Risk modeling for cyber-physical system security planning

by

Carmen Haseltine

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)

at the
University of Wisconsin-Madison
2025

Date of final oral examination : 7/14/2025
The dissertation is approved by the following members of the Final Oral Committee:
Laura Albert, Professor, Industrial and Systems Engineering
Bernie Lesieutre, Professor, Electrical and Computer Engineering
Ramya Korlakai Vinayak, Assistant Professor, Electrical and Computer Engineering
Justin Boutilier, Assistant Professor, Telfer School of Management (Uni. of Ottawa)

Risk modeling for cyber-physical system security planning

Threat modeling and analysis techniques
An exploration into risk analysis for modern critical infrastructure

Carmen Haseltine

Abstract

Our society depends on critical infrastructures increasingly composed of interdependent cyber-physical systems (CPS). Securing these infrastructures requires understanding the dynamic interactions across cyber, physical, and human domains, as well as how adversaries can exploit these interdependencies. In this work, I present a risk modeling framework for CPS security that leverages attack graphs and directed acyclic graphs (DAG) to characterize system vulnerabilities and adversarial pathways using a layered approach. Motivated by the temporal and stochastic nature of CPS operations, I incorporate discrete-time Markov chains (DTMC) to enable time-dependent risk analysis.

The structured modeling framework for CPS security integrates attack graphs with operational process models to support risk-informed decision-making. I apply this framework to two case studies in critical infrastructure: (1) a cyber-physical-human system (CPHS) representing the vote-by-mail (VBM) election process, and (2) a cyber-physical energy system (CPES) incorporating a small modular reactor (SMR).

A time-inhomogeneous discrete-time Markov chain (DTMC) model is used for VBM risk modeling to capture ballot flow, adversarial interference, and policy mitigations with a case study using absentee voting data from Milwaukee, WI. The SMR station has a layered architecture and construct attack graphs across functional layers to represent potential attack paths and system interdependencies. I introduce new metrics to assess vulnerability and component criticality under adversarial conditions. By framing these systems as CPES/CPHS, the framework effectively bridges the gap between abstract threat modeling and tangible operational realities, ensuring a more robust understanding of potential vulnerabilities. Together, these case studies demonstrate how temporal and structural extensions of attack graphs can provide scalable and interpretable tools for CPS risk analysis and prescriptive security planning.

Acknowledgments

I am genuinely grateful to my advisor, Prof. Laura Albert, and co-advisor, Prof. Bernie Lesieutre, for their steadfast mentorship and generous guidance throughout my doctoral studies. Their insight, patience, and belief in my work have been instrumental to my growth as a scholar. I also extend my sincere gratitude to the rest of my committee, whose thoughtful engagement and support have been invaluable in shaping my academic journey.

The GERS program, my advisor's research grants, and the Grainger award for which my co-advisor nominated me have played a crucial role in supporting this thesis. I am incredibly thankful for these vital resources that made my work possible.

Several key people have supported me on this journey. To my family: Your prayers and belief in my abilities made this journey possible. To my husband: Your unwavering faith in me inspired me to begin. You encouraged me to pursue what is meaningful rather than just safe.

Finally, I want to thank all my friends, colleagues, and mentors who have been a part of this journey. Your kindness, strength, and collaboration have made all the difference. Thank you!

Contents

List of Figures

List of Acronyms

CPS Cyber-Physical System

CPES Cyber-Physical Energy System

CPHS Cyber-Physical-Human System

DTMC Discrete-Time Markov Chain

DAG Directed Acyclic Graph

ICS Industrial Control System

BES Bulk Electric System

SCADA Supervisory Control and Data Acquisition

SMR Small Modular Reactor

VBM Vote-by-Mail

OT Operational Technology

IT Information Systems

PLC Programmable Logic Controller

RTU Remote Terminal Unit

Chapter 1

Introduction

1.1 Introduction

In an era defined by the convergence of digital intelligence and physical infrastructure, the systems most essential to daily life (i.e., power grids, water networks, transportation systems, and emergency services) are increasingly comprised of cyber-physical systems (CPS). CPS integrates computation, control, and physical processes to deliver critical services at scale. However, this integration also introduces new and complex vulnerabilities. Disruptions in one cyber, physical, or human interaction layer can propagate across domains, producing cascading failures. Traditional risk models, which treat cyber and physical threats in isolation, fail to capture these interdependencies. A holistic modeling approach is needed to assess risk and support decision-making in modern critical infrastructure.

As the threat landscape expands, cyber-physical systems face malicious and non-malicious disruptions. Adversaries may exploit software vulnerabilities, misconfigured human-machine interfaces, or outdated control equipment. Additionally, non-malicious attacks such as equipment failure, operator error, and natural hazards pose other performance risks. Attack graphs have emerged as a structured tool for modeling how threats (both malicious and non-malicious) can propagate through a system. These graphs represent sequences of attacker actions or system-level transitions, offering a visual and mathematical representation of threat escalation pathways.

This dissertation builds on the attack graph paradigm by incorporating directed acyclic graphs (DAGs) to represent logical and temporal dependencies and embedding the resulting models within a time-inhomogeneous discrete-time Markov chain (DTMC) framework. This integrated approach enables the dynamic modeling of risk evolution over time, taking into account attacks, mitigations, and system processes. I apply the methodology to two domains of critical infrastructure: (1) a cyber-physical energy system (CPES), modeled through a layered analysis of a small modular reactor (SMR) station, and (2) a cyber-physical-human system (CPHS), applied to vote-by-mail (VBM) election infrastructure. In both applications, the model supports the evaluation of cascading failures and the effectiveness of mitigation through newly developed system performance metrics.

1.2 Motivation: Securing cyber-physical critical infrastructures

The integration of systems in modern critical infrastructure enables the delivery of goods, services, and energy at scale, ensuring societal and economic stability [65]. At the same time, the interconnectedness of critical infrastructure introduces vulnerabilities to various hazards, which can result in cascading failures [19]. This vulnerability motivates threat modeling to identify risks and fortify the system's critical components. Threat modeling frameworks, such as attack graphs, have emerged as essential tools for safeguarding these systems and supporting enterprise risk management. Attack graphs visually represent the paths an attacker might take to compromise a system, helping to identify vulnerabilities, understand potential attack vectors, and prioritize security measures.

Defining critical infrastructure Critical infrastructure in the United States (US) encompasses a wide range of systems and assets that are essential for the nation's security, economic stability, and public health and safety [84]. These infrastructures are categorized into sixteen sectors, each playing a vital role in maintaining the functionality and resilience of the country. Table 1.1 lists the official US critical infrastructure sectors alongside the US government agency responsible for the secure operation of that sector.

Table 1.1: Critical infrastructure sectors and their responsible federal agencies

	Critical Infrastructure Sector	**Responsible Federal Government Agency**
1	Chemical	Department of Homeland Security
2	Commercial Facilities	Department of Homeland Security
3	Communications	Department of Homeland Security
4	Critical Manufacturing	Department of Homeland Security
5	Dams	Department of Homeland Security
6	Defense Industrial Base	Department of Defense
7	Emergency Services	Department of Homeland Security
8	Energy Infrastructure	Department of Energy
9	Financial Services	Department of the Treasury
10	Food and Agriculture	Department of Agriculture
11	Government Facilities	Department of Homeland Security and General Services Administration
12	Healthcare and Public Health	Department of Health and Human Services
13	Nuclear Reactors, Materials, and Waste	Department of Homeland Security
14	Information Technology	Department of Homeland Security
15	Transportation Systems	Department of Homeland Security and Department of Transportation
16	Water and Wastewater Systems	Environmental Protection Agency

1.2.1 Threats to cyber-physical infrastructures

Now that we know that modern critical infrastructures are comprised of CPS, we must understand the risk to CPS. There are cyber, physical, and cyber-physical threats to a CPS. Cyber attacks are often related to communication protocols of various vendor components comprised in a CPS. These attacks arise in the form of spoofing attacks, spying attacks, false data injection, and glitches in software updates for computational components. An often referenced incident of a cyber attack on a CPS was STUXNET which targeted programmable logic controllers at an Iranian nuclear facility. The cyber attack involved malware which injected false data into the monitoring system. [44] The cyber-physical part of the STUXNET attack is defined to be the part of the code which gave incorrect commands to equipment controls incorrect control commands were sent to physical equipment at the facility. [44]

Cyber-physical attacks use both computational and physical components to disrupt a CPS. An example of a cyber-physical attack would be the Maroochy Shire attack in Australia which resulted in 800 kiloliters of raw sewage spilling into local roadways and parks. The attacker had insider information which allowed them to use a radio transmitter and a laptop to send malicious commands to specific pumps[79]. Cyber-physical attacks can be summarized as adversarial access to control systems in a CPS.

Similarly physical attacks have been largely considered to involve unapproved access to physical components of CPS[33]. Physical attacks are defined as damage of the CPS physical components by nature, vandalism, or accidents. One notable example of a physical attack on a power grid is the Metcalf attack. Early morning on April 16, 2013, seventeen of the California Metcalf substation transformers were shot over 100 times by a .30-caliber rifle in less than 20 minutes[88]. These attack types are varied in execution and each type must be analyzed to determine all potential vulnerabilities of a CPS.

Modern critical infrastructure are comprised of highly interdependent CPS, which are essential for the functioning of various sectors. For example, transportation roadways rely on information technology (IT) communication networks with cyber components, which in turn depend on power grids. Power grids are interconnected with other infrastructures, such as natural gas pipelines and water systems. CPS in the manufacturing sector often referred to as Industrial Control Systems (ICS), while in utilities, they are referred to as Supervisory Control and Data Acquisition (SCADA) protection systems [11]. Today's critical infrastructure incorporates cyber-physical components to monitor, control, and secure operations, data, and services. CPS also include automated processes controlled by relays, breakers, switches, and valves that facilitate daily operations. These systems rely on both human inputs and computational logic to manage the complexity of modern networks.

1.2.2 Role of attack graphs in modeling CPS risk

Threat modeling in cybersecurity systematically examines how an adversary might exploit vulnerabilities in a system. Traditionally rooted in computer science, threat modeling methods are now relevant to security challenges in CPS and critical infrastructures, especially in the context of performing ERM.

Attack trees, introduced by Schneier [72] in the late 1990s, provide a hierarchical approach to structuring threats. Attack trees provide a map of adversarial goals and entry points, highlighting how initial vulnerabilities could evolve into more significant security breaches. However, attack trees are limited by their strict hierarchical format, making it challenging to represent scenarios involving interdependencies, feedback loops, or simultaneous attack paths. Attack graphs, first proposed by Phillips and Swiler [62], are represented as directed acyclic graphs (DAGs) and allow for the modeling of complex relationships, parallel paths, feedback loops, and cross-domain propagation [38]. These features make attack graphs especially suited to CPS environments where cross-layer interactions are prevalent.

1.3 Dissertation contributions and outline

This dissertation advances the modeling and analysis of CPS security through a unified framework that captures the complex interdependencies across cyber, physical, and human domains. The main contributions of this work are:

1. *CPS risk modeling framework based on attack graphs:* A structured methodology is introduced that combines attack graphs, directed acyclic graphs (DAGs), and time-inhomogeneous discrete-time Markov chains (DTMCs) to model risk in CPS. This framework enables dynamic analysis of both adversarial and non-adversarial disruptions and supports multi-layered, time-dependent risk assessment.

2. *Application to a cyber-physical energy system (CPES):* The framework is applied to a small modular reactor (SMR) station, modeling layered attack paths mapped to the MITRE ATT&CK for ICS framework. This study develops new metrics for device-level criticality, incorporating system topology, functional dependencies, and attack progression logic.

3. *Application to a cyber-physical-human system (CPHS):* The framework is also applied to a vote-by-mail (VBM) election infrastructure system, demonstrating how layered attack and mitigation interactions can be modeled within a unified probabilistic framework. This case illustrates how process disruptions and security mitigations interact dynamically over time and highlights the need for system-specific performance metrics in CPS risk evaluation.

The remainder of this dissertation is organized as follows. Chapter 2 presents background literature on CPS security modeling and foundational definitions. Chapter 3 details the methodological framework, including the attack graph construction, DAG encoding, and DTMC implementation. Chapter 4 presents an application of the framework to CPES performing an SMR station risk analysis. Chapter 5 presents the method applied to CPHS performing a risk analysis of the voting-by-mail election system. Chapter 6 concludes with a discussion of frameworks limitations, contributions to CPS security planning, and opportunities for future research.

Chapter 2

Background

2.1 CPS risk modeling

Early approaches to securing critical infrastructure operated under the principle of "security through obscurity," in which systems were considered secure due to their isolation and limited connectivity [40]. This paradigm began to erode in the early 2000's as critical infrastructures adopted increasingly integrated cyber-physical systems (CPS) to optimize performance, improve efficiency, and enable remote monitoring and control. As interconnectivity increases, so does the potential for cascading failures and systemic vulnerabilities across sectors. This shift prompted a move toward "layered security," which assumes interdependencies across multiple domains and demands coordinated defense across cyber, physical, and operational layers.

Rinaldi, Peerenboom, and Kelly were among the first to formally characterize interdependencies in critical infrastructure by formally identifying cyber, physical, and organizational linkages as vulnerability pathways [65]. Building on this work, subsequent studies proposed multi-layered modeling techniques to better capture the functional, structural, and informational interdependencies of modern CPS [32, 64]. These works inform the layered modeling approach adopted in this dissertation.

Humayed et al. [33] published a security survey of CPS defined by four system types; industrial control systems, smart grid, medical devices, and smart cars, with each type having cyber, human, and physical components. Humayed et al. inventories associated threats along with threat types. The identified main threats to a CPS were determined to be caused by assumptions of network isolation, an excess of system access, and component heterogeneity. [33]

This trend of cyber-centric analysis continues in much of the CPS literature, including works like [96], which emphasize network-level protections and software vulnerabilities. Even in emerging research on cyber-physical-human systems (CPHS), such as Ganesan et al.'s machine learning-based analysis of human-in-the-loop systems, the human is modeled only as an input signal rather than as an autonomous agent with decision-making capabilities [21].

These limitations highlight a persistent challenge in CPS risk modeling: the tendency to focus on cyber vulnerabilities in isolation, while underrepresenting the physical oper-

ation and human layers and their interactions. This dissertation addresses this gap by advancing a modeling framework that explicitly integrates cyber, physical, and human components through a layered attack graph-based methodology.

Humayed et al. gives a general overview of several CPS, however it limits physical system interaction to automation without human intervention. Physical threats are defined as an adversary attaining access to computational CPS components (i.e. PLC, RTU, TPMS parts, or similar devices). [33] This becomes a reoccurring trend for CPS studies to focus primarily, if not exclusively, on the cyber and automation components of the system rather than the physical sensors and actuators. The assumption is that physical threats are assumed to only require physical security, a holdover of the 'silo' coverage impact of security through obscurity approach. In 2020, this trend continues with Yaacoub et al. publishing an article on CPS security limitations. In the article Yaacoub et al. describes CPS security as having a hierarchical structure of several layers, inventories existing security measures and associated limitation, and provides a risk assessment method[96]. More recent, a study of CPS with "humans in the loop," (i.e. cyber-physical human system *CPHS*) looks to model and predict the functionality of a CPHS through machine learning techniques, however the human is not an agent it is more of an input as the model is only learning from human interaction. [21] Prior to this work, CPHS analysis focusing on the cyber functions of the CPS.

In this chapter, we review the current best practices of CPS risk analysis from a control systems perspective and the limitations of this approach, which is often used in computer science and electrical engineering papers. Next, we review the operational risk analysis methods operations researchers use to model CPS risk with attack graphs and system models. We then compare static versus dynamic risk modeling, and the chapter will conclude with a summary of the gaps in risk modeling CPS in critical infrastructures.

2.2 Conventional and control-oriented approaches to CPS risk modeling

Since the early 2000s, there has been increased interest in the protection of CPS as they make up many of utilities necessary for modern life. These studies largely utilize cybersecurity tools for analysis. Many of the studies into cyber-physical systems build from the core attack graph cybersecurity model to incorporate physical attacks. Cheh et al. [10] studies the use of "meta model" attack graph known as "ADversary VIew Security Evaluation (ADVISE)" to evaluate the cyber-physical security of a railway station. Zheng et al. [106] studies the selection of mitigations to manage cybersecurity risks in resource constrained systems using attack paths. The dependencies of the mitigations are modeled using multiple choice constraints directly and coverage models indirectly. Zheng and Albert [105] build on this research to study temporal issues in infrastructure protection. They introduce a bi-level network interdiction model to study how to maximally delay the total weighted attack times of multiple adversarial, however, they do not consider the timing of security mitigations. In fact, we found no studies that consider the timing of the attacks or the security mitigation dependencies. Garcia et al. provides a review

of HiLCPS literature and points out there exist no predictive CPS model considering non-autonomous human actions [22]. This knowledge gap is addressed through the investigation of vote-by-mail processes, a CPS comprised of multiple human and automatic agents with temporal constraints.

2.3 Modeling CPS using attack graphs

Threat modeling in cybersecurity involves identifying and analyzing how adversaries might compromise a system and the potential consequences of those compromises. Traditional threat modeling methods, such as STRIDE [82], categorize threats such as spoofing or denial of service, providing checklists for vulnerability identification.

However, as systems grow more interconnected, especially in cyber-physical and industrial contexts, modeling isolated threats is no longer sufficient. Analysts need to understand how individual attacks can combine, propagate, and evolve to reach critical system goals. Attack graphs represent sequences of attacker actions and dependencies between them. They model not just whether an attack exists, but how it could progress through a system over time. By visualizing attack logic and reachability, attack graphs support both qualitative reasoning and quantitative risk analysis. This modeling approach has gained traction as a way to systematically evaluate cybersecurity risks in complex, interconnected systems [3].

Recent studies have explored the application of automatically defining attack surfaces to enhance cybersecurity in CPS. Paridari et al. [61] utilize the definition of attack surfaces to reflect any cyber attack on an industrial control system. They introduce a new approach for CPS security modeling through automatic attack detection to inform system reconfiguration. Though the physical components of the CPS are referenced, threats are considered to be cyberattacks affecting physical equipment. Similarly, [104] present a security model that considers performance of the physical part of the system. Protection of the CPS physical components are sampled during "normal operation" and this is compared to the real-time operation that informs the attack surface model. These papers both define attack surfaces as the points of ingress for the CPS that directly translates to the terminal attack nodes of an attack graph.

2.3.1 Generating attack graphs

Attack graphs are vital tools for identifying vulnerabilities in CPS, however, creating attack graphs requires substantial effort. Attack graphs can be created automatically using algorithms or manually with subject matter experts (SMEs). All attack graphs require detailed information about the CPS and its functionality. The use of attack graphs to model CPS have been explored by Few et al., who take a high-level overview of how such a system can be implemented.

In some applications, attack graphs are obtained through automatic attack graph generation programs [24, 77]. Automatically generated attack graphs can be created quickly; however, they may face limitations in scalability and robustness [20]. Automated tools typically use algorithms to scan and analyze configurations, assets, and known vulnera-

bilities within a network, and these resources are not available for all application. These tools can generate attack graphs by leveraging databases of known vulnerabilities, such as the Common Vulnerabilities and Exposures (CVE) database, and applying machine learning techniques to predict potential attack paths [41].

Attack graphs can also be created by sourcing subject matter experts for the systems involved, including red teaming (attacker perspective) and purple (attack-defense) teams. Human expert-generated attack graphs usually take a much longer time to construct and can have gaps in information if sufficient background is not documented. Nevertheless, the depth of insight SMEs provide ensures that the attack graphs are highly accurate and tailored to the specific system being analyzed [5].

The size of attack graphs can vary significantly depending on the complexity of the system being modeled and the level of detail included. Generally, attack graphs can range from a few nodes to thousands, depending on the scope and scale of the network or system as well as how it was generated. Attack graphs generated with SMEs tend to be smaller comparative to automatically generated models.

2.4 Limitations of Existing CPS Risk Models and Research Gaps

Attack graphs and their variants provide a structured way to represent adversarial behavior in cyber-physical systems; however, most existing implementations suffer from limitations that constrain their applicability in enterprise-scale CPS environments. This section identifies three critical gaps in the current literature and practice that motivate the modeling framework of this research.

Lack of Dynamic and Process-Aware Models Most attack graph literature adopts a static view of system vulnerabilities, often treating attack progression as a purely topological traversal problem. In earlier research during my doctoral studies Haseltine and Roald [28] conducted a small review of wildfire risk from power system arcs associated with automatic reclosing. In this study, we see the implications of time. However, it does not take into consideration any process components related to the system or external vulnerabilities other than transient faults. These models fail to incorporate operational dynamics or process-aware logic, which are essential factors in CPS, where time, control flow, and system workload influence risk exposure. As noted by Enoch, conventional attack graphs are insufficient for modeling dynamic systems such as cloud infrastructure or energy systems with temporal constraints. CPS infrastructures require models representing how threats evolve during continuous operations, how mitigations interact with time-sensitive processes, and how cascading effects can emerge from small perturbations.

Limitations of Boolean-Only Logic Representations Traditional attack graphs use Boolean logic to represent the preconditions for successful exploits, meaning a given step in an attack path is either "enabled" or "disabled." While useful for basic depen-

dency modeling, this logic cannot accommodate probabilistic state transitions, partial mitigation effectiveness, or variable system performance. Recent studies argue that this binary approach oversimplifies attacker behavior and underrepresents system complexity [49]. Evolving threat landscapes—especially in systems with human and physical components—necessitate richer logical and stochastic representations to capture uncertainty, failure modes, and temporal transitions.

Need for New Metrics from Layered Risk Models Several recent studies propose decomposing CPS into functional layers to better capture interdependencies between physical processes, control logic, and communication networks [108, 60]. These layered approaches allow for the derivation of system-specific risk metrics, such as Support Subgraph Participation (SSP) and Load Support Share (LSS), which provide more granular insights into node criticality and system resilience. However, these works remain limited to specific domains (e.g., power systems) and do not generalize to broader CPS architectures. There remains a need to develop flexible, layered modeling approaches that can extract interpretable and actionable metrics across multiple CPS contexts.

Looking forward Together, these gaps underscore the need for a CPS risk modeling framework that (1) captures process dynamics and operational flow, (2) employs probabilistic and temporal logic beyond static Boolean gates, and (3) supports the derivation of system-specific performance metrics to inform enterprise-level decision-making. The framework introduced in the next chapter addresses these challenges by integrating layered attack graphs with time-inhomogeneous discrete-time Markov chains (DTMCs), offering a structured and adaptable approach to analyzing risk in complex, evolving CPS environments.

Chapter 3

Methodology

3.1 Overview and Modeling Framework

Recent studies and national standards on cyber-physical systems (CPS) security, including those from NIST [25] and other thought leaders [96], emphasize the need for new approaches to risk modeling that account for the unique properties of CPS—namely their integration of cyber, physical, and operational dynamics. Traditional security analysis methods often fall short in these contexts due to their static assumptions and limited ability to model adversarial behavior over time. This thesis introduces a novel modeling approach for CPS security risk assessment based on three foundational components: (1) the normal operation process, (2) the space of adversarial attacks, and (3) the available mitigations or recourse actions. Together, these elements form the basis of a layered attack graph modeling framework, extended with time-inhomogeneous discrete-time Markov chains (DTMCs) to support systems with dynamic or policy-sensitive behaviors.

This methodological approach has evolved through a series of studies and public presentations. Early work focused on the limitations of traditional attack graphs in representing vulnerabilities in election infrastructure, which we presented in a collaborative poster at the ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO), where it was awarded the Best Poster in 2021 [29]. Subsequent research revealed functional layers are needed to represent the operational levels of a CPS. HaseltineAlbert2025Tutorial define the functional layers of the CPS in a tutorial.

3.2 Structure and properties of a CPS

Cyber physical systems are those systems which exist in the physical world however are governed by control systems and monitoring networks. Figure 3.1 shows the physical part of the CPS could be a plant, process, or network [101]. In chapter 2, I explain that much of the literature related to CPS security and analysis focuses on the 'Cyber' components.

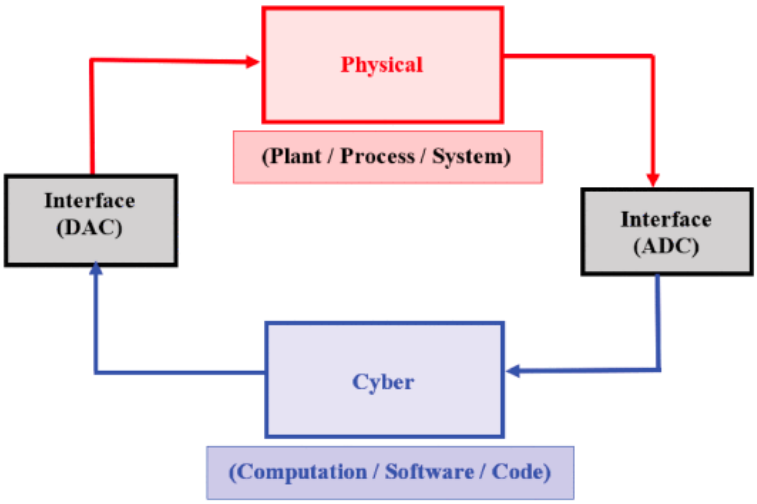

Figure 3.1: CPS Properties

3.2.1 Functional layered model of CPS

The National Institute of Standards and Technology (NIST) defines a CPS as an interaction of digital, analog, physical, and human components engineered to function through integrated physics and logic [56]. Under this definition, a CPS is a blend of these elements designed to work together seamlessly. CPS play a crucial role in critical infrastructure and essential services. Currently, CPS are integral to most critical infrastructures, including healthcare, transportation, energy, and commerce.

We present a general framework to consider CPS functional layers. A layered approach to risk modeling helps quantify threats at different operational levels and assess how failures propagate across the system. For specific applications there can be a need to segment the layers further. However, it is important to consider that the functionality of CPS is more that two (cyber and physical). We consider analyzing each layer—physical, network, control, and application—individually and in relation to one another to understand potential points of failure and their impacts. Since it is impossible to fully protect a system, adopting a systems-level perspective allows one to mitigate when failures occur to enhance resilience.

CPS systems comprise various devices and technologies, making it difficult to model [102]. Challenges include the heterogeneity of components, the complexity of interactions, and the dynamic nature of these systems. A layered approach to modeling CPS is used to better capture CPS vulnerabilities, modeling them in singular function layers to aid in risk modeling. We adopt the convention by [4] that segments layers of a CPS into general categories: (1) applications layer, (2) network transmission layer, and (3) physical perception layer. We briefly describe each of these layers.

1. **Physical Perception Layer**: The physical layer represents parts that interact with the physical world. This includes components such as sensors collect data (i.e., temperature, pressure, motion) to collect information, and actuators to perform actions on the physical system (i.e., operate valves, switches, arms).

2. **Network Transmission Layer**: CPS frequently requires interaction among distributed elements, necessitating data transfer across networks. These networks may be wired or wireless, employing technologies like Ethernet, Wi-Fi, fiber optics, or specific industrial protocols (e.g., Modbus, PROFIBUS, DNP3). The network/transmission layer facilitates dependable and prompt data communication among sensors, controllers, and actuators, supporting real-time monitoring and control.

3. **Application Layer**: The application layer manages command execution and control logic for the CPS [108]. It processes sensor data from the physical perception layer to facilitate real-time decision-making. This layer includes the software and algorithms that outline the system's capabilities, including control strategies, optimization routines, and specific applications tailored to the domain.

There are linkages between the CPS layers that capture interdependencies. For example, sensor data from the physical perception layer influence decisions made in the application layer, while control commands from the application layer impact the physical perception layer.

A network's control logic can have a greater influence on a CPS's vulnerabilities than the type of components used. Control logic refers to the algorithms and decision-making processes that govern the interactions between CPS layers. It determines how data is processed, how actions are executed, and how the system responds to various inputs. Control logic ensures the effective operation, security, and optimization of cyber-physical systems. It ensures that the various components of the system work together harmoniously, respond to real-time changes, and maintain resilience against threats and disruptions.

An example of control logic is a water treatment facility's ICS. In such a facility, the programmable logic controller (PLC) regulates chemical dosing pumps, monitors water levels in storage tanks, and adjusts valves to ensure proper flow between treatment stages. Turbidity, pH, and chlorine levels are the sensory data collected in real time and fed into control logic routines set into the PLC. These routines execute automatic responses. For example, if the pH falls below a safe threshold, the control logic increases the lime dosing rate to neutralize acidity. SCADA control terminals provide operators with dashboards to monitor these values and override commands when necessary.

In a threat modeling context, this environment presents several places where adversarial attacks could be initiated: adversaries may exploit networked PLC and relays, manipulate sensor data to cause misoperation, or interfere with control logic to trigger unsafe states. Each of these potential exploits can be represented as steps in an attack sequence that models how an attacker could escalate from network access to physical system damage.

3.3 Layered Attack Graph Modeling Framework

The complexity of CPS requires a layered modeling approach to segment and organize the attack surface of the system for holistic risk assessment. The core modeling framework

introduced in this thesis uses layered attack graphs to represent security risks across different domains of a cyber-physical system. This section details the components and structural logic of these graphs, including how nodes, edges, and layers are defined and interact.

3.3.1 Attack graph components and structures

Next, we introduce terminology common to attack graph risk modeling that is used throughout this tutorial. We frame threat modeling concepts hierarchically, from individual elements to system-wide models:

Attack Nodes An attack node represents a discrete condition task that an adversary must complete within the CPS to progress toward their goal, such as gaining initial network access or manipulating control logic. There are three types of attack nodes we consider in attack graphs:

1. *Terminal Nodes*: Represent the starting point of an attack, often where an attacker gains initial access.
2. *Intermediate Nodes*: Represent intermediate states or steps in the attack path.
3. *Goal Node*: Represent the attacker's ultimate objective, such as gaining administrative access or extracting sensitive data.

Edges Edges represent the transitions between nodes, indicating the actions or exploits that an attacker can perform to move from one state to another. These transitions are often labeled with the specific exploit or attack technique used.

Attack Paths An attack path is an ordered sequence of nodes capturing how an adversary progresses through a system. Transitions between nodes (edges) may be annotated with exploit descriptions, probabilities, or logical conditions (e.g. AND/OR).

Attack Graphs An attack graph integrates multiple attack paths into a full-state model of adversarial behavior across a system. Attack graphs capture complex relationships, parallel paths, and dependencies inherent to CPS environments, making them particularly valuable for analyzing and mitigating risks in interconnected and dynamic systems.

Attack Surfaces The attack surface encompasses all entry points (terminal nodes) into a system that adversaries could potentially exploit. Attack graphs clearly identify these entry points, helping analysts prioritize defense measures and systematically mitigate vulnerabilities to reduce overall system risk.

Probabilities Some attack graphs incorporate probabilities to represent the likelihood of successful exploits. This helps quantify the risk associated with different attack paths and prioritize mitigation efforts based on the probability of occurrence.

As an example, consider an attack graph V with three attack nodes $\{v_1, v_2, v_3\}$, where:

- v_1: "Attacker has prior knowledge of SCADA RTU protocols,"
- v_2: "Attacker exploits a vulnerability via remote terminal,'
- v_3: "Attacker modifies control logic of critical component."

Node v_1 is the terminal node and is on the attack surface, v_2 is an intermediate node, and v_3 is the goal node. There are edges between v_1 and v_2 and between v_2 and v_3. This example has a single path.

Boolean logic A crucial element of capturing the interdependencies of CPS functional layers is understanding the role of Boolean logic. Attack graphs are structured representations of how adversaries can compromise a system. These attack graphs rely on Boolean logic to model the dependencies and sequences that define successful attack paths. Boolean operators such as AND and OR capture how individual attack steps must combine conjunctively or disjunctively to enable further progression through the graph. This logical foundation provides a natural way to reason about the conditions under which an attacker can achieve specific objectives. In this section, we focus on the role of Boolean logic in structuring attack graphs and refer the reader to formal logic frameworks such as Computational Tree Logic (CTL) for a deeper mathematical treatment [43, 73].

A central focus of an attack graph is the goal node, which represents the ultimate target of the adversary [98]. Attack graphs can be interpreted as logic paths that begin at terminal attack nodes and proceed toward the goal node. Along these paths, Boolean logic gates—AND and OR—define the required conditions for the attack to propagate from one exploit to the next [74]. In this way, the graph encodes the alternative and conjunctive sequences by which an attacker might achieve their objective. In the context of CPS, Boolean logic provides the foundation for modeling how dependencies between components (e.g., relays, communication links, operator actions) influence the feasibility and progression of an attack. In the next chapter, the attack graphs is extended into a layered framework using DAG representations, which are further integrated into a dynamic stochastic process via a time-inhomogeneous DTMC.

Boolean logic gates operate on binary values: low (0) and high (1). Within an attack graph, a node set to high (1) indicates that the attacker has successfully completed the associated task. Conversely, a low value (0) indicates that the step has not yet been completed. The default state of all nodes in a secure system should be low, since no attacks are assumed to be initially in progress.

- **AND-gate:** The output Y is high (1) only if *all* input nodes are high. This models a situation where multiple conditions must be satisfied before the next attack step can proceed.

- **OR-gate:** The output Y is high (1) if *any* of the input nodes are high. This models situations where multiple paths can independently enable the next stage of attack.

Thus, a gate output of $Y = 1$ confirms that the attacker has successfully transitioned past that gate in the attack graph. Figure 3.2 illustrates the input-output behavior of the AND and OR gates. The left-hand truth table shows that for the AND-gate, $Y = 1$ only when both $X_1 = 1$ and $X_2 = 1$. The right-hand truth table confirms that for the OR-gate, $Y = 1$ as long as either $X_1 = 1$ or $X_2 = 1$.

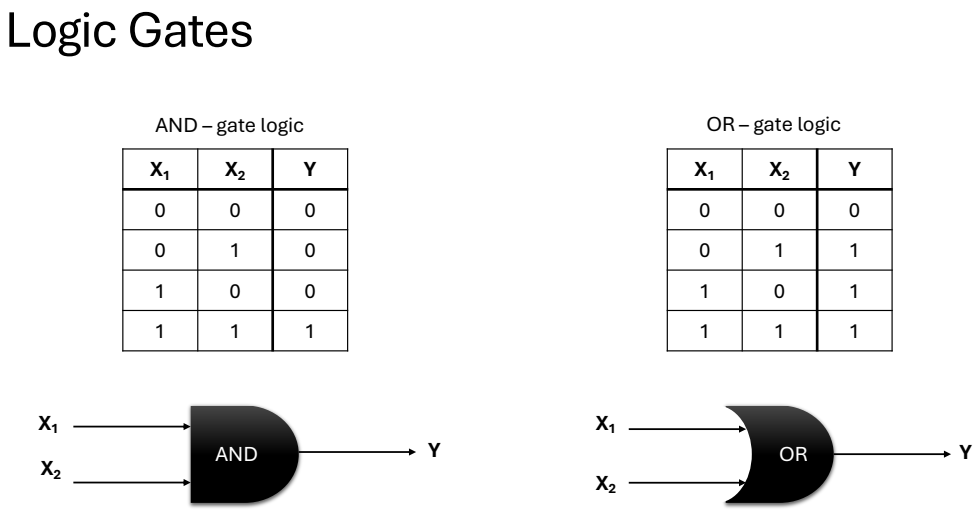

Figure 3.2: Boolean logic operators

Figure 3.3 illustrates an example of an attack graph with its Boolean logic represented using symbols connecting attack nodes. Both nodes X_1 and X_2 must be set high (1) to set Y_1 high (1), i.e.., the attacker must complete both $X_1 = 1$ and $X_2 = 1$ for Y_1 to activate. Similarly, there is a second AND-gate connecting X_3 and X_4 to node Y_2. The goal node has an OR-gate, indicating that an adversary can reach the goal with either Y_1 or Y_2 being set high (1), denoted by the OR-gate between $Y_1 \rightarrow Goal$ and $Y_2 \rightarrow Goal$. Therefore, there are two attack paths in the attack graph.

The *attack surface* represents the set of points on the boundary of a system where attackers can attempt to enter, exploit, or extract data [57]. It comprises all terminal nodes in the attack graph, capturing potential entry points for attackers and frequently utilized in cybersecurity for risk quantification. A smaller attack surface generally indicates a lower difficulty in protecting a system, helping us understand the "points of ingress" that may require more protection. In the attack graph illustrated in Figure 3.3, the attack surface is composed of X_1, X_2, X_3, X_4.

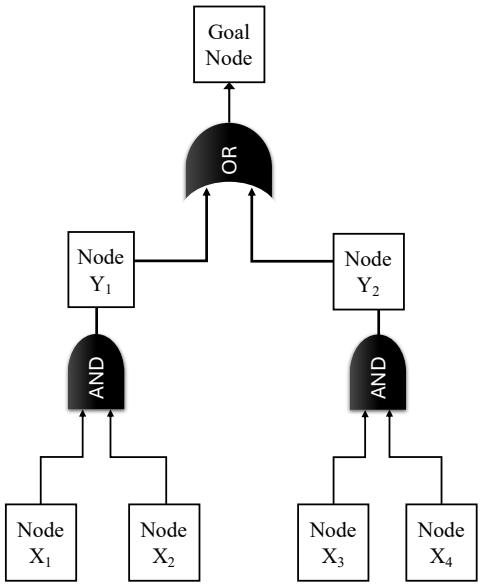

Figure 3.3: Attack graph logic operator representations

3.3.2 Threat propagation in a CPS using attack graphs

Threat modeling in cybersecurity involves identifying and analyzing how adversaries might compromise a system and the potential consequences of those compromises. Traditional threat modeling methods, such as STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege), categorize threats such as spoofing or denial of service (DoS), providing checklists for vulnerability identification [82].

Consider an attack-defense graph for a communications network illustrated in Figure 3.4 [87]. The goal node—"Block communication" for a network—is at the top of the figure. The red nodes denote attack conditions, and the green nodes represent mitigations. The mitigation tasks cover two attack nodes. Recall from Section **??** that we read the graph from the bottom points of ingress (terminal attack nodes) through several paths to the top goal to block communication [87]. The Radio-Frequency Identification (RFID) attack graph shows that some attack paths have multiple dependencies that can impact the effectiveness of mitigations. In this example, we assume that the mitigations completely counteract the terminal attack nodes that make up the attack surface of the attack graph. Therefore, the mitigations in this example reduce the attack surface from seven ingress nodes to three. The two nodes mitigated include the attack nodes directly blocked by mitigations (i.e., connected to green blocks in Figure 3.4. The two attack nodes associated with "Faraday Cage" logically depend upon on the attack node "Be in Vicinity of Tag" to progress toward the goal node. This dependency is reflected as the logical AND (arc connecting attack nodes), by mitigating "Be in Vicinity of Tag" we inhibit the attack paths associated with "Shield Tag" from asserting. However, the mitigation of "Secure Warehouse" does not affect any other attack path. The logical AND dependence of attack nodes correlates to the effectiveness of mitigations.

Attack graphs are valuable tools for modeling cybersecurity threats and CPS vulnera-

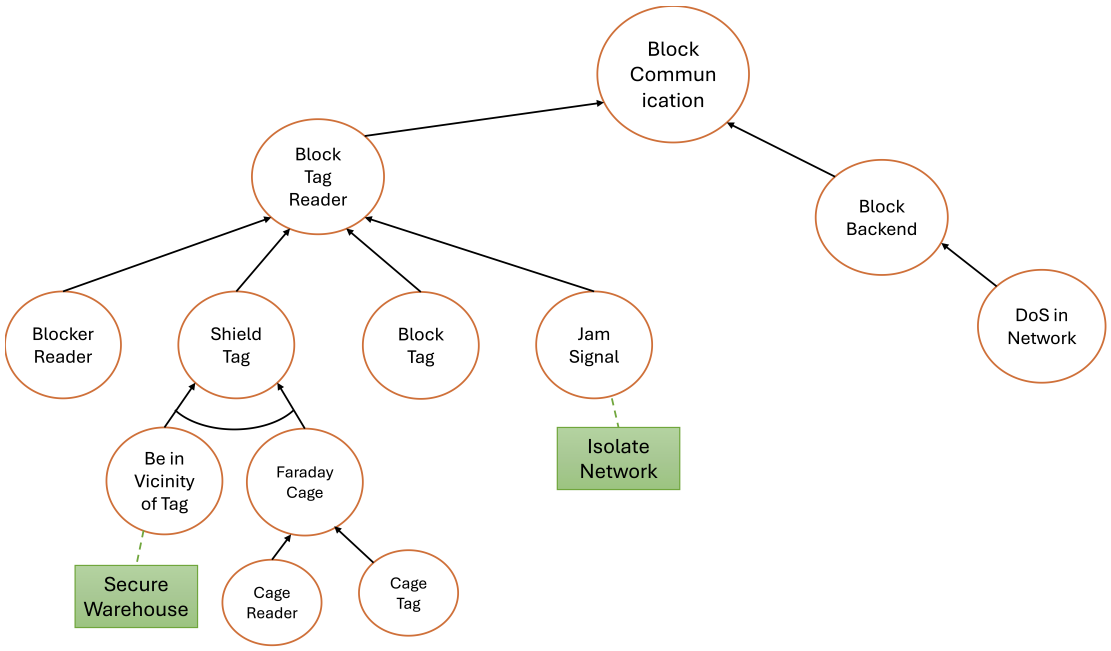

Figure 3.4: Example of RFID Attack graph. Source: [87]

bilities. However, to effectively mitigate known system risks, we must carefully integrate security controls with attack graphs. Security controls are mitigative actions taken as preventive or reactive measures to protect systems from specific threats. Using attack-defense graphs to set these controls allows organizations to assess their defenses and prioritize resource allocation.

Various methods have been employed to generate and analyze attack-defense graphs and to integrate security controls effectively. One notable approach is by [39] who model the functional interdependencies of CPS using graphical techniques. The result is a directed acyclic graph (DAG) with attacks modeled as compromised functions/nodes in the graph. This approach allows for the iterative identification of compromised nodes and enables rapid adjustments to defenses in response to intrusions. The computational experiments demonstrate the model's effectiveness in addressing real-time cyber threats.

However, as systems grow more interconnected, especially in cyber-physical and industrial contexts, modeling isolated threats is no longer sufficient. Analysts need to understand how individual attacks can combine, propagate, and evolve to reach critical system goals. Attack graphs represent sequences of attacker actions and dependencies between them. They model not just whether an attack exists, but how it could progress through a system over time. By visualizing attack logic and reachability, attack graphs support both qualitative reasoning and quantitative risk analysis.

3.4 Time-Based View of Attack Graphs Using a Markov Chain Model

Although attack graphs and attack–defense trees are widely used in cybersecurity modeling, they are typically Boolean in nature and limited to static representations of system vulnerability. Even probabilistic extensions of these models, which estimate the likelihood of an attacker progressing through a given path, fall short in capturing the temporal dynamics critical to cyber-physical energy systems (CPES) or cyber-physical human systems (CPHS). In these complex systems, risk is not only a function of topology or vulnerability overlap, but also of timing, sequencing, and operational duration. Markov chains provide a principled way to introduce time into the analysis, enabling stochastic modeling of how adversarial influence may evolve over system states. This time-based view is essential for capturing delayed effects, persistent exposure windows, and the probabilistic unfolding of cascading failures—factors that static attack graphs alone cannot adequately represent.

In Chapter 5, a discrete-time Markov chain (DTMC) model is developed to capture the evolution of a cyber-physical process in the context of vote-by-mail (VBM) operations. The DTMC integrates physical process transitions, adversarial interventions, and mitigation effects into a unified probabilistic framework. Each system state encodes a snapshot of the ballot process, and transitions represent either operational progress or security events. The DTMC structure leverages the memoryless property to model this progression: the probability of transitioning to a future state depends only on the current state, not on the full historical path. This assumption is particularly well suited to processes like VBM, where delays, disruptions, and recoveries occur in well-defined stages, and the focus is on forward progression rather than path tracing.

The DTMC model allows us to compute not only the probability of successful system operation, but also the expected time to failure, the likelihood of recovery, and the relative impact of various mitigations over time. The model captures adversarial persistence, control delays, and the time-varying nature of risk exposure in ways that static attack graphs cannot. While attack graphs provide the necessary topological and logical foundation, embedding them into a time-indexed stochastic model enables a far more realistic analysis of risk in cyber-physical and cyber-human systems.

Below is a brief overview of the discrete-time Markov chain framework used in this thesis. Given a random process V_t with n finite states, the one-step transition probability of the process moving from state i in time step t to state j in a single time step is

$$P_t(i,j) = P(V_{t+1} = j | V_t = i) \quad \forall i, j \in \{1, \cdots, n\}. \tag{3.1}$$

We define a transition probability matrix P_t for each time step ($t = 1, 2, \cdots, T-1$) to determine the state of the system at the end of each time step [76].

Figure 3.5 illustrates the three core dimensions that must be jointly represented in a comprehensive risk model of a CPS. On the left, the operational process layer captures the sequence of activities and normal system states. On the right, the attack surface is depicted as a collection of adversarial entry points—each represented as a node—linked by directional edges to the physical and logical processes they can impact. These attack nodes reflect potential disruptions stemming from cyber, physical, or human-originated

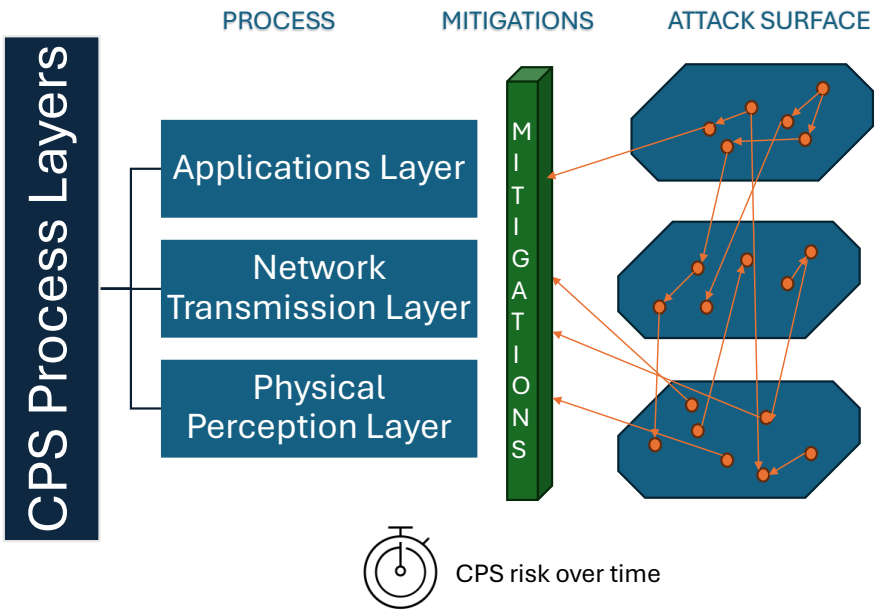

Figure 3.5: Visualization of the CPS risk modeling framework

threats. Centered between these layers is the mitigation layer, which represents recourse actions that can block or alter adversarial pathways before system damage occurs. Notably, mitigations can be interdependent and conditional, meaning their effectiveness may rely on co-deployment or sequential activation. All three layers evolve over time, necessitating a dynamic framework that models not only state transitions within each layer, but also the temporal interactions across them.

3.5 Risk Analysis Outputs and Interpretation

By integrating normal operating conditions, known vulnerabilities, and regulatory protections into a unified framework, this risk model provides a more comprehensive view of cyber-physical system (CPS) risk than traditional attack graphs alone. The core objective is not merely to identify potential attack paths, but to evaluate how system design, maintenance activities, and mitigation strategies interact over time to influence the likelihood and impact of compromise.

The outputs of this framework enable both qualitative and quantitative assessments. On the quantitative side, the model provides scenario-specific probabilities of failure, residual risk estimates for each attack node, and comparative rankings of mitigation effectiveness. These results highlight which components are most susceptible under real-world operational conditions, particularly in scenarios that involve overlapping tasks between legitimate maintenance and adversarial behavior.

On the qualitative side, the model offers structural insights into risk propagation patterns, high-risk/low-coverage nodes, and the temporal alignment of threats with routine system functions. Together, these outputs support security planning by identifying critical weaknesses, prioritizing mitigations, and informing operating procedures that reduce

exposure during sensitive tasks. By combining static structural modeling with scenario-based activation and time-aware reasoning, the framework advances CPS risk analysis toward actionable, context-aware security planning.

Chapter 4

Risk modeling for cyber-physical energy systems: SMR station application

4.1 Introduction

The modern power grid has evolved into a cyber-physical energy system (CPES). The incorporation of microprocessor relays, digital communications for Supervisory Control and Data Acquisition (SCADA) protocols and optimal power flow supervisory dispatch have created a multi-layered system. Research on CPES and risk modeling utilizes multi-layered strategies for identifying system vulnerabilities, since the combination of cyber and physical components increases the complexity and potential attack surfaces of these systems [108]. Several studies demonstrate the use of C-ID to model and analyze a CPS for increasing the resilience and security of these complex systems [75]. Others highlight the need for CPES systems to have an integrated framework that captures the interdependencies that exist between CPES components [100].

The integration of SMR stations will alter the present regulatory and security needs of the power system. The current North American Bulk Electric System (BES) relies heavily on centralized, utility-scale power plants ranging from 100 MW to 1,000 MW. The integration of SMRs as distributed energy resources would mark a fundamental shift in grid architecture, offering operational benefits but introducing new cyber-physical vulnerabilities [103]. These vulnerabilities include increased exposure to cyberattacks, tighter coupling between cyber and physical layers, and the potential for cascading disruptions due to system interdependencies. Without proactive risk modeling, such vulnerabilities may lead to energy supply interruptions, equipment damage, or broader grid instability [36]. Despite the advantages of a centralized control scheme for managing microreactor fleets, the interconnected nature of these systems makes them vulnerable to cybersecurity risks. Threats such as lateral movement across networks, digital exploitation of vulnerabilities, and wireless-based attacks necessitate a re-evaluation of existing security guidelines.

Modern CPES, including emerging Small Modular Reactor (SMR) installations, pose

new challenges for risk modeling. Traditional methods such as Probabilistic Risk Assessment (PRA) and fault trees remain the dominant tools in both commercial nuclear safety analysis and regulatory guidance. These methods are well-suited to quantifying the likelihood of core damage or physical failure but offer limited insight into multi-layered cyber-physical interdependencies or adversarial behavior.

SMRs are an emerging nuclear energy technology offering the potential to reduce industrial loads on the electric grid to improve frequency stability and power quality. These compact nuclear facilities provide reliable energy, especially in remote areas. However, integrating SMRs into the bulk electric system introduces new cyber-physical vulnerabilities, including increased exposure to cyberattacks and potential cascading disruptions. This chapter introduces a Cyber-Informed Design framework tailored to CPES that function as SMR stations. The framework employs attack trees to model threat scenarios and identify vulnerabilities across four functional layers of SMR stations. We apply the threat modeling approach to a reference architecture and map the attack paths to the MITRE ATT&CK for Industrial Control Systems to provide insights into improving the resilience of cyber-physical energy systems.

The chapter is organized as follows: Section 4.2 reviews prior work on CPES risk modeling and highlights the gap in adversarial CPS analysis for SMR applications. Section 4.3 describes the architecture and attack surface of the reference SMR station. Section 4.4 presents the layered attack graph formulation. Section 4.5 formalizes the risk quantification model using scenario-triggered activation, path-based logic, and mitigation mapping. Section 4.6 summarizes findings from the COINS paper case study. Sections **??** and 4.7 conclude with mitigation evaluation and transition toward dynamic CPS risk modeling.

4.2 Background

Small Modular Reactor stations represent a significant evolution in nuclear power system design, offering modularity, scalability, and potential economic advantages over conventional large-scale reactors. However, the integration of SMRs into modern power grids introduces a new class of cyber-physical interdependencies that existing risk frameworks are ill-equipped to model. Unlike traditional nuclear facilities, SMR stations are increasingly integrated within networked control environments, leveraging digital instrumentation, remote access, and smart grid connectivity. These features position SMRs within the broader class of Cyber-Physical Energy Systems (CPES), where vulnerabilities emerge not only from component-level failures but also from interactions across cyber, control, and physical layers. In this context, we need a reassessment of risk modeling strategies that move beyond classical reliability models to account for layered architectures and adversarial threats. The following sections examine how current methods address this complexity, where they fall short, and how a dual-layered DAG approach may address these gaps.

4.2.1 Existing approach to CPES risk modeling

Probabilistic Risk Assessment (PRA) is commonly used to to estimate the probability of reactor core damage [42]. This is a comparative analysis performed between traditional large capacity reactors versus smaller versions such as the small modular type reactor sites. Dennis et al. [17] expand the PRA framework to model shared dependencies across the physical components of the SMR station creating a multi-module PRA (MMPRA). In 2021 Zhou et al. [107] perform a survey of several MMPRA approaches and conclude that dynamic models are preferable to capture evolving risk in complex systems.

4.2.2 Limitations of PRA and alternative approaches

No existing MMPRA models considers the communications and controls of a CPES – SCADA systems, or protection and control. Cox [13] expresses the limitations of PRA based analysis methods for complex systems. For this reason, we consider less conventional methods for risk modeling of complex systems such as CPES.

Directed acyclic attack graphs(DAG) have been used as a means to create two new risk metrics associated with CPES outside of conventional PRA analysis. Palomino and Zhang [59, 60] use this method to model the digital communication and controls separately from the power load flow of the system, capturing an interdependent layered approach to modeling CPES. The two layered approach is a significant step towards capturing the functional layers of a CPES.

Alderson, Brown, and Carlyle, Cox [2, 13] critique the DHS-style probabilistic terrorism risk models and points out the inability of these type frameworks to realistically model adaptive adversaries or structural interdependence such as those seen in CPES between the physical components and SCADA. Haimes emphasizes that cyber-physical infrastructures like SMRs are best modeled as interdependent systems-of-systems (SoS), requiring risk modeling to account for emergent behaviors, shared states, and cascading effects. Backing the use case for a model that considers CPES as a multiple layer system with various vulnerabilities and threat paths associated.

4.2.3 Adversarial CPS risk propogation models

There have been several PRA based approaches to risk propagation in CPS. Zhou et al., Dennis et al. [107, 17] emphasize the importance of modeling multi-stage failure propagation and human-system interactions in multi-module SMR accident scenarios. These complexities become more pronounced under intentional adversarial actions, where attacker adaptation can drive non-obvious dependencies. In response to such challenges, the National Academies recommend augmenting traditional probabilistic risk assessment with attacker-defender game theory and Bayesian networks to account for adaptive threat behaviors and system interdependencies [48]. However, Bayesian network-based models face limitations in the SMR context. Their robustness is hindered by structural sensitivity and limited observability, particularly when prior probabilities and conditional dependencies are not grounded in empirical incident data [95]. The need for historical data is a critical shortcoming for emerging technologies like SMR stations, where histor-

ical adversarial data is scarce or non-existent. Moreover, traceability of causal pathways in large-scale Bayesian models can be opaque, making model verification and validation challenging in operational or regulatory environments.

Recent studies demonstrate the usefulness of threat modeling in identifying and managing cybersecurity risks in CPES [1]. Threat modeling is a process of identifying and analyzing potential attacks. Attack models such as attack trees and attack graphs are structured approaches to model the steps required to successfully carry out an attack [63]. In these models, the root node represents the goal of an attack, nodes represent vulnerabilities that are exploited in an attack (e.g., a step that an attacker may take to achieve their goal), and edges represent relationships between different actions. Boolean logic models the relationship between attack nodes. By mapping out these potential attack paths, organizations can better understand their security posture, identify critical vulnerabilities, and develop effective mitigation strategies to protect against cyber threats. There are various taxonomies for threat modeling, such as the MITRE ATT&CK for Industrial Control Systems (ICS) framework that categorizes adversary tactics and techniques specifically tailored to operational technology environments [54]. It organizes attack behaviors into high-level tactics and correlates each tactic with corresponding techniques observed in real-world ICS contexts. This structured representation enables analysts to model threat progression, evaluate system exposure, and align detection or mitigation efforts with standardized threat behavior, and it is widely adopted in ICS security for threat identification, system evaluation, and security validation [23]. We align attack paths with corresponding MITRE ATT&CK tactics and techniques to ensure ontological consistency and support traceable, evidence-based risk analysis.

4.2.4 Gaps and need for novel framework

The current SMR threat models consider physical risk and adversarial separately. Often focusing soley on the physical components such as MMPRA analysis ofZhou et al., Dennis et al. [107, 17].We see that the framework proposed in chapter 3 is a necessary consideration to account for all layers of functionality in a CPES.

By extending the CPES DAG framework to support (1) adversarial path modeling and (2) structural dependency assessment, we propose a dual-DAG approach tailored to high-consequence systems like SMRs. Using the proposed risk modeling framework enables the simultaneous quantification of attack likelihood and functional impact through the use of attack graphs.

4.3 Threat modeling framework for SMR stations

In this paper, we use a hierarchical terminology to distinguish between different levels of threat modeling granularity. An attack path refers to a single linear sequence of actions (i.e. attack nodes) that an adversary could follow to achieve a specific malicious goal. A collection of such paths, typically structured using Boolean logic (e.g., AND/OR dependencies), forms an attack tree, which models the possible strategies for compromising a particular functional layer of the system. Finally, we define an attack graph as the

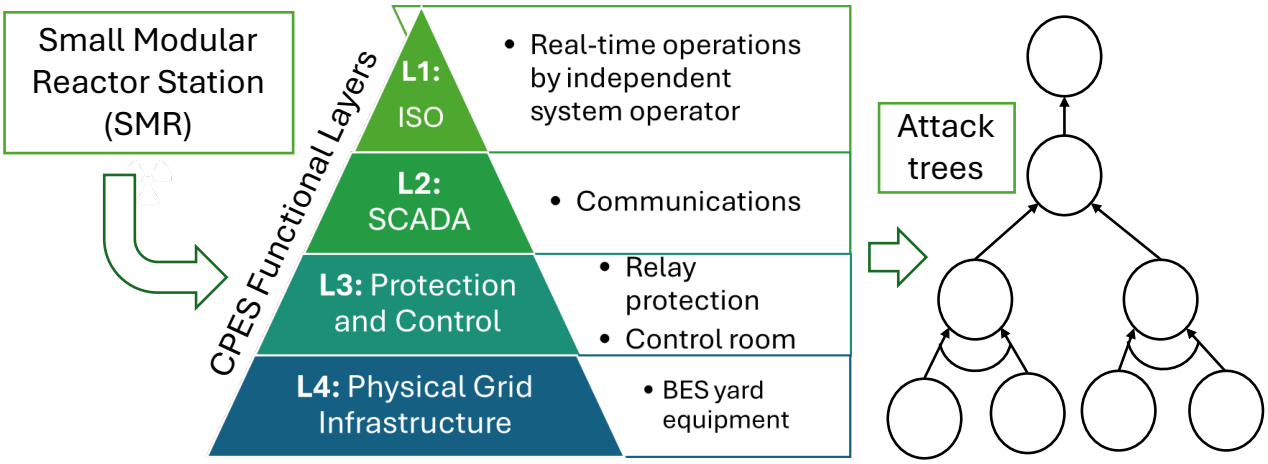

Figure 4.1: Overview of threal modeling framework

full model representing all attack trees across the functional layers of the SMR-station CPES. This layered attack graph captures cross-domain dependencies and provides a system-wide view of potential adversarial behavior. This hierarchical structure allows us to analyze both localized and system-level risks within the same framework. In the following sections, we introduce the modeling approach and the reference architecture used for layered attack graphs that reflect the unique architectural and operational features of SMR-based CPES.

4.3.1 Modeling CPES with layers

CPES are complex with multiple processes functioning simultaneously, which requires a high-level modeling approach. Attacker actions that pose a threat, such as those in an SMR station, can be modeled as comprising four interdependent layers as illustrated in Fig.4.1. The layers have a hierarchical relationship with the highest layer associated with BES oversight and the lowest layer associated with the physical grid components[46]. We review the functional differences between each layer and how attacks could propagate differently in each layer.

Application and Oversight Layer (L1): The Application Layer, L1, is the highest level of regulation, where the Independent Electric System Operator (ISO) monitors the BES. Real-time operators at the ISO continuously monitor the BES, ensuring compliance with the North American Reliability Corporation (NERC) technical standards and the Federal Energy Regulatory Commission (FERC) energy market rules. Attacks propagate in this layer by means of cyber-attacks or insider attacks due to the level of clearance needed to influence the SMR station at this layer.

Communications Layer (L2): The Communications Layer, L2, is where SCADA infrastructure resides. This layer contains operations and commands that are performed in ten to fifteen minutes or less based on the emergency rating of the equipment on the

BES. Metering and steady-state operation of the grid operate in this layer. The ISO on layer L1 communicates through layers L2 and L3 if commands need to be sent to the field. For example, suppose the ISO needs to de-energize a transmission line. In this case, a signal is sent from the real-time operator (RTO) to the local substation protection relays to open the circuit breakers at the associated substation. SCADA operations in the US must comply with NERC Critical Infrastructure Protection (CIP) standards. Typically, this layer requires the most coordination between the ISO and generator owner/operators.

Relay Protection and Control Layer (L3): Local to the substations is the Protection and Control Layer, L3, at terminal buses. The substation owner is usually responsible for safeguarding equipment during both planned and unplanned events, such as system faults or equipment failures. This protection is generally achieved through relays and actuators that operate circuit breakers within a few cycles (1 cycle = 1/60 of a second). Relay control logic serves as the primary protective measure for the power grid, ensuring that remote operation requests go through the relay protection scheme instead of directly to field equipment. The protection scheme connects to the BES through measuring transformers (current transformers, potential transformers) and other field sensors.

Physical Infrastructure Layer (L4): Lastly, the Physical Infrastructure Level, L4, consists of the physical copper lines and infrastructure of the BES. This includes autotransformers, switches, circuit breakers, and all physical structures needed to transport power. There are several interactions between the immediate layers. For example, the communications layer L2 interacts with both the applications layer L1 and the relay protection layer L3. The physical connections between layers L4 and L3 represent the measuring instruments (current and potential transformers) and physical control elements such as breakers, switches, and other actuators. Attacks propagate in layer L4 through physical interference at the SMR station by attackers. These attack paths generally require at least one task to be executed at the station and in the control room

4.3.2 SMR station reference architecture

We identify attack graphs and perform C-ID in reference to an SMR fleet reference model that provides operational context and ensure that threat modeling is grounded in realistic assumptions and capture cyber-physical dependencies in potential next-generation CPES. The supporting scenario for the SMR reference model is based on publicly available documentation on nuclear power plant design and corresponding safety and control systems. The content of these documents was further evaluated and expanded upon via several discussions with stakeholders who design, install, own, and operate nuclear power plants to ensure that research assumptions underlying the resultant reference model align with the current state of the practice.

The microreactor deployment scenario consists of a remote mining town of no more than 2000 residents. The facilities of significance in this town are residential housing, a hospital, a refinery, a mine, and the reactor housing. A facility containing several SMRs is located near the refinery to assist in refining heavy metals. Both the reactor housing

and refinery are located near the mine to reduce the transportation requirements to bring ore to the refinery. In addition, the electrical power and heat provided by this facility also may be used to support the region more broadly. Both the residential housing and the hospital are located several miles away to mitigate potential health issues among residents. An electrical power substation nearby distributes electricity generated by the SMR to the refinery and surrounding neighborhood. The reference model is illustrated in Fig. 4.2. The reference model is a single-line diagram of the SMR station, reflecting the major equipment that will be installed. The single-line diagram illustrates a substation configuration featuring a single-bus transmission connection with one breaker, a step-up transformer for the generator, and an additional transformer tapped from the BES to provide start-up capabilities for the reactor. A three-phase transformer is included to isolate the system from the BES if maloperation is detected by the protection relays or PLC of the SMR unit. The blue devices shown in Fig. 4.2 represent the protection relays or PLC of the SMR unit, which are located in the station's control room.

The Fig. 4.2 single-line substation diagram shows the general layout and essential equipment at an SMR station in the reference architecture. The turbine of the SMR unit requires a start-up motor to get the turbine up to speed for power generation. Once the turbine is up to speed and is generating electricity in phase with the grid, a breaker is thrown at the switchyard in the distribution substation to feed electricity into the larger BES. In Fig. 4.2 this is reflected by the need for two transformers at the station with on generator step up transformer and one start-up transformer. We consider the companies that design, install, operate, and maintain SMR facilities on behalf of other companies with respect to attack surface exposure. Third-party relations are often reflected in valid access to systems and may increase attack surface exposure. For example, the reference architecture assumes that different companies maintain and operate the SMR fleet and the distribution substation for generated electrical power, since this is typical in renewable energy. Moreover, although SMRs are envisioned as being remotely managed, they still require routine physical maintenance that is outsourced to a third party and conducted locally.

The MITRE ATT&CK for ICS framework was selected since the operational environment of a SMR station more closely resembles that of an industrial control system (ICS) than a conventional enterprise network. Unlike IT-centric threat models, the ICS framework accounts for the physical processes, real-time control requirements, and specialized assets found in generation stations. Additionally, the MITRE ATT&CK for ICS includes tactic and technique mappings that align directly with the functional tasks defined in the layered attack paths, enabling consistent ontology-based threat modeling grounded in real-world adversary behavior [9].

4.4 Layered attack graph for SMR station

In this section, we will thoroughly investigate the attack paths associated with the reference architecture introduced earlier. By individually analyzing each of the four layers, I will create attack graphs that reveal critical vulnerabilities. This work presents a comprehensive overview of the attack paths for layers L1 through L4, incorporating the MITRE

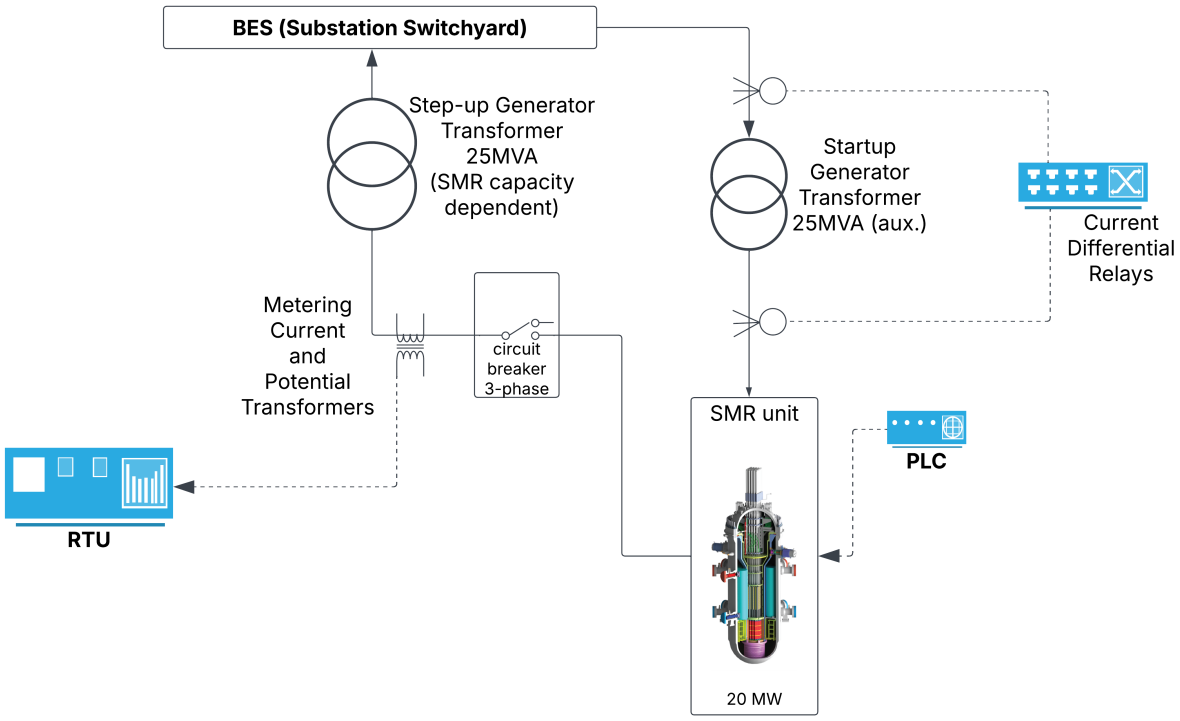

Figure 4.2: Single -line substation diagram that shows the layout and essential equipment at an SMR single bus single breaker station

ATT&CK for ICS technique mappings for both layers L1 and L4. The approach highlights several significant security challenges that researchers must address to enhance the overall resilience of CPES.

4.4.1 Physical Perception Layer (L4)

Fig. 4.3 presents a visual representation of the attack graph associated with the physical infrastructure layer L4 for an SMR station under the constraints of a CPES. Each circle represents an attack node represented by a task/event that leads to the root attack node of successful physical damage and environmental disruption to the SMR station and surrounding area. The specific task is described in Table 4.1. The red nodes represent actions that require the attacker to be physically present at the SMR station, while the SME symbol represents direct access to the reactor room at the SMR station. Table 4.1 describes the attack nodes shown in the SMR attack graph visualized in Fig. 4.3. The label aligns with the attack graph, directly indicating the DAG representation. Each attack node is aligned with the MITRE ATT&CK ICS reference architecture specific technique. Note that an attack initiated in the physical layer would require the attacker to be on-site the most.

We denote attack nodes that require on-site access and those that require direct access to the SMR unit in boldface text. However, there are attack paths that lead to the root node which do not require an attacker to be at the SMR station. Note that remote threats

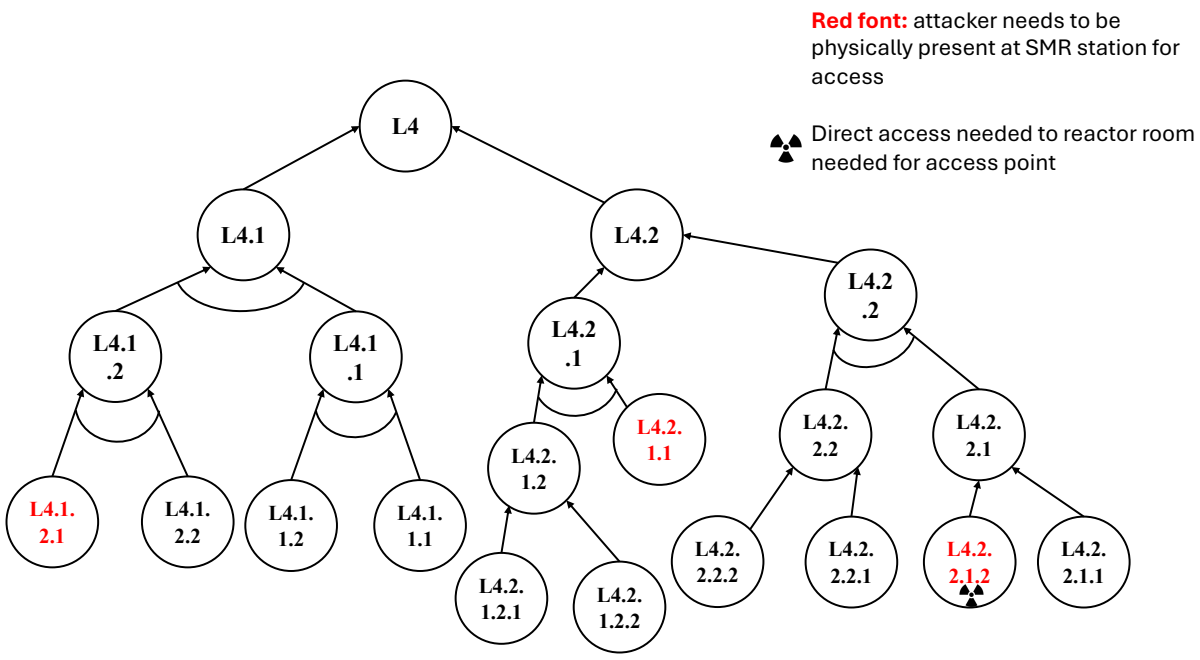

Figure 4.3: Visualization of the attack paths associated with the physical infrastructure of an SMR station

can access the CPES through SCADA communications protocols. Examine attack node L4.1.1 and terminal attack nodes L4.1.1.1 and L4.1.1.2 that only need remote terminal unit (RTU) access (via SCADA with higher permissions) to initiate an interference attack. Additionally, there is only one attack path that requires the attacker to have direct access to the SMR reactor room. This highlights the fact that physical security is only a small part of securing reactive equipment of a CPES.

4.4.2 Relay Protection and Control layer (L3)

Fig. 4.4 presents a visual representation of the attack graph associated with the protection and control layer L3 for an SMR station under the constraints of a CPES. The graph follows the same notation as the prior attack graph in Fig. 4.3. The protection and control layer houses some of the most sensitive devices in a CPES. The protection and control layer L3 contains the programmable logic controllers (PLCs) and microprocessor relays responsible for the automatic operation of the grid to prevent catastrophic failures. The devices in this layer operate very fast, in cycles (1 cycle=1/60 second). It is also important to note that these devices are found in the control room/house of the SMR station. It is for this reason that all attack paths on this layer require an attacker being onsite. The protection and controls in layer L3 necessitate greater complexity for attackers, since multiple attack paths are interconnected through dependent tasks (i.e. Logical -AND). The connectivity of layer L3 leads to interdependencies in other CPES layers in an SMR station. For example, if an operator in layer L1 would like to influence

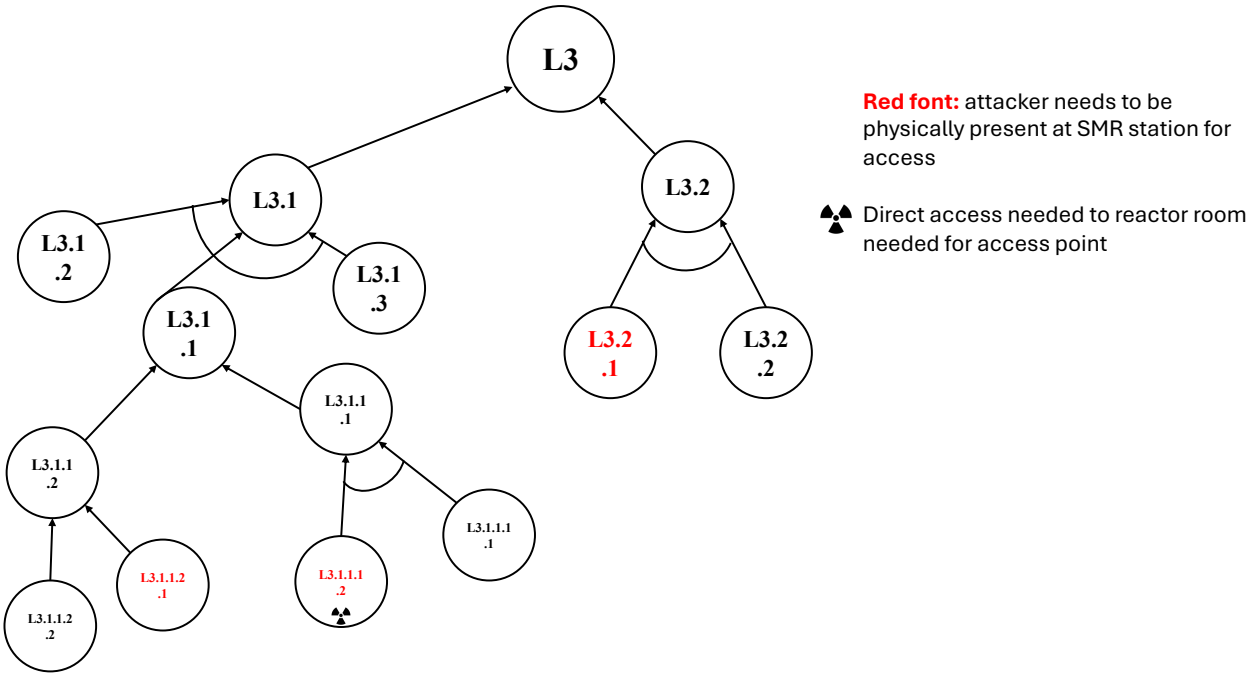

Figure 4.4: Visualization of the attack paths associated with the relaying and SCADA of an SMR station

operations directly at the SMR station, the command protocols to operate any circuit breaker, switch, or transformer tap changer traverse through the microprocessor relays to ensure that no manual command supersedes the protection of the physical infrastructure.

Table 4.2 describes the attack nodes shown in the SMR attack graph visualized in Fig. 4.4. The label aligns with the attack graph, directly indicating the DAG representation. Each attack node is aligned with the MITRE ATT&CK ICS reference architecture specific technique. Note that two nodes require on-site access and one requires direct access.

4.4.3 Communications Layer - SCADA (L2)

The SCADA communications layer for the SMR station is modeled to function similar to existing large scale reactors on the BES, governed by NERC CIP protocols. Fig. 4.5 presents a visual representation of the attack graph associated with the protection and control layer L3 for an SMR station under the constraints of a CPES. The graph follows the same notation as the prior attack graph in Fig. 4.3. Many of the attack paths in the communications layer do not require physical access to the SMR unit to reach the root attack node of a successful attack on the reactor. The SCADA communication terminals provide direct access from the local and remote operator to the SMR station equipment. The equipment at this layer is not directly connected to PLC's at the SMR station, rather provides the communication via remote terminal units (RTU) and switches. Table **??** describes the attack nodes shown in the SMR attack graph visualized in Fig. 4.5. The label aligns with the attack graph, directly indicating the DAG representation. Each

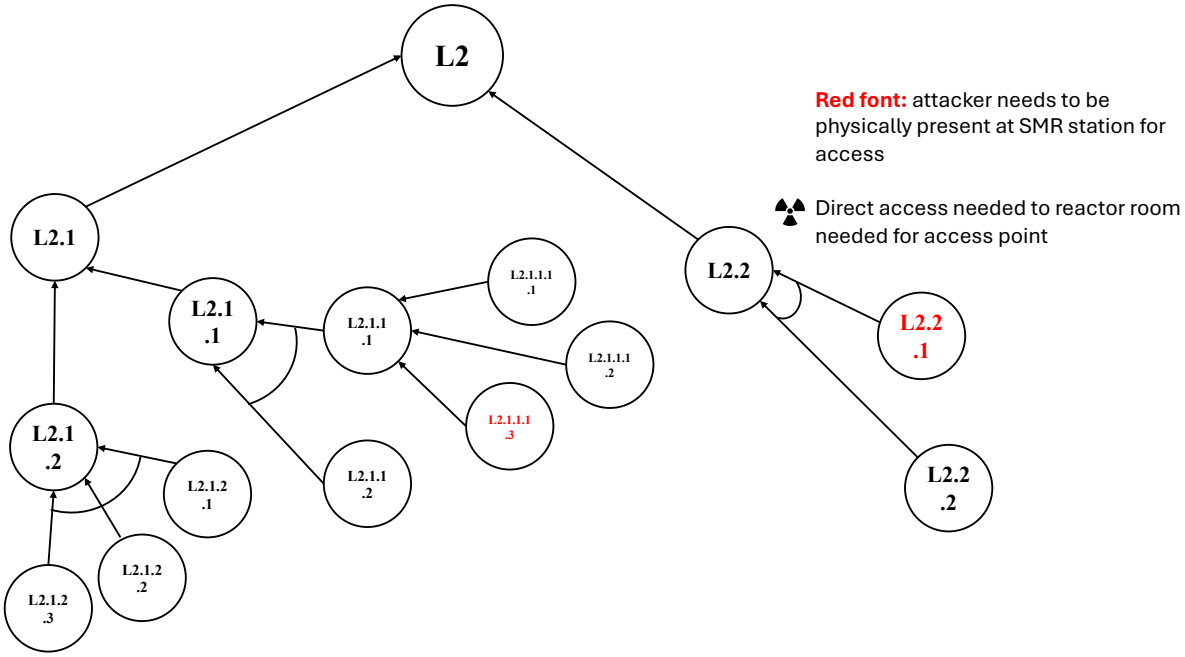

Figure 4.5: Visualization of the attack paths of communications and SCADA functionality

attack node is aligned with the MITRE ATT&CK ICS reference architecture specific technique. Note that two nodes require on-site access and one requires direct access.

4.4.4 Application and oversight layer (L1)

The application and oversight layer for the SMR station is modeled to reflect the NERC Transmission Operator Protocols (TOP), which require the ISO to maintain real-time monitoring of the BES and its critical components. Note, SMR stations are more likely to be associated as distribution voltage level units with even less oversight from the ISO than we infer here. In this model, the ISO has authority over voltage and frequency setpoints to ensure overall grid stability. This operational assumption forms the basis for constructing attack paths associated with layer L1. It is important to note that the ISO has limited control and communication with the SMR unit. The ISO has operational access that is primarily restricted to the ability to actuate the breaker at the point of interconnection. Fig. 4.6 represents the attack paths associated with vulnerabilities at the ISO oversight of the SMR station. We can observe that none of these attack paths require any direct manipulation of the SMR unit. Nevertheless, the techniques needed to complete these attacks are comparatively complex.

Table 4.4 describes the attack nodes shown in the SMR attack graph visualized in Fig. 4.6. Note that no attack nodes in layer L1 can be executed remotely. Instead, attacks propagate in this layer by means of cyber-attacks reflected by attack nodes parent to node L1.1.1. Here we see MITRE defined tactics such as “Initial access” and “privilege escalation” leading to high impact tactic that can destabilize/damage an SMR unit. Furthermore, the potential for insider attacks, reflected by attack nodes parent to L1.2,

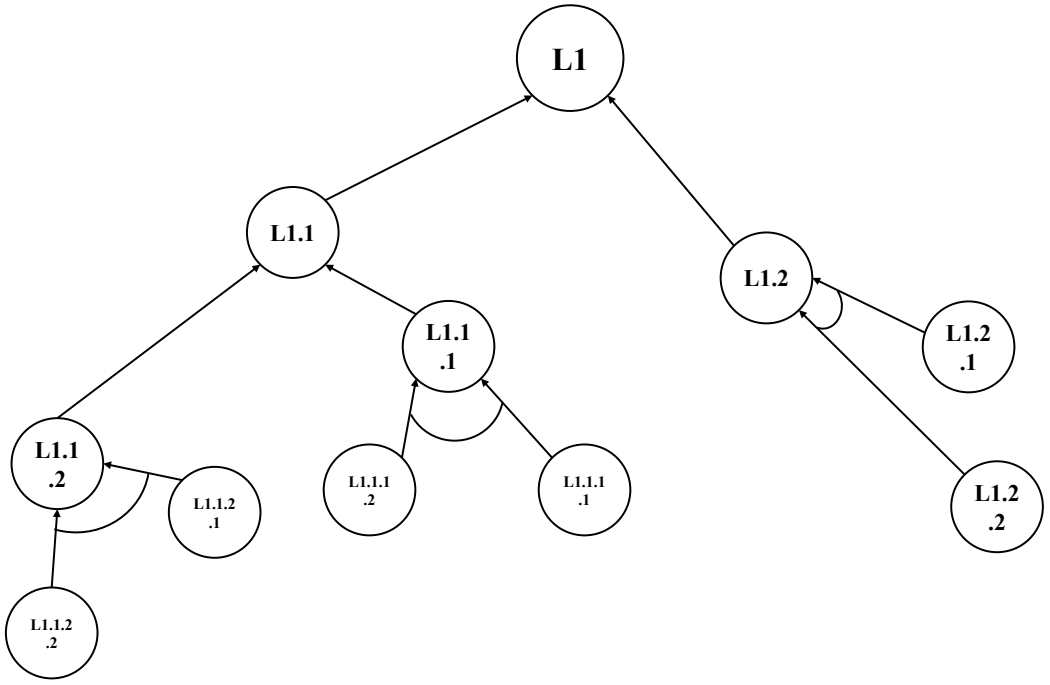

Figure 4.6: Visualization of the attack paths associated with application and oversight by the ISO and owner/operator of an SMR station

is present in layer L1 due to the level of clearance needed to create physical damage to the reactor.

4.4.5 CPES mitigations and defensive controls

Table 4.5 lists the mitigation strategies $M \in \mathcal{M}$ derived from NERC regulatory guidelines. These regulatory controls are designed to reduce the likelihood that an attacker can successfully execute a task associated with an attack node. Each mitigation provides a specific form of protection relevant to high-voltage Bulk Electric System (BES) operations.

We categorize the type of coverage provided for each attack node by each mitigation. Table 4.6 illustrates the regulatory mitigation mapping for selected Layer 3 attack nodes. Each entry represents the strength of coverage provided by the corresponding NERC regulation: **M** (Minimal), **P** (Partial), and **T** (Total). These mappings inform the classification of high-risk, low-coverage nodes.

To quantify the extent of coverage provided by each regulation, we assign the following symbolic indicators:

- **M** — Minimal coverage: regulation offers limited, indirect, or situational protection against the attack node task.
- **P** — Partial coverage: regulation addresses key components or processes associated with the attack node but does not fully prevent its execution.

- **T** — Total (terminal) coverage: regulation directly mitigates or prohibits the attack task, serving as a strong barrier against path progression.

Coverage, in this context, refers to the regulation's ability to prevent the activation of a given attack node and thereby block propagation through associated attack paths.

4.4.6 SMR station Maintenance activities

The maintenance of small modular reactors (SMRs) is crucial for their safety, efficiency, and competitiveness in the energy market. Replacing spent fuel is a key part of this maintenance, and innovations in operation and maintenance (O&M) technologies are being explored to improve this process. There are several technologies in-progress to streamline the maintenance of SMR stations [37]. In this work, I identify five common maintenance activities that will need to be performed on an annual or semi-annual basis to ensure performance, even after the newer technologies are online. A maintenance scenario represents as a set of tasks that are are needed to complete the maintenance activity. This is a list of the maintenance activities considered in this study:

- **Spent Fuel Replacement:** Fuel rods are typically replaced every 12 to 24 months, depending on the reactor design and operational cycle.
- **Relay Calibration:** Regular updates and recalibration of relay protection settings are critical due to evolving performance parameters and grid interaction requirements.
- **Cooling System Maintenance:** Includes inspections, heat exchanger cleaning, coolant chemistry monitoring, and leak checks—essential for thermal regulation and safety.
- **Monitoring and Maintenance of Fuel:** Ensures integrity and cooling of stored spent fuel assemblies; includes radiation monitoring, water chemistry checks, and structural inspections.
- **Ancillary Services Testing:** Ensures that the unit is able to follow the governing ISO voltage setpoints and frequency stability (fast ramping) when operating on the BES

Once the maintenance scenarios are defined we can determine overlap with attack node tasks to find the risk associated with each maintenance activity. Table 4.7 represents the overlap between maintenance scenario tasks and attack node–related adversarial tasks. This overlap forms the basis of the trigger function $\delta(v_i, s_k)$, which is defined as:

$$\delta(v_i, s_k) = \begin{cases} 1 & \text{if the maintenance task set } \mathcal{T}_k \text{ overlaps with the attack task set } \mathcal{T}_{v_i} \\ 0 & \text{otherwise} \end{cases}$$

Each "X" in the table denotes a maintenance scenario s_k that could unintentionally trigger the assertion of attack node v_i, due to task-level similarity or system access overlap. In this context, a maintenance action does not directly cause a compromise, but it

creates operational conditions that lower the barrier for adversarial progression through the attack graph.

The trigger function $\delta(v_i, s_k)$ serves as a key input into the node activation and risk propagation model. Nodes triggered by multiple maintenance scenarios are considered to have elevated real-world exposure, and are classified as high-risk. These nodes may require additional scrutiny, redesign, or mitigation coverage to reduce systemic vulnerability.

4.5 Model Formulation

In this section, we formalize the graph-based risk model used to evaluate cyber-physical vulnerabilities in an SMR station. This formulation captures the structural propagation of adversarial actions, the external influence of maintenance scenarios, and the effect of regulatory mitigations.

4.5.1 Attack graph structure

We define the SMR station attack model as a directed graph $G = (V, E)$, where:

- $V = \{v_1, v_2, \dots\}$ is the set of attack nodes, each representing a discrete adversarial task or compromised process step.
- $E \subseteq V \times V$ is the set of directed edges, where $(v_i, v_j) \in E$ indicates that node v_i is a prerequisite for activating node v_j.

Let $\mathcal{P} = \{p_1, p_2, \dots\}$ denote the set of valid attack paths in the graph, where each path $p_j \subseteq V$ is a topologically valid sequence of attack nodes.

4.5.2 Maintenance scenarios and task overlap

Recall from Section 4.4.6 that there are overlaps in the tasks required to complete a maintenance activity and tasks that need to be performed by an attacker that can increase the vulnerability of an SMR station to an attack.

Let $\mathcal{T}$ be the universe of operational tasks relevant to either adversarial or maintenance behavior. Each maintenance scenario $s_k \in \mathcal{S} = \{s_1, \dots, s_K\}$ is associated with a task set $\mathcal{T}_k \subseteq \mathcal{T}$, and each attack node $v_i \in V$ is associated with an adversarial task set $\mathcal{T}_{v_i} \subseteq \mathcal{T}$.

The *trigger function* can then be expressed as:

$$\delta(v_i, s_k) = \begin{cases} 1 & \text{if } \mathcal{T}_{v_i} \cap \mathcal{T}_k \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

The set of attack nodes triggered by scenario s_k is defined as:

$$A_k = \{v_i \in V \mid \delta(v_i, s_k) = 1\}$$

4.5.3 Attack path activation

For a given attack graph $G = (V, E)$ we define an *attack path* p_j to be the set of nodes and edges required for an attacker to initiate an attack at a terminal attack node v_i and progress through the graph to a goal node v_{goal}. For a given maintenance scenario we identify the fraction of all paths $p_j \in G$ that are activated. This activation of the attack path means that there are attack nodes in the path with an increased probability of completion during a maintenance scenario s_k. We define the scenario-specific path activation fraction as:

$$\alpha(p_j, s_k) = \frac{|\{v_i \in p_j \mid \delta(v_i, s_k) = 1\}|}{|p_j|}$$

The OR-logic precedence creates a situation where an attack node $v_i \in V$ is a part of more than one attack pathp_j. This construction captures the idea that an attack node is considered active if it is part of at least one path significantly influenced by scenario s_k. Since a single node may appear in multiple paths, we define the node-level scenario activation:

$$\delta^*(v_i, s_k) = \max_{p_j} \alpha(p_j, s_k) \tag{4.1}$$

4.5.4 Mitigation coverage

We now have a set of mitigations as *coverage* for terminal attack nodes. Let $\mathcal{M} = \{M_1, M_2, \ldots, M_L\}$ denote the set of mitigation controls derived from regulatory guidance. The coverage function is defined as:

$$\mu : V \times \mathcal{M} \to [0, 1] \quad \text{where } \mu(v_i, M_\ell) \text{ is the effectiveness of } M_\ell \text{ on } v_i$$

Let $\mathcal{M}_i \subseteq \mathcal{M}$ be the set of mitigations applied to node v_i. The combined mitigation effectiveness at node v_i is:

$$M_i = 1 - \prod_{M_\ell \in \mathcal{M}_i} (1 - \mu(v_i, M_\ell)) \tag{4.2}$$

4.5.5 System failure probability

We define the residual risk for node v_i under scenario s_k as:

$$R(v_i, s_k) = \delta^*(v_i, s_k) \cdot (1 - M_i) \tag{4.3}$$

Finally, the system-level failure probability under scenario s_k is:

$$P_{\text{fail}}(s_k) = 1 - \prod_{v_i \in V} (1 - R(v_i, s_k)) \tag{4.4}$$

This metric represents the likelihood that **at least one** unmitigated attack path becomes feasible due to maintenance-induced activation and insufficient defensive coverage. The probability of system failure is directly linked to the likelihood of completion attack path p_j in a scenario s_k. It is defined in the following relation:

$$P_{\text{fail}}(p_j \mid s_k) = \prod_{v_i \in p_j} \mathbb{P}(v_i = 1 \mid s_k) \tag{4.5}$$

Illustrative example: In a simple case, assume scenario s_1 activates three terminal attack nodes: v_1, v_2, and v_3, each with $\delta(v_i, s_1) = 0.5$. Mitigations M_j are applied to some nodes, with associated effectiveness values $\mu(v_i, M_j)$. The residual risk is calculated as:

$$R(v_i, s_1) = \delta^*(v_i, s_1) \cdot (1 - M_i)$$

where M_i is the combined effect of all mitigations applied to v_i. If v_3 receives no mitigation ($M_i = 0$), it contributes most significantly to the system failure risk. The final system failure probability is computed as:

$$P_{\text{fail}}(s_1) = 1 - (1 - 0.05)(1 - 0.2)(1 - 0.5) = 0.62$$

This example demonstrates how unmitigated high-assertion nodes under s_1 raise the overall risk of critical failure.

4.6 Case study for Single bus Single Breaker SMR station

4.6.1 Quick observations:

We can do a quick overview of attack node coverage just from the tables generated in this study. Let's classify an attack node as *high-risk, low-coverage* when it meets two criteria:

1. **High risk:** The node is externally triggered by three or more of the five modeled maintenance scenarios. This is indicated by an "X" appearing in three or more maintenance columns (e.g., Spent Fuel Replacement, Relay Calibration, etc.), suggesting that routine operational activities increase the likelihood of the attack task being enabled.

2. **Low coverage:** The node lacks any associated mitigation rated "T" (Total), and has fewer than three combined "M" or "P" entries across the regulatory mitigation set.

Nodes that meet both conditions (high-risk, low-coverage) are considered operationally vulnerable and are not adequately protected by current regulatory frameworks. Identifying these nodes provides a targeted view into structural weaknesses where policy, procedural, or technical intervention may deliver the most meaningful security improvements.

Evaluating attack graph $G = (V, E)$ in Fig. 4.7 we see that there are two attack nodes, in this particular configuration, that would be considered critical nodes. However, we want to know exactly what is the risk associated with each maintenance scenario.

Given the conditional path failure probability defined in Eq. 4.5, the expected risk for a given maintenance scenario s_k is calculated as:

$$\text{Risk}(s_k) = \sum_{p_j \in \mathcal{P}} P_{\text{fail}}(p_j \mid s_k) \cdot \max_{v_i \in p_j} C_i \tag{4.6}$$

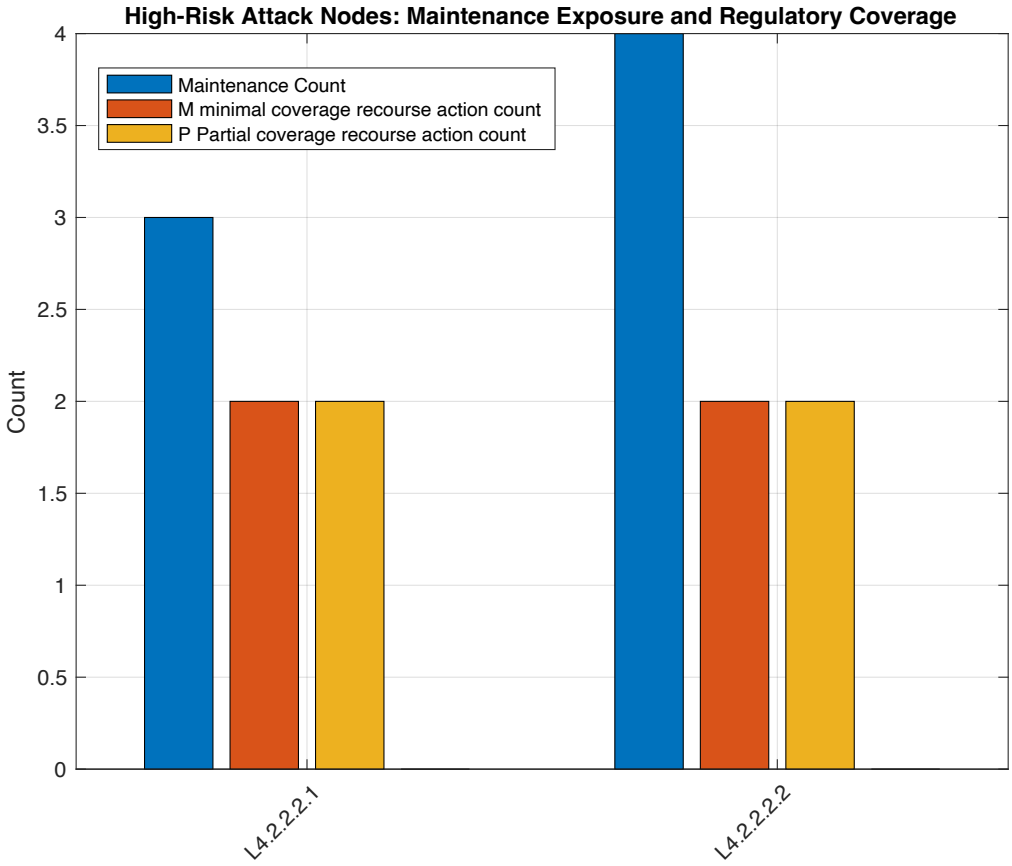

Figure 4.7: Bar chart to compare the number of triggering maintenance scenarios compared to the number and type of recourse actions available. We see there are no total (T) coverage actions available for either attack node

Here, the consequence assigned to each path is defined as the maximum impact of any attack node on that path:

$$C_{p_j} = \max_{v_i \in p_j} C_i$$

Substituting the path failure probability from Eq. 4.5 yields:

$$\text{Risk}(s_k) = \sum_{p_j \in \mathcal{P}} \left(\prod_{v_i \in p_j} \mathbb{P}(v_i = 1 \mid s_k) \right) \cdot \max_{v_i \in p_j} C_i \tag{4.7}$$

This formulation captures how elevated assertion probabilities at individual nodes—due to maintenance-induced exposure and incomplete mitigation—aggregate along a path to contribute to overall system risk.

We visualize this relationship in Fig. 4.8, which shows how the system failure probability varies under different maintenance scenarios as uniform mitigation levels M_i are increased. Despite identical mitigation coverage across all nodes, the different triggering profiles of maintenance scenarios lead to distinct $P_{\text{fail}}(s_k)$ trajectories. This illustrates that risk is not solely a function of defensive effort, but also of operational context.

To explore which maintenance scenarios respond more to increased mitigation coverage, we compute the derivative of $P_{\text{fail}}(s_k)$ with respect to uniform mitigation level M. This local sensitivity indicates how rapidly system risk declines as defenses are strengthened. Figure 4.8 shows this slope for each scenario at select mitigation levels (e.g., $M = 0.6, 0.8$). Scenarios with steeper slopes are more responsive to control investments, while flatter profiles suggest structural vulnerabilities that may require operational redesign or selective hardening at critical nodes.

4.6.2 Layered Threat Modeling Framework for SMR-Based CPES

Figure 4.8 illustrates the layered attack graph framework used to model cybersecurity risks in Small Modular Reactor (SMR) stations, conceptualized as Cyber-Physical Energy Systems (CPES). This framework decomposes the SMR station architecture into four interdependent functional layers, allowing for a structured, multistage representation of adversarial attack paths. Each layer reflects a distinct operational domain and exposes specific vulnerabilities, and the inter-layer dependencies enable modeling of cascading and cross-domain cyber-physical risks.

Functional Layer Structure. The vertical columns in the figure represent four hierarchical layers of the SMR station:

- **L1 – Application and Oversight Layer:** This topmost layer models threats originating from Independent System Operators (ISOs) and regulatory oversight mechanisms. Attack vectors at this level include compromised operator terminals, improper coordination, and malicious market signaling. These actions typically require elevated access privileges and may influence reactor behavior via market dispatch or interconnection setpoints.

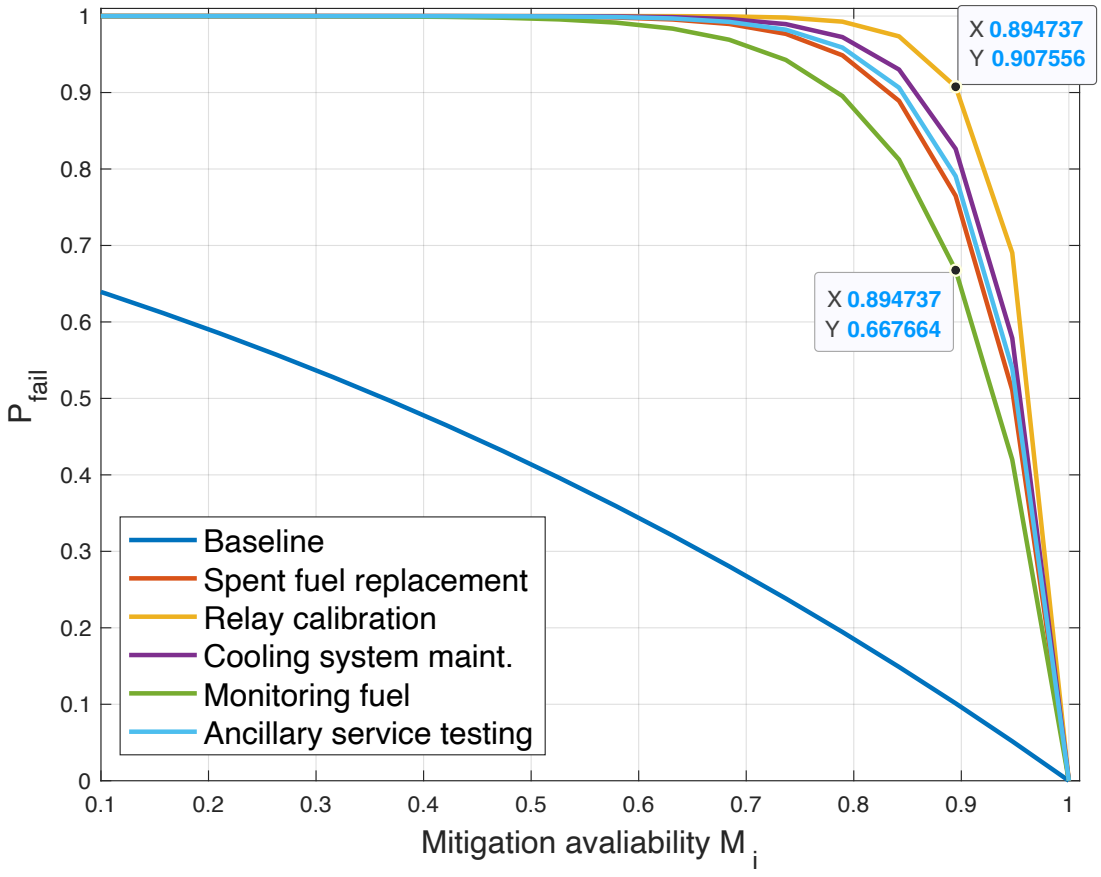

Figure 4.8: System failure probability $P_{\text{fail}}(s_k)$ for different maintenance scenarios, as a function of uniform mitigation coverage M_i. Each curve represents a distinct scenario s_k, highlighting how varying maintenance task overlap affects overall risk.

- **L2 – Communications Layer:** This layer captures SCADA-based command and control infrastructure, including remote terminal units (RTUs), human-machine interfaces (HMIs), and data exchange protocols. Attacks in this layer involve firmware manipulation, unauthorized software injection, or network enumeration techniques to distort control messaging and intercept field-level commands.
- **L3 – Relay Protection and Control Layer:** This layer includes programmable logic controllers (PLCs) and microprocessor-based relays that safeguard equipment through automated protection schemes. Vulnerabilities here involve logic manipulation, sensor spoofing, and override of hardcoded safety interlocks. Notably, due to the critical nature of L3 devices, most attack paths in this layer require physical or on-premise access.
- **L4 – Physical Infrastructure Layer:** This layer includes physical assets such as transformers, switches, generator sets, and control rods. Attacks at this level entail direct sabotage or interference with power-electronics infrastructure. Many paths originate in higher cyber layers and culminate in physical actions at L4, highlighting the dual cyber-physical nature of SMR system risk.

Graph Structure and Semantics. The attack model is encoded as a directed acyclic graph (DAG), where each node represents a discrete attacker action, condition, or goal. Edges capture logical dependencies—AND/OR semantics—between attack steps. Nodes are color-coded and symbolically marked to indicate access constraints:

- **Red nodes** indicate actions requiring *on-site access* to the SMR station.
- Nodes marked with the **SMR icon** require *direct access to the reactor room*, representing the most restricted and high-impact steps.
- Unmarked nodes denote attacks executable via *remote cyber access*, such as protocol exploitation or credential compromise.

The horizontal orientation of the layers corresponds to the temporal and operational hierarchy of a CPES. An adversary may progress laterally across a single layer or escalate vertically across layers to achieve more impactful outcomes. For example, a phishing-based credential theft in L1 may lead to SCADA command injection in L2, which in turn manipulates protection logic in L3 and triggers actuator faults in L4.

Cyber-Informed Design (C-ID) and Risk Applications. This framework supports the goals of Cyber-Informed Design (C-ID) by aligning attack steps with known tactics and techniques in the MITRE ATT&CK for Industrial Control Systems (ICS) ontology. By mapping each node to real-world adversarial behavior, the model enables:

- Traceable identification of *high-risk attack paths* and their dependencies;
- Evaluation of which combinations of cyber and physical access yield critical system compromise;

- Prioritization of *layer-specific mitigations* that interrupt multi-layer attack progression;
- Structured support for *scenario-based risk assessment*, including integration with temporal models such as time-inhomogeneous DTMCs.

Conclusion. Overall, this layered attack graph framework provides a formal and modular basis for modeling the complex interactions that govern risk in SMR-based CPES. By structurally linking attacker behavior to both operational architecture and real-world cyber-physical control logic, the model serves as both a diagnostic and prescriptive tool in critical infrastructure cybersecurity planning.

4.7 Lessons Learned and Transition to Dynamic risk modeling

The SMR case study demonstrates the value of a layered attack graph framework in mapping system vulnerabilities, identifying critical components, and exploring mitigation strategies within a cyber-physical energy system (CPES). By modeling the system's architecture and threat surface through the lens of structured attack and defense components, this chapter highlights how such graphs can support high-level risk analysis and system redesign.

However, this analysis remains inherently **static**—it does not account for the temporal progression of threats or the dynamic response behaviors of the system over time. In real-world CPS environments, especially those with human or policy-driven components, time plays a critical role in shaping both risk exposure and the effectiveness of mitigation strategies.

The next chapter applies the same core framework *process, attacks, mitigations* to a socio-technical CPHS system: vote-by-mail (VBM) infrastructure. There, temporal dynamics are integrated into the modeling via a time-inhomogeneous discrete-time Markov chain (DTMC). This extension enables a more granular risk assessment, capturing how vulnerabilities evolve and interact across process stages. The VBM case not only completes the risk modeling framework introduced in this thesis, but also demonstrates its capacity to inform both technical design and public policy in complex, real-world CPS contexts.

Table 4.1: Attack Nodes and Associated MITRE ATT&CK ICS Techniques

Label	**Attack Node Description**	**MITRE ATT&CK ICS Technique (Tactic)**
L4	Physical layer attack causes physical damage to SMR unit	T0879: Damage to property (Impact)
L4.1	Interference attacks on connections between sensors, PLCs, and control actuators	T0880: Loss of safety (Impair process control)
L4.1.1	Access gained to devices via RTU to a remote terminal	T0866: Exploitation of remote services (Initial access)
L4.1.1.1	Insider knowledge of remote terminal addresses gained	T0831: Manipulation of process control (Impair process control)
L4.1.1.2	Attain permissions and passwords to access RTU	T0890: Exploitation of privilege escalation (Privilege escalation)
L4.1.2	Executes PLC rootkit attack to manipulate controls logic	T0803: Command message injection (Execution)
L4.1.2.1	Access to SMR PLC directly (on-site access)	T0848: Rogue master (Initial access)
L4.1.2.2	Gains knowledge of the protection and controls scheme	T0888: Remote system information (Discovery)
L4.2	Physical sabotage of equipment in the field	T0837: Loss of protection (Impact)
L4.2.1	Physical damage to the adjacent reactor rooms	T0831: Manipulation of control (Impact)
L4.2.1.1	Physical access to auxiliary room (on-site access)	T0862: Supply chain compromise (Initial access)
L4.2.1.2	Attain knowledge of interactions between the auxiliary room and SMR unit	T0816: Device restart/shutdown (Inhibit response function)
L4.2.1.2.1	Attain knowledge of Generator operation	T0811: Data from information repositories (Collection)
L4.2.1.2.2	Attain knowledge of turbine operation	T0811: Data from information repositories (Collection)
L4.2.2	Physical damage to SMR unit	T0879: Damage to property (Impact)
L4.2.2.1	Gain access to controls or physical location	T0863: User execution (Execution)
L4.2.2.1.1	Gain access to SMR reactor control room	T0863: User execution (Execution)
L4.2.2.1.2	Gain access to SMR reactor rooms (direct access)	T0859: Valid accounts access (Lateral movement)
L4.2.2.2	Malicious controller commands sent to PLC of individual SMR unit	T0821: Modify Controller Tasking (Execution)
L4.2.2.2.1	Tools obtained to operate controls of SMR manually or just destructive devices	T0807: Command Line Interface (Execution)
L4.2.2.2.2	Knowledge of SMR control rod configurations and overrides	T0888: Remote system information (Discovery)

Table 4.2: Protection and Control Layer (L3) Attack Nodes and MITRE ATT&CK ICS Techniques

Label	**Attack Node Description**	**MITRE ATT&CK ICS Technique (Tactic)**
L3	Protection and Control Layer attack causes physical damage of SMR unit	T0879: Damage to property (Impact)
L3.1	Direct interference with protection and control scheme	T0837: Loss of protection (Impact)
L3.1.1	Attain access to SMR station RTU	T0890: Exploitation of privilege escalation (Privilege escalation)
L3.1.1.1	Attain physical access to the adjacent reactor rooms	T0862: Supply chain compromise (Initial access)
L3.1.1.1.1	Attain physical access to auxiliary room	T0862: Supply chain compromise (Initial access)
L3.1.1.1.2	Attain ability to impact or damage auxiliary room (direct access)	T0881: Service stop (Inhibit response function)
L3.1.1.2.1	Damage generator	T0807: Command Line Interface (Execution)
L3.1.1.2.2	Damage turbine	T0807: Command Line Interface (Execution)
L3.1.1.2	Attain access to PLC of reactor	T0835: Manipulate I/O image (Inhibit response function)
L3.1.1.2.1	Attain access via physical plant control room (on-site access)	T0859: Valid accounts access (Lateral movement)
L3.1.1.2.2	Attain access via SCADA/HMI port	T0866: Exploitation of remote services (Initial access)
L3.1.2	Attain knowledge of the reactor control system and safety rods configuration	T0888: Remote system information (Discovery)
L3.1.3	Gain ability to override the built-in protection and control scheme of the SMR	T0836: Modify parameters (Impair process control)
L3.2	Physical damage to equipment sensors or control equipment that would affect control scheme	T0837: Loss of protection (Impact)
L3.2.1	Attain physical access to substation measuring equipment (on-site access)	T0855: Unauthorized command message (Impair process control)
L3.2.2	Gain knowledge of relay control scheme	T0888: Remote system information (Discovery)

Table 4.3: Communications Layer (L2) Attack Nodes and MITRE ATT&CK ICS Techniques

Label	**Attack Node Description**	**MITRE ATT&CK ICS Technique (Tactic)**
L2	Communications layer attack causes physical damage of SMR unit	T0879: Damage to property (Impact)
L2.1	SCADA equipment attack results in misoperation of plant protection and control schemes	T0837: Loss of protection (Impact)
L2.1.1	Firmware attack to network commands (remote terminal control)	T0839: Module firmware (Impair process control)
L2.1.1.1	Access to a SCADA terminal on the system	T0883: Internet accessible device (Initial access)
L2.1.1.2	Inject malware onto RTU or downstream relays	T0807: Command Line Interface (Execution)
L2.1.1.1.1	HMI port accessed	T0823: Graphical user interface (Execution)
L2.1.1.1.2	SCADA remote terminal accessed	T0848: Rogue master (Initial access)
L2.1.1.1.3	Introduction of unauthorized software by employees (on-site access)	T0831: Manipulation of control (Impact)
L2.1.2	Initiates RTU malware attack	T0890: Exploitation for privilege escalation (Privilege escalation)
L2.1.2.1	Attain access to RTU terminal port	T0866: Exploitation of remote services (Initial access)
L2.1.2.2	Gains knowledge of location-specific SCADA protocols	T0840: Network connection enumeration (Discovery)
L2.1.2.3	Attain access to RTU permissions for modifying settings	T0889: Modify program (Persistence)
L2.2	Physical damage to SCADA sensors and monitors	T0837: Loss of protection (Impact)
L2.2.1	Gain access to the control room of the SMR station (on-site access)	T0863: User execution (Execution)
L2.2.2	Attain knowledge of equipment and specifications	T0888: Remote system information (Discovery)

Table 4.4: Application Layer (L1) Attack Nodes and MITRE ATT&CK ICS Techniques

Label	**Attack Node Description**	**MITRE ATT&CK ICS Technique (Tactic)**
L1	Application Layer attack causes physical damage of SMR unit	T0879: Damage to property (Impact)
L1.1	Real-time operator terminal compromised	T0813: Denial of control (Impact)
L1.1.1	RTO gives bad instruction to SMR controls	T0831: Manipulation of control (Impact)
L1.1.1.1	Exploits past organizational relationship	T0865: Spearphishing attachment (Initial access)
L1.1.1.2	Operator not monitored by redundant operators on the floor	T0890: Exploitation for privilege escalation (Privilege escalation)
L1.1.2	RTO monitoring is suspended	T0815: Denial of view (Impact)
L1.1.2.1	Executes attacks on firewalls, including IP address spoofing, source routing	T0822: External remote services (Initial access)
L1.1.2.2	Gains knowledge of specific IP and terminal information specific to the SMR	T0893: Data from local system (Collection)
L1.2	Ancillary service controls have interference	T0835: Manipulation of I/O image (Inhibit response function)
L1.2.1	Improper coordination with neighboring generation stations	T0804: Block reporting message (Inhibit response function)
L1.2.2	Attains access and ability to influence voltage and frequency setpoints of the SMR stations	T0821: Modify controller tasking (Execution)

Table 4.5: NERC Regulatory Controls Considered as Mitigations in SMR Security Planning

Regulation ID	**Description**
NERC.NUC.001.4.R1–R5	Nuclear Plant Interface Requirements (NPIRs): Coordination for normal operation between Generator Operator and Transmission Entities.
NERC.NUC.001.4.R6–R7	Generator operator must coordinate all maintenance actions and plant changes with Transmission Entities.
NERC.NUC.001.4.R8	Transmission Entities must coordinate all changes that impact the unit with the Generator Operator.
NERC.CIP.006.6.R1–R2	Visitor control programs requiring escorts and automated logging of non-personnel in sensitive BES areas.
NERC.CIP.004.7.R1–R2	Cybersecurity and Security Awareness training for all personnel at Generator and Transmission Entities.
NERC.CIP.004.7.R3	Personnel risk assessment program covering all BES communication and control-related workers.
NERC.CIP.004.7.R4–R6	Access management program for physical locations and documents linked to Generator and Transmission Entities.
NERC.FAC.014.3.R4–R5	Annual system studies of voltage and frequency stability to establish operating limits for Generator Operators.
NERC.FAC.014.3.R6–R7	Transmission planning must issue corrective action plans to address abnormalities impacting Generator Operators.
NERC.PRC.005	Coordination between Transmission Entities and Generator Operators to maintain protective relay controls.

Table 4.6: Representative M–P–T Mapping for Selected Layer 3 Attack Nodes

Node Label	Attack Description	CIP-006 R1–R2	CIP-004 R1–R2	CIP-004 R3	CIP-004 R4–R6
L3.1.1.1.1	Physical access to auxiliary room	P	M	P	T
L3.1.1.1.2	Damage to auxiliary room (direct)	T		M	P
L3.1.1.2.1	Damage generator via command injection	P	P	P	P
L3.1.1.2.2	Damage turbine via command injection	P	P	P	P
L3.1.1.2.1	Access via plant control room (onsite)	P	P	T	T
L3.1.1.2.2	Access via SCADA/HMI port	M	P	T	
L3.1.2	Learn control system and safety rod logic	M	P	P	T
L3.1.3	Override built-in protection and control	P	P	T	

Table 4.7: Maintenance scenario activation of select attack nodes

Node Label	Attack Description	Spent Fuel	Relay Calib.	Primary Coolant	Ancillary Testing	Cooling Pool
L3.1.1.2.1	Access via physical plant control room	X	X	X		
L3.1.3	Override SMR control and protection logic	X		X		X
L4.1.1.2.2	Compromise auxiliary control firewall		X		X	X
L4.2.2.2	SCADA configuration tampering	X	X	X		
L4.2.1.1	Exploit auxiliary terminal credentials			X	X	

Chapter 5

Risk modeling CPHS - Vote-by-Mail security modeling with temporal risk analysis

5.1 Introduction

The legitimacy of a political system fundamentally depends on the security, transparency, and accuracy of its electoral processes. In the United States, absentee voting has become increasingly popular over the past three decades, with 57-percent of ballots cast early in the 2024 General Election [15]. Among absentee methods, voting by mail (VBM) has emerged as a prominent modality, prompting intensified scrutiny of its security and resilience. In this work, I model VBM as a Cyber-Physical-Human System (CPHS) wherein human actors, physical infrastructure, and digital technologies interact to carry out election procedures. A CPHS, such as VBM operations, spans the cyber, physical, and procedural domains. The complexity of these systems renders them susceptible to a wide range of threats. These threats include both malicious cyber intrusions and non-malicious process failures. Because elections are designated as part of the Government Facilities sector under U.S. critical infrastructure classifications, we must ensure the integrity of VBM systems is not only a matter of procedural correctness but also of national security.

This chapter presents a novel application of attack graph modeling for risk analysis in election systems, with a specific focus on vote-by-mail. We adopt the modeling approach introduced by Haseltine and Albert, which integrates attack graphs with a discrete-time Markov chain (DTMC) to provide a dynamic, multi-layered risk assessment framework. The DTMC formulation includes three interacting layers: a process layer that models the typical sequence of VBM operations, a threat layer that encodes both malicious and non-malicious attack vectors, and a mitigation layer that captures the available recourse mechanisms. The layered framework creates a more holistic representation of the temporal dynamics, interdependencies, and mitigation effects throughout the election cycle. The DTMC is time-inhomogeneous, allowing the model to capture variations in system behavior and the impact of attacks over time.

The remainder of this chapter introduces the VBM process components and details the layered DTMC framework. The chapter's case study illustrates how this modeling paradigm supports rigorous and time-sensitive risk analysis in election systems. Key insights into policy and procedures for VBM are given at the end of the chapter.

5.2 Background

The vast majority of election security modeling research focuses on in-person voting. A stream of papers in this area assesses the impact of operational decisions and resource allocation on election performance using methodologies such as discrete event models and model optimization [81, 80]. However, these approaches are not adaptable to address VBM, which requires its own analytical framework due to its unique operations and vulnerabilities that differ from in-person voting. An exception is Schmidt and Albert [71] who formulate an integer programming model to determine the locations of drop boxes used in VBM. However, Schmidt and Albert [71] narrowly focus on the location of drop boxes and the collection of ballots from drop boxes. In contrast, our paper focuses on the entire VBM process, including operations leading up to an election.

Research on the allocation of voting machines to polling locations for in-person voting has been extensively explored through various methodologies. One notable example is a study by Yang, Fry, and Kelton [97] who uses discrete event simulation to model election day processes, combining queuing theory and optimization to address resource concerns. Similarly, Li, Allen, and Akah [45] employ simulation optimization to assess the impact of voting machine distribution on voter queue times. The significance of these studies extends to a tailored risk assessment for VBM. Schmidt and Albert apply discrete event simulation to understand the effects of safety measures implemented during the COVID-19 pandemic on wait times and other election performance indicators. In addition, McIntyre focus on optimizing the polling locations to reduce voter waiting times. These approaches underscore the importance of modeling and optimization in improving election processes, informing the development of risk assessment methodologies for election infrastructure. Specific to the VBM system, Scala et al. develop a detailed model of the VBM process, which meticulously outlines its various components and the physical journey of ballots. The Cybersecurity and Infrastructure Security Agency (CISA) conducted a comprehensive assessment of mail-voting security for the 2020 General Election [14], contributing to efforts to identify and mitigate VBM-related risks during the pandemic.

A growing body of literature characterizes threats to the voting process using attack trees. Attack trees help identify and prioritize potential vulnerabilities in a system, making them an essential tool in cybersecurity planning and risk mitigation. Attack trees are fundamental in cybersecurity for visualizing vulnerabilities and potential attacker paths [72]. An attack tree starts with the primary goal of the attacker (the root) and branches out into different methods or steps that the attacker might employ (the leaves). This branching structure often incorporates logic gates like AND and OR to demonstrate how different actions might combine to lead to the attack goal. Terminal attack nodes, are

the leaf nodes of an attack tree that initiate the attack on the system. The U.S. Election Assistance Commission present a risk analysis report for election systems and models security threats to the VBM using attack trees that capture all possible vulnerabilities in the VBM system. Scala et al. expand on the VBM threat trees and adds to tree logic enumerating all terminal attack nodes to account for the new mechanisms available, such as drop boxes and in-person absentee voting. Haseltine, Wang, and Albert build upon this body of literature to identify mitigation strategies and policies that protect the VBM process, accounting for linkages to attack trees relevant to VBM processes. Our DTMC framework utilizes these attack trees and mitigations.

Attack and fault trees have been used for the risk analysis of in-person voting processes. Yasinsac and Pardue [99] illustrate how attack trees have been deployed to analyze risks, including malicious cyber-attacks targeting voting machines. These methodologies consider both equipment failures and cybersecurity vulnerabilities, offering a broad view of potential threats. In a similar vein, Simidchieva et al. [78] use fault trees to evaluate the in-person election process, with a particular emphasis on ballot counting and vulnerability detection. However, these papers do not differentiate between malicious and non-malicious attacks, solely focus on in-person voting, and represent risks as static by excluding temporal analysis.

Motivation: This work aims to fill the gaps in knowledge by introducing a time-inhomogeneous DTMC model to support VBM risk analysis over an election cycle. The DTMC model differentiates between malicious and non-malicious threats, considers risk in a dynamic system, and evaluates the effectiveness of mitigation strategies. This DTMC modeling approach employs layered networks to model interdependent components within the VBM process. The DTMC model is uniquely tailored to address the inherent complexities of VBM, efficiently accommodating simplified process components, diverse attack scenarios, and corresponding mitigation strategies. Furthermore, the integration of VBM procedures within a cyber-physical systems (CPS) framework highlights the processes and threats that span cyber and physical components as well as the interdependence of the overall system. This is consistent with the observations of Rinaldi, Peerenboom, and Kelly [66], who examine the interconnected nature of modern critical infrastructure. Voting systems are integral to social and political structures and therefore require a holistic security approach to maintain their integrity [67]. Our paper aligns with these perspectives to provide a comprehensive and dynamic approach to safeguarding the voting process.

5.3 Markov Chain Framework

We introduce a DTMC framework to evaluate VBM performance and assess its associated risks which captures the dynamic nature of the VBM process that unfolds over several months. We adopt a layered approach in the DTMC model, comprising a *process layer*, an *attacks layer*, and a *mitigations layer*. This layered structure reflects the physical components, process, and informational interdependencies among the ballots affected by attacks, mitigations, and recourse actions. Furthermore, the DTMC allows for a nuanced

understanding of the impact of policy implementations on the VBM process. In this section, we describe these three layers and discuss VBM performance. Although exact details of VBM operations may vary slightly according to a particular municipality in the US, the overall VBM process is similar between municipalities. Therefore, our high-level model of the VBM process effectively captures the fundamental dynamics and can be used to identify insights under different settings.

5.3.1 Process Layer

First, we introduce seven process states that capture the movement of ballots in the process layer and define the operation of the VBM system. Each of these process states is also a state in the DTMC, and the states represent the change in physical location of the ballot over time. The voter requests a ballot in state I, the election office then fulfills this request in state II and mails an unmarked ballot to the voter. In state III the unmarked ballot is handled by the United States Postal Service (USPS) and is in transit to the voter. In state IV the unmarked ballot reaches the voter, however, it is up to the voter when the marked ballot is filled out and returned. In state V the voter selects the method of return for the marked ballot. Standard operation for returning a ballot is a transition to state VI via the USPS. Alternatively, the voter could return the ballot by drop box, if drop boxes are available. The final process state VII occurs when the marked ballot reaches the election office and is held until processing on election day. The ballot process is summarized by the following states:

- **I:** Ballot requested by voter.
- **II:** Unmarked ballot sent from election office.
- **III:** Unmarked ballot in-transit via USPS.
- **IV:** Voter marks ballot.
- **V:** Voter returns marked ballot via USPS or drop box.
- **VI:** Marked ballot in-transit via USPS.
- **VII:** Marked ballot processed at election office and held to be counted on election day.

5.3.2 Final Ballot States

At the end of the VBM process, ballots move to a final absorbing state that reflects the ballot status and is used to evaluate election performance. The final states capture all possible final states the voter ballot can take on at the end of the process, and they are defined to be mutually exclusive. The voter ballots can be counted (C) or not counted (NC) by the election officials. There are several reasons a ballot may not be counted, and not all result from attacks to the process. Some ballots are not counted simply

because they arrive at the election office after the required date for processing (L), which is typically an election day.

In traditional VBM processes the only ballot states that are recorded are not counted and counted ballots, and all ballots are assumed to be unaltered (U). Some ballots can be altered (A) by masquerade attacks. Since an altered ballot may not be observable, altered ballots could be counted or not counted. Additionally, ballots may be lost, and all lost ballots are not counted and listed as not returned (NR).

Together, a ballot can be in one of the six following final ballot states at the end of the time horizon considered in the DTMC:

- **(C,U):** Voter ballot is received on-time unaltered in-transit, accepted, and counted on election day.
- **(NC,U):** Voter ballot arrives on-time and unaltered, however it is rejected as incomplete.
- **(NC,L):** Voter ballot is returned and does not arrive at the election office by election day, designated late ballot.
- **(C,A):** Voter ballot is maliciously altered, accepted, and counted on election day
- **(NC,A):** Voter ballot is maliciously altered, however it is rejected and not counted on election day.
- **(NC,NR):** Voter does not return the ballot, includes ballots lost in transit and ballots never delivered [92].

These final ballot states allow us to quantify the impact of various risks on an election. Note that the preferred ballot status is for ballots to reach the (C,U) state.

5.3.3 Attacks Layer

The attacks layer is comprised of active vulnerabilities to the VBM process and can reflect different attacker goals, including changing the outcome of an election or eroding trust in political systems. Vulnerabilities in the VBM system are documented in attack trees by the US Election Assistance Commission [85] and have been expanded upon since the 2020 General Election [68]. The terminal nodes of these attack trees represent "access" points to additional vulnerabilities within the process. For this reason, we model each of these terminal leaf nodes of the attack trees. Attack trees, a valuable tool in cybersecurity, help assess threats to a system [72]. They depict potential vulnerabilities and attacker paths in a Boolean logic tree structure, where the root node represents the ultimate goal of an attacker. Attack trees use a combination of Boolean AND($\times$) and OR($+$) logic gates to trace paths from terminal attack nodes back to the root node. Figure 5.1 shows an example of an attack tree that visualizes two attacks corresponding to the malicious loss ($X13$) and accidental loss ($X14$) of a ballot. In Figure 5.1, terminal attack nodes $X13$ and $X14$ form part of an attack path leading to a successful insider attack, represented by root node 1 with only OR-gates along the paths. These attack paths offer a static view

of threats within the VBM process, with the terminal leaf nodes $X13$ and $X14$ serving as entry points.

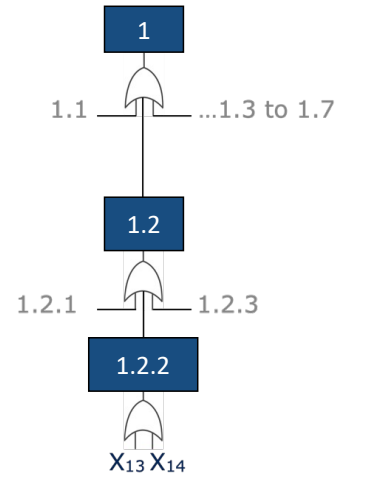

Figure 5.1: Example of a portion of the VBM attack tree

Haseltine, Wang, and Albert explore the impact of attacks on the VBM process, focusing on how each attack could affect voter ballots in terms of being lost, late, or maliciously altered. Expanding upon this, here we classify attacks not only based on their impact on ballots but also on the VBM process as a whole. Terminal attack nodes are classified into three categories: *fail rate increase*, where attacks change the probability associated with an undesirable outcome; *ballot altering*, which involves direct tampering with the voter ballot; and *process-altering*, where attacks cause voter ballots to deviate from the standard physical VBM process. This broader perspective allows for a more comprehensive understanding of the attacks' implications.

Any attack that affects the probability associated with an outcome, e.g., a transition probability in the DTMC model, is considered to be a *fail-rate* increase. These attacks can be non-malicious, e.g., caused by voter errors when filling out a ballot. As a result, the election office has a higher rejection rate for returned ballots, resulting in uncounted votes. However, these attacks can also be malicious, e.g., caused by a bad actor at the election office erroneously failing ballots. For example, consider a malicious insider attack ($X9$) in which a VBM ballot is erroneously deemed insufficiently filled out (errant failed signature). If the malicious attack occurs, it increases the probability that a completed ballot is rejected.

Changing the vote on a marked ballot alters the information in the original ballot we consider a "ballot-altering attack." Masquerade attacks involve a bad actor maliciously

altering a ballot. These attacks are more complex in that they create alternate ballot states in the DTMC that are needed to reflect the ballot's change in composition. For example, a masquerade attack in which a bad actor votes on behalf of someone in a central housing situation requires several steps. The malicious actor must first register to vote on behalf of the original voter, intercept their mail, mark the ballot, and return the altered ballot to the VBM process. We designated this type of attack as ballot modifying. It is important to note that modified ballots still have the potential to be counted. For this reason, there is an alternate path for the modified ballots in the DTMC. This is later illustrated in the DTMC model attack layer in Section 5.4, which reflects the VBM process states in the process layer to indicate that the ballots of those states were maliciously altered during transit.

In the VBM system, there are seven distinct states that an original ballot must pass through to be counted and remain unaltered (see Section 5.3.1). When someone attempts to disrupt this process, it is known as a *process-altering* attack. For example, suppose an election volunteer steals the marked ballot while it is in transit to the election office. This malicious attack alters the ballot process by holding the ballot in an "attack state" outside of the normal stages of the election process until the voter is notified and uses a recourse action such as requesting a replacement ballot, or it is deemed not returned and is not counted. The notable implication of process-altering attacks is that they can delay or prevent ballots from reaching the election office.

Table 5.1 presents the list of terminal attack nodes to the VBM system from Scala et al. We focus on those attacks that require voter or election office recourse actions to be countered. The first column lists the updated terminal attack nodes in the VBM attack tree, while the second column identifies the specific stage in the VBM process affected by each attack. The third column indicates the final state of the ballot if the attack is successful. Table 5.1 delineates the intent and classification of each terminal node attack in the last two columns, offering insights into the nature of these threats. Table 5.1 boldface rows are considered the most significant risks to VBM. These attacks are considered significant attacks, since they introduce substantial variations in the final ballot states, thereby highlighting the relevance to our analysis.

There are malicious and non-malicious attack types. Malicious attacks are targeted and have a high local impact for a limited period of time (e.g., a day) of being active. Conversely, non-malicious attacks (e.g., voter error and accidental loss) are accidental and could occur any time in the DTMC model time horizon. For example, in the 2020 General Election, some ballots were not counted due to voters failing to sign or bundle the ballots correctly (attack $X67$). This non-malicious attack results in ballots being rejected by the election office, thus inhibiting the VBM process if not countered. We express this attack as a small probability of occurrence for each ballot over the entire evaluation period.

5.3.4 Mitigations Layer

The mitigations layer in the DTMC model incorporates processes and actions designed to counteract both malicious and non-malicious threats. A mitigation is a recourse action available to counter any attacks to ballots in the VBM process. We build on previously

Table 5.1: Modeled VBM terminal attack nodes and their attributes

Attack tree	Terminal attack node linkage to Process Layer	Final ballot state for successful attack	Intent	Classification
T 1.2.1.1.1 (X8) Judge misinterprets rule	VII	NC, U	Malicious	Fail rate increase
T 1.2.1.1.2 (X9) Errant failed signature	VII	NC, U	Malicious	Fail rate increase
T 1.2.1.2.1 (X10) Challenge signature	VII	NC, U	Malicious	Fail rate increase
T 1.2.1.2.2 (X11) Challenge postmark	VII	NC, U	Malicious	Fail rate increase
T 1.2.1.2.3 (X12) Challenge intent	VII	NC, U	Malicious	Fail rate increase
T 1.2.2.1 (X13) Malicious loss	VI	NC, NR	Malicious	Process altering
T 1.2.2.2 (X14) Accidental loss	VI	NC, NR	Non-malicious	Process altering
T 1.5.1.1 (X28) Fail to stuff envelope	II	NC,A	Malicious	Ballot modifying
T 1.5.1.2 (X29) Send wrong or pre marked ballot	II	C,A	Malicious	Ballot modifying
T 1.5.1.3 (X30) Mis-address envelope (to voter)	II	NC,NR	Non-malicious	Process altering
T 1.5.3.1 (X36) accidentally lost in the mail room	VII	NC, NR	Non-malicious	Process altering
T 1.5.3.2 (X37) Mailbox attack	VI	NC, NR	Malicious	Process altering
T 1.7.2 (X84) Vote denied	VII	NC,U	Malicious	Process altering
T 2.3.1 (X43) Identify target residents	IV	C,A	Malicious	Ballot modifying
T 2.3.2 (X44) Register them	IV	C,A	Malicious	Ballot modifying
T 2.3.3 (X45) Intercept, mark, and return their ballot	IV	C,A	Malicious	Ballot modifying
T 2.3.4.1 (X46) Register as the voter	IV	C,A	Malicious	Ballot modifying
T 2.3.4.2 (X47) Forge the signature	IV	C,A	Malicious	Ballot modifying
T 2.4.1 (X48) Identify target	IV	C,A	Malicious	Ballot modifying
T 2.4.2 (X49) Steal blank ballot from mailbox	IV	C,A	Malicious	Ballot modifying
T 2.4.3 (X50) Receive, mark, return their ballots	IV	C,A	Malicious	Ballot modifying
T 2.4.4.1 (X51) Register as the voter	IV	C,A	Malicious	Ballot modifying
T 2.4.4.2 (X52) Forge the signature	IV	C,A	Malicious	Ballot modifying
T 2.5 (X53) Malicious "messenger ballots"	II	C,A	Malicious	Ballot modifying
T 2.8.1 (X93) Steal blank ballot from mailbox	IV	C,A	Malicious	Ballot modifying
T 2.8.2 (X94) Mark and return their ballot	IV	C,A	Malicious	Ballot modifying
T 2.8.3 (X95) Defeat signature check	IV	C,A	Malicious	Ballot modifying
T 4.1.1 (X65) Failure to sign correctly	IV	NC,U	Non-malicious	Process altering
T 4.1.2 (X66) Signature mismatch	IV	NC,U	Non-malicious	Process altering
T 4.1.3 (X67) Failure to bundle correctly	IV	NC,U	Non-malicious	Process altering
T 4.1.4 (X68) Failure to meet time requirements	IV	NC, L	Non-malicious	Process altering

published models of VBM mitigations to identify those that counteract attacks on the VBM process [30].

The mitigation layer connects with both the process and attack layers. We model structural mitigations both the physical pathways (arcs) and the probabilities (transitions) of a ballot's progression. These require specific responsive measures from voters to negate the effects of attacks. The DTMC model consists of five structural mitigations $M3, M4, M5, M6, M7$; Table 5.2 details each mitigation. We derived these mitigations from prior work that condense countermeasures published by CISA into practical logic intertwined with the VBM process [30, 14]. Table 5.2 presents these mitigations, categorized according to their operational nature along with the controlling entity, election office or voter.

This framework necessitates a thorough understanding of the connections between terminal node attacks in the VBM process and the available mitigation strategies designed to counteract them. To facilitate this, we employ a framework that examines the impact of terminal node attacks on the VBM process and incorporates recourse actions available to counter these attacks. Table 5.3 illustrates the relationship between attacks and mitigations. In this table, an "O" indicates that a mitigation strategy can counter the threat without impacting the ballot processing time. In contrast, a "D" signifies that while mitigation can counter the threat, it introduces a delay in the VBM process. For example, the scenario of a lost ballot ($X13$), a type of malicious attack, causes a delay resulting from the mitigation logic that includes the time required to notify the voter

Table 5.2: Mitigations available for the VBM process

Mitigation name	**Label**	**Mitigation description**	**Controlling entity**
Automatic ballot notifications	**M3**	The ability of a voter to attain the status of their ballot. This has a high probability if automatic notifications are provided via BalloTrax/Ballot Scout.	Election office
Replacement ballots	**M4**	Replacement ballot package request	Voter
Automatic ballot reminders	**M5**	Notify voter to send ballot back before deadline	Election office
Early voting	**M6**	In-person absentee voting	Voter
Drop boxes	**M7**	Return ballot via drop box	Voter

about their missing ballot followed by the time it takes for the voter to decide on an appropriate recourse action, such as M4 or M6. This scenario exemplifies how Table 5.3 summarizes key linkages in the DTMC model introduced in the next section. Consistent with the attacks in Table 5.1, terminal attack nodes listed in bold are incorporated into the DTMC model and are included in the computational results.

Table 5.3: Linkages between terminal attack nodes and mitigations

Terminal Attack Nodes of VBM Attack Tree	**M3 Automated notification of ballot status**	**M4 Replacement ballot voter request**	**M5 Notify voters to send ballot back earlier**	**M6 In-person absentee voting**	**M7 Return ballot via drop boxes**
T 1.2.1.1.1 (X8) Judge misinterprets rule	D	D		D	
T 1.2.1.1.2 (X9) Errant failed signature	**D**	**D**		**D**	
T 1.2.1.2.1 (X10) Challenge signature	D	D		D	
T 1.2.1.2.2 (X11) Challenge postmark	D	D		D	
T 1.2.1.2.3 (X12) Challenge intent	D	D		D	
T 1.2.2.1 (X13) Malicious loss	**D**	**D**		**O**	**O**
T 1.2.2.2 (X14) Accidental loss	**D**	**D**		**O**	**O**
T 1.5.1.1 (X28) Fail to stuff envelope		D			
T 1.5.1.2 (X29) Send wrong or pre marked ballot	**D**	**D**			
T 1.5.1.3 (X30) Mis-address envelope (to voter)	**D**	**D**		**O**	
T 1.5.3.1 (X36) accidentally lost in the mailroom					
T 1.5.3.2 (X37) Mailbox \ Dropbox attack	D	D		O	
T 1.7.2 (X84) Vote denied	D	D			
T 2.3.1 (X43) Identify target residents	D	D			
T 2.3.2 (X44) Register them	D	D			
T 2.3.3 (X45) Intercept, mark, and return their ballot	D	D			
T 2.3.4.1 (X46) Register as the voter	D	D			
T 2.3.4.2 (X47) Forge the signature	D	D			
T 2.4.1 (X48) Identify target	D	D		O	
T 2.4.2 (X49) Steal blank ballot from mailbox	D	D		O	
T 2.4.3 (X50) Receive, mark, return their ballots	D	D		O	
T 2.4.4.1 (X51) Register as the voter	D	D			
T 2.4.4.2 (X52) Forge the signature	D	D			
T 2.5 (X53) Malicious “messenger ballots”	D	D			
T 2.8.1 (X93) Steal blank ballot from mailbox	**D**	**D**			
T 2.8.2 (X94) Mark and return their ballot	**D**	**D**			
T 2.8.3 (X95) Defeat signature check	**D**	**D**			
T 4.1.1 (X65) Failure to sign correctly	**D**	**D**			
T 4.1.2 (X66) Signature mismatch	**D**	**D**			
T 4.1.3 (X67) Failure to bundle correctly	**D**	**D**			
T 4.1.4 (X68) Failure to meet time requirements			**O**	**O**	

5.4 Time inhomogeneous DTMC model

In this section, we introduce a DTMC model of the VBM system based on a multi-layer configuration of the process, attacks, and mitigations layers. The DTMC model captures the stochastic movement of ballots through various stages from a ballot request to the counting of ballots. This approach allows us to analyze how different factors influence overall election performance. A crucial aspect of the DTMC model is its ability to delineate the interaction between the terminal attack nodes, the mitigation layer, and the VBM process.

The time-inhomogeneous DTMC operates over a finite time horizon starting at time step $t = 1$ and continues until the final time step T. The time between time steps is one day, with the state reflecting the system's state at the end of day t. Let $t = 1$ capture the earliest time election officials process requests for absentee ballots. The election is held at time step $T - 1$, under the assumption that ballots are not accepted after an election day. The model can easily be adapted to consider accepting ballots postmarked by election day by adding extra time steps.

On election day, there are transition probability arcs re-positioned to move a ballot to its final post-election status in the last period, $t = T$, to evaluate the performance measures. Given a random process V_t with n finite states, the one-step transition probability of the process moving from state i in time step t to state j in a single time step is

$$P_t(i, j) = P(V_{t+1} = j | V_t = i) \quad \forall i, j \in \{1, \cdots, n\}. \tag{5.1}$$

We define a transition probability matrix P_t for each time step ($t = 1, 2, \cdots, T - 1$) to calculate the ballot states at the end of each time step [76]. Figure 5.2 illustrates the DTMC model state diagram for $t < T-1$, showing the final ballot states along with three distinct layers: the process layer, the attack layer, and the mitigation layer. The DTMC model comprises 30 finite states $S \in V_t$, categorized into six recurrent final ballot states and 24 transient states. Next, we describe the DTMC states and transition probabilities, starting with the process layer and then adding the attack and mitigation layers. The voting process initiates in the "I" state within the process layer, where voters begin the process by requesting a ballot. The ballot then navigates through various states in the process layer until it reaches one of the final ballot states. Table 5.1 details the interactions and connections between these states in the VBM process layer and the attacks layer. In Figure 5.2, different shapes represent distinct Markov states of the VBM system. Triangles depict attacks while circles depict the physical states of the ballots. Triangles labeled in black denote malicious attacks, whereas those in gold represent non-malicious attacks. When an attack is active, ballots transition from the process layer state to one of the triangular attack states. The "mitigations layer" encompasses recourse actions to counter attacks. Both $M5$ and $M7$ are illustrated in the process layer as these are *process-altering* mitigations. Mitigation $M5$ affects the rate of ballot returns from voters, and $M7$ enables voters to return their ballot in a drop box.

Additionally, Figure 5.2 uses line styles to convey information about the arc transition probabilities in the DTMC. Solid lines indicate transitions with non-zero probabilities at all steps, except at the final time step $T-1$. In contrast, dotted and dashed lines represent transitions with non-zero probabilities only at specific time steps, such as those associated

with transient malicious attacks. The complexity of the model leads to overlapping lines in the figure; intersections marked by a dot signal a connection to the intersecting line. In the model, masquerade attacks change a ballot's status and add a hidden attribute. Process layer states are mirrored in the attacks layer and labeled with an 'A' in Figure 5.2 to indicate the altered status of these ballots.

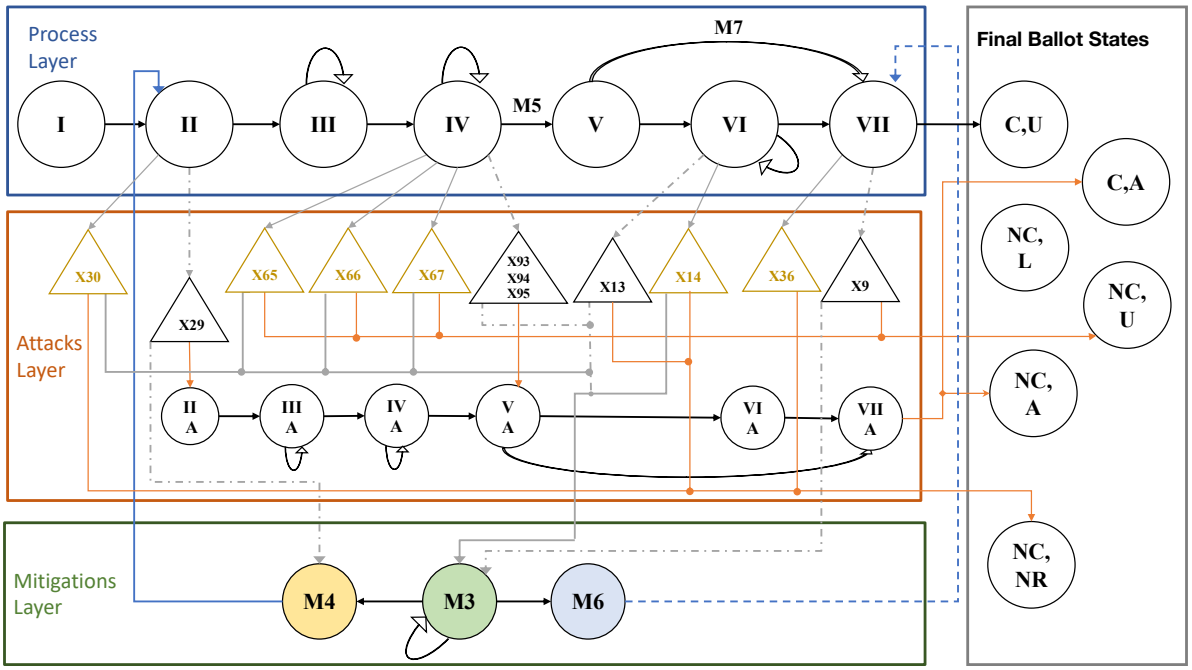

Figure 5.2: Layered Network for time intervals $1 \leq t < T - 1$

The DTMC model allows us to study VBM system performance as reflected by the final states of ballots over the course of an election cycle. We can determine the number of ballots that end up in the final desired state of "Counted, Unaltered" (C, U) and the other final ballot states.

At the end of the time horizon, the ballots move to the final ballot states. Figure 5.3 shows the non-zero transition probabilities on election day at time step $T - 1$. If ballots are not returned to the election office, they are "not counted, not returned" (NC, NR). These six final ballot states are recurrent in the DTMC, and all others are transient. If ballots are returned to the election office at $t = T$ they are "not returned, late"(NC, L).

Next, we summarize the transitions. Let P_t capture the transition probability matrix immediately after the time steps $t = 1, 2, ..., T - 1$. Voters can request ballots at different times. Let β_t capture the number of ballots requested at time $t = 1, 2, ..., T - 1$, which reflects the distribution of times when voters request absentee ballots. For the ballots requested at time t, let α_t be the vector of the probability mass function for the starting states of the ballot, where all the ballots are initiated in state I, that is, $\alpha_t(I) = 1$ and $\alpha_t(s) \geq 0$ for all other states $s \neq I$. For a ballot requested at time t, we can compute the vector of state probabilities at the end of the time horizon ω_t at time step T after all

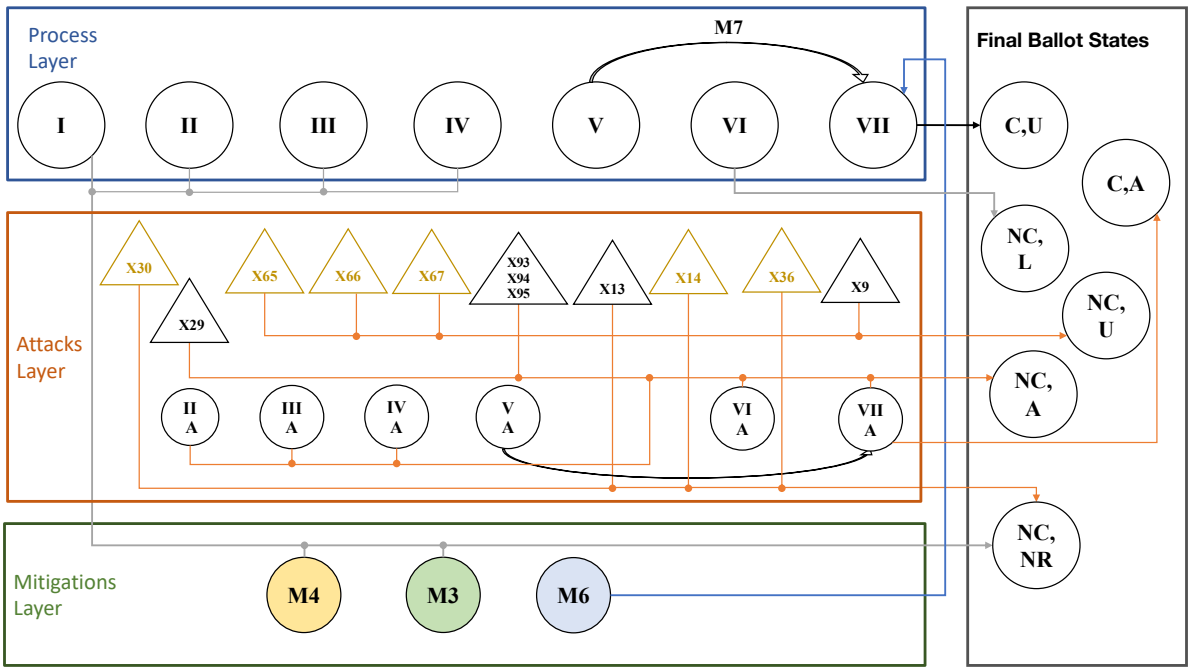

Figure 5.3: Layered Network for VBM on Election Day at time interval $t \geq T-1$

ballots have been evaluated and transitioned to one of the ballot status states:

$$\omega_t = \alpha_t \prod_{t'=t}^{T-1} P_{t'}, t = 1, 2, ..., T-1. \tag{5.2}$$

Then, we can compute the overall distribution of final ballot statuses as $\sum_{t=1}^{T-1} \beta_t \omega_t$ that reflects the expected number of ballots in each final ballot state.

5.5 Wisconsin 2020 General Election Case Study

We conduct a case study based on data from the 2020 General Election in Milwaukee County, Wisconsin. In 2020 Milwaukee County, the largest county in Wisconsin, had 939,489 residents and 478 voting wards [86, 12, 52]. Milwaukee County election officials used multiple absentee voting mechanisms during the 2020 General Election, including VBM with designated drop boxes and in-person absentee voting. The state of Wisconsin allowed absentee voting with "no excuses" and sent voter reminders to return ballots [55]. At that time, the county had 550,132 registered voters and experienced a turnout of 83.67 percent [89]. There are no known malicious attacks on VBM in Milwaukee County in the 2020 General Election, although there were two convictions elsewhere in Wisconsin associated with independent instances of VBM election fraud, each affecting a single ballot [83].

We assembled a data set for Milwaukee County using detailed, publicly reported information regarding absentee voting rates [91, 51]. The Wisconsin Elections Commission reports a daily tally of the following events for each county prior to the election [90]:

1. absentee ballot requests,
2. ballots mailed out by an election office,
3. absentee ballots returned by mail, and
4. in-person absentee ballots cast.

Using this data, we determined the daily number of absentee ballots requested, β_t, for $t = 1, 2, \ldots, T-1$, with 94.1 percent of these ballots being returned. The procedures and recourse actions available to voters during the General Election on November 3, 2020 are represented in Table 5.2 as mitigations. All requested ballots correspond to the starting state of the DTMC state I. Additional data were obtained from the MIT Election Data + Science Lab and the United States Office of the Inspector General [53, 35]. The case study begins with establishing transition probabilities for each arc in the process for every time interval $t = 1, 2, ..., T$ in the DTMC model.

5.5.1 Time

In the VBM process, ballot states of the DTMC model transitions on a daily basis. The case study begins on September 17, 2020 ($t = 1$), the first day ballots are mailed to voters, and ends after election day, November 4, 2020 ($T = 49$). The DTMC transition probabilities are time-dependent in accounting for procedure changes over the VBM timeline. For example, some mitigations are not available at all times due to VBM procedure, and malicious attacks occur at various times. However, many of the transition probabilities are homogeneous across time intervals. As a result, we partition the time horizon into four segments with transition probabilities that are time-homogeneous within an interval.

In the first interval, absentee voting opens and ballots are mailed to voters. The second interval begins when in-person absentee voting opens for the election. To reflect this procedure, mitigation $M6$ (related to in-person absentee voting) is unavailable during the first time interval. In the third time interval, ballots are no longer mailed to voters requesting absentee voting. During this interval, there is no longer a connection between the process states I and II. Instead, the connection from state I changes to $M6$, indicating that voters can only vote in-person absentee during this period. The fourth-time interval is election day and the days following, during which ballots are processed. Ballots that have not reached the USPS or the election office by election day are not counted. The exact final destination of the ballots depends on their location on election day. The model configuration in Figure 5.3 outlines all arc connections from the voter requesting a ballot, I of the process layer to one of the "final ballot states." In summary, the case study defines the time intervals as follows:

- interval 1 starts on September 17, 2020 ($t = 1$ to $t = 34$),
- interval 2 starts on October 21, 2020 ($t = 35$ to $t = 42$),
- interval 3 starts on October 29, 2020 ($t = 43$ to $t = 47$), and
- interval 4 starts on November 3, 2020 ($t = 48$ to $t = 49$).

Table 5.4: Arc transition probabilities

From node	To node	Function	Interval 1	Interval 2	Interval 3	Justification
I	II	Voter requests ballot to election office	1	1	1	All voter requests show being received by election office
II	III	Unmarked ballot sent from election office	1	0.2	0	All ballot requests filled by the election office
II	M6	Ballots requests are deferred to in-person absentee	0	0.8	1	WI policy for 2020 General Election [90]
II	X30	Ballot envelope mis-addressed to voter (undelivered)	0.0343	0.0343	0.0343	Calibration, Wisconsin Elections Commission [92]
III	IV	Unmarked ballot in transit via USPS	0.938	0.938	0.938	Office of Inspector General' [35]
III	III	Unmarked ballot remains at USPS past one cycle	0.062	0.062	0.062	Office of Inspector General' [35]
IV	IV	Voter waits one day to return ballot	0.947	0.797	0.947	Calibration, Wisconsin Elections Commission [92]
IV	V	Voter returns marked ballot	0.05	0.79	0.79	Calibration, Wisconsin Elections Commission [92]
IV	X65+X66+X67	Ballot subject to non-malicious attack of voter error voter error	0.000162	0.000162	0.000162	Calibration, Wisconsin Elections Commission [92]
V	VI	Marked ballot returned via USPS	0.485	0.485	0.485	MIT Election Data + Science Lab [53]
V	VII	Marked ballot returned via Dropbox	0.515	0.515	0.515	MIT Election Data + Science Lab [53]
VI	VII	Marked ballot in transit and received at election office	0.938	0.938	0.938	Inspector General' [35]
VI	X14	Marked ballot lost at USPS, non-malicious	0.0343	0.0343	0.0343	Calibration, Wisconsin Elections Commission [92]
VI	VI	Marked ballot remains at USPS past one cycle	0.061	0.061	0.061	Office of Inspector General' [35]
VII	(C,U)	Marked ballot processed at election office	0.966	0.966	0.966	Ballots are be counted and unaltered if not subject to attack
VII	X36	Marked ballot lost at election office, non-malicious	0.0343	0.0343	0.0343	Calibration, Wisconsin Elections Commission [92]
M4	II	Ballot replacement request	1	1	1	Once mitigation is reached, request are be made to election office
M3	M4	Voter is notified ballot has problem and chooses replacement	0.5	0.33	0.33	Equal probability for active mitigations
M3	M6	Voter is notified ballot has problem and chooses in-person vote	0	0.33	0.33	Equal probability for active mitigations
M3	M3	Voter is notified ballot has problem and chooses no recourse action	0.5	0.34	0.34	Equal probability for active mitigations
M6	VII	In-person absentee voting	1	1	1	Ballot successfully submitted
II-A	III-A	Modified ballot sent to the voter	1	1	1	WI timeline for absentee voting [93]
III-A	IV-A	Modified ballot received by the voter	0.9	0.9	0.9	Altered ballot received by the voter
III-A	III-A	Modified ballot remains in transit beyond one cycle	0.1	0.1	0.1	Attacked ballots movement low
IV-A	IV-A	Modified ballot is not filled out in one cycle	0.1	0.1	0.1	Approximation (fast return for malicious intent)
IV-A	V-A	Modified ballot returned	0.9	0.9	0.9	Approximation (fast return for malicious intent)
V-A	VI-A	Modified ballot returned via USPS	0.485	0.485	0.485	MIT Election Data + Science Lab [53]
V-A	VII-A	Modified ballot returned via Dropbox	0.515	0.515	0.515	MIT Election Data + Science Lab [53]
VI-A	VII-A	Modified ballot returned to election office on-time	1	1	1	Office of Inspector General' [35]
VII-A	(C,A)	Modified ballot counted	0.5	0.5	0.5	Equal probability of being counted unless other mitigations are in place
VII-A	(NC,A)	Modified ballot rejected at election office	0.5	0.5	0.5	Equal probability of being counted unless other mitigations are in place
X14	M3	Voter notified of non-malicious attack of lost ballot	0.0265	0.0265	0.0265	Calibration, Wisconsin Elections Commission [92]
X14	(NC,NR)	Marked ballot lost in process	0.974	0.974	0.974	Inverse of M3 availability
X36	(NC, NR)	Ballot lost in election office	1	1	1	No monitoring available, no recourse
X65+X66+X67	M3	Voter made aware of errors in ballot package	0.0265	0.0265	0.0265	Calibration, Wisconsin Elections Commission [92]
X65/X67	(NC,A)	Marked ballot rejected at election office	0.974	0.974	0.974	Inverse of M3 availability
X93-X95	V-A	Masquerade attack leading altered ballot path	0.10	0.10	0.10	Test various strengths of attack on various days
X93-X95	M4	Mitigation M4 availability	0.90	0.90	0.90	Test various strengths of attack on various days
X9	(NC,A)	Malicious challenge of ballot signature successful	0.974	0.974	0.974	Test various strengths of attack on various days
X9	M3	Voter notified of failed ballot submission	0.0265	0.0265	0.0265	Calibration, Wisconsin Elections Commission [92]
X29	III-A	Malicious incomplete ballot sent to voter leading to altered ballot path	0.10	0.10	0.10	Test various strengths of attack on various days
X29	M4	Voter requests replacement ballot	0.90	0.90	0.90	Test various strengths of attack on various days

5.5.2 Transition probabilities

We define the transition probability matrices as follows. We first define the transition probabilities for the *baseline model* that only considers non-malicious attacks. Non-malicious attacks are represented as consistent threats with time-inhomogenous probabilities of occurrence. Later, we consider malicious attacks with a duration of one day.

Table 5.4 outlines the transition probabilities for each arc active in the baseline model. The first and second columns define the two DTMC states associated with a non-zero transition probability. The next column provides a brief functional description of the ballot state transition that the arc represents. The columns labeled Interval 1, 2 and 3 represent the values of associated transition probability across time intervals 1, 2, and 3. We omit interval 4, since these transition probabilities are 0 or 1 with arcs illustrated in Figure 5.3. The justification column reports a reference for each transition probability value and denotes which arcs were adjusted through calibration. The arc configuration aligns with the model shown in Figure 5.2.

We discuss several transition probabilities. The transition probability from process state III to IV is 0.938, which reflects the proportion of election mail that is processed on-time as reported by the USPS Inspector General [58]. The rest of the ballots remain in state III. The transition from state IV to V reflects the return of completed ballots by voters (that is, the voter rate of ballot return). This transition captures the proportion of voters who return the marked ballot within a day of receiving it in the mail, which is determined as a part of the model calibration since this value is not directly recorded. MIT Election Data + Science Lab provide the values for utilization of drop boxes over USPS to return ballots. Transitions from state V to state VII, which occur with a probability of 0.515, represents the proportion of ballots returned by drop boxes. [53] Transitions at the end of the time interval move ballots to their final ballot states with probability 1.0. These probabilities can be gleaned from Figure 5.3. All other transition probabilities are zero. We calibrated the DTMC using historical data from the WEC 2020 General Election report [92]. This involved adjusting the voter rate of ballot return, the arc probability from node IV to V, to align the output of (C,U) and (NC,U) ballots with the observed ballot return rate seen in aggregate by the state of Wisconsin. Following calibration, we validate the model's predictive accuracy by testing it against independent data validation points for Milwaukee County. This ensures blind prediction input data such that our model fits historical data and ensures the model is capable of making accurate predictions in varying scenarios.

5.5.3 Calibration

We calibrated the model using Wisconsin state-level data to align the recorded values of the ballots returned on various days with the calculated number of ballots counted and unaltered (C, U) final ballot state of the DTMC model. To accomplish this, we set the *voter rate of ballot return* to reflect the influence of mitigation $M5$ (reminders); the arc

Table 5.5: Baseline values for mitigation deployment level

Mitigation name and label	**Mitigation strength**	**Justification**
Automatic ballot notifications (M3)	0.0265	Automatic voter notifications of ballot status were not implemented
Replacement ballots (M4)	0.900	Voters had option of requesting replacement ballots
Automatic ballot reminders (M5)	0.740	Region used ballot return reminders
Early voting (M6)	0.400	In-person absentee was implemented
Drop boxes (M7)	0.520	Survey of the Performance of American Elections Dataverse [8]

transition probability from node IV to node V reflects the voter rate of ballots to return in intervals 2 and 3.

Figure 5.4 illustrates the cumulative number of ballots returned by day (solid line) compared to the expected number of ballots returned in the DTMC model (dotted line) using the calibrated return rates. The comparison of actual and modeled returned ballots for Milwaukee County is aligned with actual daily ballots returned (blue) and the baseline of the DTMC model for daily ballots returned (dashed). Note that all mitigations are not active in all time intervals. For example, the mitigation $M5$ for reminders to return ballots was not implemented until time $t = 35$ (interval 2). Figure 5.4 shows that the modeled baseline model aligns closely with real-world data.

We then consider the mitigations of the model to determine a starting point for policies active in the 2020 Wisconsin General Election. We study the availability of mitigations M3, M4, M5, M6, and M7. Mitigation strength refers to the effect of each mitigation on the magnitude of the associated arc transition probabilities. Table 5.5 lists the inputs associated with mitigation strength in the baseline model. Low strength (less than 0.05) is associated with a minimal impact of that mitigation on the VBM system. Wisconsin voters were able to view ballot notifications manually through myvote.wi.gov, Wisconsin did not implement automatic ballot notification in 2020, so this value is low to reflect that few voters manually checked their ballot statuses. We set mitigation $M3$ strength to 0.0265, which resulted in a mean absolute deviation of 0.24-percent from the observed number of (NC,U) ballots. Next, we set mitigation $M4$ to 0.90 to represent the ability of voters to request replacement ballots. We set mitigation $M5$ to 0.74 to represent the sent and advertised reminders to return ballots in the late second interval. In Table 5.4 we see the arc from node IV to node V increases in interval 2 to account for the increased rate of ballot return caused by implementation of mitigation $M5$. Next, we set mitigation $M6$ to 0.40 to represent the rate voters choose to submit an absentee ballot in person. Finally, we set the strength of $M7$ to the average use in the state of Wisconsin [53]. This results in a mean absolute deviation of 0.25-percent from the observed number of (NC,U) ballots.

The calibration of the baseline DTMC model also requires consideration of non-malicious attacks; we model as part of normal system operations. The *ballot rejection rate* of the model is associated with the arc transition probability from node IV to

$(X65 + X67 + X68)$. It correlates directly with the number of ballots rejected by the election office due to voter error. We set the ballot rejection rate to 0.000162 to closely match the observed number of returned ballots that were unaltered and not counted (NC,U). Next, we vary the impact of the remaining non-malicious attacks, representing the ballots that never returned to the election office. Recall terminal attack node $X14$ represents the accidental loss of ballots in the mail. The terminal attack node $X30$ is the misaddressing of ballots to the voter, and $X36$ represents the accidental loss of ballots in the election office. These non-malicious attacks all result in the same penalty, and the ballots impacted are considered not counted and not returned (NC,NR). Unlike the ballot rejection rate of the election office, there is a lack of evidence to support the different magnitudes for the non-malicious attacks $X14, X30, X36$. These non-malicious attack strength values are equivalent for all periods $t = 1, ..., T - 1$ [94]. Therefore, we set the terminal attacks $X14, X30, X36$ to equal strength. As a result, the following arcs have the same transition probability of 0.0343: arc from IV to $X14$; arc from II to $X30$; arc from VII to $X36$. These arcs result in a mean absolute deviation of 0.14-percent from the observed (C,U) ballots.

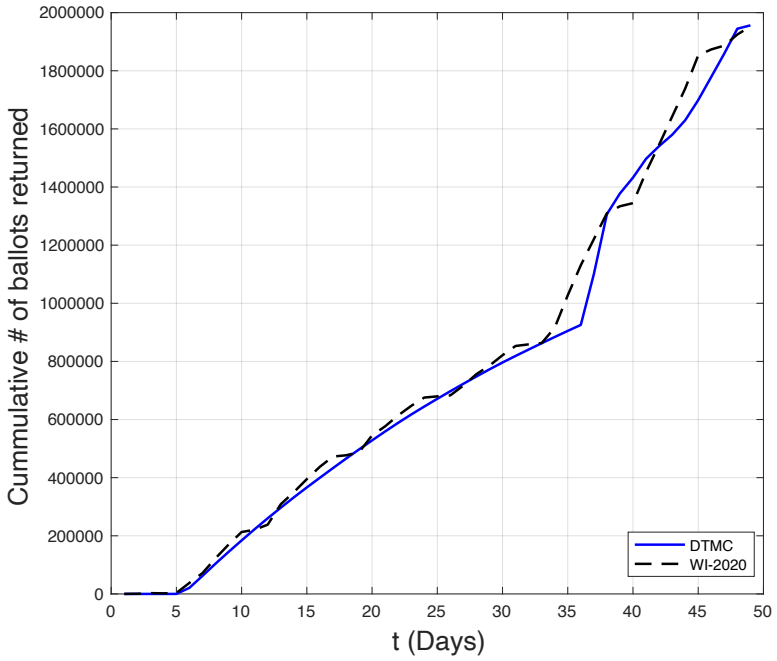

Figure 5.4: Comparison of recorded and modeled returned ballots for the state of Wisconsin for the 2020 General Election. The graph shows daily ballots returned in blue and the DTMC model baseline for daily ballots returned as a dashed line.

5.5.4 Validation

To validate the DTMC, we align the model outputs with five validation points from the 2020 WEC report scaled to Milwaukee County by ratio of ballots returned [92]. The

Table 5.6: Model validation using WI Election Commission (WEC) data

Validation point	WEC ballot status	Corresponding DTMC final ballot states	WEC ballot counts	WEC ballot counts scaled to Milwaukee County	DTMC ballot counts
v1	Ballots returned and counted	(C,U)+(C,A)	1,969,274	324,896	324,792
v2	Ballots not returned	(NC, NR)	85,586	20,349	20,371
v3	Ballot rejected by election office	(NC,U)+(NC,A)	3,225	532	559
v4	Ballot not returned before polls closed	(NC,Late)	1,045	176	173
v5	All sent ballots	County specific sum of all ballot states	2,059,130	325,547	325,351

validation points are as follows:

- **v1**: Captures all returned and counted ballots. WEC reports these values for Milwaukee County on each day of the election cycle.
- **v2**: Captures all ballots sent and not returned to the election office. WEC reports these values for Milwaukee County on each day of the election cycle.
- **v3**: Reflects the ballots returned to the election office and rejected due to non-malicious attacks ($X65$, $X66$, $X67$). WEC reports the state level for the rejection rate of ballots; population ratios are used to scale to Milwaukee County.
- **v4**: Reflects all ballots deemed late as of day 48 of the election cycle. Again, WEC reports the state level for late ballots; population ratios are used to scale for Milwaukee County.
- **v5**: Captures the sum of all sent ballots. WEC reports these values for Milwaukee County on each day of the election cycle.

Table 5.6 summarizes the validation points and relates them to the final ballot states of the model. These points were chosen based on their relevance to the predictive accuracy of the DTMC model, accounting for potential variations between state-level and county-level data. We expected some differences between case study values and validation points, since the validation values are aggregate for the state. Ratios indicating the proportion of ballots affected by various conditions were employed as a margin of error in the DTMC model. The last two columns of Table 5.6 show the number of ballots under each validation point, comparing real world ballot counts "WEC ballot counts scaled to Milwaukee" and the DTMC model ballot counts. The validation process, aligning the DTMC model with real-world data from the WEC report, reinforces the model's reliability in reflecting absentee voting behaviors and provides valuable insights into vote-by-mail dynamics in Wisconsin.The last column of Table 5.6 shows the DTMC model output values for the given validation points. Comparative analysis aligned the model's final ballot states with the WEC report's metrics. Once calibrated and validated the model, most transition probabilities were established for all time intervals. Milwaukee County data points are then used to establish a baseline model that can be used to consider multiple scenarios.

Table 5.7: Baseline DTMC model values and final ballot states

Scenario	Election office rejection rate	$X14, X30, X36$ attack strength	**Final Ballot States (Expected number of ballots)** (C,U)	(NC,U)	(C,A)	(NC,L)	(NC,NR)	(NC,A)
Baseline	0.000162	0.0343	324,792	559	0	173	20,371	0
Baseline that varies election office rejection rate from node IV to $(X65 + X67 + X68)$	0.0000707	0.0343	325,091	245	0	174	20,386	0
	0.000111	0.0343	324,958	385	0	174	20,379	0
	0.000172	0.0343	324,759	594	0	173	20,369	0
	0.000424	0.0343	323,933	1,464	0	173	20,325	0
	0.000818	0.0343	322,653	2,812	0	172	20,258	0
Baseline that varies non-malicious attack strength associated with $X14, X30, X36$	0.000162	0.00202	339,228	559	0	198	5,911	0
	0.000162	0.0192	331,493	559	0	184	13,660	0
	0.000162	0.0384	323,025	559	0	171	22,141	0
	0.000162	0.0576	314,732	560	0	158	30,445	0
	0.000162	0.0939	299,469	560	0	140	45,728	0

5.6 Computational Results

This section discusses the baseline DTMC model results for the Milwaukee County case study. The model was coded and implemented in Matlab, and the code and data is publicly available [27]. We then study the impact of several types of malicious attacks on the established baseline. We report the expected number of ballots in each of the final ballot states—rounded to the nearest ballot—under multiple scenarios.

5.6.1 Baseline model

We first analyze the VBM system under normal conditions with only non-malicious attacks in the baseline model introduced in Section 5.5. Non-malicious attacks (see Table 5.4) are included in the baseline model, since they are an uncontrolled, continuous part of the VBM process. Additionally, we perform a sensitivity analysis of four of the non-malicious attacks by varying their corresponding probabilities of occurring.

Table 5.7 summarizes the expected number of ballots in each final ballot state: counted and unaltered (C,U), not counted and unaltered (NC,U), correctly counted but altered (C,A), not counted due to late arrival (NC,L), not counted due to not being returned (NC,NR), and not counted and altered (NC,A). Table 5.7 also reports a sensitivity analysis that varies the election office rejection rate and three other non-malicious attack transition probabilities used in the model. The election office rejection rate varies based on ballots returned and rejected by the election office on election day due to non-malicious attacks $X65, X66$, and $X67$, according to subject matter expert testimony [6]. The non-malicious attack strength of attacks $X14, X30$, and $X36$ represent ballots not returned due to either not being delivered to the voter or being discarded at the election office before counting the valid returned ballots on election day [92].

The top row of Table 5.7 represents the baseline model, and it reports the expected number of ballots in each final ballot state. Each subsequent row represents a different scenario with varying attack parameters. The incremental increase of the rejection rate associated with the arc transition probability from node IV to $(X65 + X67 + X68)$ by a probability of 0.00001 increases the number of (NC,U) ballots by 6-percent (35 total ballots), suggesting that efforts to reduce the rejection rate can decrease the number of ballots that are not counted. Higher attack strengths associated with $X14, X30, X36$ increase the number of ballots not returned (NC,NR). For example, when the attach

strength increases from 0.0343 to 0.0576, the number of (NC,NR) ballots increase by $10,074$. Additionally, we note that the election office rejection rate ($X65 + X66 + X67$) for Milwaukee County is substantially lower than that of the other non-malicious attacks. These insights underscore the need for additional attention to unreturned (NC,NR) ballots in the VBM process and maintaining minimal non-malicious attack strengths to ensure that ballots are returned and counted.

5.6.2 Malicious attack scenarios given baseline

Next, we introduce malicious attacks to the baseline. We focus on malicious attacks $X9, X13$, and $X29$. Malicious attacks are modeled to last a single day during the 49-day time horizon to reflect feasible real-world election interference events, and we vary the attack strength of each attack to consider different scenarios. Malicious attacks therefore, modify the transition probability matrix (P_{t^*}) for a single day associated with the attack t^*. This approach allows us to assess the impact of attack timings and the mitigation strategies for countering attacks.

X9: Malicious attack to challenge the signature of a valid ballot

Malicious $X9$ attack captures bad actors in an election office erroneously rejecting marked ballots by challenging their signatures. This scenario, with a base arc transition probability of 0.055, is set to affect roughly one of 19 precincts in Milwaukee. Referencing Figure 5.2, malicious attack $X9$ impacts process state VII, where ballots are collected and verified by the election office. On a specific day, this could lead to a significant number of ballots being improperly discarded.

Table 5.8 details the impact of the $X9$ attack, showing variations in the expected number of ballots across the final ballot states under varying attack strengths and days. The top section reports the scenario of a medium-strength attack on different days. The following four sections of Table 5.8 selects a day in the interval and varies the attack strength. These comparisons show the impact of attack strength versus attack day. We pay particular attention to ballot state (NC,U), since attack $X9$ causes ballots to not reach this state. Recall that there are 559 ballots in the (NC,U) final ballot state (see Table 5.7). Table 5.8 reports how the attack day and attack strength affects the ballot outcomes. In the top section we find Day 38 to be particularly critical, in that 2,415 ballots end in the (NC,U) final ballot state. This increase in affected ballots is caused by increased ballot inflow due to active mitigations $M5$ and $M6$, initiated on day 36 (October 21, 2020). This finding underscores a vulnerability when attack strength is minimal but timed with peak ballot presence, highlighting critical periods when attacks have a larger scope of impact.

X13: Malicious attack discard ballot in transit via USPS

Malicious $X13$ attack represents bad actors in a postal office who discard completed ballots at the sorting or storage point. Referring back to Figure 5.2, we see that malicious attack $X13$ directly impacts process state VI where the voter returns the completed ballot

Table 5.8: Attack X9: The malicious attack of erroneously rejecting of ballots

Scenario	Day of malicious attack	Malicious attack strength	**Final Ballot States (Expected number of ballots)** (C,U)	(NC,U)	(C,A)	(NC,L)	(NC,NR)	(NC,A)
X9 active at medium strength varying attack days	10	0.055	324,341	933	0	173	20,449	0
	30	0.055	324,495	779	0	173	20,449	0
	36	0.055	324,524	751	0	173	20,449	0
	37	0.055	323,092	2,180	0	173	20,451	0
	38	0.055	322,857	2,415	0	173	20,451	0
	39	0.055	324,094	1,180	0	173	20,449	0
	40	0.055	324,286	988	0	173	20,449	0
	45	0.055	324,106	1,161	0	173	20,457	0
	48	0.055	324,053	1,221	0	173	20,449	0
Vary attack strength X9 with attacks in interval 1	10	0.050	324,375	899	0	173	20,449	0
	10	0.055	324,341	933	0	173	20,449	0
	10	0.075	324,205	1,069	0	173	20,450	0
	10	0.100	324,035	1,239	0	173	20,450	0
	30	0.050	324,515	759	0	173	20,449	0
	30	0.055	324,495	779	0	173	20,449	0
	30	0.075	324,415	859	0	173	20,449	0
	30	0.100	324,315	959	0	173	20,449	0
Vary attack strength X9 with attacks in interval 2	40	0.050	324,325	949	0	173	20,449	0
	40	0.055	324,286	988	0	173	20,449	0
	40	0.075	324,130	1,144	0	173	20,449	0
	40	0.100	323,935	1,339	0	173	20,450	0
Vary attack strength X9 with attacks in interval 3	45	0.050	324,161	1,106	0	173	20,456	0
	45	0.055	324,106	1,161	0	173	20,457	0
	45	0.075	323,884	1,380	0	173	20,460	0
	45	0.100	323,607	1,653	0	173	20,464	0
Vary attack strength X9 with attacks in interval 4	48	0.050	324,114	1,161	0	173	20,449	0
	48	0.055	324,053	1,221	0	173	20,449	0
	48	0.075	323,813	1,462	0	173	20,449	0
	48	0.100	323,512	1,763	0	173	20,449	0

Table 5.9: Attack X13: Ballot maliciously discarded in transit via USPS

Scenario	Day of malicious attack	Malicious attack strength	**Final Ballot States (Expected number of ballots)** (C,U)	(NC,U)	(C,A)	(NC,L)	(NC,NR)	(NC,A)
X13, are active at medium strength varying attack days	10	0.033	324,495	559	0	173	20,551	0
	36	0.033	324,602	559	0	173	20,502	0
	37	0.033	323,059	559	0	173	21,219	0
	38	0.033	324,300	559	0	173	20,642	0
	40	0.033	324,544	559	0	173	20,529	0
	45	0.033	324,649	559	0	173	20,480	0
Vary attack strength X13 with attacks in interval 1	10	0.025	324,521	559	0	173	20,525	0
	10	0.030	324,505	559	0	173	20,541	0
	10	0.040	324,473	559	0	173	20,574	0
	10	0.065	324,392	559	0	173	20,654	0
	30	0.025	324,601	559	0	173	20,494	0
	30	0.030	324,592	559	0	173	20,503	0
	30	0.040	324,573	559	0	173	20,522	0
	30	0.065	324,525	559	0	173	20,570	0
Vary attack strength X13 with attacks in interval 2	40	0.025	324,564	559	0	173	20,508	0
	40	0.030	324,551	559	0	173	20,521	0
	40	0.040	324,526	559	0	173	20,546	0
	40	0.065	324,464	559	0	173	20,609	0

Table 5.10: Attack X29: Intercept and pre-mark ballot to voter

Scenario	Day of malicious attack	Malicious attack strength	**Final Ballot States (Expected number of ballots)** (C,U)	(NC,U)	(C,A)	(NC,L)	(NC,NR)	(NC,A)
X29 are active at medium strength varying attack days	7	0.055	324,086	558	332	173	20,416	332
	8	0.055	324,466	559	132	173	20,436	132
	10	0.055	324,495	559	116	173	20,437	116
	20	0.055	324,528	559	99	173	20,439	99
	30	0.055	324,573	559	75	173	20,441	75
	40	0.055	324,519	560	99	173	20,438	109
Vary attack strength X29 with attacks in interval 1	20	0.025	324,630	559	45	173	20,444	45
	20	0.050	324,545	559	90	173	20,440	90
	20	0.075	324,460	559	135	173	20,435	135
	20	0.100	324,375	559	180	173	20,431	180
	30	0.025	324,651	559	34	173	20,445	34
	30	0.050	324,586	559	68	173	20,442	68
	30	0.075	324,521	559	102	173	20,439	102
	30	0.100	324,457	559	136	173	20,435	136
Vary attack strength X29 with attacks in interval 2	40	0.025	324,626	559	45	173	20,444	50
	40	0.050	324,536	559	90	173	20,439	90
	40	0.075	324,447	559	134	172	20,435	149
	40	0.100	324,358	559	179	172	20,430	198

via USPS. In Milwaukee County, a large-scale attack affects one of the 36 USPS offices for a day [50]. Consequently, there is a probability of 0.033 that ballots in process state VI are impacted by the $X13$ attack.

Table 5.9 summarizes the impact of attack $X13$, showing variations in the expected number of ballots across different final states under varying attack strengths and days. The top section reports the scenario of a medium strength attack on different days. The following two sections of Table 5.9 selects a day in the interval and varies the attack strength to determine the impact of attack strength versus attack day. Attack $X13$ affects the number of ballots that are not returned, leading to an increase in the (NC,NR) final ballot state. The highest number of (NC,NR) ballots occurs when there is a malicious attack on day 37 of the election cycle. There are 21,219 ballots expected to not be returned (NC, NR), an 848 additional ballots not returned when compared to the baseline.

X29: Masquerade Attacks, Ballot Modifying

We then consider the malicious attack $X29$, which alters voters' original ballots and directly impacts process states II. The consequence of this attack leads to altered ballots that are both counted and uncounted (final ballot states (C,A) and (NC,A)). The arc probability strengths from attack $X9$ are reflected here, since there exists no known reference of an executed masquerade attack.

We assume that there is an equal probability that the altered ballot are counted or uncounted. The analysis regarding attack $X29$ underscores the vulnerability of the voting process to pre-altered ballot distributions. This manipulation targets the early stages of ballot circulation, redirecting ballots from the process layer state II to the attack layer state II, A.

Table 5.10 summarizes the results associated with various days and strengths of attack $X29$. The first column lists the scenarios considered, and the next two columns report the attack day and strength associated with different scenarios. The remaining columns report the expected number of ballots in each of the final ballot states. The final ballot states (C, U) and (C, A) reflect the impact of attack $X29$. Table 5.10 shows the variations

in the intensity of the attack during two distinct intervals, namely intervals 1 and 2. Intervals 3 and 4 are not studied, since ballots are not mailed to voters in these intervals. The attack day greatly affects the impact of the attack on the VBM process, affecting up to 664 ballots on a day 7 attack. Attacks carried out in the initial stages of the election have a large impact, since many ballots are mailed at these times.

Overall, these results reveals a clear trend: initiating an attack in a high-demand phase of the election cycle results in a higher number of affected ballots.

Attack timing

We examine the impact of attack timing on three malicious attack scenarios: $X9$, $X13$, and $X29$. To do so, we vary the day of each attack from Day 7 to Day 49 and evaluate the change in the expected number of returned ballots that are counted and unaltered (C,U) compared to the baseline (see Table 5.6.2).

Figure 5.5 illustrates the deviations in the expected number of (C,U) ballots associated with each attack scenario as a function of the day of the attack. All values are negative, indicating that each attack reduces the number of ballots in the preferred (C,U) state. Attack $X9$, which models the false rejection of valid ballots at the election office (process state VII), experiences its largest deviation in (C,U) ballots when the attack is initiated on day 38. This attack peak aligns with a surge of ballots routed to the election office via early voting (M6), which begins on day 36 and introduces a direct arc from process state II to state VII, in addition to those being returned by mail or drop box. Similarly, attack $X13$, representing the loss of ballots in USPS transit, peaks on day 37 in terms of its deviation in the expected number of (C,U) ballots. This deviation increases due to ballot return reminders (M5) that start to alert voters on Day 36. This results in an increase in the flow of ballots in transit, which results in more ballots that are available to be maliciously stolen.

Attack $X29$, a masquerade attack involving stolen or pre-marked ballots (state II), experiences its largest deviation in (C,U) ballots earlier in the election cycle on day 7, which results in 706 fewer (C,U) ballots. This occurs due to the initial surge in ballot issuance and early returns at this time. A secondary deviation of 614 (C,U) ballots occurs on day 36, coinciding with an increase in voters returning ballots due to automatic ballot reminders (M5). These results indicate that attack timing——specifically its alignment with the flow of ballots——is crucial for determining the consequences of an attack. The consequences of an attack corresponds to its interaction with process dynamics during periods of high ballot flow.

5.6.3 Worst-Case Scenario Modeling

We build on the attack scenarios in Section 5.6.2 to create a "worst-case scenario" to evaluate various policy implementations in the VBM process under severe conditions as well as the impact on mitigations. The worst-case scenario involves the simultaneous execution of three malicious attacks within the 49-day election cycle. Figure 5.5 illustrates the highest impact for each attack as: $X9$ on day 38, $X13$ on day 37, and $X29$ on day 7. The following arc probabilities are all set to 0.10 on these days: arc from node IV

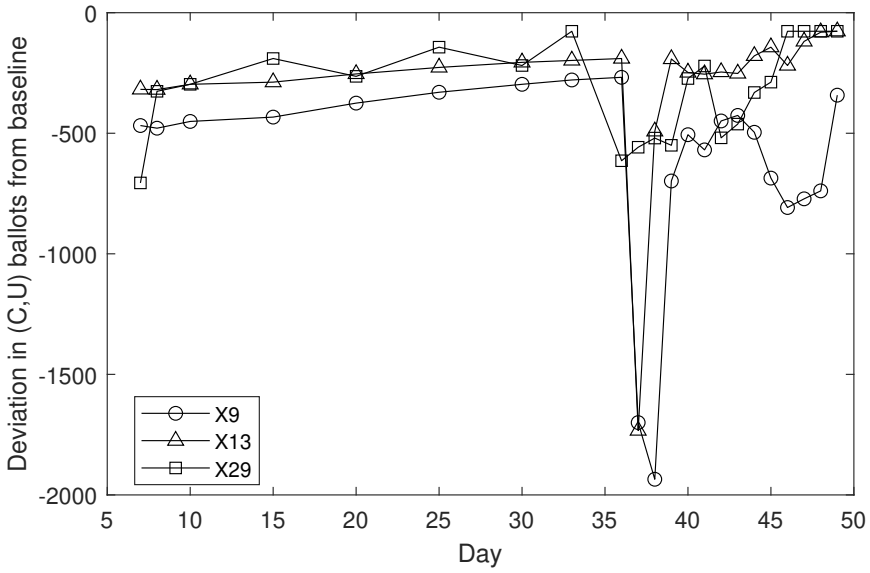

Figure 5.5: Deviation in counted and unaltered (C, U) ballots from the baseline under three moderate malicious attack scenarios ($X9$, $X13$, and $X29$).

to $X13$ on day 37; arc from node II to $X29$ on day 7; arc from node VII to $X9$. We also consider a sensitivity analysis that evaluates the impact of mitigation strength and associated strength against the worst-case attack scenario. Referencing Table 5.3 in Section 5.3.4, four mitigations counter the attacks in the worst-case scenario (see Table 5.3):

- $X9$ is countered by automatic ballot notifications M3, replacement ballots M4, and early voting M6,
- $X13$ is countered by automatic ballot notifications M3, replacement ballots M4, early voting M6, and drop boxes M7,
- $X29$ is countered by automatic ballot notifications M3 and replacement ballots M4.

Table 5.11 reports final ballot states associated with various mitigation scenarios, reflected by the first five columns. Then, it outlines the expected number of ballots in each final ballot state in the remaining six columns.

We first vary drop box ($M7$) availability from its base value of 0.520 using an availability range from 0.1 to 0.95. We do not consider drop boxes to have zero availability, since voters can return their absentee ballots directly to their local municipality election administration office. The remaining mitigations ($M6, M4, M3$) vary in availability, ranging from 0.01 to 0.95 to capture a wide range of operating conditions based on current federal laws and reports of mitigation effectiveness [7].

The first row of Table 5.11 shows the impact of the worst-case malicious attack scenario on the DTMC model for Milwaukee County. We see 8,950 fewer ballots counted and unaltered (C,U) than the baseline. Further, we see 1,208 altered ballots in final ballot states (C,A) and (NC,A). The number of ballots that are not counted increases

significantly, with 3,593 (NC,U) ballots compared to 559 in the baseline. The results in Table 5.11 show a substantial impact in the performance of the VBM process when compared to the baseline results with no malicious attacks in Table 5.7. Next, we vary mitigation strength to find which mitigations might be effective in countering the affects of the worst-case scenario. The goal is to align the worst-case with the baseline through only modifying mitigation strengths.

Next, we vary mitigation strength. Drop box $M7$ availability significantly impacts the number of ballots in the final ballot states. Increasing the availability of $M7$ to 0.95 increases the counted ballots (C,U) to 324,867, which is higher the 324,792 (C,U) ballots in the baseline scenario without malicious attacks. Reducing $M7$ availability to less than 0.10 results in 17,776 (NC,U) ballots, the highest across all worst-case scenarios. However, M7 does not counter masquerade attacks to reduce the number of altered ballots (C,A) or (NC,A).

The following eight rows in Table 5.11 examine the availability of early voting $M6$. Changes in early voting $M6$ availability has substantially less impact on final ballot states than drop boxes $M7$ availability. Similar to drop boxes $M7$, the availability of early voting $M6$ fails to counter masquerade attacks by reducing the number of altered ballots (C,A) or (NC,A).

Next, we examine varying mitigations automatic ballot notifications $M3$ and replacement ballots $M4$ simultaneously to reflect their interdependency. We find that deploying automatic ballot notifications $M3$ with the same availability as replacement ballots$M4$, at 0.90, results in 325,584 counted (C,U) ballots, an increase of 9,742 ballots as compared to the baseline, worst-case performance in the first row of Table 5.11. An automatic ballot notifications $M3$ availability of 0.90 results in 62 expected (C,A) ballots. Lastly, since automatic ballot notifications $M3$ and early voting $M6$ can function interdependently, we examine the impact of simultaneously varying automatic ballot notifications $M3$ and early voting $M6$. When deployed at 0.90 availability, we observe 325,485 (C,U) ballots, similar to the previous observations. Additionally, we observe fewer ballots not returned, with 17,243 (NC,NR) ballots compared with 20,371 (NC,NR) ballots in the Milwaukee baseline without malicious attacks. We conclude that in-person absentee voting gives voters a powerful recourse option to counter malicious and non-malicious attacks.

Sensitivity Analysis: We study the sensitivity of the DTMC model mitigation strength and efficacy. The two final ballot states we study are the desired final ballot state (C,U) and the unreturned ballots (NC,NR) since these two final ballot states are the most impacted by mitigations. However, other ballot states are impacted by mitigations to a lesser degree. We perform a one-way sensitivity analysis by changing each mitigation strength by ± 0.01 from its baseline value (see Table 5.2). Figure 5.6 shows a tornado graph illustration of the results. The blue half of Figure 5.6 on the left represents the difference in the number of correctly counted and unaltered ballots when the mitigation strength decreases. The red half is the difference in the number of ballots that are not counted and not returned, which occurs when mitigation strength increases. The tornado graph shows that the DTMC model is most sensitive to drop boxes $M7$ and automatic ballot notifications $M3$. In contrast, replacement ballots $M4$ and early voting $M6$ (inde-

Table 5.11: Policy performance for a worst-case election cycle

Scenario	M3 available	M4 available	M6 available	M7 available	Final Ballot States (Expected number of ballots) (C,U)	(NC,U)	(C,A)	(NC,L)	(NC,NR)	(NC,A)
Milwaukee County baseline performance for worst case (X13, day 37) + (X29, day 7) + (X9, day 38)	0.027	0.900	0.400	0.520	315,842	3,593	604	391	22,913	604
Vary drop box M7 availability for worst case	0.027	0.900	0.400	0.100	307,016	4,865	604	735	28,901	604
	0.027	0.900	0.400	0.200	309,118	4,562	604	653	27,475	604
	0.027	0.900	0.400	0.520	315,842	3,593	604	391	22,913	604
	0.027	0.900	0.400	0.750	320,671	2,897	604	203	19,637	604
	0.027	0.900	0.400	0.850	322,769	2,594	604	122	18,213	604
	0.027	0.900	0.400	0.950	324,867	2,291	604	41	16,789	604
Vary in-person absentee M6 for worst case	0.027	0.900	0.010	0.520	316,160	3,198	604	423	23,099	604
	0.027	0.900	0.100	0.520	316,086	3,289	604	416	23,056	604
	0.027	0.900	0.200	0.520	316,005	3,390	604	407	23,009	604
	0.027	0.900	0.300	0.520	315,923	3,492	604	399	22,961	604
	0.027	0.900	0.400	0.520	315,842	3,593	604	391	22,913	604
	0.027	0.900	0.500	0.520	315,760	3,694	604	382	22,865	604
	0.027	0.900	0.750	0.520	315,557	3,948	604	362	22,745	604
	0.027	0.900	0.950	0.520	315,393	4,150	604	345	22,650	604
Vary Ballot Trax/Scout implementation M3 with mail-in availability M4	0.010	0.010	0.400	0.520	315,661	3,652	614	389	23,016	614
	0.027	0.027	0.400	0.520	315,842	3,593	604	391	22,913	604
	0.050	0.050	0.400	0.520	316,099	3,508	590	393	22,766	590
	0.400	0.400	0.400	0.520	319,963	2,236	372	423	20,558	372
	0.500	0.500	0.400	0.520	321,078	1,868	310	432	19,921	310
	0.600	0.600	0.400	0.520	322,197	1,499	248	441	19,281	248
	0.800	0.800	0.400	0.520	324,451	753	124	459	17,992	124
	0.900	0.900	0.400	0.520	325,584	378	62	468	17,344	62
	0.950	0.950	0.400	0.520	326,153	189	31	472	17,018	31
Vary Ballot Trax/Scout implementation M3 with in-person absentee availability M6	0.010	0.900	0.010	0.520	315,984	3,250	614	422	23,205	614
	0.027	0.900	0.027	0.520	316,146	3,214	604	422	23,092	604
	0.050	0.900	0.050	0.520	316,379	3,162	590	422	22,931	590
	0.400	0.900	0.400	0.520	319,963	2,236	372	423	20,558	372
	0.500	0.900	0.500	0.520	321,030	1,921	310	424	19,889	310
	0.600	0.900	0.600	0.520	322,115	1,582	248	426	19,222	248
	0.800	0.900	0.800	0.520	324,343	837	124	430	17,900	124
	0.900	0.900	0.900	0.520	325,485	430	62	433	17,243	62
	0.950	0.900	0.950	0.520	326,063	218	31	434	16,916	31

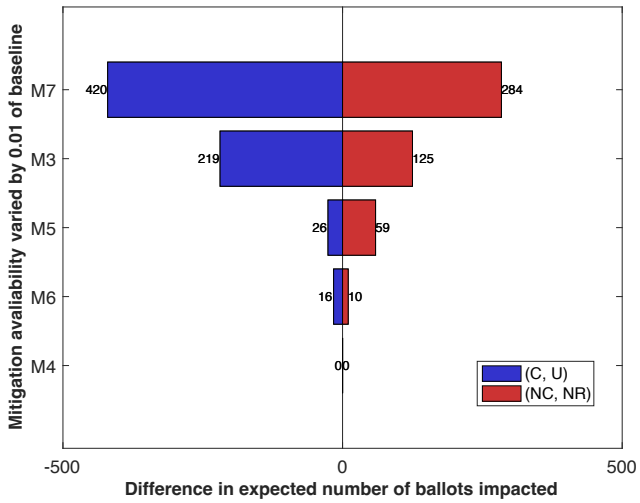

Figure 5.6: Results of a sensitivity analysis that evaluates the impact of mitigations M3, M4, M5, M6, and M7 on the DTMC model's final ballot outputs in the presence of multiple malicious attacks. We compare the changes in the expected number of (C, U) and (NC, NR) ballots to the baseline.

pendently) do not significantly affect the outputs in a significant manner. However, their impact can be larger when their availability is considered within the context of other mitigations (see Table 5.11). These results demonstrate that modeling attack timing in conjunction with mitigation availability provides crucial insight into the consequence of each attack.

5.7 Key risk modeling insights

The VBM study offers several key insights into the role of attack graph models in CPS risk analysis. First, while Boolean logic attack paths provide an integral starting point for expressing attack conditions, they are insufficient for assessing the performance of dynamic system. For this reason, integrating attack graphs with a DTMC enables a detailed risk assessment of VBM, since the DTMC model has time-dependent process transitions, adversarial attack sequences, and voter-driven mitigations. Integrating malicious, non-malicious, and normative operating conditions into a single framework advances security modeling. Additionally, the complexity of CPS demands a functional layered approach. Structuring models across physical, cyber, and human operational layers help ensure visibility into where attacks occur, how they propagate, and what mitigations are viable at each stage. These principles can guide future work applying DAG-based attack graphs and stochastic models to other CPHS domains, such as autonomous vehicle networks, distributed control networks, or supply chain infrastructure.

5.7.1 Time sensitivity of threats

This research highlights the fact that risk in VBM process is dynamic and must be managed through a systems approach. We note that different malicious attacks may have different scopes of impact based on the timing of the attacks. Our findings suggest that mitigations that provide voters with information and recourse actions are crucial for VBM security. In particular, drop boxes enable voters to cast a second ballot when their first ballot is affected by a malicious or non-malicious attack.

5.7.2 Effective mitigations

Additionally, automatic ballot tracking can significantly affect VBM performance. Both drop boxes and automatic ballot tracking provide broad protection against attacks. Two notable examples of ballot tracking programs that are currently available are BallotTrax and BallotScout. BallotTrax allows voters to track the status of their mail-in ballots, providing real-time updates when ballots are collected, received, and accepted by election offices [34]. Similarly, BallotScout integrates with USPS to provide voters with detailed information on their ballots' whereabouts throughout the election process [16].

Chapter 6

Conclusion and Future Work

6.1 Overview of Contributions

6.1.1 A holistic risk modeling framework for CPS

This research makes several key contributions to the modeling and protection of cyber-physical systems (CPS). Foremost, it introduces a layered risk assessment framework that emphasizes not only the importance of decomposing a CPS into distinct functional layers but also the critical need to define the contents and interactions within those layers thoughtfully. The framework incorporates not only system vulnerabilities, but also the available mitigations and expected patterns of normal system operations, which together provide a comprehensive basis for assessing both everyday operational resilience and resilience to cyber-physical threats, including those posed by malicious actors, hackers, or foreign adversaries. A significant innovation of this work is its explicit incorporation of temporal dynamics, highlighting the necessity of time-aware modeling in CPS risk analysis. The benefits of temporal constraint modeling in risk are proven through the vote-by-mail (VBM) case study, a cyber-physical-human system that demonstrates how temporal modeling enables detailed insights into system behavior under both normal and adversarial conditions. These contributions are explored further through their temporal implications in the following subsection.

6.1.2 Temporal constraints in CPS risk modeling

We see in the CPES application for SMR that having a risk modeling framework that incorporates the three key components of process, attacks, and mitigations can provide new insights into the risk associated with CPS. However, in the CPHS application to VBM we see that when the temporal constraints of a system are accounted for in the risk model yield significantly more insights. The VBM model uses a time-inhomogeneous DTMC to capture the impact of malicious and non-malicious attacks to ballots cast via absentee voting. The impact of mitigations from policies such as Drop boxes and automatic ballot tracking have quantifiable benefits to improving security of VBM. This framework can be applied to other critical infrastructures as a new view of CPS security. The framework presented in this thesis provides a vital key to the improvement of decision sciences and

analysis of risk.

6.1.3 Benefit of considering CPES/CPHS

The inclusion of both CPES and CPHS case studies demonstrates the utility of extending cybersecurity modeling beyond conventional IT frameworks to encompass operational, physical, and human-dependent systems. In the SMR CPES case, the layered attack graph reveals vulnerabilities that emerge from cross-domain interactions (e.g., operator commands at the oversight level that propagate down to field-level actuators). The framework captures risks that traditional flat, static models would obscure. Additionally, the VBM system benefits from a CPHS lens by explicitly modeling human behavior (e.g., ballot return decisions), temporal constraints (e.g., voting deadlines), and layered recourse mechanisms (e.g., ballot replacement or drop box usage). By framing these systems as CPES/CPHS, the framework effectively connects abstract threat modeling to tangible operational realities, ensuring a more robust understanding of potential vulnerabilities. The modeling of temporal constraints enhances the model's ability to forecast the consequences of specific adversarial actions, assess mitigation timing, and support dynamic policy decision-making. The result is a more holistic view of system risk, specifically one that accounts for both cyber threat vectors and the physical and procedural states through which those threats must propagate.

6.1.4 Adaptability and scalability of risk modeling methodology

A key strength of the proposed methodology is its adaptability across various domains and its scalability in terms of both system complexity and temporal resolution. This work builds the risk modeling framework using modular components, including layered system decomposition, attack graph modeling, and discrete-time Markov chains (DTMCs) or other stochastic and dynamic risk representations. Each component is customizable according to the unique operational profile and security architecture of the target system. The SMR model illustrates vertical scalability, handling deeply layered technical infrastructures from high-level ISO commands to field-level relay protection schemes. In contrast, the VBM model emphasizes horizontal scalability by integrating process stages across months of operation, showing how threats and mitigations evolve. Importantly, both models use the same core structure of attack surface, process, and mitigation layers to enable consistent interpretation and reuse of modeling logic. Furthermore, the methodology supports different levels of granularity. A modeler can scale the temporal step size (e.g., daily for elections, sub-second for power systems), select threat representation formats (e.g., MITRE ATT&CK mappings or Boolean logic trees), and incorporate domain-specific mitigations or human responses. This flexibility makes the framework a suitable candidate for broader CPS security applications across various sectors, including water treatment, transportation, and supply chain systems.

6.2 Limitations and Model Modifications

Given the identification of high-risk, low-coverage attack nodes, the next step is to develop targeted mitigation strategies that reduce system exposure and increase resilience. In the context of Small Modular Reactor (SMR) operations, mitigation can be approached from multiple angles: enhancing physical security controls, refining operational procedures during maintenance, and addressing regulatory coverage gaps. Since both of the most critical nodes occur in Layer 4 (the physical system layer), and are associated with control rod and cooling subsystems, mitigation efforts should prioritize access restrictions, audit logging, and hardware-based interlocks to prevent unauthorized overrides or physical interference. Procedurally, more robust verification during multi-party maintenance activities—especially involving shared relay access or cooling diagnostics—may reduce the likelihood of simultaneous exposure.

On the regulatory side, these findings suggest a need to expand the scope of existing cybersecurity and coordination protocols to explicitly include protections for control rod override and field-level actuator manipulation. This may involve revisiting the technical applicability of existing NERC or NRC guidance, or drafting SMR-specific modifications that address the unique architecture of smaller, modular systems. The combination of architectural hardening and policy refinement offers a layered defense model that can substantially lower the risk posed by these exposed nodes, ultimately improving the station's ability to withstand both insider threats and targeted cyber-physical attacks during routine operations.

6.2.1 Updated SMR security risk modeling architecture

In the current model, the mitigation effectiveness for a given attack node is computed as a probabilistic complement over independent mitigations. The $\mu(v_i, M_j$ represents the overall mitigation coverage for attack nodev_i:

$$M_i = 1 - \prod(1 - \mu(v_i, M_j)). \tag{6.1}$$

While this formulation captures the cumulative protective effect of multiple mitigations, it implicitly assumes that all mitigations act independently and with diminishing marginal returns. However, in many real-world settings (including work on VBM election systems) we observed that mitigations can interact in unexpected ways. Some defenses may be redundant, while others may reinforce one another synergistically. As such, this product-based model may underestimate or overestimate the true defensive impact when mitigations are interdependent. Future work should explore more sophisticated mitigation impact models, potentially drawing on dependency graphs, logic-based synergy functions, or game-theoretic formulations. Capturing these interactions will be critical for producing realistic and actionable risk assessments in cyber-physical systems such as SMR stations. While this work uses fixed probability distributions for each scenario-node pair, future work will incorporate subject-matter expertise or empirical incident reports to calibrate these distributions. Such an approach will improve alignment with operational realities and support dynamic threat modeling under uncertainty.

A promising future research direction for the small modular reactor (SMR) modeling work is the integration of layered graph structures that jointly capture adversarial behavior and system dependencies. Specifically, we propose developing a dynamic risk assessment framework by mapping adversarial execution logic to infrastructure components through the fusion of an Adversarial Execution Graph (AEG) and an Infrastructure Dependency Graph (IDG). The AEG encodes the sequence of attack steps an adversary might take—aligned with the MITRE ATT&CK for ICS framework—where nodes represent discrete attack techniques and edges represent temporal or logical dependencies. Each node is enriched with attributes such as CVSS scores, detection probabilities, and kill chain phases.

In parallel, the IDG captures the hierarchical and functional interdependencies among SMR devices and subsystems, where nodes denote physical or cyber-physical assets (e.g., relays, HMIs, RTUs) and edges encode control, communication, or physical support relations. By establishing a formal mapping between nodes in the AEG and devices in the IDG, this fusion framework enables the conditional propagation of risk: attack node activations can dynamically escalate threats across the infrastructure depending on the state and connectivity of the underlying system. This dual-graph construct supports a predictive modeling capability for SMR security by allowing practitioners to simulate adversarial campaigns and observe cascading effects through the control hierarchy. Future work will focus on refining this mapping mechanism, incorporating stochastic state transitions informed by CVSS vectors and device reliability, and implementing simulation studies to validate the model against SMR reference architectures.

6.2.2 Integrate with real-world policy

There are several limitations of the VBM study that provide insight into topics for future research. First, this analysis considers malicious attacks that occur on a single-day and is limited by a lack of historical data regarding malicious attacks. Future research could diversify attack scenarios and incorporate a broader range of data to enhance the model's sensitivity to resource constraints and varying electoral processes. Second, future research could account for resource constraints and system congestion. Lastly, we did not examine the likelihood of attack scenarios altering election outcomes. Successful attacks on voting systems compromise the integrity of elections and undermine public trust in political systems, regardless of whether they change the outcomes. Although not all attacks aim to change election results, this remains a significant concern that future research could investigate.

6.3 Future Research

This thesis presents a structured approach to modeling security risks in CPS by integrating operational processes, attack graphs, and mitigation strategies and accounting for the development of risk over time. The framework developed in this research addresses key limitations found in existing cyber-physical system models, specifically by considering both cyber and physical elements rather than relying on a static interpretation of system

performance. The proposed methodology demonstrates how a more dynamic, layered modeling approach can provide actionable insights for both technical system design and public infrastructure policy.

In the near future, my work will continue to advance this line of research within the energy sector, particularly in response to emerging concerns about grid reliability. The North American Electric Reliability Corporation (NERC) and several Independent System Operators (ISOs) have issued warnings of high load scenarios and tightening reserve margins, underscoring an urgent need for more sophisticated risk models that can simulate system performance under stress, account for evolving threats, and guide investment in resilience. By applying and extending the methodology developed in this thesis, I aim to contribute to the development of risk-informed planning tools that support both operational stability and long-term energy security. In a world where CPSs are increasingly integral to daily life, risk modeling must evolve in tandem with system complexity. This work is a step toward that evolution.

Bibliography

[1] W. Ahn et al. "Development of cyber-attack scenarios for nuclear power plants using scenario graphs". In: *International Journal of Distributed Sensor Networks* 11.836258 (2015).

[2] D. L. Alderson, G. G. Brown, and M. Carlyle. "Operational Models of Terrorism Risk for Homeland Security". In: *Risk Analysis* 35.4 (2015), pp. 562–576. DOI: 10.1111/risa.12341.

[3] Paul Ammann, Duminda Wijesekera, and Saket Kaushik. "Scalable, graph-based network vulnerability analysis". In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*. CCS '02. Washington, DC, USA: Association for Computing Machinery, 2002, pp. 217–224. ISBN: 1581136129.

[4] Yosef Ashibani and Q. Mahmoud. "Cyber physical systems security: Analysis, challenges and solutions". In: *Comput. Secur.* 68 (2017), pp. 81–97.

[5] AttackIQ. *The CISO's guide to using attack graphs and MITRE ATT&CK*. White paper. Los Altos, CA: AttackIQ, 2022.

[6] Barry C. Burden. *The Experiences of Municipal Clerks and the Electorate in the November 2020 General Election in Wisconsin*. Sept. 2021. URL: https://thompsoncenter.wisc.edu/wp-content/uploads/%20sites/509/2021/09/Burden-2020-Wisconsin-Election-%20Report-PUBLIC.pdf.

[7] Bipartisan Policy Center. *The 2020 Voting Experience: Lessons Learned and Recommendations for Reform*. Accessed: 2024-09-02. 2021. URL: https://bipartisanpolicy.org/download/?file=/wp-content/uploads/2021/04/EPP-Voting-Experience_RV1.pdf.

[8] Blessing, Jenny et al. "Security Survey and Analysis of Vote-by-Mail Systems". In: *Computers and Society, Cryptography and Security MIT* (Sept. 2020).

[9] H. Çetinay et al. "Comparing destructive strategies for attacking networks". In: *Guide to Disaster-Resilient Communication Networks*. Heidelberg: Springer, 2020, pp. 117–140.

[10] C. Cheh et al. "Developing Models for Physical Attacks in Cyber-Physical Systems". In: Dallas, TX: Association for Computing Machinery, 2017, pp. 49–55.

[11] Yulia Cherdantseva et al. "A review of cyber security risk assessment methods for SCADA systems". In: *Computers & Security* 56 (2016), pp. 1–27. ISSN: 0167-4048.

[12] City of Milwaukee Common Council. *Final Voting Ward Demographics*. Accessed on: Aug. 4, 2020. [Online]. Available: https://city.milwaukee.gov/ImageLibrary/Groups/ccCouncil/2012PDF/FinalVotingWardDemographics-Ju.xls. 2012.

[13] Louis A. T. Cox. "Some Limitations of "Risk = Threat × Vulnerability × Consequence" for Risk Analysis of Terrorist Attacks". In: *Risk Analysis* 28.6 (2008), pp. 1749–1761. DOI: 10.1111/j.1539-6924.2008.01174.x.

[14] Cybersecurity and Infrastructure Security Agency. *Mail-in Voting in 2020 Infrastructure Risk Assessment and Infographic*. July 2020.

[15] Melissa De Witte. *Examining effects, challenges of mail-in voting*. Sept. 2020.

[16] Democracy Works. *BallotScout*. https://www.democracy.works/ballotscout. Accessed: 2024-05-21. 2021.

[17] A. Dennis et al. "Dynamic Multi-Module PRA (MMPRA) Approach for Small Modular Reactors". In: *Proceedings of the PSA Conference* (2016).

[18] Simon Enoch. "Dynamic Cybersecurity Modelling and Analysis". PhD thesis. Nov. 2018. DOI: 10.26021/1578.

[19] Yiping Fang, N. Pedroni, and E. Zio. "Resilience-Based component importance measures for critical infrastructure network Systems". In: *IEEE Transactions on Reliability* 65 (2016), pp. 502–512. DOI: 10.1109/TR.2016.2521761.

[20] Chris Few et al. "A case study in the use of attack graphs for predicting the security of cyber-physical systems". In: *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*. 2021, pp. 1–7.

[21] Anusha Ganesan et al. "Human-in-the-Loop Predictive Analytics Using Statistical Learning". In: *Journal of Healthcare Engineering* 2021 (2021).

[22] Manuel A. Ruiz Garcia et al. "A human-in-the-loop cyber-physical system for collaborative assembly in smart manufacturing". In: *Procedia CIRP* 81 (2019). 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019, pp. 600–605. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2019.03.162. URL: https://www.sciencedirect.com/science/article/pii/S2212827119304676.

[23] A. Georgiadou, S. Mouzakitis, and D. Askounis. "Assessing MITRE ATT&CK risk using a cyber-security culture framework". In: *Sensors* 21 (2021), p. 3267.

[24] Alaa T. Al Ghazo et al. "A2G2V: Automatic Attack Graph Generation and Visualization and Its Applications to Computer and SCADA Networks". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50 (2020), pp. 3488–3498.

[25] Edward Griffor et al. *Framework for Cyber-Physical Systems: Volume 1, Overview*. en. 2017-06-26 2017. DOI: https://doi.org/10.6028/NIST.SP.1500-201.

[26] Yacov Y. Haimes. "Risk Modeling of Interdependent Complex Systems of Systems: Theory and Practice". In: *Risk Analysis* 38.1 (2018), pp. 84–98. DOI: 10.1111/risa.12804.

[27] Carmen Haseltine. *VBM DTMC Github Repository*. https://github.com/HaseltineC/VBM-DTMC. Accessed: 2025-04-27. 2024.

[28] Carmen Haseltine and Line Roald. "The Effect of Blocking Automatic Reclosing on Wildfire Risk and Outage Times". In: *2020 52nd North American Power Symposium (NAPS)*. 2021, pp. 1–6. DOI: 10.1109/NAPS50074.2021.9449670.

[29] Carmen Haseltine, Adam Schmidt, and Laura Albert. "Poster: On designing and operating Vote-by-Mail processes". In: *ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization EAAMO'21*. Sept. 2021.

[30] Carmen Haseltine, Shaonan Wang, and Laura Albert. "Dynamic Cyber-Physical System Security Planning using Attack Graphs". In: *Proceedings of the IISE Annual Meeting*. Institute of Industrial and Systems Engineers. Seattle, WA, May 2022.

[31] Carmen A. Haseltine and Laura A. Albert. *Risk Analysis and Dynamic Security Planning for Voting-by-Mail*. arXiv:2410.13900 [cs.CR]. 2024. arXiv: 2410.13900 [cs.CR]. URL: https://arxiv.org/abs/2410.13900.

[32] Jin B. Hong and Dong Seong Kim. "Towards scalable security analysis using multi-layered security models". In: *Journal of Network and Computer Applications* 75 (2016), pp. 156–168. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2016.08.024. URL: https://www.sciencedirect.com/science/article/pii/S1084804516301928.

[33] Abdulmalik Humayed et al. "Cyber-Physical Systems Security—A Survey". In: *IEEE Internet of Things Journal* 4.6 (Dec. 2017), pp. 1802–1831. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2703172.

[34] i3logix. *BallotTrax*. https://www.ballottrax.com/. Accessed: 2024-05-21. 2024.

[35] 'Office of Inspector General'. *United States Postal Service Performance of Election and Political Mail During the November 2020 General Election*. Audit Report. Mar. 2021.

[36] A. Jahromi et al. *Cyber-physical interdependence for power system operation and control*. Tech. rep. PES-TR119. IEEE Power & Energy Society, Dec. 2023.

[37] Ik Jae Jin and In Cheol Bang. "The time for revolutionizing small modular reactors: Cost reduction strategies from innovations in operation and maintenance". In: *Progress in Nuclear Energy* (2024). DOI: 10.1016/j.pnucene.2024.105288.

[38] John R. Johnson and Emilie Hogan. "A graph analytic metric for mitigating advanced persistent threat". In: 2013, pp. 129–133.

[39] Maxim Kalinin, Evgenii Zavadskii, and Alexey G. Busygin. "A Graph-Based Technique for Securing the Distributed Cyber-Physical System Infrastructure". In: *Sensors (Basel, Switzerland)* 23 (2023).

[40] Hakan Kayan et al. "Cybersecurity of Industrial Cyber-Physical Systems: A Review". In: *ACM Computing Surveys (CSUR)* 54 (2021), pp. 1–35. DOI: 10.1145/3510410.

[41] Arash Negahdari Kia et al. "A cyber risk prediction model using common vulnerabilities and exposures". In: *Expert Systems with Applications* 237 (2024), p. 121599. ISSN: 0957-4174.

[42] Tony Koonce, Curtis Smith, and S. Prescott. "Advanced Small Modular Reactor (SMR) Probabilistic Risk Assessment (PRA) Demonstration". In: *Engineering, Physics* (2014). DOI: 10.2172/1149017.

[43] B. Kordy et al. "Foundations of attack-defense trees". In: Pisa, Italy: Springer, 2010, pp. 80–95.

[44] David Kushner. *the real story of stuxnet*. accessed: 4/14/2022. July 2021. URL: https://spectrum.ieee.org/the-real-story-of-stuxnet.

[45] Jingsheng Li, Theodore T Allen, and Kimiebi Akah. "Could simulation optimization have prevented 2012 central Florida election lines?" In: *2013 Winter Simulations Conference (WSC)*. IEEE. Washington, DC, USA, 2013, pp. 2088–2096.

[46] R. Liu et al. "Analyzing the cyber-physical impact of cyber events on the power grid". In: *IEEE Transactions on Smart Grid* 6 (2015), pp. 2444–2453.

[47] Colin McIntyre. *"What queueing means; polling places COVID-19?" - MIT Election Lab*. Aug. 2020. URL: https://electionlab.mit.edu/sites/default/files/%202020-08/WhatQueueingMeansPollingPlacesCOVID19.pdf.

[48] Jason R. W. Merrick and G. Parnell. "A Comparative Analysis of PRA and Intelligent Adversary Methods for Counterterrorism Risk Management". In: *Risk Analysis* 31 (2011). DOI: 10.1111/j.1539-6924.2011.01590.x.

[49] Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. "Optimal Defense Policies for Partially Observable Spreading Processes on Bayesian Attack Graphs". In: *Proceedings of the Second ACM Workshop on Moving Target Defense*. MTD '15. Denver, Colorado, USA: Association for Computing Machinery, 2015, pp. 67–76. ISBN: 9781450338233. DOI: 10.1145/2808475.2808482. URL: https://doi.org/10.1145/2808475.2808482.

[50] Milwaukee County. *Municipalities*. June 2023. URL: https://county.milwaukee.gov/EN/Municipalities.

[51] Milwaukee Elections Commission. *November 3, 2020 - General Election*. Nov. 6, 2020, Accessed on: June 5, 2021 [Online]. Available: https://city.milwaukee.gov/election/ElectionInformation/ElectionResults/2020/November. 2020.

[52] Milwaukee OpenData. *Voting Wards 2020*. Accessed on: Aug. 6, 2020. [Online]. Available: https://data.milwaukee.gov/dataset/voting-wards/resource/01139d6b-b65a-4d63-89da-b87f3986ff0d?inner_span=True, April 6, 2020. 2020.

[53] MIT Election Data + Science Lab. *Voting by mail and absentee voting*. Mar. 2021. URL: https://electionlab.mit.edu/research/voting-mail-and-absentee-voting.

[54] MITRE. *ATT&CK for Industrial Control Systems*. https://attack.mitre.org/matrices/ics/. 2023.

[55] National Conference of State Legislatures. *Report voting outside the Polling Place: Absentee, all-mail and other voting at home options*. Updated on: July 12, 2022. [Online]. Available: https://www.ncsl.org/elections-and-campaigns/voting-outside-the-polling-place. 2022.

[56] National Institute of Standards and Technology. *Assessing Enhanced Security Requirements for Controlled Unclassified Information*. Tech. rep. National Institute of Standards and Technology (NIST), Mar. 2022. DOI: 10.6028/NIST.SP.800-172A. URL: https://doi.org/10.6028/NIST.SP.800-172A.

[57] National Institute of Standards and Technology. *Security and Privacy Controls for Information Systems and Organizations*. Tech. rep. NIST SP 800-53 Rev. 5. Revision 5. Gaithersburg, MD: U.S. Department of Commerce, 2020.

[58] Office of Inspector General. *United States Postal Service Performance of Election and Political Mail During the November 2020 General Election*. Audit Report. United States Postal Service, Mar. 2021.

[59] A. Palomino and Y. Zhang. "Layered Attack Graphs for CPES: A Structural Vulnerability Framework". In: *IEEE Transactions on Industrial Informatics* (2025). Forthcoming.

[60] Alejandro Palomino, Jairo Giraldo, and Masood Parvania. "Graph-Based Interdependent Cyber-Physical Risk Analysis of Power Distribution Networks". In: *IEEE Transactions on Power Delivery* 38.3 (2023), pp. 1510–1520.

[61] Kaveh Paridari et al. "A Framework for Attack-Resilient Industrial Control Systems: Attack Detection and Controller Reconfiguration". In: *Proceedings of the IEEE* 106.1 (Jan. 2018), pp. 113–128. ISSN: 1558-2256. DOI: 10.1109/JPROC.2017.2725482.

[62] C. Phillips and L. P. Swiler. "A graph-based system for network-vulnerability analysis". In: *Proceedings of the 1998 workshop on New security paradigms*. ACM. 1998, pp. 71–79.

[63] A. Presekal et al. "Attack graph model for cyber-physical power systems using hybrid deep learning". In: *IEEE Transactions on Smart Grid* 14 (2023), pp. 4007–4020.

[64] Megha Quamara, Gabriel Pedroza, and B. Hamid. "Multi-layered Model-based Design Approach towards System Safety and Security Co-engineering". In: *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)* (2021), pp. 274–283. DOI: 10.1109/models-c53483.2021.00048.

[65] S.M. Rinaldi, J.P. Peerenboom, and T.K. Kelly. "Identifying, understanding, and analyzing critical infrastructure interdependencies". In: *IEEE Control Systems Magazine* 21.6 (2001), pp. 11–25.

[66] S.M. Rinaldi, J.P. Peerenboom, and T.K. Kelly. "Identifying, understanding, and analyzing critical infrastructure interdependencies". In: *IEEE Control Systems Magazine* 21.6 (2001), pp. 11–25. DOI: 10.1109/37.969131.

[67] R. G. Saltman. "Accuracy, Integrity and Security in Computerized Vote-Tallying". In: *Commun. ACM* 31.10 (Oct. 1988), pp. 1184–1191. ISSN: 0001-0782. DOI: 10.1145/63039.63041. URL: https://doi.org/10.1145/63039.63041.

[68] N.M. Scala et al. "A Process Map and Risk Assessment for Mail-based Voting". In: *Proceedings of the 2021 Institute of Industrial and System Engineers (IISE) Annual Conference*. Ed. by A. Ghate, K. Krishnaiyer, K. Paynabar, eds. 2021.

[69] Natalie M Scala et al. "Evaluating mail-based security for electoral processes using attack trees". In: *Risk Analysis* 42.10 (2022), pp. 2327–2343.

[70] Adam Schmidt and Laura A. Albert. "Designing pandemic-resilient voting systems". In: *Socio-Economic Planning Sciences* 80 (2022), p. 101174. ISSN: 0038-0121. DOI: https://doi.org/10.1016/j.seps.2021.101174. URL: https://www.sciencedirect.com/science/%20article/pii/S003801212100166X.

[71] Adam Schmidt and Laura A. Albert. "The drop box location problem". In: *IISE Transactions* 56.4 (2023), pp. 424–436. DOI: 10.1080/24725854.2023.2213754. URL: https://doi.org/10.1080/24725854.2023.2213754.

[72] B. Schneier. "Attack trees: Modeling security threats". In: *Dr. Dobb's Journal* (Dec. 1999).

[73] B. Schneier. "Attack trees: Modeling security threats". In: *Dr. Dobb's Journal* (Dec. 1999).

[74] B. Schneier. "Attack trees: Modeling security threats". In: *Dr. Dobb's Journal* (Dec. 1999).

[75] S. Seshia et al. "Design automation of cyber-physical systems: Challenges, advances, and opportunities". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36 (2017), pp. 1421–1434.

[76] Jiarou Shen. "Merge Times and Hitting Times of Time-inhomogeneous Markov Chains". Thesis. Duke University, 2013.

[77] O. Sheyner et al. "Automated generation and analysis of attack graphs". In: *Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on*. IEEE. 2002, pp. 273–284.

[78] B. I Simidchieva et al. "Modeling and Analyzing Faults to Improve Election Process Robustness". en. In: *In Proceedings of the 2010 USENIX/ACCURATE Electronic Voting Technology Workshop*. 2010. URL: https://escholarship.org/uc/item/0cg3b5vb#article_main (visited on 05/22/2021).

[79] Jill Slay and Michael Miller. "Lessons Learned from the Maroochy Water Breach". In: vol. 253. Mar. 2007, pp. 73–82. ISBN: 978-0-387-75461-1. DOI: 10.1007/978-0-387-75462-8_6.

[80] Charles Stewart. "2016 Survey of the Performance of American Elections". In: *2020 Survey of the Performance of American Elections*. Harvard Dataverse, 2021. DOI: 10.7910/DVN/Y38VIQ/SXXGGV. URL: https://doi.org/10.7910/DVN/FSGX7Z.

[81] Charles Stewart III and Stephen Ansolabehere. "Waiting to Vote". In: *Election Law Journal* 14.1 (2015), 47–53.

[82] Frank Swiderski and Window Snyder. *Threat Modeling*. 1st. Redmond, WA: Microsoft Press, 2004.

[83] The Heritage Foundation. *Election fraud map*. accessed: 3/12/2025. Washington, D.C., 2025. URL: https://electionfraud.heritage.org/.

[84] The White House. *Presidential Policy Directive 21: Critical Infrastructure Security and Resilience*. https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil. Accessed: March 10, 2025. 2013.

[85] by Univ. South Alabama U.S. Election Assistance Commission. "Elections Operations Assessment; Threat Trees and Matrices TIRA". In: *24 Boston University Journal of Science and Technology Law* (Dec. 2009).

[86] United States Census Bureau. *QuickFacts Milwaukee city, Wisconsin*. Accessed on: Aug. 5, 2020. [Online]. Available: https://www.census.gov/quickfacts/fact/table/milwaukeecitywisconsin/PST045219. July 2019.

[87] Donatas Vitkus and et al. "Method for Attack Tree Data Transformation and Import into IT Risk Analysis Expert Systems". In: *MDPI, Multidisciplinary Digital Publishing Institute* (Nov. 2020). URL: https://www.mdpi.com/2076-3417/10/23/8423/htm,.

[88] Homeland Security News Wire. "Grid, Cyberattacks, Infrastructure Protection". In: (Oct. 2015). URL: https://www.homelandsecuritynewswire.com/dr20151019-2013-attack-on-metcalf-california-power-grid-substation-committed-by-an-insider-dhs..

[89] Wisconsin Elections Commission. *2020 General Election Voting and Registration Statistics Report, EL-109F*. Report. Accessed: 1-18-2024. Nov. 2020. URL: https://elections.wi.gov/statistics-data/voting-statistics?combine=2020&field_subject_target_id=All.

[90] Wisconsin Elections Commission. *Absentee Ballot Report*. Oct. 2020. URL: https://elections.wi.gov/.

[91] Wisconsin Elections Commission. *Absentee Ballot Report - November 3, 2020 General Election*. Accessed on: June 5, 2021 [Online]. Available: https://elections.wi.gov/node/6862. 2020.

[92] Wisconsin Elections Commission. *November 3, 2020 Election Data Report*. White Paper. Accessed: 1-17-2024. Feb. 2021. URL: https://elections.wi.gov/sites/default/files/legacy/2021-01/D.%2520November%25202020%2520Election%2520Data%2520Report.pdf.

[93] Wisconsin Elections Commission. *Voting by Mail: Note on Absentee Ballot Return*. Oct. 2020. URL: https://elections.wi.gov/%20voters/voting-mail#230548828-2254551794.

[94] Wisconsin Elections Commission. *WEC Releases Analysis of November 2020 Election Data*. https://elections.wi.gov/news/wec-releases-analysis-november-2020-election-data. [Online; accessed 28-December-2023]. 2020.

[95] Hongqi Xu et al. "Application of Bayesian Networks in Reliability Evaluation". In: *IEEE Transactions on Industrial Informatics* 15 (2019), pp. 2146–2157. DOI: 10.1109/TII.2018.2858281.

[96] Jean-Paul A. Yaacoub et al. "Cyber-physical systems security: Limitations, issues and future trends". In: *Microprocessors and Microsystems* 77 (2020), p. 103201. ISSN: 0141-9331. DOI: https://doi.org/10.1016/j.micpro.2020.103201. URL: https://www.sciencedirect.com/science/article/pii/S0141933120303689.

[97] Muer Yang, Michael J Fry, and W David Kelton. "Are all voting queues created equal?" In: *Proceedings of the 2009 Winter Simulation Conference (WSC)*. IEEE. 2009, pp. 3140–3149.

[98] Pardue, J. Yasinisac, A. "A Process for Assessing Voting System Risk Using Threat Trees". In: *Journal of Information Systems Applied Research* (2010).

[99] Alec Yasinsac and Harold Pardue. "A process for assessing voting system risk using threat trees". In: *Conference on Information Systems Applied Research*. Citeseer. 2010.

[100] R. Yohanandhan et al. "Cyber-physical power system (CPPS): A review on modeling, simulation, and analysis with cyber security applications". In: *IEEE Access* 8 (2020), pp. 151019–151064.

[101] Rajaa Vikhram Yohanandhan et al. "Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis With Cyber Security Applications". In: *IEEE Access* 8 (2020), pp. 151019–151064. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3016826.

[102] Ye Yuan et al. "Data driven discovery of cyber-physical systems". In: *Nature Communications* 10.1 (2019), p. 4894.

[103] P. Zarebski and Dominik Katarzyński. "Small modular reactors (SMRs) as a solution for renewable energy gaps: Spatial analysis for Polish Strategy". In: *Energies* 16 (2023), p. 6491.

[104] Bowen Zheng et al. "Cross-Layer Codesign for Secure Cyber-Physical Systems". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.5 (2016), pp. 699–711. DOI: 10.1109/TCAD.2016.2523937.

[105] K. Zheng and L.A. Albert. "A budgeted maximum multiple coverage model for cybersecurity planning and management". In: *Naval Research Logistics* 66.5 (2019), pp. 441–429.

[106] K. Zheng et al. "A budgeted maximum multiple coverage model for cybersecurity planning and management". In: *IISE Transactions* 51.12 (2019), pp. 1303–1317.

[107] T. Zhou et al. "A Review of Probabilistic Risk Assessment for Multi-Module Nuclear Power Plants". In: *Progress in Nuclear Energy* 135 (2021), p. 103686. DOI: 10.1016/j.pnucene.2021.103686.

[108] Ioannis Zografopoulos et al. “Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies”. In: *IEEE Access* 9 (2021), pp. 29775–29818.