i^{\phi*})\Bigg]-\Bigg[C_F+\sum_{(i,j)\in\mathcal{E}_s}C_{e_t}^{ij}e_{ij}^t + \sum_{(i,j)\in\mathcal{E}_s}C_{e_k}^{ij}e_{ij}^k+\sum_{i\in\mathcal{D}_s}C_d^i(1-z_i)\Bigg] \tag{4.2}$$

$$\text{s.t.}\quad C_F+\sum_{(i,j)\in\mathcal{E}_s}C_{e_t}^{ij}e_{ij}^t+\sum_{(i,j)\in\mathcal{E}_s}C_{e_k}^{ij}e_{ij}^k+\sum_{i\in\mathcal{D}_s}C_d^i(1-z_i)\le B \tag{4.3}$$

$$e_{ij}^k\le 1-e_{ij}^t\quad\forall(i,j)\in\mathcal{E}_s \tag{4.4}$$

$$\sum_{i\in\mathcal{D},\phi\in\Phi_i}P_i^{\phi*}=\arg\max_{P,y,s}\sum_{i\in\mathcal{D},\phi\in\Phi_i}P_i^{\phi} \tag{4.5}$$

$$P_i^{\phi}=\sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}}P_{ij}^{\phi}\quad\forall i\in\mathcal{S},\phi\in\Phi_i \tag{4.6}$$

$$P_i^{\phi}=\sum_{j:(j,i)\in\mathcal{E},\phi\in\Phi_{ij}}P_{ji}^{\phi}-\sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}}P_{ij}^{\phi}\quad\forall i\in\mathcal{D},\phi\in\Phi_i \tag{4.7}$$

$$-s_{ij}F_{ij}\le P_{ij}^{\phi}\le s_{ij}F_{ij}\quad\forall(i,j)\in\mathcal{E}_s,\phi\in\Phi_{ij} \tag{4.8}$$

$$-F_{ij}\le P_{ij}^{\phi}\le F_{ij}\quad\forall(i,j)\in\mathcal{E}\backslash\mathcal{E}_s,\phi\in\Phi_{ij} \tag{4.9}$$

$$0\le P_i^{\phi}\le z_iP_{ic}^{\phi}\quad\forall i\in\mathcal{D}_s,\phi\in\Phi_i \tag{4.10}$$

$$0\le P_i^{\phi}\le P_{ic}^{\phi}\quad\forall i\in\mathcal{D}\backslash\mathcal{D}_s,\phi\in\Phi_i \tag{4.11}$$

$$y_{ij}\le 1-(e_{ij}^t+e_{ij}^k)\quad\forall(i,j)\in\mathcal{E}_s \tag{4.12}$$

$$s_{ij}=e_{ij}^t(1-s_{ij}^c)+e_{ij}^ks_{ij}^c+y_{ij}\quad\forall(i,j)\in\mathcal{E}_s \tag{4.13}$$

$$e_{ij}^{t}, e_{ij}^{k}, y_{ij} \in \{0,1\} \quad \forall (i,j) \in \mathcal{E}_s \tag{4.14}$$

$$z_i \in \{0,1\} \quad \forall i \in \mathcal{D}_s \tag{4.15}$$

Objective function (4.2) maximizes the net reward of the attacker. Constraint (4.3) requires that the total cost of attack (i.e., sum of fixed cost and variable costs) is within their budget. Constraint (4.4) ensures that a disconnect switch selected to be toggled is not also selected to be kept in its current state. Constraint (4.5) and the remaining constraints form the inner level optimization problem that characterize what the attacker believes will be the response of the operator to their attack. Constraint (4.5) maximizes the load served. In (4.6), power generated is the sum of line flows on the feeder, and in (4.7) power flow is balanced at each demand node. Line flow must be within acceptable limits in (4.8) and (4.9). However, (4.8) also constrains the line flow according to the state of the disconnect switch of that line. Similarly, the served load at each demand node must be within its limits in (4.10) and (4.11), with (4.10) also constraining according to the state of the load disconnect switch. Constraint (4.12) implies that the operator can toggle only line switches that are not attacked. Finally, the current state of a disconnect switch of a line is determined by (4.13), and (4.14)-(4.15) provide binary constraints. It is noted that radiality is not ensured here; the attacker has no need to ensure this constraint, and the operator may be unable to do so for a certain attack configuration. Since power flow is allowed to be negative, in this model, $(i,j) \in \mathcal{E}$ implies $(j,i) \notin \mathcal{E}$.

It is noteworthy that the bi-level mixed integer linear program (4.2) – (4.15) is computationally demanding especially when solved for a large distribution system. Thus, the method proposed in [40] is used to convert the problem into a single level mixed integer linear program: dual constraints of the inner level problem are added to the problem, and so is the equality constraint for the inner level dual and primal objectives.

4.2 Summary of Proposed Algorithm

The proposed algorithm consists of two parts: a decentralized system in the first part, and a centralized system in the second part.

The decentralized system is a multi-agent system (MAS), with an agent installed at each remote-control switch in the network. The agent is an autonomous software module, implemented on a computing device with RTU functionality. The agent is, therefore, the cyber interface of the remote-control switch, and is aware of attack quality, q, as well as its own operational properties. In this chapter, the operational properties of an agent include the type of SCADA communication network it connects to (e.g., cellular network A, etc.), and the firmware. Each agent of the MAS executes a three-stage intrusion prevention algorithm, which includes attack detection, attack target prediction, and communication level mitigation.

On the other hand, the centralized system is located at the operations center and performs the second level of mitigation. It is activated when a user-defined K alerts have been received at the operations centers. An optimal reconfiguration of the network is performed to minimize potential loss of load. Following this, a command is issued by the central agent to switches to turn off remote control ability.

The proposed algorithm is hierarchical, enforcing a hybrid mitigation, and is summarized in Fig. 4.1. The next sections detail the parts of the proposed algorithm.

4.3 Multi-Agent (Decentralized) Level

An agent is installed at each remote-control switch. Each agent implements the following three-stage algorithm.

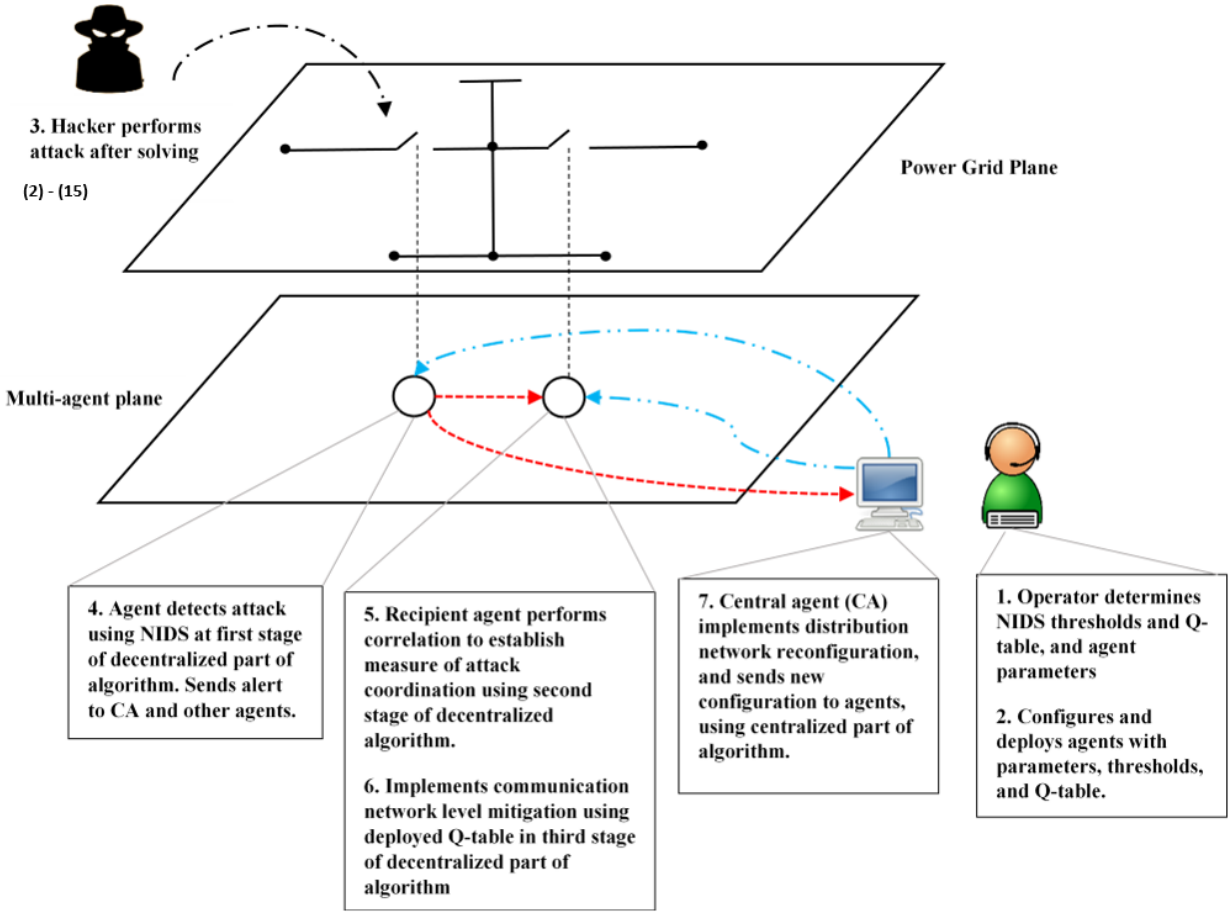

Figure 4.1: Summary of proposed algorithm

4.3.1 Detection Stage

In this first stage, the agent implements a network-based intrusion detection system (NIDS), which performs real-time checks in a sliding window of time, W. It monitors for a set of attacks $\{d, p, l, f\}$, where d, p, l, and f are floo**d**ing, **p**assword login attempts, rep**l**ay and packet **f**alsification attacks, respectively. An event is classified as an attack if and when the number of anomalies within W exceeds pre-determined threshold values. In this chapter, the Poisson distribution is used to determine such thresholds in an offline process; the arrival of communication packets (e.g., SCADA master requests, and operator login requests), X, at a node is an independent random process. Data is collected to establish records for the average frequency of such requests within W, and a Poisson distribution is plotted. Since the Poisson distribution is plotted using data from the utility, the minimum occurrence of x events such that $Pr(X = x) \approx 0$ may be considered anomalous with respect to the utility's operations. Thus, x is the threshold, and the Poisson method may be used to obtain thresholds for password login attempts, p_t, and flooding, f_t.

With the execution of packet authentication schemes, replay and packet falsification can be monitored. Since the utility sets up the necessary security keys for this, replayed and falsified packets are unexpected. Consequently, the threshold for replay and packet falsification, l_t and f_t respectively, may each be set to 1.

Should an intrusion be detected, the agent sends an alert, $\mathcal{L}$, to the operations center and to all other agents in the network. Denote the alerting agent by n_L, and an agent that receives $\mathcal{L}$ by n_R. Among others, $\mathcal{L}$ contains the operational properties of agent n_L. All other agents activate the second stage of the decentralized algorithm when $\mathcal{L}$ is received. Meanwhile, n_L shuts down the remote control capability by blocking all incoming connections to itself and dropping all packets addressed to itself for a user-defined time period $T \leq W$. Only outgoing

communication is allowed.

4.3.2 Correlation and Target Prediction Stage

Once an alert is received, n_R proceeds to determine if its switch will be a target of the (ongoing) attack. Ideally, this process requires an agent to solve the attacker's problem (4.2) – (4.15). However agents are unable to do so since the reward, budget, and cost of attack are known only by the attacker. Consequently, a machine learning process for predicting the targets of an attack is used. The result is the attack likelihood index, ρ, which measures the certainty of agent n_R that its switch will be a target. It is assumed that nodes that possess the same operational properties have the same cost. In the substitute machine learning process, an agent correlates its own properties to those in received alerts, establishing a measure of the cost to attack. Variability in the operational properties in received alerts signifies variability in the associated cost to attack. Subsequently, the metric of entropy is employed.

Define $\mathcal{N}_L$ and $\mathcal{F}_L$ as the subsets of unique communication network types and firmware types in the set of received alerts. As more alerts are received, $\mathcal{N}_L$ and $\mathcal{F}_L$ are updated. Let $\mathcal{N}_N$, and $\mathcal{F}_N$, be the sets of unique communication network types, and firmware types in the electric distribution network respectively. It is noted that $\mathcal{N}_L \subseteq \mathcal{N}_N$ and $\mathcal{F}_L \subseteq \mathcal{F}_N$.

Also, let $u^n_{n_R}$, and $u^f_{n_R}$ be the communication network type, and firmware of the recipient agent n_R, respectively. Define the following:

$$I_n \coloneqq I(u^n_{n_R} \in \mathcal{N}_L) \tag{4.16}$$

$$I_f \coloneqq I(u^f_{n_R} \in \mathcal{F}_L) \tag{4.17}$$

where $I(\cdot)$ is an indicator function. A similarity index for operational properties is calculated

as follows:

$$\psi_{n_R} = I_n\Big[I_f + H_f(1 - I_f)\Big] + H_n(1 - I_n)\Big[I_f + H_f(1 - I_f)\Big] \tag{4.18}$$

where H_n is Shannon's entropy [41] of $\mathcal{N}_L$ relative to that of $\mathcal{N}_N$, and H_f is Shannon's entropy of $\mathcal{F}_L$ relative to that of $\mathcal{F}_N$. They are given in (4.19) and (4.20), respectively. Here, a and b are arbitrary elements of the given sets.

$$H_n = \frac{-\sum_{a\in\mathcal{N}_L} Pr(a)\log_2 Pr(a)}{-\sum_{b\in\mathcal{N}_N} Pr(b)\log_2 Pr(b)} \tag{4.19}$$

$$H_f = \frac{-\sum_{a\in\mathcal{F}_L} Pr(a)\log_2 Pr(a)}{-\sum_{b\in\mathcal{F}_N} Pr(b)\log_2 Pr(b)} \tag{4.20}$$

Since $\mathcal{N}_L$, $\mathcal{F}_L$, $\mathcal{N}_N$, and $\mathcal{F}_N$ contain only unique elements, it follows that for $a \in N_k$, $Pr(a) = \frac{1}{|\mathcal{N}_k|}$, for $k \in \{L, N\}$. Thus, (4.19) and (4.20) are simplified to (4.21) and (4.22) respectively.

$$H_n = \frac{\log_2 |\mathcal{N}_L|}{\log_2 |\mathcal{N}_N|} \tag{4.21}$$

$$H_n = \frac{\log_2 |\mathcal{F}_L|}{\log_2 |\mathcal{F}_N|} \tag{4.22}$$

From (4.21) and (4.22), it is clear that:

$$\lim_{|\mathcal{N}_L|\to|\mathcal{N}_N|} H_n = 1 \tag{4.23}$$

$$\lim_{|\mathcal{F}_L|\to|\mathcal{F}_N|} H_f = 1 \tag{4.24}$$

As $\log_2 \mathcal{N}_N$ and $\log_2 \mathcal{F}_N$ are constants, and the log function is monotonically increasing, (4.19) - (4.24) imply that H_n and H_f increase monotonically to 1. H_n and H_f represent the degree of randomness in the behavior of the attacker concerning their choice of communication network and firmware types. The following observations can be made from (4.18) - (4.24):

1. For $|\mathcal{N}_L| = 1$, $H_n = 0$. This indicates that the attacker has leverage from only one network type. A similar expression and interpretation can be derived for firmware type. When $H_n = H_f = 0$, only agents with the exact same properties will have $\psi = 1$, whereas all others will have $\psi = 0$.

2. $|\mathcal{N}_L| = |\mathcal{N}_N|$ implies that $H_n = 1$ and $I_n = 1$. Again, a similar expression can be written for firmware type. In this scenario, it is clear that the attacker has no preference for any operational property, indicating that the attack quality influences their choices.

3. $H_n = 0$ and $I_n = 0$ imply that $\psi = 0$, irrespective of the values of H_f and I_f. This is reasonable since an attacker must first gain access to a communication network before they are able to leverage any firmware vulnerabilities present.

The similarity index for operational properties found using (4.18) is used in Algorithm 2 to determine an attack likelihood index ρ_{n_R}, which measures the certainty of agent n_R of the likelihood of attack of its switch.

In Algorithm 2, the normalized quality, q_i^n, is used to ensure that the effect of ψ_{n_R} is not masked. Both q_i^n and ψ_{n_R} form a vector $\mathbf{v_{n_R}}$. It is noteworthy that $\mathbf{v_{n_R}} = \mathbf{1_2}$ implies that agent n_R has the highest quality and the exact same operational properties as is present in received alert(s). This is the worst case scenario for the agent. Therefore, in step 6 of Algorithm 1, $\mathbf{v_{n_R}}$ is compared to the worst case scenario vector using the Jaccard similarity index [42], which performs an element-wise comparison of the two vectors.

4.3.3 Communication-network-level Mitigation

In the final stage of the decentralized level of the algorithm, n_R finds new security thresholds so that the relative level of protection offered by using the new thresholds matches the level

Algorithm 2 Predicting the likelihood of attack

if Load switch **then**
  if $z_i = 1$ **then**
    Continue
  else
    $\rho = 0$
    Break
  end if
end if
Determine normalized quality $q_i^n = \frac{q_i}{q_{max}}$
Determine correlation index ψ_{n_R} for operational properties by (4.18).
$\mathbf{v_{n_R}} = [q_i^n \quad \psi_{n_R}]$
Determine worst case vector: $\mathbf{v_w} = \mathbf{1_2}$
$\rho = \frac{\sum_j \min\{v_{n_R j}, v_{wj}\}}{\sum_j \max\{v_{n_R j}, v_{wj}\}}$

of threat perceived, as measured by ρ. The new thresholds must neither be too restrictive, increasing false alarms, nor too lax, overly exposing the agent to malicious activities. A combinatorial problem must be solved to select such new thresholds. However, this can be computationally intensive for the agent, which is a simple computing device. Thus, Q-Learning, a reinforcement learning (RL) algorithm, is used to determine the optimal new thresholds.

Offline Derivation of Q-table

In this application, the RL environment is the NIDS executed by the agent. It is desired that the added level of protection offered by new thresholds matches the level of threat perceived, as measured by ρ. Therefore, the RL state, $s \in S$, is defined as the differnce between ρ and the level of security offered by new thresholds. The state is given as in (4.25).

$$s = \rho - \left(1 - \frac{1}{4}\left(\frac{d_n}{d_t} + \frac{p_n}{p_t} + \frac{f_n}{f_t} + \frac{l_n}{l_t}\right)\right) \tag{4.25}$$

In (4.25), c_n represents new threshold values, while c_t represents the default threshold values,

where $c \in \{d, p, l, f\}$. The thresholds, both default and new, for packet falsification and replay are each set to 1 for aforementioned reasons.

Ideally, it is desired that new thresholds are selected to render $s = 0$. Note that if the current threshold values are the same as the default, no extra protection is offered and $s = \rho$. The state, s, is continuous and finite, ranging from -1 to 1. For simplicity, s is discretized into intervals of 0.05. The action set A contains eight actions: an increment action for each of d and p, a decrement action for each of d and p, a no-change action, set-to-default-thresholds action, set-to-median-thresholds action (i.e., $\{d_n, p_n, l_n, f_n\} \approx \{0.5d_t, 0.5p_t, l_t, f_t\}$), and shut-down-remote-control action (i.e., $\{d_n, p_n, l_n, f_n\} = \{0, 0, 0, 0\}$).

As mentioned, it is desired that new thresholds are selected to render $s = 0$. However, the discrete and finite nature of the thresholds implies that this not always achievable. Hence, some tolerance, τ, is allowed. A decaying exploration rate, ϵ, is used to balance exploration and exploitation. The reward in the next time step r_{t+1}, is as given in (4.26).

$$r_{t+1} = \begin{cases} 0 & |s_{t+1}| \leq \tau \\ -|\tau - |s_{t+1}|| & \text{otherwise} \end{cases} \tag{4.26}$$

Equation (4.26) gives a reward of 0 when the next state s_{t+1} after performing an action a_t is within a certain τ deviation from 0, and a penalty if it does not.

Q-Learning is employed to find the optimal action value policy. As learning progresses, each action value $Q(s_t, a_t)$ in the Q-table is updated according to (4.27).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{4.27}$$

The learning rate, α, is a hyper-parameter in the unit interval which determines how fast

learning occurs. An α too small results in slow convergence. On the other hand, when selected too large, it leads to fast convergence to a potentially not optimal policy, or no convergence at all. To achieve convergence on a stationary environment as is the case in this application, α must be such that $\sum_{t=1}^{\infty} \alpha(t) = \infty$ and $\sum_{t=1}^{\infty} \alpha^2(t) < \infty$ [43]. Hence, a decaying α guarantees convergence. The discount factor, γ, also between 0 and 1, discounts future returns and ensures that the expected value is finite. The Q-table is updated according to (4.27) until the iteration converges.

Online Application of Q-table

The Q-table obtained from the offline simulation is deployed with MAS agents, and a small ϵ is set to encourage exploration. When ρ is found in the second stage, an agent proceeds to the third stage of the decentralized algorithm and obtains its state using (4.25). The optimal action for that state is selected from the Q-table with a probability of $1 - \epsilon$. Upon applying the selected action, a new state is obtained. This process is repeated until the optimal action is to make no change to the thresholds. The final thresholds are enforced until T has elapsed, after which they are reset to their default values. The sliding time window W is also reset to start from when new thresholds are enforced. It is emphasized that the new thresholds are selected and enforced by n_R, which has not yet been attacked but is anticipating an attack with certainty ρ. This is, therefore, proactive communication level mitigation.

4.4 Physical-Level Mitigation and Contingency Analysis

The decentralized algorithm implemented by the dispersed agents allows for attack target prediction and communication level mitigation. Suppose K alerts have been received at the operations center. It may be necessary to further perform a reconfiguration of the

distribution system in order to minimize the impact of attacks. Contingency analysis is also useful to understand how many nodes and lines could be lost to the attacker without impacting distribution service. The central agent (CA) solves the following optimization problem:

$$\max \sum_{i\in\mathcal{D},\phi\in\Phi_i} P_i^\phi + \delta\Bigg(\sum_{(i,j)\in\mathcal{E}_s\setminus\mathcal{E}_sp} \rho_{ij}\Big[e_{ij}^t s_{ij}^c + e_{ij}^k(1-s_{ij}^c)\Big] + \sum_{i\in\mathcal{D}_s\setminus\mathcal{D}_sp}\rho_i(1-z_i)\Bigg) \tag{4.28}$$

$$\text{s.t.}\quad P_i^\phi = \sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}} P_{ij}^\phi \quad \forall i\in\mathcal{S},\phi\in\Phi_i \tag{4.29}$$

$$Q_i^\phi = \sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}} Q_{ij}^\phi \quad \forall i\in\mathcal{S},\phi\in\Phi_i \tag{4.30}$$

$$P_i^\phi = \sum_{j:(j,i)\in\mathcal{E},\phi\in\Phi_{ij}} P_{ji}^\phi - \sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}} P_{ij}^\phi \quad \forall i\in\mathcal{D},\phi\in\Phi_i \tag{4.31}$$

$$Q_i^\phi = \sum_{j:(j,i)\in\mathcal{E},\phi\in\Phi_{ij}} Q_{ji}^\phi - \sum_{j:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}} Q_{ij}^\phi \quad \forall i\in\mathcal{D},\phi\in\Phi_i \tag{4.32}$$

$$Q_i^\phi = P_i^\phi \tan\theta_i^\phi \quad \forall i\in\mathcal{D},\phi\in\Phi_i \tag{4.33}$$

$$\begin{aligned} U_j^{\phi_j} = \sum_{i:(i,j)\in\mathcal{E},\phi\in\Phi_{ij}} w_{ij}\Bigg(U_i^\phi - 2\sum_{\phi\in\Phi_{ij}}\Big[r_{\phi_j\phi}^{ij}(\Gamma_{\phi_j\phi}^{re}P_{ij}^\phi + \Gamma_{\phi_j\phi}^{im}Q_{ij}^\phi) \\ + x_{\phi_j\phi}^{ij}(\Gamma_{\phi_j\phi}^{re}Q_{ij}^\phi - \Gamma_{\phi_j\phi}^{im}P_{ij}^\phi)\Big]\Bigg) \quad \forall j\in\mathcal{N},\phi_j\in\Phi_j \end{aligned} \tag{4.34}$$

$$\underline{V}^2 \le U_i^\phi \le \overline{V}^2 \quad \forall i\in\mathcal{N},\phi\in\Phi_i \tag{4.35}$$

$$0 \le P_{ij}^\phi \le w_{ij}F_{ij}^p \quad \forall (i,j)\in\mathcal{E},\phi\in\Phi_{ij} \tag{4.36}$$

$$0 \le Q_{ij}^\phi \le w_{ij}F_{ij}^q \quad \forall (i,j)\in\mathcal{E},\phi\in\Phi_{ij} \tag{4.37}$$

$$0 \le P_i^\phi \le z_i P_{ic}^\phi \quad \forall i\in\mathcal{D}_s,\phi\in\Phi_i \tag{4.38}$$

$$0 \le P_i^\phi \le P_{ic}^\phi \quad \forall i\in\mathcal{D}\setminus\mathcal{D}_s,\phi\in\Phi_i \tag{4.39}$$

$$0 \leq Q_i^{\phi} \leq z_i Q_{ic}^{\phi} \quad \forall i \in \mathcal{D}_s, \phi \in \Phi_i \tag{4.40}$$

$$0 \leq Q_i^{\phi} \leq Q_{ic}^{\phi} \quad \forall i \in \mathcal{D} \backslash \mathcal{D}_s, \phi \in \Phi_i \tag{4.41}$$

$$e_{ij}^k \leq 1 - e_{ij}^t \quad \forall (i,j) \in \mathcal{E}_s \tag{4.42}$$

$$s_{ij} = e_{ij}^t (1 - s_{ij}^c) + e_{ij}^k s_{ij}^c \quad \forall (i,j) \in \mathcal{E}_s \tag{4.43}$$

$$s_{ij} = s_{ij}^c \quad \forall (i,j) \in \mathcal{E}_{sp} \tag{4.44}$$

$$z_i = 1 \quad \forall i \in \mathcal{D}_{sp} \tag{4.45}$$

$$w_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{E} \tag{4.46}$$

$$w_{ij} = 0 \quad \forall j \in \mathcal{S}, (i,j) \in \mathcal{E} \tag{4.47}$$

$$w_{ij} + w_{ji} = 1 \quad \forall (i,j) \in \mathcal{E} \backslash \mathcal{E}_s \tag{4.48}$$

$$w_{ij} + w_{ji} = s_{ij} \quad \forall (i,j) \in \mathcal{E}_s \tag{4.49}$$

$$\sum_{i:(i,j)\in\mathcal{E}} w_{ij} = 1 \quad \forall j \in \mathcal{D} \tag{4.50}$$

$$e_{ij}^t, e_{ij}^k \in \{0,1\} \quad \forall (i,j) \in \mathcal{E}_s \tag{4.51}$$

$$z_i \in \{0,1\} \quad \forall i \in \mathcal{D}_s \tag{4.52}$$

In (4.28), the CA, operating on behalf of the operator, seeks to find a new configuration that maximizes the total active power that can be served. However, the second term of (4.28) ensures that switches with high attack likelihoods are placed in the state desirable by the attacker, i.e., to keep open switches open and closed switches open. Hence, together, the two terms of (4.28) maximize the load served and the number of switches lost to the attacker. In (4.28), δ is a scaling parameter. It has a unit of kW, and is chosen such that it

is smaller than the magnitude of the smallest kW demand in the network. The expressions and parameters in (4.28)-(4.52) have similar interpretations as in (4.2) – (4.15). Constraint (4.33) relates active and reactive power using the load angle, θ. Constraint (4.34) is the squared voltage magnitude expression adapted from the linearized power flow model [44], which is constrained by (4.35) within the square of voltage magnitude limits. In (4.34), $\Gamma^{re}_{\phi_j\phi}$ is the element in row ϕ_j column ϕ of $\Gamma^{re} = Real\{\Gamma\}$. The interpretation is similar for $\Gamma^{im} = Im\{\Gamma\}$, where Γ is given by (4.53).

$$\Gamma = \begin{bmatrix} 1 & e^{-j4\pi/3} & e^{-j2\pi/3} \\ e^{-j2\pi/3} & 1 & e^{-j4\pi/3} \\ e^{-j4\pi/3} & e^{-j2\pi/3} & 1 \end{bmatrix} \tag{4.53}$$

In (4.44) and (4.45), switches whose agents have shut down remote control capability remain in their current state. The variable w_{ij}, adapted from [45] is introduced to determine the direction of power flow and ensure radiality in (4.46) – (4.50). Hence, in this model, $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$. Clearly, (4.28) – (4.53) is a non-linear integer programming problem (due to (4.34)). To ensure faster solution, (4.34) is linearized using big M notation.

4.5 Setup and Preliminary Data

The IEEE 123-Node Test Feeder, shown in Fig. 4.2, is used as a test system. The network comprises four substation feeders, switches, and load. For simulation purposes, the load at nodes 2, 24, 70, 88, and 109 are assumed connected through remote-control switches. Also, all three-phase disconnect switches are assumed to be remote-control. Each load in the network has been assumed to be composed of some combination of critical, highly critical, and non-critical load. Hence, each load is assigned a criticality level. In addition, fictitious operational properties, shown in Table 4.1, are assigned to each remote-control switch.

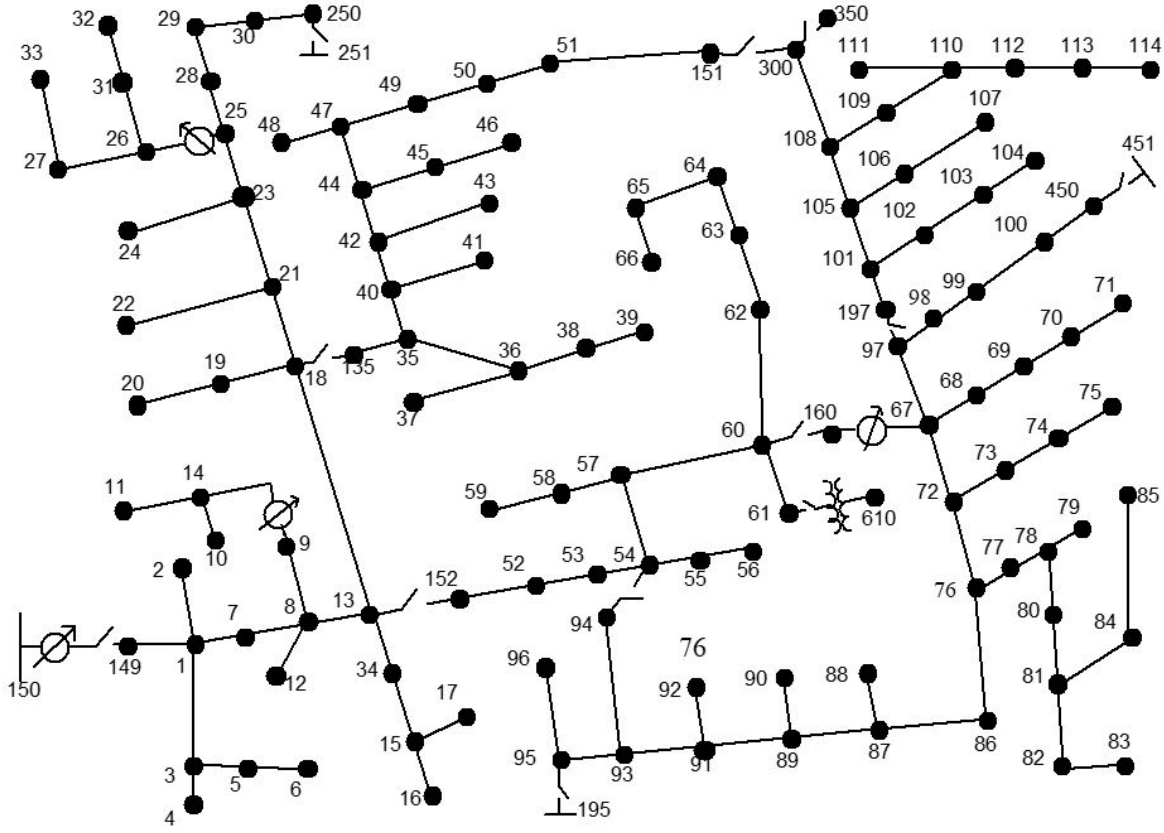

Figure 4.2: IEEE 123-Node Test Feeder

Table 4.1: Operational properties assigned to agents

Agent Location	Network Type	Firmware Type
Node 2	Network D	OS1
Node 24	Network D	OS4
Node 70	Network A	OS3
Node 88	Network B	OS2
Node 109	Network C	OS4
Line (451-450)	Network C	OS2
Line (300-350)	Network B	OS3
Line (251-250)	Network A	OS1
Line (151-300)	Network D	OS2
Line (150-149)	Network C	OS4
Line (97-197)	Network A	OS3
Line (61-610)	Network A	OS1
Line (60-160)	Network B	OS4
Line (195-95)	Network B	OS1
Line(18-135)	Network C	OS2
Line (13-152)	Network D	OS3
Line (54-94)	Network C	OS1

Agents are implemented in Volttron [32], a Pacific Northwest National Lab (PNNL) agent development platform. A packet manipulation tool, Scapy, is used to create an attack platform. The power system model is implemented in OpenDSS, and the attacker's and central agent's optimization problems are modeled using GAMS and solved using CPLEX solver. All simulations are performed on a computer with Intel Core i7 processors and 16GB of RAM.

4.6 Simulations and Results

Simulations are grouped into three main sections. In the first section, a direct switching attack is performed on a distribution system not implementing the proposed method. In the second section, offline planning stage simulations are performed. Here, the Q-table used at the communication network level mitigation stage is obtained, and so are threshold values used for attack detection. The Q-table and threshold values obtained are deployed with

agents in the third section and the working of the proposed algorithm demonstrated on the IEEE123-Node Test Feeder. For this third section, $K = 3$, and $T = W = 1$ hour. The attack reward, R, is 1 MU per kW power disrupted.

4.6.1 Responding to Direct Switching Attack According to Interdiction Studies in the Literature

In this simulation (E1) no detection and mitigation strategy is enforced. The aim here, as suggested by interdiction studies in the literature, is to restore as much load as is possible after an attack. The attacker solves (4.2) – (4.15) in its single level form, knowing only the current state of switches, the network topology, and the current active demand being served. By using an aerial map of the location served by the target distribution system, the attacker is also able to form judgement of the criticality of the load. As indicated by (4.2) – (4.15), the attacker has no need for information on voltages, currents, impedances and reactive demand. All load switches are closed, and the states of all other disconnect switches are as shown in Fig. 4.3. The attacker's budget is 50 monetary units (MU), and the fixed cost is 30 MU. Attack variable costs are shown in Table 4.2. For this configuration, the attacker finds that toggling the switch connecting line 18-135 and keeping the switch connecting line 151-300 inaccessible, i.e., performing DoS, will yield a maximum net return of 1010.01 MU. The targets are shown with red crosses in Fig. 4.3. This is a total of 755 kW of disrupted load. The shaded area in Fig. 4.3 shows the outage region. Since the attacked switches are inaccessible to the operator, the operator is unable to restore power to the outage region.

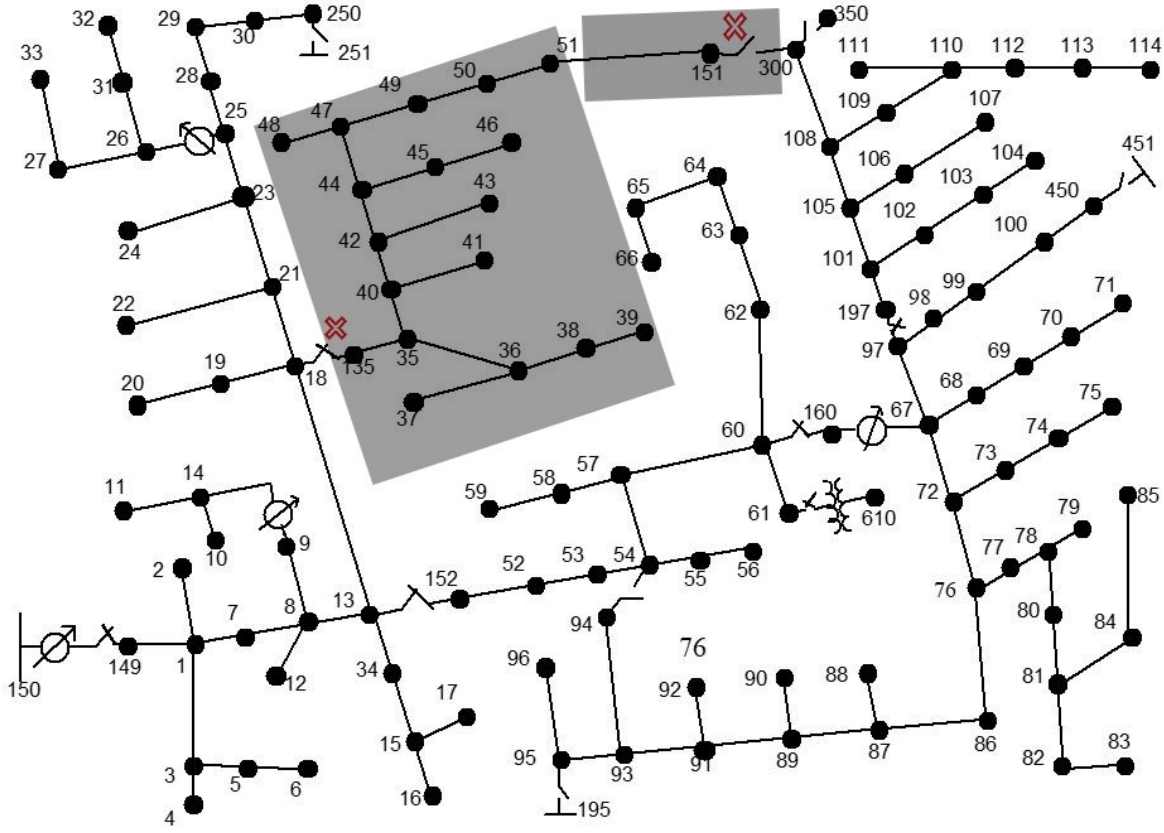

Figure 4.3: Power outage caused by cyberattack

Table 4.2: Variable costs to attack switches (in MU) for different experiments

	E1		E2		E3		E4	
Line Switch	C_{e_t}	C_{e_k}	C_{e_t}	C_{e_k}	C_{e_t}	C_{e_k}	C_{e_t}	C_{e_k}
Line (451-450)	15.60	5.60	14.50	5.60	67.00	25.00	70.00	25.00
Line (300-350)	12.00	7.00	30.00	20.00	67.00	25.00	70.00	25.00
Line (251-250)	13.00	13.00	67.00	20.00	45.00	25.00	70.00	25.00
Line (151-300)	41.90	4.90	30.00	15.00	67.00	25.00	70.00	25.00
Line (150-149)	18.90	8.90	14.50	5.60	67.00	25.00	70.00	25.00
Line (97-197)	74.50	6.50	67.00	20.00	67.00	25.00	70.00	25.00
Line (61-610)	70.10	2.10	67.00	20.00	45.00	25.00	70.00	25.00
Line (60-160)	69.34	9.34	30.00	15.00	67.00	25.00	70.00	25.00
Line (195-95)	16.30	6.30	30.00	15.00	45.00	25.00	70.00	25.00
Line(18-135)	12.00	2.00	14.50	5.60	67.00	25.00	70.00	25.00
Line (13-152)	77.90	7.90	30.00	15.00	67.00	25.00	70.00	25.00
Line (54-94)	30.00	3.00	14.50	5.60	45.00	25.00	70.00	25.00
Load Switch	C_d		C_d		C_d		C_d	
2	30.45		30.00		45.00		70.00	
24	28.90		30.00		67.00		70.00	
70	35.78		67.00		67.00		70.00	
88	79.00		30.00		67.00		70.00	
109	58.21		14.50		67.00		70.00	

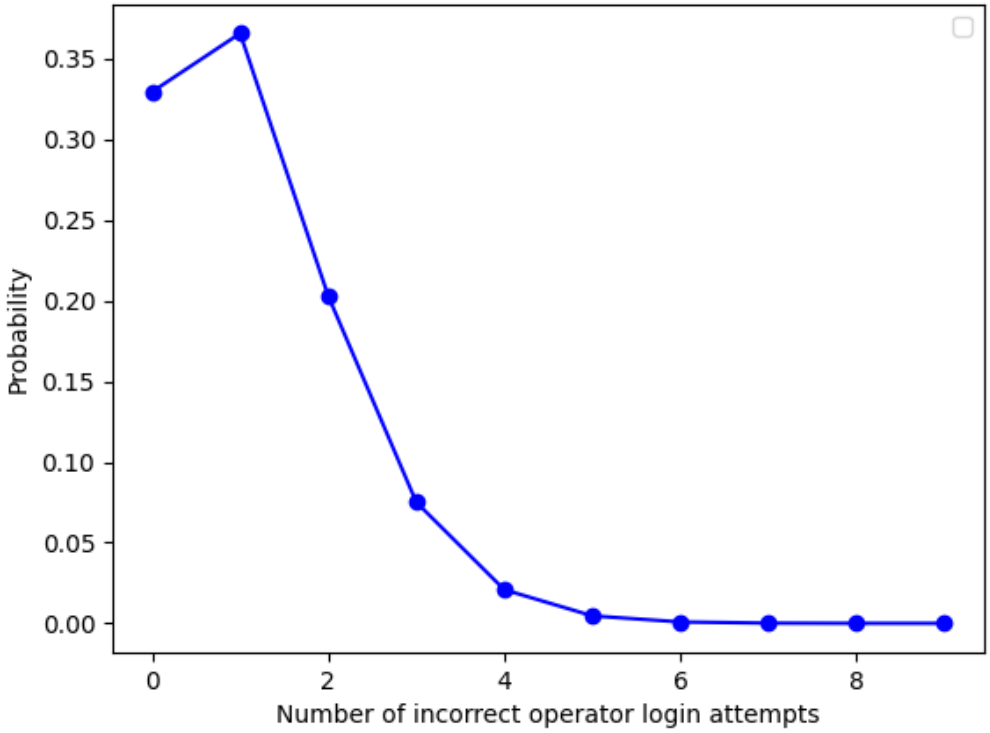

Figure 4.4: A Poisson distribution of incorrect operator login events

4.6.2 Offline Planning Stage Simulations

Obtaining NIDS Thresholds

Assume that from data collected at the distribution operations center, the average incorrect password credentials entered by authorized operators is 1.27 within the period. Using this average, the Poisson distribution plotted in Fig. 4.4 is obtained. The threshold is $p_t = 5$. This is the smallest value above which the probability of operator password error occurrence is 0. The average number of packets sent to a remote-control switch is 160. Since this mean is large, the Poisson distribution is approximated by the normal distribution. From this, the obtained threshold is $d_t = 200$.

Obtaining Q-Table

There are forty discretized states. Increments and decrements to the threshold values are made in steps of 1 for each of d_t and p_t. Certain combinations of thresholds have no logical interpretation (e.g., $[0, 0, 1, 1]$). Therefore, a set of minimum values, $[d_n, p_n, l_n, f_n] = [2, 0, 1, 1]$ is set. For this combination, $p_n = 0$ is interpreted by the agent as not displaying the login console in its RTU interface. Next to this minimum set of thresholds is $[0, 0, 0, 0]$.

As a hyper-parameter, α is found from prior empirical tests, during which an initial value of 0.1 which decays slowly to 1×10^{-5} gives good results. Also, the discount factor, γ, is set to 0.99, while ϵ is set to 1 at the beginning of each episode and decayed to a minimum of 0.05. In addition, τ is taken as 0.05. Thousand trials are performed in each of two hundred thousand episodes. Fig. 4.5 shows convergence of the Q-Learning algorithm.

At the end of this simulation, a 6 KB Q-table is obtained which is deployed with agents configured with NIDS thresholds. The table is referenced during the communication level mitigation stage.

4.6.3 Implementing the Proposed Algorithm

Here, agents have been installed at all remote-control switches and the proposed algorithm is executed.

Leverage from Communication Network Vulnerability

Here (E2), all load switches are on and the state of line switches is as shown in Fig. 4.6. The attacker's budget is 100 MU, the fixed cost is 50 MU, and all variable attack costs are shown in Table 4.2. By solving (4.2) – (4.15), the attacker determines to toggle the switch connecting line 18-135 and load switch 109, and to create DoS at the switch connecting

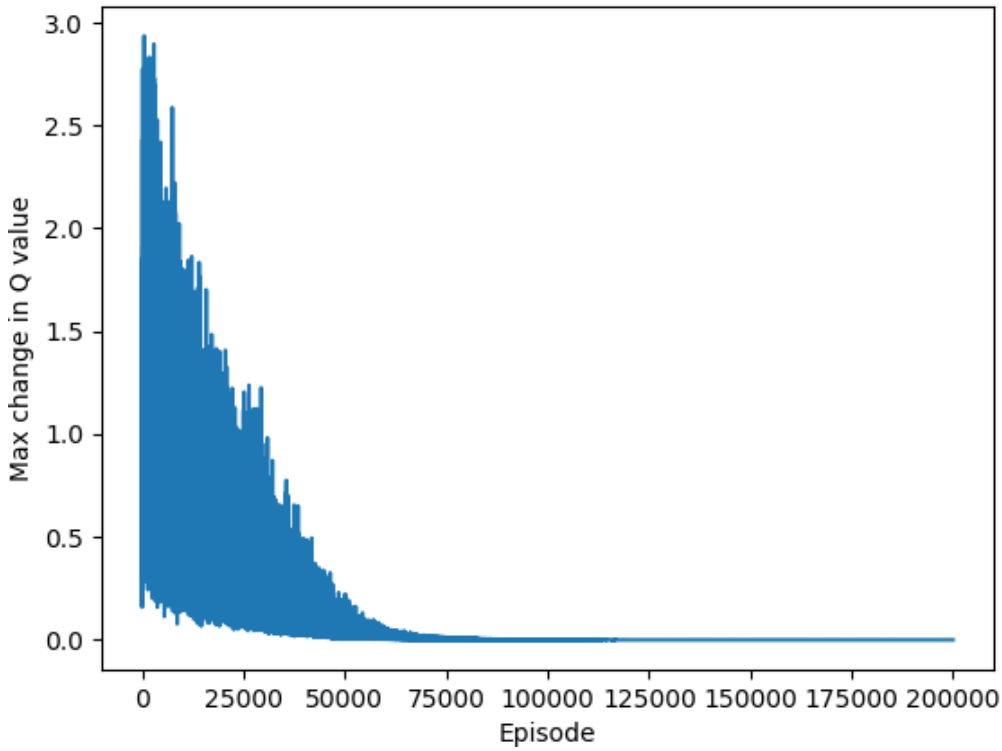

Figure 4.5: A plot showing the maximum absolute change in Q-value per episode

line 151-300. The anticipated disrupted load, shown in Fig. 4.6 as the green shaded area, is 795 kW which gives a reward of 1006.72 MU. They therefore begin a sequential attack, executing replay attack at switch 18-135, password hacks at load switch 109, and flooding switch 151-300. Since the agent at each switch is implementing the decentralized level of the proposed algorithm, each attack is detected by the NIDS at the first stage. Since the attack is sequential, the agent at switch 18-135 detects the attack first, and sends an alert to both the CA and all other agents in the distribution network. All other agents calculate their attack likelihood indices and retune the NIDS parameters to reflect this. As more alerts are received from the agents at switch 109 and switch 151-300, the index is updated accordingly, and agents retune their NIDS thresholds with each update. For instance, by the third alert, the agent at switch 197-97 sets the following NIDS thresholds: $[100, 2, 1, 1]$. Fig. 4.7 shows the change in attack likelihood indices, calculated by some agents as the alerts are received.

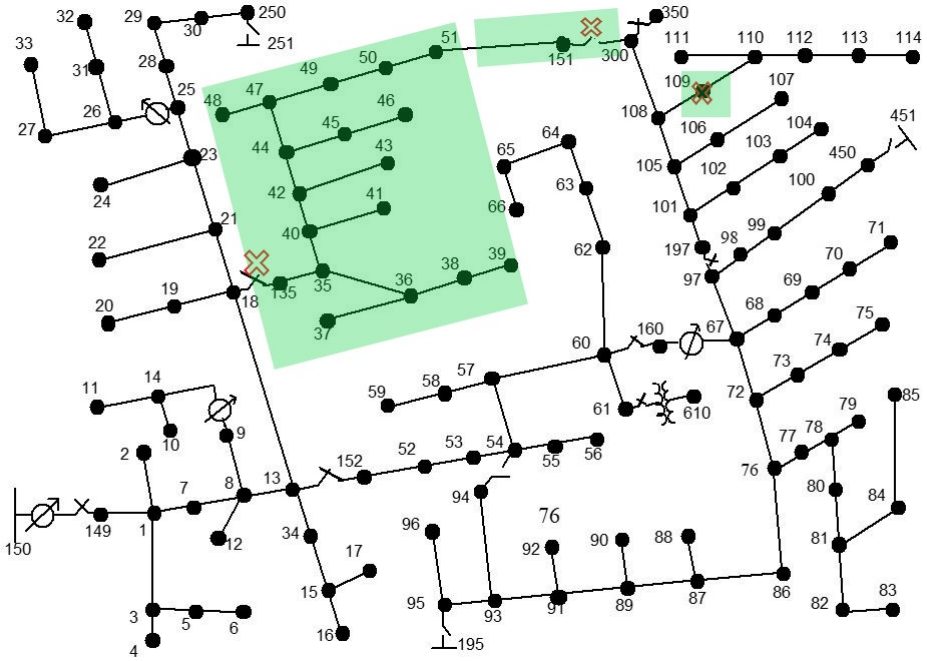

Figure 4.6: Anticipated outage area of attack in E2

When $K = 3$ alerts are received by the operator, the central agent at the operations center is triggered, which queries the dispersed agents for their attack likelihood indices. The central agent then solves (4.28) – (4.52) to determine a new configuration for the network. The new configuration, which serves all load, is shown in Fig. 4.8. The red circles represent switches whose agents have shut down remote control capabilities, whereas the green circles represent switches with attack likelihoods greater than 0.4. Having enforced the new configuration, the central agent sends a command to turn off remote control capability for all switches. No load is lost.

Leverage from Firmware Vulnerability

In this experiment (E3), OS1 has a firmware vulnerability. The attacker's budget is 150 MU, and there is a fixed cost of 50 MU. All variable costs are shown in Table 4.2. All load switches are on, and so are switches connecting lines 18-135, 60-160, 97-197, 251-250, 451-450 and 300-350. All other switches are off. While a firmware vulnerability is known, the

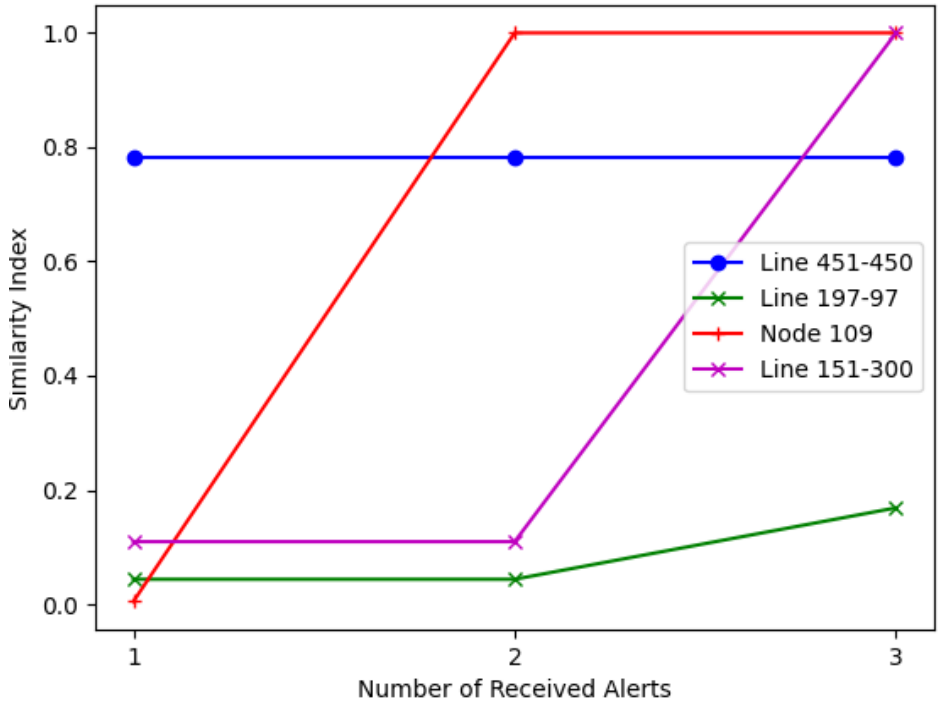

Figure 4.7: A plot showing change in attack likelihood, ρ, with receipt of alerts for some agents

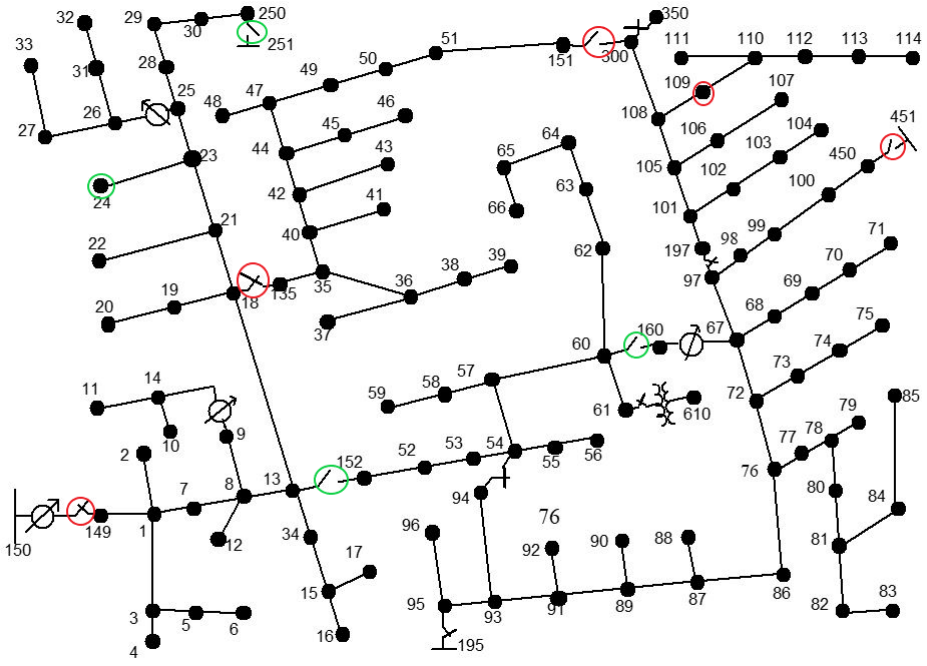

Figure 4.8: New configuration from central agent (CA) in E2

attacker must first gain access to the network before leveraging any firmware vulnerability. From solving (4.2) – (4.15), the attacker determines to toggle switch 18-135 by sending a falsified packet, and flood switch 151-300. This coordinated attack is expected to yield a net return of 912.01 MU, being a disruption of 555 kW of load. The attack is conducted as a simultaneous cyberattack. Again, the agents deployed at the attacked switches detect the attack and send an alert to the CA and to all other agents in the distribution network. The alerting agents shut down remote control capabilities while the others retune their NIDS thresholds. However, because $K = 3$ alerts are not received, the CA is not triggered and a new reconfiguration is not found. No load is lost.

Demonstrating the Impact of Criticality

In the fourth experiment (E4), there is no known operational vulnerability. Thus, the attack cost is the same for all switches, being 70 MU for replay, packet falsification and password hacks, and 25 MU for flooding. The initial state of switches in this experiment is the same as for E3, shown in Fig. 4.9. The attacker's budget is 200 MU and there is a fixed cost of 50 MU. Having solved (4.2) – (4.15), the attacker finds that they must toggle switch 451-450 which is one of the feeder switches, and keep switches 13-152, 195-95 and 151-300 inaccessible to the operator. From this, they anticipate a net return of 2748.59 MU, which is obtained from disrupting 1775 kW of load. This is shown in Fig. 4.9. A simultaneous coordinated cyberattack begins, and the agents deployed at the attacked switches detect the attacks. Although simultaneous, alerts are received in the following order: 451-450, 13-152, and 195-95. Before the central agent queries the dispersed agents for their attack likelihood indices, a fourth alert is received from the agent at switch 151-300. The dispersed agents update their attack likelihood index and retune their NIDS thresholds with each received alert. The central agent performs a new configuration, as shown Fig. 4.10, after which it sends a command to the dispersed agents to turn off remote control capability. No load is

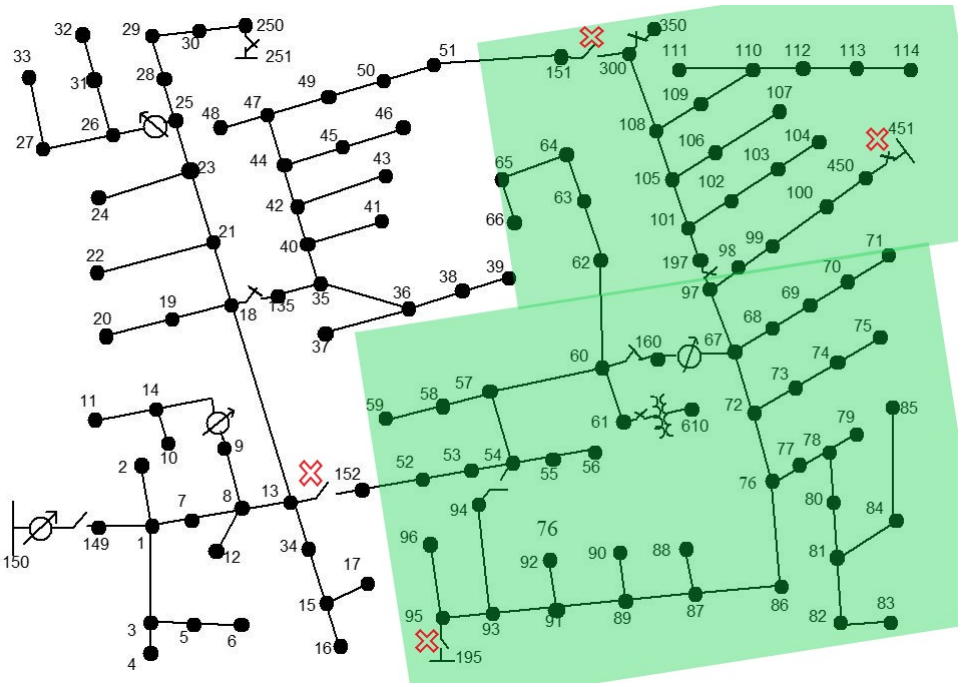

Figure 4.9: Anticipated outage area of attack in E4

lost.

4.7 Discussion

4.7.1 Discussion on Experiment E2

In Fig. 4.8, it is observed that four switches have attack likelihoods greater than 0.4. Of these, switch 251-250 remains in the same open state before and after the new configuration, while switches 13-152 and 60-160 are closed in the previous configuration, and open in the new configuration. The attacker aims to toggle switches in the closed state and keep those in the open state inaccessible. Hence, in the new configuration, the switches are assumed to be in the control of the attacker. This behavior is triggered by the second term of (4.28) as explained in an earlier section. Therefore, for this experiment, it is shown that the three switches could be lost to the attacker. It is also observed that in the new configuration, switch 195-95, which has a relatively low attack likelihood, is closed and feeder 195 serves

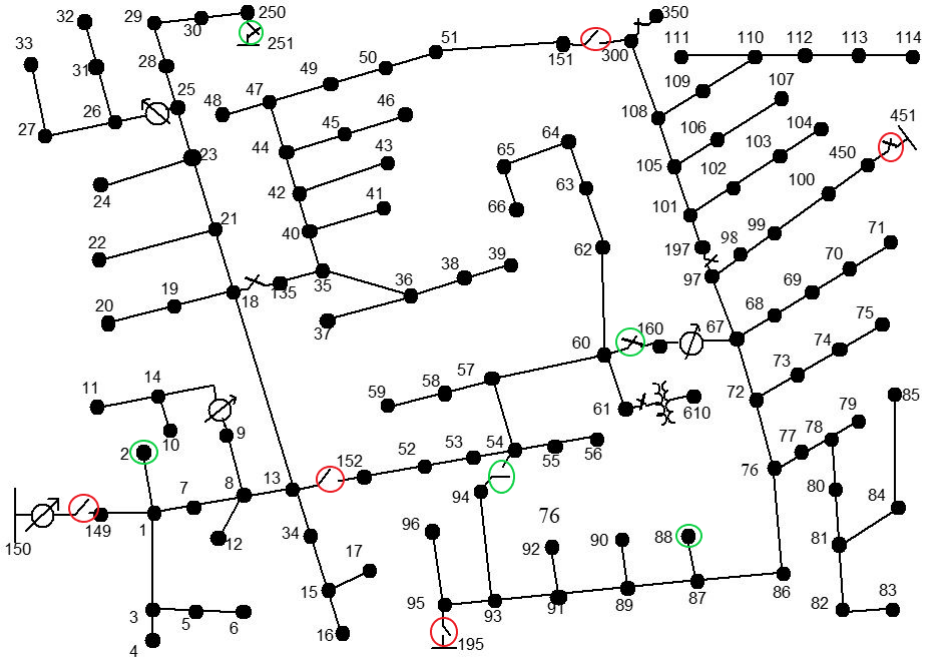

Figure 4.10: New configuration from central agent (CA) in E4

load. Again, this behavior is triggered by the second term of (4.28).

It is noteworthy that even though line switch 451-450 is not one of the targeted switches, its attack likelihood is high, as shown in Fig. 4.7. This is expected, as that switch has a high normalized quality, and its agent implements OS2 on Network C. When an alert is received from the agent at line switch 18-135 (which also implements OS2 on Network C), there is perfect operational similarity, i.e., $\psi_{451-450} = 1$. Other agents such as the one at switch 197-97 have increased attack likelihoods as more alerts are received and variation in communication network and firmware properties are observed.

4.7.2 Discussion on Experiment E3

In E3, no physical level mitigation is enforced as the threshold number of alerts are not received. The threshold, K, is therefore an important parameter that measures the willingness of the operator to reconfigure the network following the receipt of alerts. Thus, mitigation

is left at the communication level for as long as is necessary. This illustrates the need to have both communication level mitigation and physical level mitigation. For critical infrastructure such as the power grid, this is holistic cybersecurity. The provision of tune-able parameters such as K, T, and W to the operator allows for customized use of the algorithm.

4.7.3 Discussion on Experiment E4

In the final experiment, there is a significant number of switches with high attack likelihoods. This is expected since there is no variation in attack costs. Thus, alerts are received from agents with varying operational properties, triggering a high value of ψ at each of the agents. As shown in Fig. 4.10, the configuration is the same before and after the attack. This is mainly due to three of the feeder switches having shut down remote control capability. Thus, the CA must immediately shut down remote control capability for all remote-control switches without changing any state. This demonstrates that the operator must compete with the attacker to gain control of certain critical switches, with little tolerance for failure.

Chapter 5

Supply Chain Cyberattack Detection for Power Grids[1]

Compared to other forms of attacks such as man-in-the-middle attacks, supply chain cyberattack detection and mitigation in the context of power grid cybersecurity has not been fully explored. In recent times, supply chain cybersecurity has garnered a significant amount of interest from governments, industry practitioners, and academic researchers alike. This is, in part, due to the potentially far-reaching impacts of the same, as demonstrated by recent attacks. As supply chain attacks propagate through products, services, and connections, a preliminary study on the subject may focus on a product or service of interest. In this chapter, a model-based mechanism for detecting supply chain cyberattacks in a power system equipment is presented. The specific component in the power grid of interest in this study is the RTU, including feeder RTUs in the distribution systems. An RTU may be compromised in a supply chain attack as follows: a malicious hardware or software may be installed at the time of manufacture, shipping, or update. A compromised RTU may modify, falsify, drop, or delay communication signals between downstream components and the operations center. It may also engage in espionage activities. The technique presented in this chapter relies on modeling the normal behavior of the device by a Petri Net, which is then used in a series of algorithms to detect several forms of supply chain cyberattacks in real time.

[1] J. Appiah-Kubi, C. –C. Liu, R. Zhu, M. Otto, J. Vempati, "Detection of Supply Chain Cyberattacks for Power Grids", *CIGRE 2022 Kyoto Symposium*, Japan, 2022.

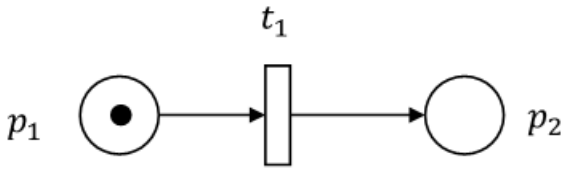

Figure 5.1: An example Petri Net

5.1 Background Information

A Petri Net (PN) models the structural and dynamic relationships among the components of a system, and a timed PN (TPN) factors in the temporal relationships as well. Therefore, a TPN is a good model for studying the normal operational pattern of an RTU.

5.1.1 Petri Nets

A Petri Net (PN) [46], an example of which is shown in Figure 5.1, is a 5-tuple $\mathcal{N} = (P, T, F, E, M_0)$, where $P = \{p_1, p_2, ..., p_m\}$ is a finite set of places, $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs or edges that connect places and transitions, $E : F \rightarrow \{1, 2, ...\}$ is a weight function that assigns weights to arcs,$M_0 : P \rightarrow \{0, 1, 2, ...\}$ is the initial marking, and $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$. In practice, a transition represents a task, or event, while a place is symbolic of a condition or an activation function for an event. A token, shown pictorially as a dot in a place, indicates which places are activated. Assuming that there are m places in the PN, a marking, M, is an m-vector in which the pth component is the number of tokens in place p.

The dynamic behavior of a PN, and therefore, of the system it models, is simulated by firing activated transitions. This removes $e(p, t)$ tokens from each input place of the transition and adds $e(t, p)$ to each of its output places, where $e(.)$ is the weight of the arc between both nodes. This modifies the marking of the PN. When not indicated, the weight of an arc is

1. For a PN with n transitions and m places, the transition matrix $A = [a_{ij}]$ is an $n \times m$ matrix. Its entries, a_{ij}, are integers and are the change in tokens at place j when transition i fires. Suppose there is an n-vector, u_i, with $n-1$ 0's and a 1 in the ith entry. This is the firing or control vector, which indicates that the ith transition is fired. Suppose at time step k, the firing vector is u_i, i.e., $u^{(k)} = u_i$. Then, the state equation for the PN is written as in (5.1).

$$M_k = M_{k-1} + A^T u^{(k)}, \quad k = 1, 2, ... \tag{5.1}$$

Starting from the initial marking M_0, the state equation may be written as in (5.2), where d is an arbitrary time step, and $x = \sum_{k=1}^{d} u^{(k)}$.

$$M_d = M_0 + A^T x \tag{5.2}$$

A Timed Petri Net (TPN) is an 8-tuple that comprises the five elements of standard PNs, and the following [47]: $D = (T \times S) \rightarrow \mathbb{R}^+ \cup \{0\}$ is a set of firing durations, $F = (T \times S) \rightarrow \mathbb{R}^+ \cup \{0\}$ is a set of firing frequencies, and $R = P \cup T \rightarrow \mathbb{P}(\{r_1, r_2, ..., r_k\})$ is a set of resources. Here, $\mathbb{P}$ is the power set and S is the set of reachable states. The firing duration is the time between the start firing event (i.e., when tokens are removed from a transition's input places), and the end firing event (i.e., when tokens are added to a transition's output places). In practice, for applications where conditions are not timed (i.e. no delay in places), the firing duration of a transition at time step k, is the time between when it is fired at time step k, and when the previous transition is fired at time step $k-1$.

5.1.2 Process Mining and Workflow Discovery

Workflow/process mining [48] discovers a structured process from a set of observations of real executions. The observations are in the form of an event log whose events are marked by at least a case identifier, activity description, and timestamp. All tasks with the same case identifier constitute a trace. The learned structured process is often a PN called workflow net (WF-net). A WF-net contains a source place, p_{src}, and a sink place, p_{snk}. A source place has no input transition, while a sink place has no output transition. There also exists a new identifier $t' \notin (P \cup T)$ such that the short-circuited net $\mathcal{N}' = (P, T \cup \{t'\}, F \cup \{(p_{snk}, t'), (t', p_{src})\})$ is strongly connected.

Workflow/process mining seeks to discover the underlying WF-net of a process from a set of workflow logs. Among the criteria used in assessing the performance of a workflow mining algorithm are fitness and generalization. Fitness requires that the model is able to reproduce the traces in the workflow log, while generalization measures how much the model is able to allow the occurrence of future behavior that is currently absent in the log. Closely related is the quality of precision, which measures the proportion of the behavior of the model not observed in the provided workflow log. A good mining algorithm is expected to not only be fitting but also generalizable and of good precision.

Several process mining algorithms exist for different needs. The α-algorithm [48] is one of the first and most well-known workflow mining algorithm. It has been shown to be able to discover a large class of WF-nets. There are also heuristic algorithms [49]–[51] which are known to handle short loops and noise well. The inductive miner algorithm guarantees a sound fitting model in finite time [52], [53].

5.2 Proposed Methodology

The proposed detection mechanism, summarized in $\mathcal{PT}$, comprises three algorithms. The first algorithm is a planning stage algorithm and is implemented offline. The second and third algorithms are installed on a simple computing device with access to real-time RTU logs, and are implemented online for real-time detection of supply chain attacks.

The first algorithm, $\mathcal{A}_1$, focuses on discovering the PN model of an operational RTU from a workflow log, W, using a process mining algorithm $\mathcal{MA}$. In $\mathcal{A}_1$, only the firing duration is of interest. Thus, a much simpler six-tuple TPN is obtained. Source and sink places are marked by an infinite number of tokens. To produce a tractable marking, both source and sink places are removed in $\mathcal{A}_1$. In step 11 of $\mathcal{A}_1$, a mapping, U, is obtained from the model, which maps each observed transition to its corresponding firing vector. Since the mining algorithm may introduce unobserved transitions to learn the underlying process, it is important that U accounts for this.

The second algorithm, $\mathcal{A}_2$, implements (5.1) as events occur. This process monitors for occurrences that are suggestive of incorrect operational sequence, packet falsification, packet drop, packet delay and hidden tasks. In the kth time step, only M_{k-1} and the timestamp of the previous transition, $timestamp(t^{(k-1)})$, are stored in memory. When transition i is fired (i.e., event i is detected) in time step k, the corresponding firing vector, $u^{(k)} = u_i$, is looked up in U and used to calculate for M_k. In the case where U returns no corresponding mapping, an unknown task is detected, which may be an espionage activity. The marking, M_k, must satisfy the condition in step 5 of $\mathcal{A}_2$. This is called safeness of the PN and is when the maximum number of tokens in each place is 1. This is a reasonable condition, as RTUs typically operate in cycles. In the event that the condition does not hold, an incorrect sequence is detected, which is the case when data is fabricated, and/or when hidden tasks

are present. In addition, the time taken to implement a task is required to be less than or equal to the maximum time. This is useful for detecting packet delay and packet drop as a result of a supply chain security breach.

In the third algorithm, $\mathcal{A}_3$, the content of communication packets is inspected for consistency. For a received message, the intended logical node and its control command or status information are translated if a mapping file $\mathcal{M}$ exists. For instance, a message may be received from the operations center with control signal for a specific IED. The translated data is, therefore, compared to the contents of the corresponding outgoing message to ensure that neither recipient nor instruction is altered.

Algorithm 3 $\mathcal{PT}$: Proposed Technique

1: From a given workflow log, apply $\mathcal{A}_1$ to learn underlying Petri Net model and its incidence matrix A, initial marking, M_0, duration set, D, and transition function, U.
2: Install $\mathcal{M}$, A, M_0, D and U, with $\mathcal{A}_2$ and $\mathcal{A}_3$ on the RTU.
3: Deploy RTU
4: **for each**
5: **do**
Execute $\mathcal{A}_2$ and $\mathcal{A}_3$
6: **end for**

Algorithm 4 $\mathcal{A}_1$: Planning stage algorithm for obtaining PN model and other parameters

1: Collect a complete workflow logs W
2: $PN = \mathcal{MA}(W)$
3: $P' = P(PN) \backslash \{p_{src}, p_{snk}\}$
4: **for each** $t_i, t_j \in T$, where t_i is fired at time step k, and t_j is fired at time step $k-1$ **do**
5: $\tau_{t_i}^{max} = \max\{(timestamp(t_i) - timestamp(t_j))\}$
6: **end for**
7: $D = [\tau_{t_i}^{max}], \quad i = 1, 2, ..., |T|$
8: Set initial marking $M_0 = \mathbf{0}$
9: $E : F \rightarrow \{1\}$
10: $\mathcal{N} = (P', T, F, E, M_0, D)$
11: Obtain $U : T \rightarrow \{u_i\}, \quad i = 1, 2, ..., |T|$
12: Derive incidence matrix A
13: Return A, M_0, T, D and U

Algorithm 5 $\mathcal{A}_2$: Online algorithm for detecting communication pattern anomalies

for each $t_i \in T$, fired at time step k, and $t_j \in T$ fired at time step $k-1$ **do**
  if $\exists\ U(t_i)$ **then**
    Get corresponding firing vector at the kth time step $u^{(k)} = U(t_i) = u_i$
    $M_k = M_{k-1} + A^T u^{(k)}$
    if $\mathbf{0} \preceq M_k \preceq \mathbf{1}$ **then**
      $\tau_{t_i}^{(k)} = timestamp(t_i) - timestamp(t_j)$
      if $\tau_{t_i}^{(k)} \leq \tau_{t_i}^{max}$ where $\tau_{t_i}^{max} = D[t_i]$ **then**
        No alert
        Implement $\mathcal{A}_3$
      else
        Send alert $L($ *packet delay* | *packet drop*$)$
        Set current marking $M_k = \mathbf{0}$
      end if
    else
      Send alert $L($*incorrect sequence* | *packet falsification* | *hidden task*$)$
      Set current marking $M_k = \mathbf{0}$
    end if
  else
    Send alert $L($*unknown task*|*potential espionage*$)$
  end if
end for

Algorithm 6 $\mathcal{A}_3$: Online algorithm for detecting modification to communication content

for each received message m_r **do**
  Get intended logical node of the message, $n_l^{m_r}$ and command/measurement, c^{m_r}
  if $\exists\ \mathcal{M}$ **then**
    Get translated logical node and command/measurement $\{n_l^t, c^t\} = \mathcal{M}(\{n_l^{m_r}, c^{m_r}\})$
  else
    $n_l^t = n_l^{m_r}$
    $c^t = c^{m_r}$
  end if
end for
for each sent message m_s **do**
  Get intended logical node in the message, $n_l^{m_s}$, and command/measurement, c^{m_s}
  if $n_l^{m_s} = n_l^t$ **then**
    if $c^{m_s} = c^t$ **then**
      No alert
    else
      Send alert $L(\textit{packet modification})$
    end if
  else
    Send alert $L(\textit{packet modification})$
  end if
end for

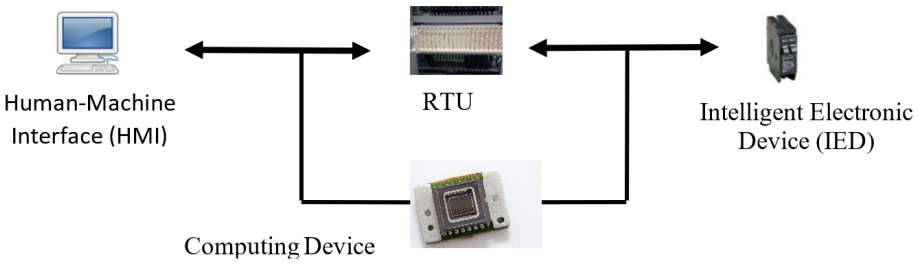

Figure 5.2: Setup for simulations

5.3 Simulations

Simulations are grouped into two main sections. In the first section, a Petri Net model is mined from an event log. Algorithm $\mathcal{A}_1$ is thus implemented in the first section. Three process mining algorithms are compared, using Python's pm4py library, and the best selected for the next section. The focus of the second section is to illustrate algorithms $\mathcal{A}_2$ and $\mathcal{A}_3$. Thus, different scenarios are created in which an RTU, compromised from the supplier side, has been deployed in a simple setup. The setup implemented is shown in Fig. 5.2. Reference [54] provides a packet capture which, in the absence of real devices, is used in this study. Thus, logs comprise communication packets. Algorithms $\mathcal{A}_2$ and $\mathcal{A}_3$ are installed by the user on a computing device with access to the communication packets of the RTU.

5.3.1 Mining Underlying Process from Event Log

Since the data available is a packet capture from Wireshark, it is first processed into an event log mineable by a process mining algorithm. The case ID of a packet is the stream (i.e., conversation) number assigned to the packet by Wireshark, and the timestamp is the timestamp of the packet. The name of the activity is derived using the source and destination addresses and the properties of the message contained in the packet (e.g., TCP flags, Modbus function code). In addition, incomplete streams are removed. Having processed the data,

Table 5.1: Comparison of performance of different mining algorithms

Mining Algorithm	Number of Places	Number of Transitions	Log Fitness	Precision	Generalization
Alpha	8	13	0.9892	0.2104	0.9916
Inductive	17	22	1.0000	0.7225	0.9918
Heuristic	12	17	1.0000	0.7225	0.9913

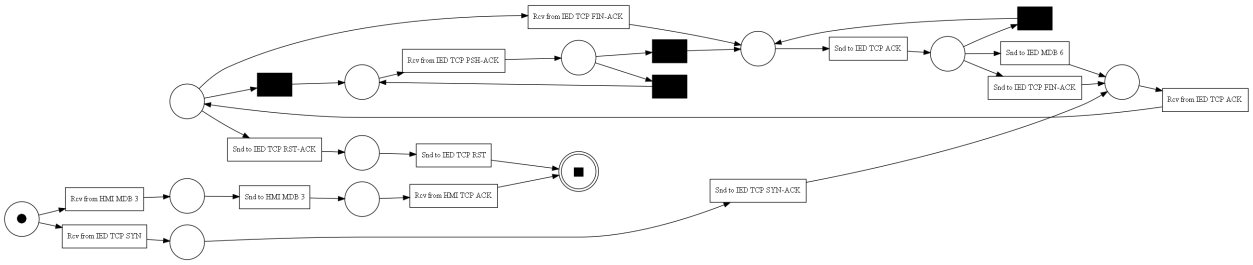

Figure 5.3: Output model of heuristic mining algorithm

thirteen unique tasks are obtained. The three mining algorithms are applied in turn and their performance summarized in Table 5.1.

From Table 5.1, it is observed that the heuristic miner has a performance comparable to that of the inductive miner, with perfect fitness and high precision. However, the model it produces, shown in Fig. 5.3, is simpler. It is therefore the mining algorithm of choice.

In Fig. 5.3 the heuristic mining algorithm mines a Petri Net model with four management tasks (i.e., black boxes). Management tasks are events that are not observed in the log but are learned by the mining algorithm to obtain a fitting model for the log. Their real-time detection is difficult until they are clearly identified and labelled. To label such tasks, more descriptive information is provided so as to clearly differentiate events. The final model is shown in Fig. 5.4. In Table 5.2, the mapping function U is provided. Ignoring the source and sink places, the initial marking is $M_0 = \mathbf{0}_{(10\times1)}$.

One may notice, from Table 5.2, that certain tasks have two 1's in their transition matrix (e.g., Snd to IED TCP ACK < 13). This is because they are originally management tasks,

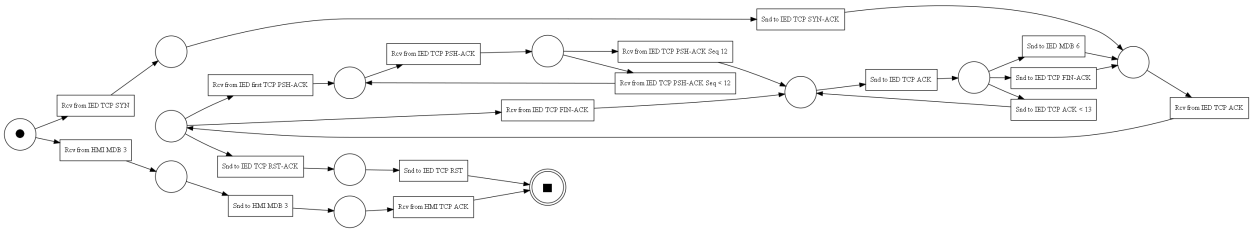

Figure 5.4: Final model with all management tasks identified and labelled

and were labelled by a more detailed description of another task, t. Thus, their occurrence implies that t has also occurred. Again, ignoring the source and sink places, the transpose of the incidence matrix is given by (5.3).

$$A^T = \begin{bmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & -1 & 0 & 0
\end{bmatrix} \tag{5.3}$$

5.3.2 Online Deployment and Real-time Detection of Supply Chain Attacks

In this section, algorithms $\mathcal{A}_2$ and $\mathcal{A}_3$ are demonstrated. Four different scenarios have been assumed in which the RTU has malicious hardware and/or software to implement espionage,

Table 5.2: Mapping function U

Task/Event	Transition Vector
Rcv from HMI MDB 3	$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Snd to HMI MDB 3	$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from HMI TCP ACK	$[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP SYN	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Snd to IED TCP SYN-ACK	$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP ACK	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED first TCP PSH-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP PSH-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP PSH-ACK Seq 12	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Snd to IED TCP ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
Snd to IED TCP ACK < 13	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$
Snd to IED MDB 6	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
Rcv from IED TCP FIN-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$
Snd to IED TCP FIN-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$
Snd to IED TCP RST-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$
Snd to IED TCP RST	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$

Table 5.3: Sequence of events in espionage scenario

Event Number	Task	Current Marking
1	Rcv from HMI MDB 3	$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
2	Snd to HMI MDB 3	$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
3	**Snd to 172.224.30.1 MDB 3**	-

data modification, incorrect operational sequence, and packet delay.

Espionage

In this scenario, the HMI requests status information (Modbus Function Code 3) from the RTU, which updates the HMI accordingly. Nevertheless, the RTU also sends a copy of the status information to an attacker address. Algorithm $\mathcal{A}_2$ detects events as they occur and constructs a task for each. It then looks up the mapping function, U, to get a transition vector for the new task. The sequence of events is listed as in Table 5.3. The current marking is calculated from (5.1), and is valid. However, from U, event 3 is unknown (steps 2 and 19 of $\mathcal{A}_2$) because the destination address is unknown. Hence, an alert is sent to indicate the presence of an unknown task and a potential espionage.

Data Modification

Here, the HMI sends a request for status information to the RTU. In turn, the RTU, which is compromised, sends a modified response to the HMI. The aim is to mislead the human operator to react to the wrong values. This potentially leads to an operator-induced emergency situation. However, $\mathcal{A}_3$ detects the change in values by comparing what is sent to the HMI to what is received from the IED a priori.

Packet delay/drop

The RTU is compromised such that there is packet delay. In this scenario, the IED reports a change in values to the compromised RTU. Following this, the HMI sends a request for status information. The RTU does not respond immediately, but delays sending the update for 30 seconds. The sequence of events is shown in Table 5.4, and the final marking is valid. However, since the known maximum time for the task (i.e., Send to HMI Modbus packet with Function Code 3), about 200ms, is less than the observed time of 30.02s, algorithm $\mathcal{A}_2$ detects the excessive delay and alerts accordingly.

Incorrect Operational Sequence

For this scenario, the RTU has a malicious code installed which attempts to send a malicious write request (Modbus Function Code 6) to the IED. The normal sequence is for the IED to first establish a TCP handshake as in the previous scenario, and follow up with a series of messages before the RTU sends a write request. Since this procedure is not followed, $\mathcal{A}_2$ detects the invalid marking and sends an alert to notify of the incorrect sequence. The sequence of events is listed in Table 5.5.

5.4 Extension of Proposed Method

The proposed detection mechanism may be extended into a standardized certification/validation process for suppliers. Each supplier in the chain, upon delivery of a product, attaches a PN model to describe its normal operation. The normal operation of the final product is therefore a chain of the PN models of its components. The provision of a PN model by the supplier removes the need to learn the underlying model, so that $\mathcal{A}_2$ and $\mathcal{A}_3$ may be installed immediately. In the event that an alert is received, the section of the PN where the anomaly is observed is inspected and corresponded to a specific supplier. Furthermore, the technique

Table 5.4: Sequence of events in packet delay scenario

Event Number	Task	Current Marking
1	Rcv from IED TCP SYN	$[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
2	Snd to IED TCP SYN-ACK	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
3	Rcv from IED TCP ACK	$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$
4	Rcv from IED first TCP PSH-ACK	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
5	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
6	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
7	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
8	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
9	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
10	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
11	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
12	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
13	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
14	Rcv from IED TCP PSH-ACK Seq < 12	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
15	Rcv from IED TCP PSH-ACK Seq 12	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$
16	Snd to IED TCP ACK < 13	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$
17	Snd to IED TCP ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$
18	Snd to IED MDB 6	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
19	Rcv from IED TCP ACK	$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$
20	Rcv from IED TCP FIN-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$
21	Snd to IED TCP ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$
22	Snd to IED TCP FIN-ACK	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
23	Rcv from IED TCP ACK	$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$
24	Snd to IED TCP RST-ACK	$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$
25	Snd to IED TCP RST	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
26	Rcv from HMI MDB 3	$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
27	**Snd to HMI MDB 3**	$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

Table 5.5: Sequence of events in incorrect operational sequence scenario

Event Number	Task	Current Marking
1	Rcv from HMI MDB 3	$[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$
2	Snd to HMI MDB 3	$[0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$
3	**Snd to IED MDB 6**	$[0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1]^T$

may be extended to monitor a whole substation or system of equipment. Here, the proposed technique is installed on a computing device with visibility over all or key equipment in the substation.

The proposed detection mechanism is also useful without a vendor-provided PN model. For detection of compromised replacement devices, the PN model mined prior to equipment replacement can be used as the baseline PN to check if and how the mined PN model of a replacement equipment differs.

Substation networks are typically private networks built to provide uninterrupted access for control and automation. However, occasionally, benign errors may occur, such as network failure and delays, generating false alarms. To cater to this, logs including benign errors may be applied at the process mining stage so that a net that allows for both normal operation and benign error situations may be obtained.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research is focused on correlating and responding to cyberattacks on electric power distribution systems. While the work focuses on direct switching attacks on the distribution grid, it also acknowledges the viability of supply chain attacks as another attack option. Consequently, the work also provides a study on supply chain cyberattack detection.

Regarding direct switching attacks, the work first establishes a decentralized intrusion prevention architecture, which is a paradigm shift from the traditional centralized architecture often presented in the literature. The architecture provides for real-time monitoring, attack target prediction and mitigation. The framework leverages a multi-agent system (MAS), whose agents are deployed at remote-control switches in the distribution system. The agents perform local intrusion detection, and are required to send an alert on cyber intrusions to the operations center and to all other agents in the network. Agents that receive alerts compute relevant metrics, i.e., the attack likelihood index, by which to measure their certainty of being a target for the attacker. This constitutes decentralized attack target prediction. A mitigation strategy is executed following this.

In one prototype of the framework, a consensus protocol, the link drop max consensus protocol, is formulated to allow for agents to come to consensus on what communication level mitigation strategy is to be employed. The consensus protocol is shown to be more effective

than the traditional max consensus protocol, and the mitigation is performed network-wide.

In another prototype, the dispersed agents of the MAS apply machine learning to determine the attack likelihood index, and thus, form judgement of the certainty of being targets of an attack. Due to the learning capability introduced, the agents are able to improve on their certainty of prediction as more alerts are received. Communication network level mitigation is executed by the agents in proportion to the level of threat perceived as measured by the attack likelihood index. This mitigation is implemented using reinforcement learning. Thus, it is optimal, and is refined as predictions are improved. When enough alerts are received at the operations center, physical mitigation is triggered. This primarily takes the form of distribution network reconfiguration (DNR). Nonetheless, contingency analysis is also performed to provide insight to the human operator on the severity of the attack scenario. This is useful in reconfiguring tunable parameters provided by the algorithm for better performance.

The proposed MAS-based approach towards predicting the targets of a direct switching cyberattack is not prone to single-point failures. Compared to interdiction models in the literature, the approach assumes no explicit knowledge of the attacker's parameters by the defenders, which in this case, are agents. The major contributions regarding correlation and mitigation of direct switching attacks on electric power distribution systems are:

1. A novel switching attack problem is formulated, which accounts for communication network vulnerabilities and the criticality of load.

2. A novel real-time decentralized mechanism for establishing coordination of attacks and predicting the targets of an attack is proposed.

3. Decentralized and hierarchical mitigation strategies are proposed for different purposes. The hierarchical mitigation strategy is also hybrid, combining both physical level and

communication network level mitigation optimized by information learned from the attacker's behavior.

The work presented in this dissertation also applies a model-based technique leveraging the mathematical support for Petri Nets to detect supply chain cyberattacks. The proposed technique is applied to real data from an example setup that contains a power system equipment of interest. It has been shown that the proposed method is able to detect several forms of supply chain attacks in real time. The major contributions are as follows:

1. A novel real-time anomaly-based method is proposed for detecting supply chain cyberattacks in power system equipment.

2. The method detects attacks not detected by traditional intrusion detection systems.

6.2 Future Work

To enhance the implementation of the proposed algorithms for responding to direct switching attacks, further investigations are required as follows:

Machine-Learning-Based Intrusion Detection: Currently, the NIDS implemented by the proposed algorithms is anomaly-based and makes use of only communication network level thresholds. It is therefore limited to only man-in-the-middle attacks, and may be prone to false alarms. Future work may consider improving the mechanism of intrusion detection by integrating machine learning or another suitable method. Also, the inclusion of physical level checks in intrusion detection may prove useful for detecting insider attacks.

Optimal Placement of Agents: The installation and maintenance of agents will incur additional costs for system operators, especially for large distribution networks. Thus, future work may also consider optimal placement of agents in order to minimize associated costs.

Generalization to smart distribution grids: The integration of distributed renewable energy resources, battery storage technology, localized electricity markets, and special electric load such as electric vehicles, remote-controlled load, and data centers into the power grid have changed the architecture and operation of the grid. The new outlook is the smart power infrastructure. The work presented in this dissertation is a paradigm shift in the response to cyberattacks on the electric power grid. However, it is primarily focused on the traditional electric distribution system, which is assumed radial, and is homogeneous. In the face of the new smart power infrastructure, the ensuing bi-directional power flow and the heterogeneous nature warrants certain key considerations in the future. These may include categorizing agents according to the type of load or source being monitored, developing new mathematical models, and diversifying mitigation, i.e., developing different mitigation schemes for different agents.

Regarding supply chain attacks, the following are useful directions for future work.

Advanced Petri Net (PN) Simplification Techniques: The concatenation of several PNs in a substation and/or a supply chain has potential to connect cyber compromise to real suppliers in a supply chain. However, the chaining of several Petri Nets may easily become intractable. Therefore, it is necessary to apply graph simplification techniques to restrict the size of the ensuing Petri Net.

Distinction of benign error from malicious behavior: The distinction of benign errors from malicious behavior, possibly with statistical techniques, is an advanced investigation and a worthy direction for future work.

Bibliography

[1] S. Nazir, S. Patel, and D. Patel, "Assessing and Augmenting SCADA Cyber Security: A Survey of Techniques," *Computers & Security*, vol. 70, pp. 436–454, 2017, ISSN: 01674048. DOI: 10.1016/j.cose.2017.06.010.

[2] M. F. Ahern, "Cybersecurity in Power Systems," *IEEE Potentials*, vol. 36, no. 5, pp. 8–12, 2017, ISSN: 0278-6648. DOI: 10.1109/MPOT.2017.2700239.

[3] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine Blackout: Implications for False Data Injection Attacks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2017, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2016.2631891.

[4] H. Zhang, B. Liu, and H. Wu, "Smart Grid Cyber-Physical Attack and Defense: A Review," *IEEE Access*, vol. 9, pp. 29 641–29 659, 2021. DOI: 10.1109/ACCESS.2021.3058628.

[5] A. Gusrialdi and Z. Qu, "Smart Grid Security: Attacks and Defenses," in *Smart Grid Control. Power Electronics and Power Systems*, Cham: Springer International Publishing, 2018, pp. 199–223.

[6] R. Deng, P. Zhuang, and H. Liang, "False Data Injection Attacks Against State Estimation in Power Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2871–2881, 2019, ISSN: 1949-3053. DOI: 10.1109/TSG.2018.2813280.

[7] Y. Liu, P. Ning, and M. K. Reiter, "False Data Injection Attacks Against State Estimation in Electric Power Grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 1–33, 2011, ISSN: 1094-9224. DOI: 10.1145/1952982.1952995.

[8] J. Ospina, X. Liu, C. Konstantinou, and Y. Dvorkin, "On the Feasibility of Load-Changing Attacks in Power Systems During the COVID-19 Pandemic," *IEEE Access*, vol. 9, pp. 2545–2563, 2021. DOI: 10.1109/ACCESS.2020.3047374.

[9] S. Amini, F. Pasqualetti, and H. Mohsenian-Rad, "Dynamic Load Altering Attacks Against Power System Stability: Attack Models and Protection Schemes," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2862–2872, 2018, ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2622686.

[10] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Distributed Internet-Based Load Altering Attacks Against Smart Power Grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 667–674, 2011, ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2160297.

[11] J. Giraldo, A. Cardenas, and N. Quijano, "Integrity Attacks on Real-Time Pricing in Smart Grids: Impact and countermeasures," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2249–2257, 2017, ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2521339.

[12] Microsoft 365 Defender Research Team and Microsoft Threat Intelligence Center, *Analyzing Solorigate, the Compromised DLL File that Started a Sophisticated Cyberattack, and how Microsoft Defender Helps Protect Customers*, 2020. [Online]. Available: https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/.

[13] I.-S. Choi, J. Hong, and T.-W. Kim, "Multi-agent Based Cyber Attack Detection and Mitigation for Distribution Automation System," *IEEE Access*, vol. 8, pp. 183 495–183 504, 2020. DOI: 10.1109/ACCESS.2020.3029765.

[14] C. Moya and J. Wang, "Developing Correlation Indices to Identify Coordinated Cyber-attacks on Power Grids," *IET Cyber-Physical Systems: Theory Applications*, vol. 3, no. 4, pp. 178–186, 2018. DOI: 10.1049/iet-cps.2018.5002.

[15] Y. Lin and Z. Bie, "Tri-level Optimal Hardening Plan for a Resilient Distribution System Considering Reconfiguration and DG Islanding," *Applied Energy*, vol. 210, pp. 1266–1279, 2018, ISSN: 03062619. DOI: 10.1016/j.apenergy.2017.06.059.

[16] K. Lai, M. Illindala, and K. Subramaniam, "A Tri-level Optimization Model to Mitigate Coordinated Attacks on Electric Power Systems in a Cyber-Physical Environment," *Applied Energy*, vol. 235, pp. 204–218, 2019, ISSN: 03062619. DOI: 10.1016/j.apenergy.2018.10.077.

[17] A. Abedi, M. R. Hesamzadeh, and F. Romerio, "An ACOPF-Based Bilevel Optimization Approach for Vulnerability Assessment of a Power System," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106 455, 2021, ISSN: 01420615. DOI: 10.1016/j.ijepes.2020.106455.

[18] L. Wei, A. I. Sarwat, W. Saad, and S. Biswas, "Stochastic Games for Power Grid Protection Against Coordinated Cyber-Physical Attacks," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 684–694, 2018. DOI: 10.1109/TSG.2016.2561266.

[19] Z. Li, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah, "Bilevel Model for Analyzing Coordinated Cyber-Physical Attacks on Power Systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2260–2272, 2016. DOI: 10.1109/TSG.2015.2456107.

[20] C. Moya, J. Hong, and J. Wang, *Application of Correlation Indices on Intrusion Detection Systems: Protecting the Power Grid Against Coordinated Attacks*, 2018. arXiv: 1806.03544 [cs.CR]. [Online]. Available: https://arxiv.org/pdf/1806.03544.pdf.

[21] S. Lakshminarayana, E. V. Belmega, and H. V. Poor, "Moving-Target Defense for Detecting Coordinated Cyber-Physical Attacks in Power Grids," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–7. DOI: 10.1109/SmartGridComm.2019.8909767.

[22] C. Sun, J. Hong, and C. Liu, "A Coordinated Cyber Attack Detection System (CCADS) for Multiple Substations," in *2016 Power Systems Computation Conference (PSCC)*, 2016, pp. 1–7. DOI: 10.1109/PSCC.2016.7540902.

[23] Y. Li and L. Xu, "Cybersecurity Investments in a Two-Echelon Supply Chain with Third-party Risk Propagation," *International Journal of Production Research*, vol. 59, no. 4, pp. 1216–1238, 2021, ISSN: 0020-7543. DOI: 10.1080/00207543.2020.1721591.

[24] North American Electric Reliability Corporation, "CIP-013-1 – Cyber Security - Supply Chain Risk Management." [Online]. Available: https://www.nerc.com/pa/Stand/Reliability%5C%20Standards/CIP-013-1.pdf.

[25] S. Dunzer, M. Stierle, M. Matzner, and S. Baier, "Conformance Checking," in *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management - S-BPM ONE '19*, S. Betz, Ed., New York, New York, USA: ACM Press, 2019, pp. 1–10, ISBN: 9781450362504. DOI: 10.1145/3329007.3329014.

[26] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly Detection for Industrial Control Systems Using Process Mining," *Computers & Security*, vol. 78, pp. 103–125, 2018, ISSN: 01674048. DOI: 10.1016/j.cose.2018.06.002.

[27] M. Parvania, G. Koutsandria, V. Muthukumary, S. Peisert, C. McParland, and A. Scaglione, "Hybrid Control Network Intrusion Detection Systems for Automated Power Distribution Systems," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, IEEE, 22-Jun-14 - 25-Jun-14, pp. 774–779, ISBN: 978-1-4799-2233-8. DOI: 10.1109/DSN.2014.81.

[28] C.-W. Ten, J. Hong, and C.-C. Liu, "Anomaly Detection for Cybersecurity of the Substations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865–873, 2011, ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2159406.

[29] ISA Global Security Alliance, "Security of Industrial Automation and Control Systems: Security Lifecycles in the ISA/IEC 62443 Series," 2014. [Online]. Available: https://www.isasecure.org/en-US/Documents/Articles-and-Technical-Papers/ISAGCA-Security-Lifecycles-whitepaper.

[30] B. Genge, P. Haller, and I. Kiss, "Cyber-Security-Aware Network Design of Industrial Control Systems," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1373–1384, 2017, ISSN: 1932-8184. DOI: 10.1109/JSYST.2015.2462715.

[31] S. Reka, T. Dragičević, P. Siano, and S. S. Prabaharan, "Future Generation 5G Wireless Networks for Smart Grid: A Comprehensive Review," *Energies*, vol. 12, no. 11, p. 2140, 2019. DOI: 10.3390/en12112140.

[32] B. A. Akyol, J. N. Haack, S. Ciraci, B. J. Carpenter, M. Vlachopoulou, and C. W. Tews, "VOLTTRON: An Agent Execution Platform for the Electric Power System," Jun. 2012. [Online]. Available: https://availabletechnologies.pnnl.gov/technology.asp?id=369.

[33] X. Duan, J. He, P. Cheng, Y. Mo, and J. Chen, "Privacy Preserving Maximum Consensus," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec. 2015, pp. 4517–4522. DOI: 10.1109/CDC.2015.7402925.

[34] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-Consensus in a Max-Plus Algebraic Setting: The Case of Fixed Communication Topologies," in *2009 XXII International Symposium on Information, Communication and Automation Technologies*, Oct. 2009, pp. 1–7. DOI: 10.1109/ICAT.2009.5348437.

[35] S. Giannini, D. Di Paola, A. Petitti, and A. Rizzo, "On the Convergence of the Max-Consensus Protocol with Asynchronous Updates," in *52nd IEEE Conference on Decision and Control*, Dec. 2013, pp. 2605–2610. DOI: 10.1109/CDC.2013.6760275.

[36] T. Mahjabin, G. S. Y. Xiao, and W. Jiang, "A Survey of Distributed Denial-of-Service Attack, Prevention, and Mitigation Techniques," *International Journal of Distributed Sensor Networks*, vol. 13, 2017. DOI: 10.1177/1550147717741463.

[37] R. O. Saber and R. M. Murray, "Consensus Protocols for Networks of Dynamic Agents," in *Proceedings of the 2003 American Control Conference*, 2003, pp. 951–956. DOI: 10.1109/ACC.2003.1239709.

[38] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, N.J.: Prentice Hall, 1989, ISBN: 9780136487005.

[39] P. Biondi, "Scapy Documentation, Release 2.4.2-dev," Apr. 2019. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/scapy/latest/scapy.pdf.

[40] A. L. Motto, J. M. Arroyo, and F. D. Galiana, "A Mixed-Integer LP Procedure for the Analysis of Electric Grid Security Under Disruptive Threat," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1357–1365, 2005, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2005.851942.

[41] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948, ISSN: 00058580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[42] W. Wu, B. Li, L. Chen, C. Zhang, and P. S. Yu, "Improved Consistent Weighted Sampling Revisited," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2332–2345, 2019, ISSN: 1041-4347. DOI: 10.1109/TKDE.2018.2876250.

[43] R. S. Sutton and A. G. Barto, "Multi-Armed Bandits," in *Reinforcement Learning: An Introduction*, Cambridge, MA, USA: MIT Press, 2018, pp. 32–33.

[44] L. Gan and S. H. Low, "Convex Relaxations and Linear Approximation for Optimal Power Flow in Multiphase Radial Networks," in *2014 Power Systems Computation Conference*, IEEE, 2014, pp. 1–9, ISBN: 978-83-935801-3-2. DOI: 10.1109/PSCC.2014.7038399.

[45] J. A. Taylor and F. S. Hover, "Convex Models of Distribution System Reconfiguration," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1407–1413, 2012, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2012.2184307.

[46] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989. DOI: 10.1109/5.24143.

[47] M. A. Holliday and M. K. Vernon, "A Generalized Timed Petri Net Model for Performance Analysis," *IEEE Transactions on Software Engineering*, vol. 13, no. 12, pp. 1297–1310, Dec. 1987. DOI: 10.1109/TSE.1987.233141.

[48] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow Mining: Discovering Process Models from Event Logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004. DOI: 10.1109/TKDE.2004.47.

[49] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros, "Process Mining with the Heuristics Miner Algorithm," 2006. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.8288.

[50] A. J. M. M. Weijters and W. M. P. van der Aalst, "Process Mining: Discovering Workflow Models from Event Based Data," in *Belgium-Netherlands Conference on Artificial Intelligence*, 2001.

[51] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible Heuristics Miner (FHM)," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, IEEE, 2011, pp. 310–317. DOI: 10.1109/CIDM.2011.5949453.

[52] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach," in *Colom JM., Desel J. (eds) Application and Theory of Petri Nets and Concurrency. PETRI NETS 2013. Lecture Notes in Computer Science*, Springer, 2013. DOI: 10.1109/Trustcom.2015.446.

[53] A. Bogarin, R. Cerezo, and C. Romero, "Discovering Learning Processes Using Inductive miner: A Case Study with Learning Management Systems (LMSs)," *Psicothema*, 2018. DOI: 10.7334/psicothema2018.116.

[54] I. Frazao, and P. Abreu, and T. Cruz, and P. Simoes, *Cyber-security Modbus ICS Dataset*, 2019. DOI: 10.21227/pjff-1a03.