Interdependent Mission Impact Assessment of an IoT System with Hypergame-Theoretic Attack-Defense Behavior Modeling

Ashrith Reddy Thukkaraju

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

Master of Science
in
Computer Science and Applications

Jin-Hee Cho, Chair
Bo Ji
Thang Hoang

November 06, 2023
Blacksburg, Virginia

Keywords: Mission Impact Assessment, mission performance, hypergame, and attack-defense interactions

Interdependent Mission Impact Assessment of an IoT System with Hypergame-Theoretic Attack-Defense Behavior Modeling

Ashrith Reddy Thukkaraju

(ABSTRACT)

Mission impact assessment (MIA) research has been explored to evaluate the performance and effectiveness of a mission system, such as enterprise networks with organizational missions and military or tactical mission teams with assigned missions. The key components in such mission systems, including assets, services, tasks, vulnerability, attacks, and defenses, are interdependent, and their impacts are interwoven. However, the current state-of-the-art MIA approaches have less studied such interdependencies. In addition, they have not modeled strategic attack-defense interactions under partial observability. In this work, we propose a novel MIA framework that assesses measures of performance (MoP) or measures of effectiveness (MoE) based on the service requirements (e.g., correctness or timeliness) of a given mission system based on full and comprehensive modeling and simulation of the key system components and their interdependencies. Particularly, we model intelligent attack-defense strategy selections based on hypergame theory, which allows considering uncertainty in estimating each player's hypergame expected utility (HEU) for its best strategy selection. As the case study, we consider an Internet-of-Things (IoT)-based mission system aiming to accurately and timely detect an object, given stringent accuracy and time constraints for successful mission completion. Via extensive simulation experiments, we validate the quality of the proposed MIA tool in its inference accuracy of the mission performance under a wide range of different environmental settings hindering the mission performance assessment and attack-defense interactions. Our results prove that the developed MIA framework shows

a sufficiently high inference accuracy (e.g., 80%) even with a small portion of the training dataset (e.g., 20-50%). We also found the MIA can better assess the system's mission performance when attackers exhibit clearer patterns to take more strategic actions using hypergame theory.

Interdependent Mission Impact Assessment of an IoT System with Hypergame-Theoretic Attack-Defense Behavior Modeling

Ashrith Reddy Thukkaraju

(GENERAL AUDIENCE ABSTRACT)

In our increasingly interconnected world, mission systems play a crucial role, whether in organizational networks or tactical military operations. We often evaluate these systems to ensure they perform effectively, but there's more to it than meets the eye.

Imagine an intricate web of resources, tasks, services, assaults, and defenses that are intertwined and have an impact on one another. The strategic interactions of attack and defense in uncertain environments have been majorly ignored by conventional techniques for mission impact assessment (MIA).

Our research introduces a new way of thinking about MIA. We've developed a framework that delves deep into the heart of mission systems, considering how each component affects the others. This comprehensive approach considers not just what's happening but also the interplay of actions and reactions. Hypergame theory, a technique that enables us to model intelligent choices in the face of uncertainty, is at the foundation of our approach. Imagine it as a chess game in which players must predict their opponents' moves and adjust their strategies appropriately. In our case study, we used an Internet-of-Things (IoT)-based mission system tasked with timely and accurate object detection to apply this architecture. In this mission system, both cyber attackers, whose aim is to compromise the mission, and cyber defenders, whose aim is to ensure mission success, are present, and they use the proposed hypergame-based decision-making to perform intelligent actions.

What did we find? Through extensive simulations, we confirmed the effectiveness of our

MIA framework. Even with limited training data, our tool demonstrated a remarkable 80% accuracy in assessing mission performance. Moreover, it excelled when attackers followed discernible patterns, allowing us to predict and respond strategically.

In simpler terms, our research provides a valuable tool for evaluating the success of mission systems in our increasingly connected world. It goes beyond surface-level assessments, considering the intricate relationships between system components and the dynamic nature of strategic decision-making. Ultimately, our framework empowers us to ensure mission success in an ever-evolving landscape.

Dedication

To my parents, Dr. Jadageesh and Dr. Soujanya, and brother Dr. Saketh, for all your love and support.

To my advisor, Dr. Cho, for your unrelenting guidance and confidence.

Acknowledgments

This work was partly supported by a grant (U22051XF) from the Agency of Defense Development, Republic of Korea.

Contents

List of Figures xi

List of Tables xii

1 Introduction 1

1.1 Motivation & Challenges 1

1.2 Research Goal 2

1.3 Key Contributions 3

2 Related Work 5

2.1 Methodologies of Cyber MIA 5

2.2 Cyber MIA Frameworks 7

2.3 Hypergame Theory 8

3 System Model 10

3.1 Network Model 10

3.2 Asset Model 11

3.3 Attack Model 14

3.4 Defense Model 17

4 Proposed MIA Framework **21**

4.1 Mission Description 21

4.1.1 Mission Scenario 21

4.2 Game-Theoretic Attack-Defense Interaction Modeling 25

4.2.1 First-Level Hypergame 26

4.2.2 Hypergame Normal Form (HNF) 26

4.2.3 Hypergame Expected Utility 29

4.3 Reasoning Model Generation 35

4.3.1 Impact Directed Graph (IDG) 35

5 Experimental Setup **38**

5.1 Simulation Environment 38

5.2 Metrics 38

5.2.1 Assessment Accuracy Metric 38

5.2.2 Mission System's Performance Metrics 39

5.3 Datasets 40

5.3.1 Microsoft Common Objects in Context (MS COCO) Dataset 40

5.3.2 Synthetic Datasets for Evaluating Our MIA Framework 41

5.4 Object Detection Model 41

5.5 Comparing Schemes 41

6 Numerical Results & Analyses **44**

6.1 Inference accuracy of MIA under varying the size of a training dataset ($\mathcal{P}_{TD}$) 44

6.2 Inference accuracy of MIA under varying the noises of a training dataset ($\mathcal{N}_{TD}$) 45

6.3 Inferred MoPE by MIA under varying the mean asset vulnerability: 46

7 Conclusions **50**

7.1 Key Findings 50

7.2 Limitations & Future Work 51

7.3 Publications 51

Bibliography **53**

List of Figures

3.1 IODM network model 11

4.1 Mission Timestep 22

4.2 Attacker Round 23

4.3 Defender Round 24

4.4 Overview of the proposed hypergame between an attacker and a defender where they view the same game differently and make decisions based on the Hypergame Expected Utility (HEU) [42]. 32

4.5 Impact Dependency Graph 36

6.1 Effect of varying the relative size of the training dataset to the testing dataset($\mathcal{P}_{TD}$) on the inference accuracy of Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome ($\mathcal{R}_{MS}$) under the six different mission scenarios. 47

6.2 Effect of Noise in the training dataset $\mathcal{N}_{TD}$ for different schemes on the inference performance for Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome (M). 48

6.3 Effect of varying the mean node vulnerability (μ_V) on Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome ($\mathcal{R}_{MS}$) under the six different mission scenarios. 49

List of Tables

3.1 Attacker's Strategies (Subgame: S_1^A – Infiltration, S_2^A – Expansion, S_3^A – Extraction/Action) 19

3.2 Defender Strategies and its corresponding Subgame, Conditions for Successful Defense, and Defense Impact 20

4.1 Variables in Bayesian Network and their defenitions 37

5.1 Key Design Parameters, Their Definition, and Default Values 39

List of Abbreviations

APT Advanced Persistent Threat

BN Bayesian Network

CCS Central Cloud Server

CKC Cyber Kill Chain

CVE Common Vulnerability Exposures

CVSS Common Vulnerability Scoring System

EV Encrytpion Vulnerabilities

HEU Hypergame Expected Utility

IDG Impact Dependency Graph

IODM IoT-based object detection mission

MEC Multi-access Edge Computing Server

MIA Mission Impact Assessment

NIDS Network Intrusion Detection System

NV Normalized Vulnerabilities

SBN Subjective Bayesian Network

SV Software Vulnerabilities

UV Unknown Vulnerabilities

Chapter 1

Introduction

1.1 Motivation & Challenges

With the rapid growth of computer usage over the past few decades, rapid advances have been made in IT technologies. Although this enabled solving challenging problems and greatly improved the world that we live in, it also comes with its cons. This enabled malicious users of the internet to exploit and target very important mission systems and cyber infrastructures. However, with the increasing complexity and effectiveness of the cyber attacks, there is also a proportionate increase in the effectiveness and performance of the defense strategies. Although studying the attacker's actions and their impact on the mission is commonplace, the interaction between the attacker and the defender strategies is mostly unexplored in the context of mission impact assessment.

Mission Impact Assessment (MIA) research has been explored to evaluate the performance and effectiveness of a mission system, such as enterprise networks with organizational missions and military or tactical mission teams with assigned missions. Although the importance of the MIA research has been discussed for more than a decade, most MIA approaches were mainly based on conceptual models with no extensive simulation analysis [2, 13, 21, 28, 30, 32, 34, 38]. In addition, despite the common practice of studying an attacker's actions and their impact on missions in the MIA research domain, modeling attack-defense interactions and their strategy selections remain largely unexplored. The key

components in such mission systems, including assets, services, tasks, vulnerability, attacks, and defenses, are interdependent, and their impacts are interwoven. However, the current state-of-the-art MIA approaches have less studied such interdependencies. In addition, they have not modeled strategic attack-defense interactions under partial observability. Game-theoretic approaches have been dominantly considered in cybersecurity research. However, they have not been much taken in the MIA research to model attack-defense interactions. To fill this gap, we propose an MIA framework to measure mission performance based on the interdependencies between the key system components. However, modeling a complicated mission system with all the critical modules, including the mission, asset, vulnerabilities, and attack and defense models, and constructing an impact dependency graph (IDG) [18] is non-trivial. Further, the previous MIA research has not explored comprehensive design, implementation, and simulation-based validation.

Finally, despite the fact that Machine Learning (ML)/ Deep Learning (DL) has been widely used in mission systems, the impact of hypergame theoretic EMB on such missions has not been investigated or quantified by existing assessment tools. For tactical mission systems, we consider an IoT-based object detection mission system and propose a Mission Impact Assessment (MIA) framework.

1.2 Research Goal

We aim to create a cyber mission impact assessment (MIA) framework in order to evaluate the performance and effectiveness of an IoT-enabled tactical ML-based mission system. We will look into the interdependence of the mission system's key components, their vulnerabilities, their exploitability, and the defense countermeasures against the attacks. The proposed MIA approach will assess the tactical ML-based mission system's performance as a function

of its security and performance in the face of adversarial attacks and network dynamics in distributed, resource-constrained tactical environments.

1.3 Key Contributions

This work aims to develop an MIA framework to evaluate the performance and effectiveness of an IoT-enabled machine learning (ML)-based mission system. We will develop the MIA framework by considering the interdependencies of the key system components and examine how strategic attack-defense interactions and their decision-making behaviors impact the inference accuracy of the MIA under a wide range of adverse factors impacting mission performance. Specifically, this work makes the following **key contributions**:

1. **Mission performance analysis of an ML-based IoT system**: Most existing MIA studies considered mission scenarios in common enterprise networks or tactical mission teams for disaster management or military mission execution. No prior work has considered an ML-based IoT system requiring intelligent operations and decision-making. Our work is the first to propose the MIA for AI-based mission systems and its validation based on the extensive experimental results obtained from our developed simulation model.

2. **Modeling hypergame-theoretic attack-defense interactions**: Our work first introduces hypergame-theoretic attack-defense interaction modeling to realistically consider their rational decision-making process under uncertainty for their best strategy selection.

3. **Bayesian Network (BN)-based interdependent MIA**: We leverage BN for the interdependency analysis of the system's key components and perform the causality

analysis of the mission's assets, tasks, and services. This allows more accurate inference of assessed mission performance.

4. **Investigation of the interplay of different attack and defense behavior modeling**: No prior work has conducted an in-depth analysis of how different attack-defense interactions can impact the inference accuracy of assessed mission performance by MIA. Via extensive simulation study, we demonstrate the effects of differently modeling the attack-defense interactions and their strategy selection process (i.e., six different scenarios) on the inference accuracy of the MIA framework and the assessed mission performance by the MIA in a wide range of varying environmental conditions.

Chapter 2

Related Work

In this section, we provide an overview of related existing works regarding the methodologies of cyber Mission Impact Assessment (MIA), MIA frameworks, and hypergame theory.

2.1 Methodologies of Cyber MIA

There are two major approaches to cyber mission impact assessment (MIA): *Threat-centric* and *mission-centric*. *Threat-centric MIA* uses the enemy/threat point of view to estimate the impact. Some of the known works that use threat-centric approaches are the Network Risk Assessment Tool (NRAT) [43] and Mission-Oriented Risk and Design Analysis (MORDA) [7]. These frameworks provide an analysis of the security of a system from the context of an attacker's attributes. They are not, however, intended for state-based analyses where an adversarial action is depicted as a dynamic selection of a full attack vector. ADdversary VIew Security Evaluation (ADVISE) [23] is an alternative method that employs multi-step attacks and quantitative metrics to allow trade-off evaluations across alternatives. It can, however, only be utilized in a planning setting and is not ideal for supporting real-time scenarios. Often utilized in Terrorism Risk Management [12], these techniques above are used to comprehend the attacker's actions or assess the implications of an attack against key infrastructures and people. Like in the case of the September 11th attacks, if the attacker's information is inadequate or insufficiently known, the protection offered by the

threat-centric approaches is ineffectual. In general, threat-centric approaches are used due to their simplicity. However, the need to know the attacker's action, behavior, and goals in detail make this paradigm unfit for real operations.

Mission-centric MIA was initially proposed to envision the evaluation of mission impact through understanding the key requirements to accomplish mission goals, as explained in their Cyber MIA (CMIA) framework [29, 30, 31, 32]. In CMIA, the approach describes a mission in a business process language, which includes the activities necessary to achieve the mission's goals and their limitations (e.g., infrastructure, services, and dependencies). The framework's fundamental idea is to determine the most significant assets to achieve the goals. Furthermore, their monitoring aids in understanding the implications of changes to these assets on the mission. The primary limitation of this framework is its inability to analyze cascading effects, which prevents it from capturing the combined impact of simultaneous attacks. Another drawback is that its offline mode is oblivious to immediate impacts, making it difficult to prioritize resources during mission execution. Furthermore, the lack of elaboration about performing the stages of CMIA raises numerous uncertainties about whether the technique can be implemented and if it can provide the promised results. The CMIA's emphasis on consequences constitutes a significant milestone in quantifying cyber impact, enabling impact assessment for zero-day attacks. The mission and cyber domain mapping is another prominent contribution that allows a mission analyst to comprehend the effect of cyber events on a mission. CMIA framework research does not contain real-world scalable implementations because of its emphasis on system architecture design.

2.2 Cyber MIA Frameworks

MIA research has been explored for over a decade in evaluating various mission systems in enterprise organizations and military or tactical contexts. An Operationally Critical Threat, Asset, and Vulnerability Evaluation framework (OCTAVE) [2] is proposed to identify and provide relevant information on asset criticality, threats, and associated vulnerabilities. The information was used to manage information risks for an organization to implement a protection strategy for its information assets. However, OCTAVE mainly aimed to identify information risks in organizations. MITRE proposed an integrated method to understand the mission impacts of cyber attacks, called *Analyzing Mission Impacts of Cyber Actions* (AMICA) [34]. AMICA provides an integrated solution combining process modeling, simulation, attack graph-based dependency modeling, and dynamic representations. AMICA considers cyberattack and defense tactics, techniques, and procedures (TTPs) along with the mission process (i.e., how mission tasks are executed). MITRE also proposed a Cyber Command System (CyCS) tool [38] to increase cyberspace mission assurance based on the mission operations to network operations mapping for supporting missions. The CyCS is designed to provide MIA through situational awareness and impact analysis by providing relevant information about vulnerability, threat, and consequence management. MITRE further provided solutions to the Cyber Defense Situational Awareness (CDSA) capability, initiated by the NATO Communications and Information Agency (NCIA) Request for Information (RFI) (CO-14068-MNCD2) [28]. Besides, MITRE developed a Cyber Mission Impact Assessment (CMIA) tool [30, 32] to represent cyber resources, processes, and cyber incident consequences. However, CMIA did not consider the dependencies between asset vulnerabilities and attack/defense effects.

A cyber situational awareness (CSA) framework, called CRUSOE [21], is presented as an

extensible, layered data model. This work interviewed several security teams to ask how incidents are handled and evaluated present requirements on CSA. The CRUSOE data model monitors missions, assets, networks, devices, attacks, detection and mitigation capabilities, and access control in an organization's network. This work mainly demonstrated ontological relational views about seven different layers with no experiment-based validation: mission, system, network, host, threat, detection and mitigation, and access control. However, no simulation experiments are conducted to validate their framework. In 2012, A Framework for Evaluating the Impact of a Cyber Attack on a Physical Mission (FEICAPM) de Barros Barreto et al. [13]. FEICAPM consists of mission modeling, network topology building, relevant information collection, and impact assessment to provide the most pertinent information to a decision-maker. This was used as the basis of the recent mission assurance framework, called Cyber-ARGUS [3], in 2019, which is a Command and Control (C2) support system. The Cyber-ARGUS comprises technologies and tools that provide a consistent and coherent mapping between the operations and the cyber domain. It stores necessary information about the devices in a cyberinfrastructure in a knowledge base (KB), allowing functionality to construct a Bayesian Network to assess the impact between these two domains. However, it did not consider the strategic attack-defense interactions and how their strategy selection processes impact the asset capabilities of a given mission system. We consider the attack-defense interactions and their best strategies based on hypergame theory to realistically consider their strategic decision-making under uncertainty.

2.3 Hypergame Theory

Game theory [41], developed by mathematician John von Neumann and economist Oskar Morgenstern, is a model for interactive decision-making in the presence of multiple players

where the decisions of one player impact the others. It has been widely used in various application fields like economics[4], political sciences[27], biology[14], and computer science [36]. *Hypergame theory* [5] extends game theory focusing on conflict games where subjective perception (e.g., misperception) is present. It aims to explain the phenomenon of players holding varying perspectives regarding the nature of the conflict. Misperception is true in the attack-defense interactions in a cyber system [17]. Determining the strategy adopted by a player relies not only on their observation of the game but also on their perception of how their opponent views the game. Hence, it is widely used to realistically model attacker-defender interactions [10, 16, 42] in the cybersecurity domain. We leverage hypergame theory to model realistic attack-defense interactions for rational decision-making based on players' perceived subjective beliefs toward their opponents. Hypergame theory is especially useful in the context of cyber security as it has the concept of subgames, which can be aligned to the multiple stages that are usually present in the various attacker models like Advanced Persistent Threat (APT)[8, 37], Cyber Kill Chain (CKC)[1, 9, 20, 44], etc.

Chapter 3

System Model

In this section, we discuss the details of the network, mission, attack, defense, and vulnerability models used in our work. In addition, we describe the assumptions made in this work to design our framework.

3.1 Network Model

IoT-Edge networks have been widely used recently to reduce the latency of the services and also offload the computation from a central server. IoT-based edge networks have been widely used to reduce the latency of the services and offload the computation from a central server [33]. Our mission system is deployed in an IoT network consisting of various IoT devices (e.g., surveillance cameras, handheld devices, laptops, and other mobile sensors like Vehicles), and servers, as described in Fig. 3.1. For our IoT-based object detection mission (IODM), the detection happens at the edge on the multi-access edge computing servers. The sensor nodes collect the data (images) and periodically forward it to the edge-computing servers to perform the task of object detection. Each varied node in the network is part of its Virtual Local Area Network (VLAN) and provides its own set of functionalities and services. These VLANs may have one or more servers, depending on their functionalities, for the IoT devices to communicate. The goal of these servers would then be to collect, process, perform object detection, and respond with the appropriate results to the sensor nodes. The CCS

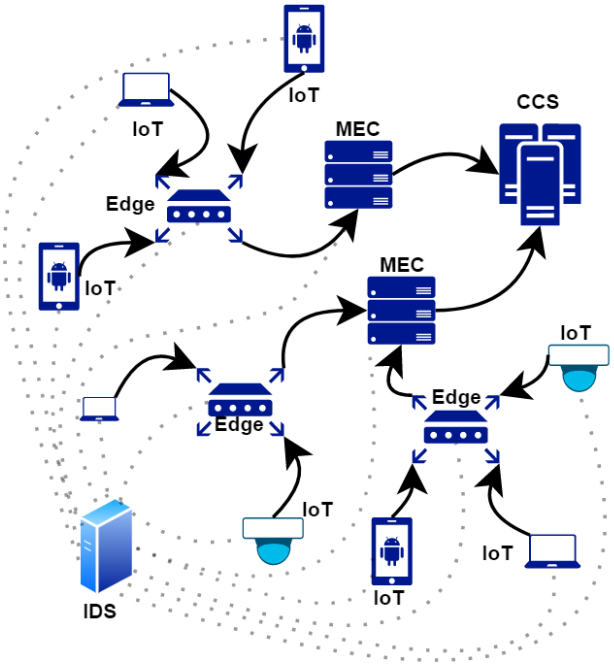


Figure 3.1: IODM network model

central server has the object detection model that will be forwarded to every MEC server at regular intervals. The cloud server is assumed to be present at a secure location safe from attackers. Hence, the CCS is assumed to be trusted. Moreover, the CCS, which has huge computation and energy resources, is assumed to have a powerful object detection model that can be used to attain ground truth predictions to measure the mission's performance. Figure 3.1 describes a network concerned in this work.

3.2 Asset Model

We consider the following assets for our considered mission system:

- **IoT Devices:** IoT devices are lightweight computing devices with sensors, processing power, and software and services that allow them to capture data and communicate with other devices on the Internet or a specific communication network. For our

mission, the sensors in the IoT devices need to have the ability to capture images or a video stream of objects in the given environment. For this purpose, we consider devices such as laptops, cameras, surveillance cameras, handheld devices like smartphones, and other miscellaneous sensors that can capture images.

- **Network Edge Devices:** Network Edge Devices include hardware at the boundary of a communication network connecting the devices in a given network with other devices in other networks. Essentially, these devices serve as network entry or exit points. Edge routers, routing switches, firewalls, and other Wide Area Network (WAN) devices are edge servers. For our mission system, to ensure simplicity and consistency in design, we only consider one layer of network edge devices as shown in Figure 3.1.

- **Multi-access edge computing (MEC) server:** MEC servers are the computing devices directly connected to the network edge devices. These intelligent servers have computing processors that allow us to run machine learning and other applications to provide services closer to the user to reduce network congestion and response delays that occur in traditional client-server-based architectures. Usually, these servers help cellular and other such networks provide edge computing abilities.

- **Cloud Central Server (CCS):** The CCS is where the pre-trained object detection model is present. The CCS distributes the trained model to all the MEC servers in the network. At regular intervals during the mission, the CCS updates the pre-trained model at the MEC since the MECs are prone to data poisoning attacks and the model could be poisoned by the attacker. The CCS is assumed to be trusted and out of reach from attackers.

In the given mission system, each asset, except the CCS, has the asset capacity, $\mathcal{R}_{AC}$, represented by a function of CPU and memory load. $\mathcal{R}_{AC}$ is given by: $\mathcal{R}_{AC} = e^{-\gamma t}$. Con-

sidering the short mission duration, we omit the effect of time (i.e., $t = 1$) and define γ by: $\gamma = \lambda(w_1 L_{\text{CPU}} + w_2 L_{\text{memory}})$, where w_1 and w_2 are the weights for the CPU and memory loads, respectively, where $w_1 + w_2 = 1$ and the weights are imposed as a system requirement. λ is initialized differently for each node type (i.e., IoT device, NED, and MEC) in our mission environment.

Asset Vulnerability In our environment, an asset's vulnerability level decides the likelihood of an attack's success it. We define three types of vulnerabilities that are associated with each node as follows:

1. **Software Vulnerability(SV):** The vulnerabilities associated with the software and services presented in an asset can be exploited by the attackers to compromise them to become inside attackers. Each asset's SV (sv_i) is modeled based on a random variable with a mean of μ_{sv} and a standard deviation of σ_{sv}, ranged in $[0, 1]$.

2. **Encryption Vulnerability(EV):** The longer an attacker stays inside the network, the more the Encryption vulnerability of an asset is. This is due to the fact that when the inside attacker stays longer in the system, it can have a better chance of getting other legitimate node's secret keys. By default, each asset is initialized with an EV value at random with a mean of μ_{ev} and a standard deviation of σ_{ev}. The EV of a node i is calculated as:

$$\hat{\text{ev}}_i = \text{ev}_i \times \text{e}^{-1/T_{\text{rekey}}}, \tag{3.1}$$

   where ev_i is the initial value of the EV of an asset i, and T_{rekey} is the time elapsed since the last time the encryption keys were updated.

3. **Unknown Vulnerability(UV):** This accounts for all the vulnerabilities that the SV

and EV can not capture. The mission analyst can assign a higher or a lower value for UV based on their perception of the system and the risk of the assets. By default, each asset is initialized with an EV value at random with a mean of μ_{uv} and a standard deviation of σ_{uv}.

Each type of asset vulnerability (SV, EV, and UV) has a value between 0 and 1. For simplicity, we consider three vulnerabilities of each asset (i.e., node) by using their mean (i.e., Normalized Vulnerability, or NV),

$$\mathrm{nv}_i = w_{sv}\mathrm{sv}_i + w_{ev}\hat{\mathrm{ev}}_i + w_{uv}\mathrm{uv}_i, \tag{3.2}$$

where $w_{sv} + w_{ev} + w_{uv} = 1$. In our case study, we modeled equal weight for each type of vulnerability. Assets normalized vulnerability score (NV) is defined as the average of its three vulnerability scores.

$$\mathrm{nv}_i = \frac{\mathrm{sv}_i + \hat{\mathrm{ev}}_i + \mathrm{uv}_i}{3} \tag{3.3}$$

3.3 Attack Model

MITRE ATT&CK [25] is a framework that captures malicious behaviors used by attackers in real-world cyber attacks at various phases. ATT&CK stands for Adversarial Tactics Techniques and Common Knowledge and it contains thorough descriptions of these groups' observed tactics and techniques and procedures. Attackers behave differently depending on the target of their attack. They use distinct Techniques, Tactics, and procedures to compromise the enterprise system than they would to mobile devices or industrial control systems, for example. MITRE ATT&CK has organized attack tactics and techniques, including 14 tactics, 185 techniques, and 367 sub-technologies. Reconnaissance, initial access,

execution, and collection are examples of tactics used by attackers to achieve attack targets and techniques define methods for achieving tactics such as man-in-the-middle and phishing attacks.

For this work, we consider some of the most common attacks in the IoT environment. These sets of attacks are chosen from the list of MITRE's ATT&CK framework and chosen so they target each of the security requirements (CIA security goals, including Confidentiality, Integrity, and Availability). Particularly, We choose a modified version of the 3-stage Advanced Persistent Threat (APT) framework [22] to model the attacker's behavior. To make this emulation plan compatible with our attacker's goal, which is to reduce the mission performance and effectiveness, we propose the 3rd stage as an Action similar to the last stage in MCKC [20] and the Lockheed Martin CKC [1]. The modified 3 stages of APT are: 1) Infiltration 2) Expansion 3) Extraction/ Action. For our hypergame, we consider each of these stages as a subgame. We choose the following 7 attack strategies for the choice of actions for the attacker in our mission.

1. **Vulnerability Scanning (T1595.002):** In a vulnerability scanning attack, the attacker can use various tools and resources to gather information about the configuration of the victim machine. More specifically, the attacker scans for the software services available and their version numbers to potentially target them using known or crafted exploits.

2. **Phishing (T1566):** Phishing is a form of social engineering where the adversary uses electronic means like emails, messages, and malicious file attachments, typically with the intention of infiltrating the victim's device. These attachments usually have hidden malware that runs malicious code on the victim's system. After infiltration using phishing, the attackers could perform various attacks to impact the mission

outcome.

3. **Exploit Public-Facing Application (T1190):** The attackers usually try to exploit the weaknesses in the internet-facing applications or services running on the victim machine. These vulnerabilities allow the attacker to inflict damage to the mission in various forms based on the type of vulnerability. The attackers can exploit a single or multiple services running to get access to the victim machine.

4. **Exploitation for Privilege Escalation (T1068):** After infiltrating into the victims system using the techniques specified above, the attacker can choose to exploit the software and other vulnerabilities to escalate privileges. Doing so would let the attacker run code controlled by the attacker in the victim machine. Escalating privileges would let the attacker perform actions that would otherwise require higher permission levels to perform.

5. **Brute Force (T1110):** Once infiltrated, the attackers can use various tools and scripts to crack the passwords for accounts, services, and applications present in the victim machine. Credential brute forcing allows the user to escalate privileges and inflict more damage to the victim machine than otherwise possible.

6. **Stored Data Manipulation (T1565.001):** After infiltration into the victim machine with the ability to execute remote code, the attacker can manipulate the contents of the local file system. The attacker can also manipulate the network traffic and the data that needs to be transmitted to other machines. The adversary can poison the data that is present in the victim machine and breach the integrity of the mission. This would directly impact the integrity of the mission outcome.

7. **Network Denial of Service (T1498):** Denial of Service is one of the most common attacks in any network environment. In this attack, the attacker could flood the

network or a particular system in the network with huge volumes of traffic to inundate the victim and impact its normal functionality. This would impact the availability of a service to genuine users. A variation of this attack is *Distributed Denial of Service (DDoS)*, in which the attacker floods the victim machine with traffic from multiple machines using botnets.

3.4 Defense Model

The defense model consists of all the possible defensive mechanisms that a defender can choose from. According to MITRE's D3FEND framework [26], a model can accurately list the capabilities of the cybersecurity defense mechanisms. It is a comprehensive knowledge graph with over 500 defensive mechanisms from the literature. It defines five defense tactics: Harden, Detect, Isolate, Deceive, and Evict. These five classes of tactics are further classified into 17 defense techniques.

The MITRE's D3FEND framework uses digital artifacts (Files, Network traffic, Software, and other Top-level artifacts) to map its ATT&CK framework to its D3FEND framework. This is very useful since once the attack is identified, we have a limited number of defensive mechanisms to choose from to defend against the said attack.

For this work, we consider the following subset of Defense mechanisms from the set of Defenses specified in the MITREs D3FEND framework. These can be categorized based on the D3FEND framework as: 1) Harden 2) Detect 3) Isolate. The defense mechanism corresponding to the 3-stage APT emulation plan are as follows

1. **Defenses against Infiltration:**

    - **Dynamic Analysis (D3-DA):** Dynamic Analysis is a method of opening a

file in a sandbox environment to contain malware or other viruses within the sandbox. This analysis helps detect and mitigate malware and viruses received from untrusted sources or attackers.

- **Software Update (D3-SU):** Most vulnerabilities that the attackers use as a means of infiltration are usually in the software services provided by the victim machines. The software vulnerabilities are often patched in the newer versions, while the device might not have the updated version installed. Hence, a simple defense mechanism to prevent attackers from infiltration is to keep the software and services up-to-date. Another way to deal with a vulnerability is to create a temporary patch or a workaround for it.

2. **Defense Against Expansion:**

- **Local File Permissions (D3-LFP):** Restricting the access of files to a set of users would make it difficult for the attacker to read, write, or delete even after infiltrating into the victim's machine. However, the attacker could achieve privilege escalation and get permission to modify or manipulate the data.
- **File Encryption (D3-FE):** File Encryption is very similar to message encryption in terms of the encryption mechanisms used. The encryption is applied to the files on a local file system. Encrypting the files in the local file system would make it harder for the attacker to read or manipulate the data in the compromised victim system.
- **Message Encryption (D3-MENCR):** This method is the most common way to ensure confidentiality of the messages sent over the network is to encrypt the messages. Symmetric (i.e., shared secret key) and asymmetric (i.e., public and private keys) cryptography can be used depending on the context. When the key is known to the attacker or when the attacker can break the encryption, the

Table 3.1: Attacker's Strategies (Subgame: S_1^A – Infiltration, S_2^A – Expansion, S_3^A – Extraction/Action)

Subgame	Attack Strategies	EXV	Attack Impact
$S_1^A - S_3^A$	Vulnerability Scanning	NV	Asset identification
$S_1^A - S_3^A$	Phishing	NV	Victim node compromise
$S_1^A - S_3^A$	Exploiting public vulnerability	SV	Victim node compromise
S_2^A	BruteForce	EV	EV = 1, credentials leaked, no rekeying
S_2^A	Privilege Escalation	NV	Increment UV by ϵ% (10)
S_3^A	Data manipulation	NV	Significant adverse impact on prediction accuracy
S_3^A	DoS	N/A	Node unavailability

quickest fix would be to change the encryption key or the encryption algorithm used.

3. **Defense against Extraction:**

   - **Network Traffic Filtering (D3-NTF):** It is to restrict traffic originating from a particular network or a device or destined towards a specific network or device. To deal with packet flooding attacks like SYN Flooding, DoS, and Distributed Denial of Service DDoS), Network traffic blacklisting or allowing listing is used.
   - **Network Traffic Community Deviation (D3-NTCD):** Identification of statistically deviant communications in a pre-defined network community helps capture peculiar network traffic patterns. A Network-based Intrusion Detection System (NIDS) is a tool that monitors the network traffic for aberrations from a predefined rule set. These rules are to capture the patterns in malicious network packets.

The subgames of the attacker and defender and their corresponding strategies in each are specified in the table below. Each item in the table corresponds to a strategies ID in the

Table 3.2: DEFENDER STRATEGIES AND ITS CORRESPONDING SUBGAME, CONDITIONS FOR SUCCESSFUL DEFENSE, AND DEFENSE IMPACT

Subgame	**Defense strategies**	**Successful defense condition**	**Defense cost**	**Defense impact**
S_1^A	Dynamic Analysis	Applied on each node with a success probability of P_{da}	Low	Reduce UV by ζ_{uv} % and Phishing fails if DA is used at attacker location
S_1^A	Software Update	Applied on each node with success probability of P_{su}	Low	Reduce SV by ζ_{sv} % for every node
S_2^A	Local File Permissions	Successful if this is applied in a non-compromised node	Medium	Reduce UV by ζ_{uv} % for every node
S_2^A	Encryption	Applied on each node with success probability of $1 - P_{ev}$ where P_{ev} refers to the probability associated with EV	Medium	EV is set to initial values
S_3^A	Network Traffic Filtering	P_e % success probability of blacklisting (evicting) the attacker node	High	Reduce the UV of all nodes by ζ_{uv} and evict the attacker node if successful
S_3^A	Network Traffic Community Deviation	Detecting a compromised node by NIDS with true positive, P_{TP} and true negative P_{TN} probabilities	High	Evict detected compromised nodes

(Subgame: S_1^A – Infiltration; S_2^A – Expansion; S_3^A – Extraction/Action; Defense cost: Low – 2.5, Medium – 5, High – 10)

MITRE ATT&CK and D3FEND Framework.

Chapter 4

Proposed MIA Framework

Most existing approaches in this domain used attacker strategies and system vulnerabilities to create a framework to assess a system's performance and effectiveness. The advances in cybersecurity technologies lead a mission system to be equipped with defenses in place to detect and respond to adversaries. This necessitates considering the impact of defense mechanisms and the interactions between the attack and the defense mechanisms on the system as a whole when assessing the performance of a complex system. The interaction between the attackers and defenses in a system or a network plays a significant role in its outcome. No prior work has considered the interactions between the attackers and the defense mechanisms when designing an MIA framework to the best of our knowledge. We shall provide models of game-theoretic interactions that can describe the attacker's and the defender's strategic decision-making, as well as the influence of their strategic decisions on the system or the network state, including its performance and efficiency.

4.1 Mission Description

4.1.1 Mission Scenario

The considered mission is to perform an object detection task on the images received from the IoT devices at the MEC servers. The IoT devices will capture the images of various

objects in the environment, and forward them to the edge server they are connected to. The task of the edge devices is to forward the images and the results to the MEC and the IoT devices respectively. The MEC devices then perform object detection on the received image using the model available. The CCS will send the detection model to the individual MEC servers at regular intervals to ensure that the mission does not fail due to local model poisoning attacks.

In the mission, multiple events occur at each time step. In a time step, the flow of events is as follows.

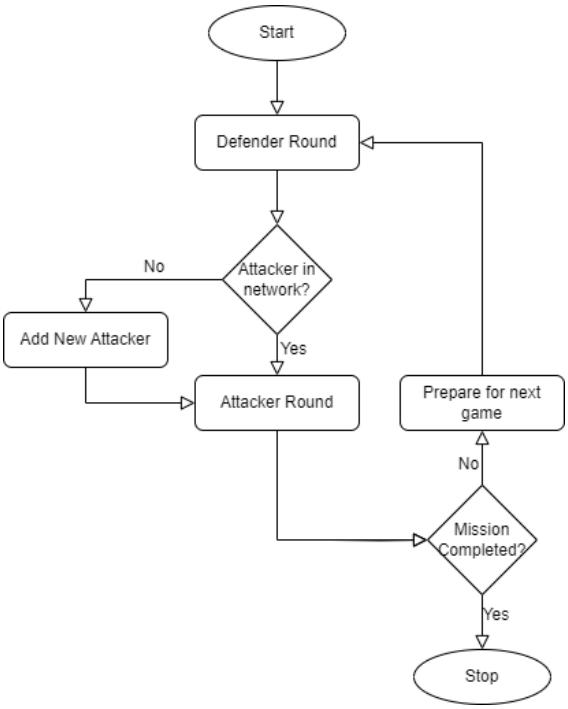


Figure 4.1: Mission Timestep

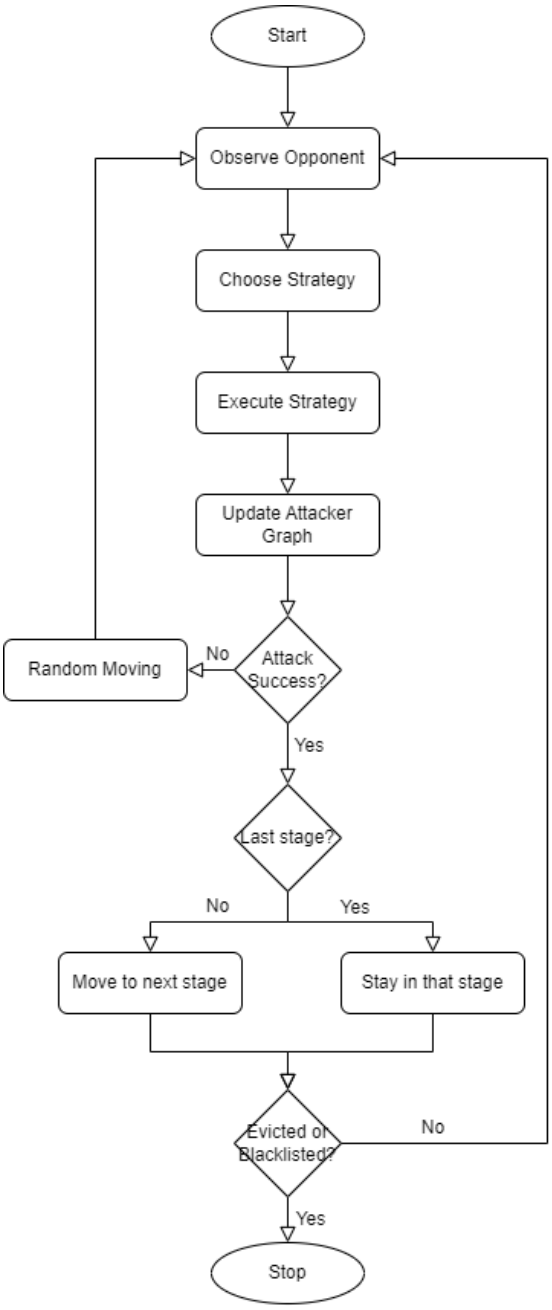


Figure 4.2: Attacker Round

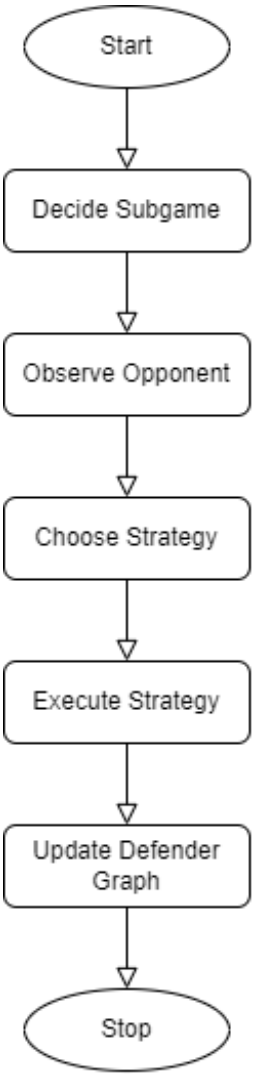


Figure 4.3: Defender Round

1. Mission continues till no images are left for object detection and all results are received.

2. Each node that has information to forward (Images for IoT nodes, results for the MEC node, and images, and results for the Edge nodes) sends it to the target node.

3. The defender performs his actions:

   (a) The defender observes the attacker and guesses the attacker's subgame.

(b) The defender chooses the best strategy in that subgame based on the Hypergame Expected Utility (HEU) and executes it.

(c) If the defense strategy chosen is a NIDS evict operation, the attacker is evicted from the network with a specified False positive and False Negative rate.

(d) The defender updates its graph based on the outcome of the actions.

4. The attacker performs his actions:

(a) If there is no attacker present in the network, a new attacker is added.

(b) The attacker observes the defender.

(c) The attacker chooses the best strategy in the current subgame based on the Hypergame Expected Utility (HEU) and executes it.

(d) The attacker updates its graph based on the outcome of the actions.

(e) If the attack succeeds, the attacker moves to the next subgame. If the attacker in the last subgame, and succeeds, then a new attacker is created

(f) If the attack fails, the attack moves to a new random location.

5. The environment is updated and the metrics are computed.

6. Prepare for the next game (next timestep).

4.2 Game-Theoretic Attack-Defense Interaction Modeling

We model the behavior of the attacker and the defender as a game-theoretic interaction between two players. In real-world scenarios, uncertainty is usually involved, and each player

has their perception of the game. Hence we choose to model the game between the attacker and the defender using the hypergame theory. A hypergame is formulated as (G, G_A, G_D) where G is the actual game, G_A and G_D are the attacker and defender's perceived games, respectively. When all these three games G, G_A, and G_D are the same, the hypergame simplifies to the standard game theory.

4.2.1 First-Level Hypergame

The simplest case is where we have two players p and q with V_p and V_q as the actual preferences of players, respectively. For this case, the game can be defined as $G = \{V_p, V_q\}$. Although, in reality, this assumption might not be valid since each player p and q might have their view about the game and their own perceived preferences. For such a case, player p's game based on the perception of q's preferences, V_{qp}, and player q's game based on the perception of player p's perception, V_{pq}, as $G_p = \{V_{qp}\}$, $G_q = \{V_{pq}\}$, respectively. The First-Level Hypergame can be formulated as:

$$H^1 = \{G_p, G_q\}. \tag{4.1}$$

4.2.2 Hypergame Normal Form (HNF)

The Hypergame Normal Form (HNF) models the hypergame based on the players' beliefs and opponent strategies. Four components constitute HNF: Full Game, Row-Mixed Strategies (RMS), Column-Mixed Strategies, and belief contexts.

A **Full Game** is defined as a matrix containing both the row and column strategies associated with their utilities $ru_{11}, \ldots, ru_{mn}$ and $cu_{11}, \ldots, cu_{mn}$, respectively. The corresponding $m \times n$

matrix U is:

$$\mathbf{U} = \begin{pmatrix} (ru_{11},\, cu_{11}) & \cdots & (ru_{1n},\, cu_{1n}) \\ \cdots & \cdots & \cdots \\ (ru_{m1},\, cu_{m1}) & \cdots & (ru_{mn},\, cu_{mn}) \end{pmatrix}. \tag{4.2}$$

The **Row-Mixed Strategies (RMSs)** are the strategies based on how a row player perceives a column player's strategies. Because the two players limit the scope of the strategy they consider based on their perceptions, the player's subgame is a subset of the full game. Suppose the row player chooses to play a subgame κ, the RMSs of the subset κ are:

$$\text{RMS}_\kappa = [r_{\kappa 1}, \cdots, r_{\kappa m}], \quad \text{where } \sum_{i=1}^{m} r_{\kappa i} = 1, \tag{4.3}$$

where each strategy i represents an estimate based on the row player's knowledge gained from prior experience. In any given subgame, the player selects one strategy to play. All the strategies not in the chosen subgame will have a zero probability (i.e., $r_{\kappa i} = 0$).

The **Column-Mixed Strategies (CMSs)** are the strategies based on how a column player perceives a row player's strategies. Suppose the column player chooses to play a subgame κ, the CMSs of the subset κ are:

$$\text{CMS}_\kappa = [c_{\kappa 1}, \cdots, c_{\kappa n}], \quad \text{where } \sum_{i=1}^{n} r_{\kappa i} = 1, \tag{4.4}$$

where each strategy i represents an estimate based on the column player's knowledge gained from prior experience. In any given subgame, the player selects one strategy to play. Even for the column player, similarly to RMSs, $c_{\kappa i} = 0$ if the strategy i is not a part of the subgame.

The action space for the attacker consists of all the attack strategies $A_{actions} = \{A_1, A_2, \ldots, A_i\}$ and the action space for the defenders $D_{actions} = \{D_1, D_2, \ldots, D_i\}$ are the set of their defense strategies against the specified attacks.

Belief contexts are the probabilities of each subgame κ being played based on the row player's belief.

$$P = [P_0, \cdots, P_K], \text{where } \sum_{\kappa=0}^{K} P_\kappa = 1, \quad \text{where } P_\kappa = \frac{\rho_\kappa}{\sum_{\kappa \in \mathbf{AT}} \rho_\kappa},$$

where P_0 is the probability of playing the full game, and for all other i's P_i is the probability of playing the ith subgame. **AT** is the set of attackers in the game, and ρ_κ is the number of attackers in the subgame κ. When the player is unsure about which subgame to play, it plays the full game.

The belief of the row player that the column player takes the strategy j is S_j, and it is given by:

$$S_j = \sum_{\kappa=0}^{K} P_\kappa c_{\kappa j} \quad \text{where } \sum_{j=1}^{n} S_j = 1. \tag{4.5}$$

We denote $C_{\sum} = [S_1, S_2, \cdots, S_n]$ by the set of row players beliefs that the column player takes all n strategies.

In a given subgame κ, the **Belief Calibration** can be computed based on the frequency of the strategies taken in the past. At the beginning of a subgame, both the players make random decisions. However, as time progresses, the players have a more shaped probability distribution to choose their actions. The probability of the row player choosing a strategy p in a subgame κ at t is denoted as $r^t_{\kappa p}$ and the column player choosing a strategy q at the same time t in the same subgame κ as $c^t_{\kappa q}$. The $r^t_{\kappa p}$ and $c^t_{\kappa q}$ can be estimated by modeling

them as a Dirichlet distribution by:

$$r^t_{\kappa p} = \frac{\gamma^t_{\kappa p}}{\sum_{p \in \mathbf{RS}_\kappa} \gamma^t_p}, \quad c^t_{\kappa q} = \frac{\gamma^t_{\kappa q}}{\sum_{q \in \mathbf{CS}_\kappa} \gamma^t_q}, \tag{4.6}$$

where $\mathbf{RS}_\kappa$ and $\mathbf{CS}_\kappa$ are the set of row and column player's strategies, respectively. At a time t in a γ^t_p and γ^t_q is the cumulative amount of evidence both the players gathered in the time interval $[0 - (t-1)]$.

Hence, in a subgame at a time t, for every player i who chooses a strategy p, HEU can be computed by:

$$\text{HEU}(rs_{ip}) = \text{HEU}(p, g^t_i), \tag{4.7}$$

where g^t_i is player i's uncertainty at time t.

4.2.3 Hypergame Expected Utility

Every player in hypergame theory chooses actions based on the Hypergame Expected Utility (HEU). The expected utility $EU(\cdot)$ and perceived uncertainty g are used to obtain HEU, which expresses the player's degree of uncertainty in interpreting the game and the strategies of the opponent. Each player observes the opponent's action based on their estimated perceived uncertainty. In our case, the attacker is uncertain about the defender's chosen strategy, and the defender is uncertain about the attacker's strategies or whether the attacker has performed any attack. If the player is completely uncertain, we set $g = 1$ while with complete certainty, $g = 0$ [40]. The HEU of a row player's strategy rs_i with a given

uncertainty g is computed by [39]:

$$\text{HEU}(rs_i, g) = (1 - g) \cdot EU(rs_i, C_{\sum}) + g \cdot EU(rs_i, CMS_w), \tag{4.8}$$

where $EU(rs_i, C_{\sum})$ is the Expected Utility of the row player when the column player chooses any one of their strategies. $EU(rs_i, CMS_w)$ denotes the row player's expected utility when the column player chooses the worst-case strategy (i.e., a strategy with the least utility) against the row player. The EU of the row player for the certain and uncertain cases is given by:

$$\text{EU}(rs_i, C_{\sum}) = \sum_{j=1}^{n} S_j \cdot u_{ij}, \tag{4.9}$$

$$\text{EU}(rs_i, \text{CMS}_w) = n \cdot S_w \cdot u_{iw}. \tag{4.10}$$

The HEU value can take both positive and negative values. To normalize HEU to take a positive value between 1-10, we use the min-max normalization method [15].

We model the perceived uncertainties of the attacker and the defender in a subgame as the following:

$$g_t^A = e^{-\lambda t}, \;\; g_t^D = e^{-\mu t}. \tag{4.11}$$

The value of these functions decreases as time t (the time for which the attacker and defender have monitored the target system) increases. In other words, the more time monitored, the more certain the attacker and the defender are about the system. In the above equations, λ and μ are parameters that control the attacker and the defender risk risk-taking ability. The higher these values are, the more steeply the uncertainties of the attacker and the defender's perception decrease, meaning that the player's confidence increases initially.

In a hypergame, each player considers themselves a row player and the opponent a column player. The **utility** when a row player chooses a strategy p and the column player chooses the strategy q can be computed as the difference between the gain and the loss.

$$u_{pq} = G_{pq} - L_{pq}, \tag{4.12}$$

Since we model the hypergame as a zero-sum game, one player's gain is the opponent's loss, and one player's loss is the opponent's gain. The gain and loss can be calculated by:

$$G_{pq} = \mathrm{r}_p^{impact} + \mathrm{c}_q^{cost}, \quad L_{pq} = \mathrm{r}_p^{cost} + \mathrm{c}_q^{impact}, \tag{4.13}$$

where r_p^{cost} and r_p^{impact} are the cost and impact respectively of the row players strategy p and c_q^{cost} and c_q^{impact} are the cost and impact for the column players strategy q. In our case, we need to model the cost and effectiveness of the attacker's and the defender's strategies.

For the attacker strategies, we can use the prior information about the exploitability to model the cost and impact to model the effectiveness of the strategies. We can leverage the Exploitability Sub Score (ESS) and the Impact Sub Score (ISS) of the CVSS base metric of the reported vulnerabilities that correspond to a particular kind of attack to get an estimate of the cost and the effectiveness of the attack strategies.

Hypergame-Theoretic Selection of Attack Strategies

We model each attack strategy (see Table 3.1)'s impact and cost based on the base metrics, *exploitability* and *impact*, of the Common Vulnerability Scoring System (CVSS) [11]. The exploitability indicates how complex the attack process is, such as the number of authentications to pass for exploiting a given vulnerability. The *impact* means the impact introduced

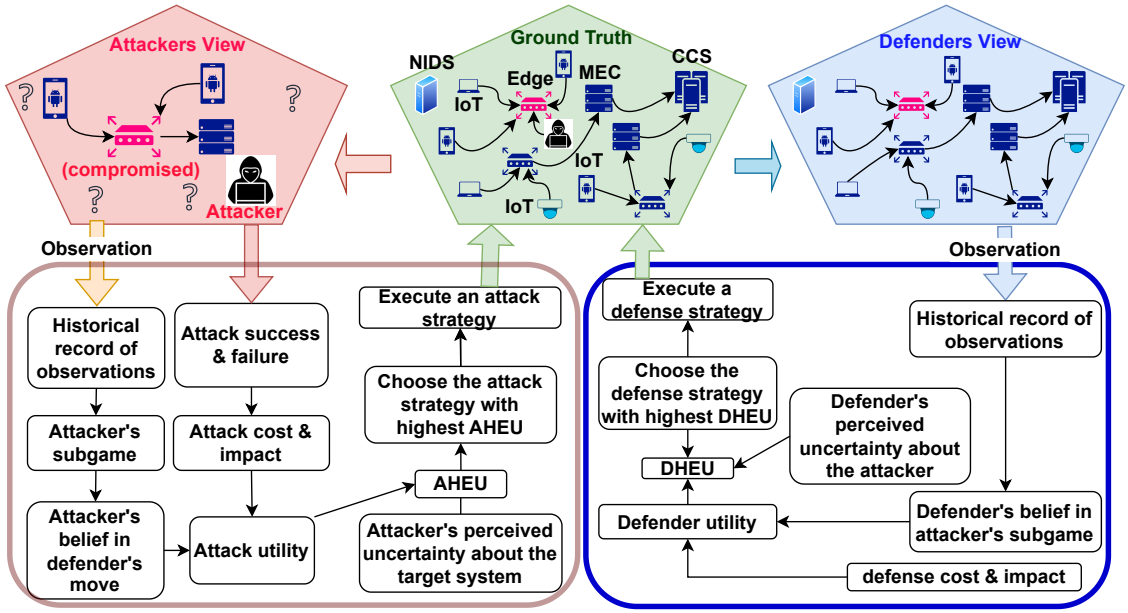


Figure 4.4: Overview of the proposed hypergame between an attacker and a defender where they view the same game differently and make decisions based on the Hypergame Expected Utility (HEU) [42].

by a successfully exploited vulnerability regarding security goals, including confidentiality, integrity, and availability (CIA). Let ac_{ik} be the attack cost when an attacker chooses strategy i on an asset k. Each asset k may have multiple vulnerabilities that can be exploited by attack strategy i, denoted by V_{ik}. To consider the attack complexity estimated by both the CVSS and a security expert, we formulate ac_{ik} by:

$$\mathrm{ac}_{ik} = \omega \times \left(\frac{\sum_{i \in V_{ik}} \mathrm{E}_i}{|V_{ik}|} \right) + (1 - \omega) \times \delta_{ik}, \tag{4.14}$$

where E_i is the exploitability score in $[0, 10]$ as an integer, δ_{ik} refers to an extra cost that may be introduced by attack complexity with attack strategy i with a range of $[0, 10]$ as an integer. A security expert will assign the value for δ_{ik} per attack for any asset. ω is a weight to consider the attack exploitability while $1 - \omega$ is to weigh the extra cost for attack complexity. When the Common Vulnerabilities and Exposures (CVEs) are unavailable, we can simply use an $\omega = 0$ and fully consider the security expert's opinion on the attack

complexity. Note that the attacker has no knowledge of the defense cost when a certain defense strategy is taken. Hence, we only consider the attack cost associated with asset k.

We define attack impact, ai_{ik}, by the average impact introduced when the attacker exploits all the vulnerabilities on asset k by taking strategy i. Thus, ai_{ik} is obtained by:

$$\text{ai}_{ik} = \omega \times \left(\frac{\sum_{i \in V_{ik}} \text{I}_k}{|V_{ik}|} \right) + (1 - \omega) \times \gamma_{ik}, \tag{4.15}$$

where ω is a weight to consider the average impact introduced by taking attack strategy i on asset k and I_k indicates the impact score on asset k obtainable in CVEs. We also consider an extra impact introduced when asset k (i.e., a node) is compromised. We consider asset k's importance with a weight

$$\gamma_k = \frac{\text{importance}_k \times \text{centrality}_k}{N}, \tag{4.16}$$

where N is the total number of nodes (i.e., assets) in the given network, importance_k is predefined based on the importance level assessed by a system analyst (e.g., 1 for IoT nodes, 2 for NED nodes, and 3 for MEC servers). Node k's centrality (centrality_k) is estimated based on the PageRank [6]. When CVEs are not obtainable, ω will be set to zero, and the attack impact will be solely estimated based on failing asset k whose importance is γ_{ik}.

Hypergame-Theoretic Selection of Defense Strategies

Similar to the attack cost, when a defender takes defense strategy j (e.g., see Table 3.2), it also incurs the defense cost, dc_j. Following the defense cost modeling in [42], we model dc_j as three levels, low, medium, and high [42] corresponding to 2.5, 5, and 10, respectively, as shown in Table 3.2. The defense impact, di_{jk}, by taking defense strategy j on asset k is

obtained by

$$\mathrm{di}_{jk} = \mathrm{di}_{j}^{\max} - \mathrm{ai}_{ik}, \tag{4.17}$$

where ai_{ik} is the impact introduced by taking attack strategy i on asset k and $\mathrm{di}_{j}^{\max}$ is the maximum impact defense strategy j can introduce (e.g., 10 in our scenario). Table 3.2 describes the conditions when a taken defense strategy j is successful and the defense impact upon the defense's success. The attack impact is influenced by each asset's vulnerability level, which can be lowered upon the success of the defense strategy.

Utilities of Attack and Defense Strategies

When an attacker takes strategy i on asset k and the defender takes strategy j, the attacker can obtain its utility, u_{ij}^{Ak}, by:

$$\begin{gathered} u_{ij}^{Ak} = G_{ij}^{Ak} - L_{ij}^{Ak}, \\ G_{ij}^{Ak} = \mathrm{ai}_{ik} + \mathrm{dc}_{j}, \quad L_{ij}^{Ak} = \mathrm{ac}_{ik} + \mathrm{di}_{jk}. \end{gathered} \tag{4.18}$$

Similarly, the defender's utility, u_{ji}^{Dk}, can be estimated by:

$$\begin{gathered} u_{ji}^{Dk} = G_{ji}^{Dk} - L_{ji}^{Dk}, \\ G_{ji}^{Dk} = \mathrm{di}_{jk} + \mathrm{ac}_{ik}, \quad L_{ji}^{Dk} = \mathrm{dc}_{j} + \mathrm{ai}_{ik}. \end{gathered} \tag{4.19}$$

Based on u_{ij}^{Ak} and u_{ji}^{Dk}, we estimate each player's HEU using Eqs. (4.8)–(4.10) to select its best strategy, respectively.

4.3 Reasoning Model Generation

4.3.1 Impact Directed Graph (IDG)

In this section, we propose how we infer the mission resilience impact on the mission system. To achieve this, we need to model the mission as an Impact Dependency Graph (IDG). We then use the IDG to propagate the belief using the Subjective Bayesian Network (SBN). For the IoT-device-based object detection mission system, the IDG is demonstrated in Figure 4.5.

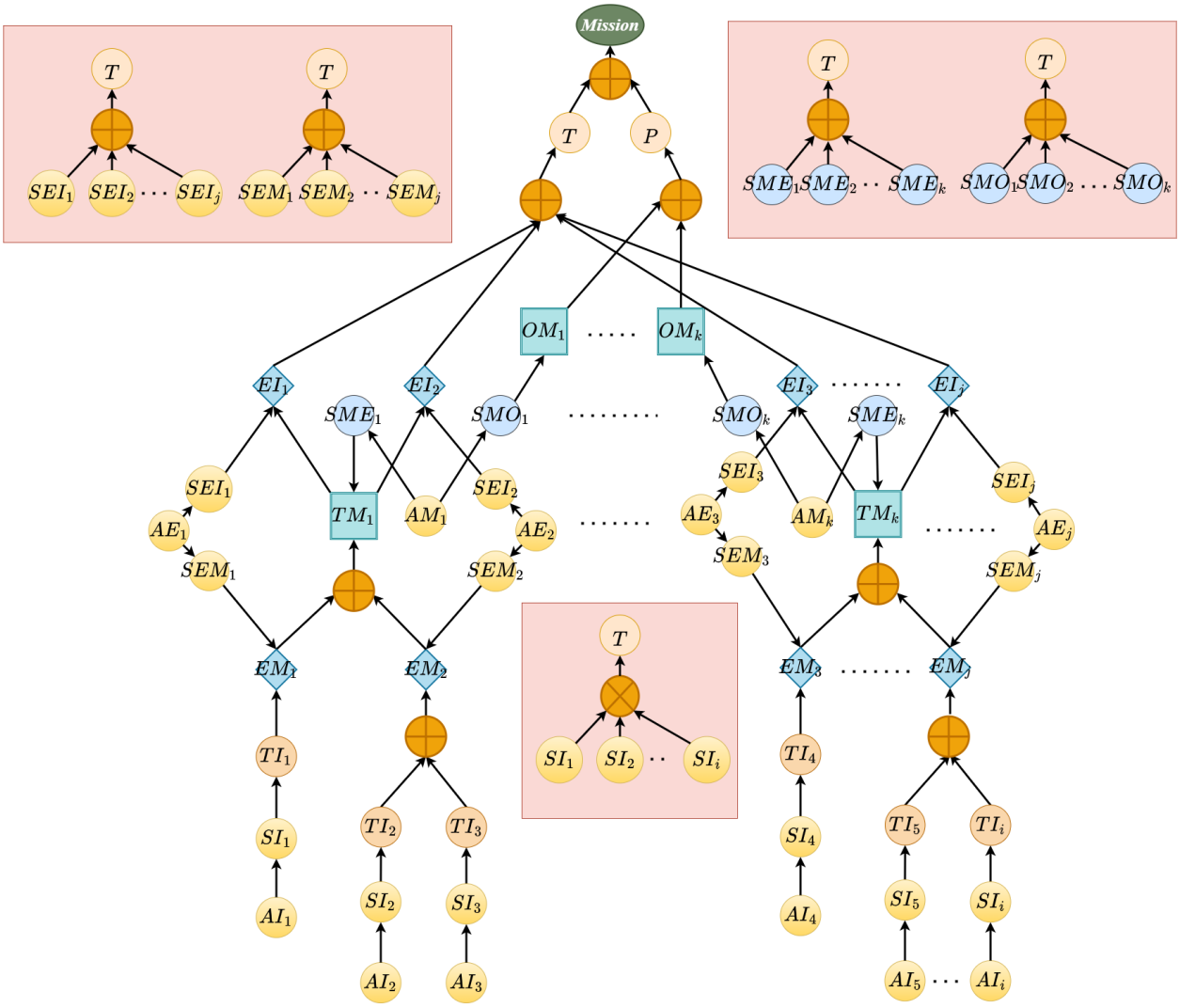


Figure 4.5: Impact Dependency Graph

Table 4.1: Variables in Bayesian Network and their defenitions

Var.	Type	Definition
AI_i	Asset	Is the IoT node i available?
AE_j	Asset	Is the Edge node j available?
AM_k	Asset	Is the MEC node k available?
SI_i	Service	Does the IoT node i have enough asset capacity to forward the test images to the Edge?
SEM_j	Service	Does the network edge node j have enough asset capacity to forward the test images to the nearest MEC server?
SEI_j	Service	Does the network edge node j have enough asset capacity to forward the results to the corresponding IoT?
SMO_k	Service	Does the MEC node k have enough asset capacity for performing object detection on the test images?
SME_k	Service	Does the MEC node k have enough asset capacity for forwarding the results back to the corresponding Edge node?
TI_i	ST	Does an IoT node i forward the test images to its nearest network edge node?
EM_j	ST	Does a network edge node j forward the test images to its nearest MEC node?
EI_j	ST	Does a network edge node j forward the object detection results back to the corresponding IoT device?
TM_k	ST	Does a MEC node k forward the object detection results back to the corresponding network edge?
OM_k	Task	Does a MEC node k perform the correct object detection on the test images?
$\mathcal{P}$	Task	Is the object detection performance better than the specified mission performance threshold $\rho\%$?
$\mathcal{T}$	Task	Does the mission system provide at least $\gamma\%$ timely service?
$\mathcal{M}$	Mission	Does a given mission succeed or fail?

Chapter 5

Experimental Setup

5.1 Simulation Environment

In our simulation of the IoT-based object detection mission system, we consider 20 IoT nodes connected to 10 network edge nodes. We consider 5 MEC nodes, each of which is connected to 2 edge nodes.

5.2 Metrics

5.2.1 Assessment Accuracy Metric

To quantify the accuracy of the MIA assessment tool, we use the Accuracy metric. Accuracy is the ratio of the number of correct predictions made to the total number of predictions. To get the prediction from the MIA framework, we provide the evidence (observations) to the trained BN reasoning model and infer the mission node to get the estimated confidence of mission success and failure. We then use this confidence value as the weight to make a stochastic decision about the estimated mission outcome. Once we have the predictions, we use the accuracy metric to quantify the performance of the assessment tool proposed.

Table 5.1: Key Design Parameters, Their Definition, and Default Values

Par.	Meaning	Value
P_{fp}	False positive rate of NIDS	0.01
P_{fn}	False negative rate of NIDS	0.05
P_{tp}	True positive rate of NIDS	0.95
P_{tn}	True negative rate of NIDS	0.99
ρ	Minimum prediction accuracy desired for mission success	0.75
γ	Minimum timeliness desired for mission success	0.9
w_1	Weight for CPU load	0.5
w_2	Weight for memory load	0.5

5.2.2 Mission System's Performance Metrics

To understand the performance of the mission, we propose using various metrics classified into Measures of Effectiveness (MoE) and Measures of Performance (MoP). We will use these metrics also to compare the performance of our proposed hypergame theoretic attack-defense interaction model with various other settings of the proposed attack-defense interaction model.

We will use the following metrics:

- *mean Intersection over Union ($\mathcal{M}_P$)*: The IoU metric, often known as the Jaccard index, is the most commonly used in object detection and to measure the similarity of the ground truth and the predicted bounding boxes. For our object detection mission, we compute the IoU score for each image as the IoU between the biggest bounding box (Most prominent object in the image) of the predicted image and the ground truth. mIoU is the average IoU score of all the test images in the mission. While mIoU is majorly used in the context of image segmentation, it can also be used to evaluate the bounding box similarity for object detection tasks.

- *Timely Service Availability ($\mathcal{A}_{TS}$)*: Timeliness is another important aspect of an effec-

tive mission. When the object detection service is not provided by the mission system to the users (IoT nodes) within a specified amount of time, we consider the timeliness of the service provider to be bad. Hence, to understand the effectiveness of the mission, we measure timeliness as the fraction of nodes that provide a service in a timestep.

- *Mission Success Ratio* ($\mathcal{R}_{MS}$): The mission success ratio is defined as the fraction of time steps in the mission where the outcome is a success. This ratio varies for different schemes and different hyperparameter settings of the simulation. Higher $\mathcal{R}_{MS}$ implies a higher likelihood of mission success.

5.3 Datasets

5.3.1 Microsoft Common Objects in Context (MS COCO) Dataset

We use the MS COCO dataset [24], a large-scale image dataset consisting of humans and everyday objects and their class annotations with their bounding box annotations of the ground truth and the predicted images to measure the mission performance (i.e., failure or success). The MS COCO dataset has been one of the most widely used benchmarks for object detection tasks for decades. It is particularly commonly used for an IoT-enabled object detection mission. To evaluate our proposed MIA's robustness, we inject noises into every observation in the training dataset based on a given error % to examine its effect on the inference accuracy.

5.3.2 Synthetic Datasets for Evaluating Our MIA Framework

We capture the observations for each of the variables specified in 4.1 at each timestep of the simulation across multiple simulation runs to generate our training and testing (ground truth) datasets for training and evaluating our proposed MIA framework. Each row of the dataset contains multiple binary (True/False) values corresponding to every node from the IDG. To simulate the noise present in the observability of a real system and the errors in the detections of asset and attack monitoring systems (NIDS), we add noise to every observation in the training dataset. However, in the test dataset, we assume perfect (error-free) observability of the mission.

5.4 Object Detection Model

For our mission, each MEC node the mission has a pre-trained YOLOv3 (You Only Look Once, Version 3) model [35]. YOLOv3 is a specially fine-tuned deep convolutional network that performs the task of object detection on everyday objects. The YOLOv3 model generates the object detection results which consist of the object classifications, the confidence of the predictions, along with the bounding box predictions. However, for the purpose of our mission, we only utilize the bounding box predictions of the biggest object in the image to compute the mIoU metric with the corresponding bounding box annotation from the ground truth dataset.

5.5 Comparing Schemes

We will compare the performance of the following schemes:

- **Hypergame-based Defender with Cost-Aware Random Attack (HD-CRA):** In this scheme, the attacker is assumed to have no knowledge about the defender and hence makes an attack cost (AC) aware random choice for the attack strategy. In the attack cost-aware strategy, the probability of an attack is inversely related to its attack cost. This deals with the drawback of a true random strategy where the attacker is not concerned about the overall cost of attacking the mission. However, it is unrealistic to assume that the attacker can spend an infinite amount of cost to perform the attacks. Hence, to model the limitation of the cost and resources in a real-world scenario, we introduced the attack cost (AC) aware random strategy-based player. In the HD-CA scheme, the defender uses the hypergame to make the decisions.

- **Hypergame-based Defender with Attack Path (HD-AP):** In this scheme, the attacker follows the same modified APT3 emulation plan defined in the Section 3.3. This means that if at least one attack is successful in any given stage of the APT3, the attacker can move to the next stage. Contrary to the hypergame-based attack-defense interaction model, where the attacker uses the Hypergame Expected Utility (HEU) to choose the best strategy in a given stage, the attackers pick a strategy from the APT stage (Infiltration, Expansion, Action) at random. In the HD-AP scheme, the defender uses a hypergame-based attack-defense interaction model to choose the strategy.

- **Hypergame-based Defender with Hypergame-based Attacker (HD-HA):** In this scheme, both the attacker and defender are using a hypergame-based attack-defense interaction model for decision-making.

- **Cost-Aware Random Defender with CRA (CRD-CRA)**: In this scheme, the defender is assumed to have little to no knowledge about the attacker's choices due to the surreptitious nature of the attacker. Hence to model this, we propose a Defense Cost (DC) aware random defender who chooses the defense strategies based on the

defense cost similar to the attacker in the HD-AC scheme. The attacker in the RD-CA scheme also uses the AC-aware random strategy defined in the HD-AC scheme.

- **CRD with AP (CRD-AP)**: In this scheme, the defender uses a defense-aware random defense strategy while the attacker uses the attack path-based strategy specified in the HD-AP scheme above.

- **CRD with HA (CRD-HA)**: In this scheme, the attacker is assumed to have some knowledge about the defender's actions while the defender is entirely unaware of the attacker's choices. Hence, the attacker uses a hypergame-based attack-defense interaction model while the defender uses a DC-aware random strategy.

Chapter 6

Numerical Results & Analyses

6.1 Inference accuracy of MIA under varying the size of a training dataset ($\mathcal{P}_{TD}$)

Fig. 6.1 shows how a different size of the training dataset ($\mathcal{P}_{TD}$) used affects the inference accuracy of the three metrics, ($\mathcal{M}_P$), ($\mathcal{A}_{TS}$), and ($\mathcal{R}_{MS}$). Note that $\mathcal{P}_{TD}$ is the ratio of the instances used for training the BN to the number of instances used to test it. Overall, we observe performance degradation as fewer training instances are used. Fig. 6.1(a) demonstrates that higher inference accuracy is observed under HD-CRA and CRD-CRA cases. Data manipulation attacks that degrade mission performance are more successful when performed on a compromised MEC node. However, CRA reduces the chances of a successful poisoning attack on a compromised MEC node due to the lack of intelligent strategy choice. As a result, fewer failure cases for $\mathcal{M}_P$ are observed in the data, making it easier for the model to learn to correctly infer $\mathcal{M}_P$.In Fig. 6.1(b), when the attack behavior is predicted based on AP (i.e., HD-AP and CRD-AP), the inference accuracy is the highest in $\mathcal{A}_{TS}$. The performance in $\mathcal{A}_{TS}$ is mainly influenced by DoS attacks which are less likely to be chosen by the AP scheme. This is due to the randomness and uncertainty in the choice of attack within a stage for an AP attacker, which makes it less likely to reach the third APT stage (Exfiltration/Action), where the DoS attack can be performed. Lastly, Fig. 6.1(c) shows

the highest inference accuracy under CRD-HA in the overall mission outcome ($\mathcal{R}_{MS}$). In our IDG (see Fig. 4.5), the Mission Outcome node ($\mathcal{M}$) is dependent on both the Timeliness node ($\mathcal{T}$) and the Mission Performance node ($\mathcal{P}$). Hence, the inference performance on $\mathcal{M}$ combines the performance of $\mathcal{T}$ and $\mathcal{P}$. The high accuracy under CRD-HA is because the attackers exhibit clear attack patterns based on a hypergame-theoretic decision-making process making it more intuitive for the BN to learn the patterns.

6.2 Inference accuracy of MIA under varying the noises of a training dataset ($\mathcal{N}_{TD}$)

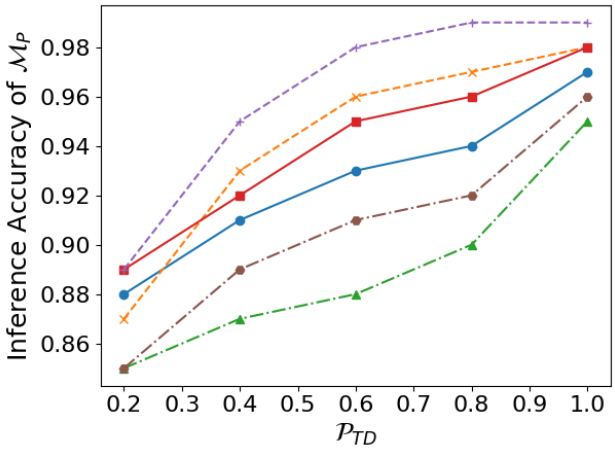
Fig. 6.2 shows how varying $\mathcal{N}_{TD}$ impacts the MIA's inference accuracy under diverse mission scenarios in terms of the three metrics. We observe that the proposed MIA has an inference accuracy of over 90% for $\mathcal{R}_{MS}$, 100% for $\mathcal{M}_P$, and around 90% for $\mathcal{A}_{TS}$ under perfect observability with no noises considered. However, the inference accuracy quickly degrades for all three metrics as the amount of noise increases. The inference performance for all three nodes does not vary significantly across different schemes under low-noise conditions. This implies that the IDG in Fig. 4.5 well represents the overall mission system and the learnability of the BN is robust to the attack-defense interaction model scheme considered.

Moreover, we can see that the HA and HD-based schemes perform better even under high noise conditions compared to the other schemes. This can be attributed to the fact that the Hypergame-based decision-making helps the players (attacker and defender) exhibit clearer, less uncertain patterns which seem to have a lesser impact due to noise in the training data. This more likely is due to the repetition of training instances caused due to the stability of the player decision making with the hypergame-based schemes.

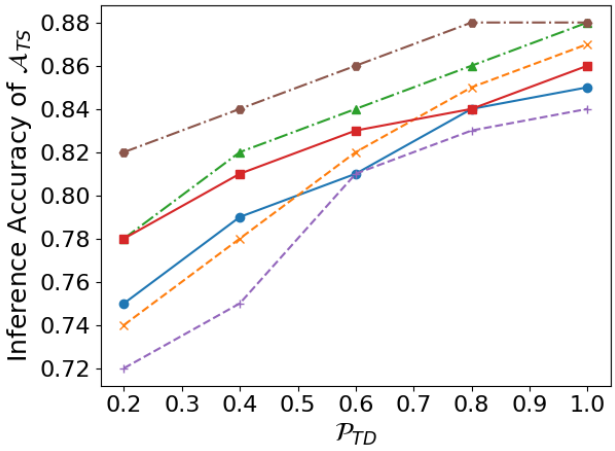
6.3 Inferred MoPE by MIA under varying the mean asset vulnerability:

In Figs. 6.1 and 6.2, we saw the quality and the robustness of MIA inference. Hence, we use it to estimate the performance of the mission for the various schemes we consider. Fig. 6.3 compares the inferred performance of the mission system under the six different scenarios. Fig. 6.3(a) shows that when the attacker uses HA or AP, $\mathcal{M}_P$ is significantly impacted compared to the case under CRA. This is because more intelligent attacks can introduce more adverse impacts in degrading mission performance. Fig. 6.3(b) shows that $\mathcal{A}_{TS}$ becomes significantly lower under HA due to the highly intelligent strategies taken by the attackers. On the other hand, the defense system may predict the attack behavior better when predicting with AP and has less impact under AP. Fig. 6.3(c) indicates the mission success ratio when both conditions of the correct and timely service provision are met, as described in Figs. 6.3(a) and 6.3(b). The overall trend of the mission outcome in Fig. 6.3(c) is well aligned with the timely service provision $\mathcal{A}_{TS}$ in Fig. 6.3(b) and mission performance $\mathcal{M}_P$ in Fig. 6.3(a).

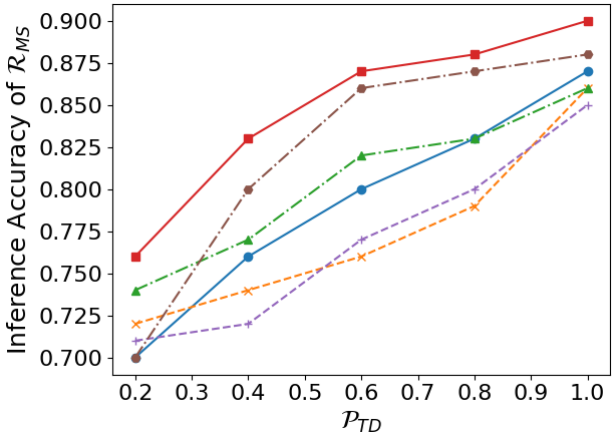
(a)



(a) $\mathcal{P}_{TD}$ vs. Inference accuracy for $\mathcal{M}_P$



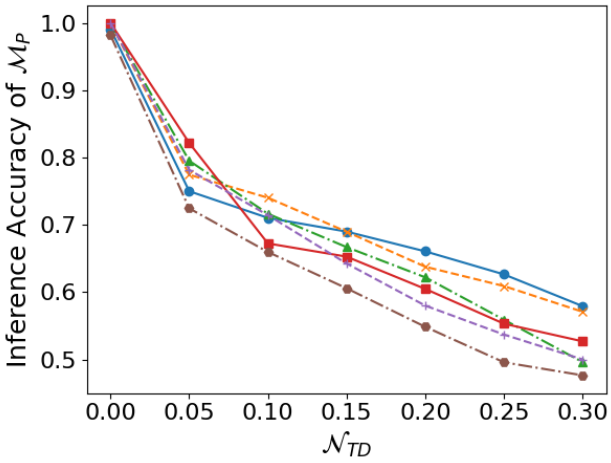
(b) $\mathcal{P}_{TD}$ vs. Inference accuracy for $\mathcal{A}_{TS}$



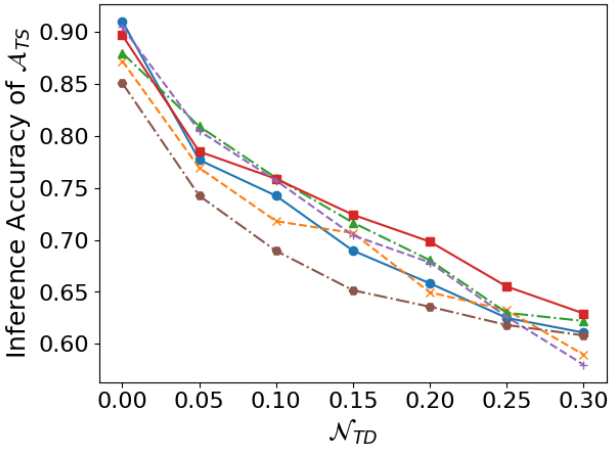
(c) $\mathcal{P}_{TD}$ vs. Inference accuracy for $\mathcal{R}_{MS}$

Figure 6.1: Effect of varying the relative size of the training dataset to the testing dataset($\mathcal{P}_{TD}$) on the inference accuracy of Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome ($\mathcal{R}_{MS}$) under the six different mission scenarios.

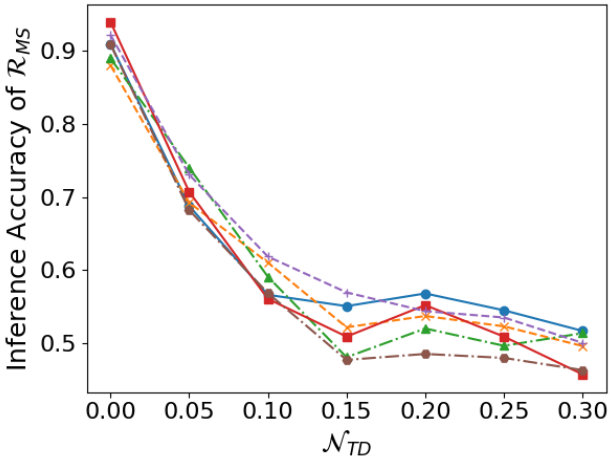
(a)



(a) Noise in training dataset vs. $\mathcal{M}_P$



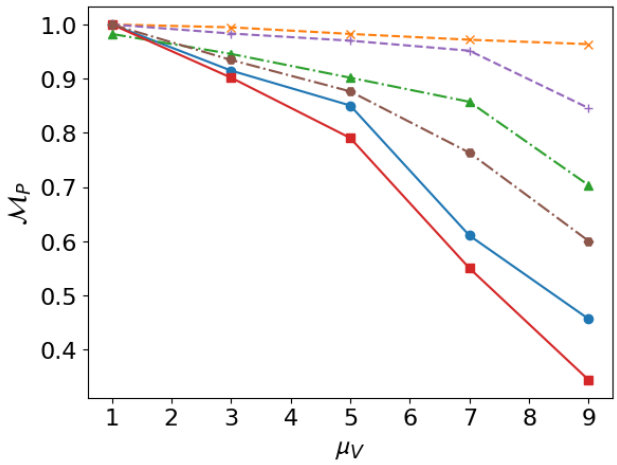
(b) Noise in training dataset vs. $\mathcal{A}_{TS}$



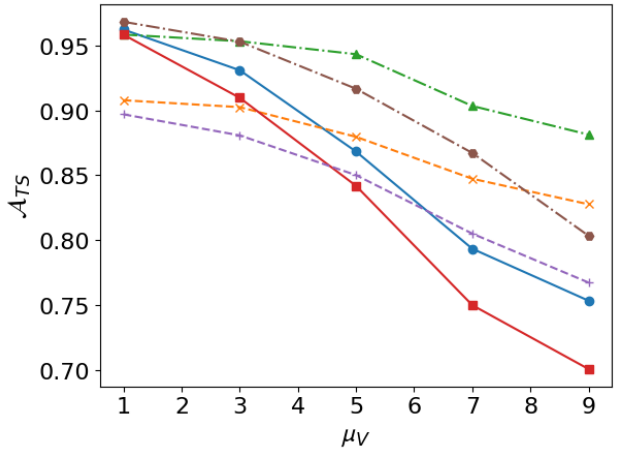
(c) Noise in training dataset vs. $\mathcal{R}_{MS}$

Figure 6.2: Effect of Noise in the training dataset $\mathcal{N}_{TD}$ for different schemes on the inference performance for Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome (M).

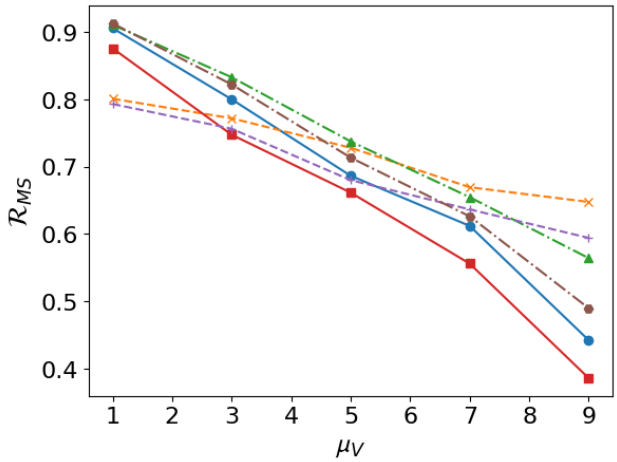
(a)



(a) $\mathcal{M}_P$ vs. μ_v



(b) $\mathcal{A}_{TS}$ vs. μ_v



(c) $\mathcal{R}_{MS}$ vs. μ_v

Figure 6.3: Effect of varying the mean node vulnerability (μ_V) on Mission Performance ($\mathcal{M}_P$), Timely Service Availability ($\mathcal{A}_{TS}$), and Mission Outcome ($\mathcal{R}_{MS}$) under the six different mission scenarios.

Chapter 7

Conclusions

7.1 Key Findings

- We proposed a comprehensive mission impact assessment (MIA) framework that considers interdependencies among key system components, including the assets, services, tasks, vulnerabilities, attacks, and defenses.

- Our framework incorporates realistic attack-defense interactions based on hypergame theory, accounting for uncertainty in strategy selection.

- This work advances existing MIA approaches by providing realistic modeling and simulation techniques and considering dynamic mission systems and complex attack-defense scenarios.

- The effectiveness and reliability of our MIA tool were demonstrated, showing high inference accuracy even with varying data quality and quantity.

- Specifically, hypergame-based attacks enable more accurate inference due to reduced randomness and uncertainty. Furthermore, clean training datasets result in significantly high inference accuracy (90%).

7.2 Limitations & Future Work

Based on the limitations of our current works, we plan to conduct the following future works:

- The current MIA design does not use a central knowledge Base (CKB) that can provide a mission system's holistic view, which will be added to the MIA in our future work.
- We will consider additional system metrics that can evaluate the mission system's dynamic quality, such as resilience or agility.
- Although BN can provide a diagnostic analysis considering the uncertainty derived from the randomness of events, it does not provide decision-making capabilities under different types of uncertainty, such as uncertainty caused by a lack of evidence or conflicting evidence.

To fill the gaps, our future work will consider Subjective Bayesian Networks (SBN) [19].

7.3 Publications

- **A. R. Thukkaraju**, H. J. Yoon, S. Matsumoto, J. F. Ferrari, D. Lee, M. K. Ahn, P. Costa and J.H. Cho, "Interdependent Mission Impact Assessment of an IoT System with Hypergame-Theoretic Attack-Defense Behavior Modeling," *The 31st International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2023)*, Oct. 2023.
- S. Matsumoto, J. F. Ferrari, H. J. Yoon, **A. R. Thukkaraju**, D. Lee, M. K. Ahn, J.H. Cho and P. Costa, "Software-Friendly Subjective Bayesian Networks: Reasoning within a Software-Centric Mission Impact Assessment Framework," *26th International Conference on Information Fusion (FUSION 2023)*, Jun. 2023.

- H. J. Yoon, **A. R. Thukkaraju**, S. Matsumoto, J. F. Ferrari, D. Lee, M. K. Ahn, P. Costa, and J.H. Cho, "Uncertainty-Aware Interdependent Mission Impact Assessment with Game-Theoretic Modeling of Attack-Defense Interactions," *Submitted to The IEEE Transactions on Emerging Topics in Computing*, Sep. 2023.

- H. J. Yoon, **A. R. Thukkaraju**, S. Matsumoto, J. F. Ferrari, D. Lee, M. K. Ahn, P. Costa, and J.H. Cho, "`iMIA`: Interdependent Mission Impact Assessment Using Subjective Bayesian Networks," *Submitted to The 37th IEEE/IFIP Network Operations and Management Symposium (NOMS 2024)*, Oct. 2023.

Bibliography

[1] Lockheed martin cyber kill chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[2] Christopher J Alberts, Sandra G Behrens, Richard D Pethia, and William R Wilson. Operationally critical threat, asset, and vulnerability evaluation (OCTAVE) framework, version 1.0. Technical report, Carnegie-Mellon University, Pittsburg, PA, Software Eng. Inst., 1999.

[3] Alexandre B. Barreto and Paulo C.G. Costa. Cyber-ARGUS - a mission assurance framework. *Journal of Network and Computer Applications*, 133:86–108, 2019. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2019.02.001.

[4] Emmanual N Barron. *Game theory: an introduction*. John Wiley & Sons, 2013.

[5] Peter G Bennett. Hypergames: developing a model of conflict. *Futures*, 12(6):489–507, 1980.

[6] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[7] Donald L. Buckshaw, Gregory S. Parnell, Willard L. Unkenholz, Donald L. Parks, James M. Wallner, and O. Sami. Saydjari. Mission oriented risk and design analysis of critical information systems. *Military Operations Research*, 2:19–38, 2005.

[8] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 Inter-*

national Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15, pages 63–72. Springer, 2014.

[9] Jin-Hee Cho and Noam Ben-Asher. Cyber defense in breadth: Modeling and analysis of integrated defense systems. *The Journal of Defense Modeling and Simulation*, 15(2): 147–160, 2018.

[10] Jin-Hee Cho, Mu Zhu, and Munindar Singh. Modeling and analysis of deception games based on hypergame theory. *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*, pages 49–74, 2019.

[11] CVSS. Common vulnerability scoring system, 2015. URL https://www.first.org/cvss/specification-document.

[12] David C Daniels, Linwood D Hudson, Kathryn B Laskey, Suzanne M Mahoney, Bryan S Ware, and Edward J Wright. Terrorism risk management. *Bayesian Networks: A Practical Guide to Applications*, pages 239–262, 2008.

[13] Alexandre de Barros Barreto, Paulo Costa, and Edgar Yano. A semantic approach to evaluate the impact of cyber actions to the physical domain. volume 966, 10 2012.

[14] Peter Hammerstein and Reinhard Selten. Game theory and evolutionary biology. *Handbook of game theory with economic applications*, 2:929–993, 1994.

[15] J. Han, J. Pei, and M. Kamber. *Data mining: Concepts and techniques*. Elsevier, 2011.

[16] James Thomas House and George Cybenko. Hypergame theory applied to cyber attack and defense. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IX*, volume 7666, pages 39–49. SPIE, 2010.

[17] Yadigar Imamverdiyev. A hypergame model for information security. *International Journal of Information Security Science*, 3(1):148–155, 2014.

[18] Gabriel Jakobson. Mission cyber security situation assessment using impact dependency graphs. In *IEEE 14th FUSION*, pages 1–8, 2011.

[19] Audun Jøsang and Lance Kaplan. Principles of subjective networks. In *19th IEEE Int'l Conf. on Information Fusion (FUSION)*, pages 1292–1299, 2016.

[20] Hyeob Kim, HyukJun Kwon, and Kyung Kyu Kim. Modified cyber kill chain model for multimedia service environments. *Multimedia Tools and Applications*, 78:3153–3170, 2019.

[21] Jana Komárková et al. CRUSOE: Data model for cyber situational awareness. In *Proceedings of the 13th Int'l Conf. on Availability, Reliability and Security*, pages 1–10, 2018.

[22] Christopher A Korban, Douglas P Miller, Adam Pennington, and Cody B Thomas. APT3 adversary emulation plan. *MITRE*, 2017.

[23] E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke. Model-based security metrics using adversary view security evaluation (advise). In *Quantitative Evaluation of Systems (QEST), 2011 Eighth Int'l Conf. on*, pages 191–200, 2011. doi: 10.1109/QEST.2011.34.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conf., Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[25] MITRE. Mitre att&ck, 2023. URL https://attack.mitre.org/.

[26] MITRE. Mitre d3fend, 2023. URL https://d3fend.mitre.org/.

[27] James D Morrow. *Game theory for political scientists*. Princeton University Press, 1994.

[28] Tamsin Moye, Reginald Sawilla, Rodney Sullivan, and Philippe Lagadec. Cyber defence situational awareness demonstration/request for information (RFI) from industry and government (co-14068-mncd2). *NCI Agency Acquisition*, 2015.

[29] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren. A systems engineering approach for crown jewels estimation and mission assurance decision making. In *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2011. doi: 10.1109/CICYBS.2011.5949403.

[30] Scott Musman and Aaron Temin. A cyber mission impact assessment tool. In *2015 IEEE Int'l Symp. Technologies for Homeland Security (HST)*, pages 1–7, 2015. doi: 10.1109/THS.2015.7225283.

[31] Scott Musman, Aaron Temin, Mike Tanner, Dick Fox, and Brian Pridemore. Evaluating the impact of cyber attacks on missions. Technical report, MITRE Corp, 2010.

[32] Scott Musman, Mike Tanner, Aaron Temin, Evan Elsaesser, and Lewis Loren. Computing the impact of cyber attacks on complex missions. In *2011 IEEE Int'l Systems Conf.*, pages 46–51, 2011.

[33] Soumyalatha Naveen and Manjunath R Kounte. Key technologies and challenges in iot edge computing. In *Third IEEE Int'l Conf. on IoT in Social, Mobile, Analytics and Cloud (I-SMAC)*, pages 61–65, 2019.

[34] Steven Noel et al. Analyzing mission impacts of cyber actions (AMICA). In *NATO IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact*, 2015.

[35] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[36] Yoav Shoham. Computer science and game theory. *Communications of the ACM*, 51 (8):74–79, 2008.

[37] Colin Tankard. Advanced persistent threats and how to monitor and deter them. *Network security*, 2011(8):16–19, 2011.

[38] The MITRE Corporation. Cyber command system (CyCS), 2015. URL http://www.mitre.org/research/technology-transfer/technology-licensing/cyber-command-system-cycs.

[39] R. Vane. *Hypergame theory for DTGT agents.* AAAI, 2000.

[40] R. Vane. Advances in hypergame theory. In *Proc. AAMAS Workshop on Game-Theoretic and Decision Theoretic Agents*, 2006.

[41] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

[42] Zelin Wan, Jin-Hee Cho, Mu Zhu, Ahmed H Anwar, Charles A Kamhoua, and Munindar P Singh. Foureye: Defensive deception against advanced persistent threats via hypergame theory. *IEEE Trans. Network and Service Management*, 19(1):112–129, 2021.

[43] Bud Whiteman. Network risk assessment tool (NRAT). *IAnewsletter*, 11:4–8, 2008.

[44] Tarun Yadav and Arvind Mallari Rao. Technical aspects of cyber kill chain. In *Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3*, pages 438–452. Springer, 2015.