iMIA: Assessing Mission Risk in Uncertain, Interdependent AI Systems

HAN JUN YOON, ASHRITH REDDY THUKKARAJU, and JIN-HEE CHO, Virginia Tech, USA
SHOU MATSUMOTO, JAIR FERRARI, and PAULO COSTA, George Mason University, USA
DONGHWAN LEE and MYUNG KIL AHN, Agency for Defense Development, Republic of Korea

Mission Impact Assessment (MIA) is critical for enhancing system effectiveness and ensuring mission success. This paper presents iMIA, an interdependent MIA framework that models relationships among mission components and enables probabilistic reasoning under uncertainty. Designed for AI-driven mission systems operating in dynamic, low-data, or poorly observable environments, iMIA addresses the limitations of traditional methods that often rely on overly confident assumptions about adversary behavior. While conventional hypergame theory (HGT) captures perceptual uncertainty from asymmetric or inaccurate views, it overlooks epistemic uncertainty arising from limited knowledge. To bridge this gap, we introduce a hybrid SL-based HGT model (SLHG), integrating Subjective Logic (SL) to represent epistemic uncertainty and HGT to account for misperceptions. This integration supports informed decision-making under both uncertain strategy beliefs and divergent environmental views. iMIA evaluates mission impact using multidimensional system quality metrics, security, trust, resilience, and agility, across diverse attacker-defender interactions. It identifies critical nodes influencing mission outcomes and quantifies performance gains from asset capacity reinforcement and asset vulnerability mitigation. Applied to a vehicle-assisted AI-based mission system, iMIA with SLHG improves performance by 16% in *ASR*, 20% in *MTBF*, 11% in *TSA*, and 14% in P_{ACC}. Designed for incremental development, iMIA supports continuous feedback and iterative refinement. Our results show that feedback-driven adjustments improve overall system performance by up to 18% in the accuracy performance.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Knowledge representation and reasoning**.

Additional Key Words and Phrases: Mission impact assessment, measures of performance, measures of effectiveness, uncertainty, and belief.

1 Introduction

Mission Impact Assessment (MIA) is well-established in mission-critical domains, particularly military systems [3, 21]. As AI-based mission systems increasingly demand security and resilience, they face growing challenges from dynamic environments, adversarial threats, and high uncertainty. Complex interdependencies among cyber attacks, defense strategies, asset capacities, and evolving contexts further complicate impact reasoning. AI also introduces novel vulnerabilities, including data poisoning and evasion.

While prior MIA efforts [1, 4, 20, 25–27, 30, 37] offer foundational insights, comprehensive frameworks for design, implementation, and validation remain limited. Notably, Cyber-ARGUS [5] advances mission assurance using Bayesian Networks but struggles with heterogeneous uncertainties like epistemic uncertainty and relies on static attack path databases. Its narrow cybersecurity focus also limits metric diversity.

Authors' Contact Information: Han Jun Yoon, godzmdi93@vt.edu; Ashrith Reddy Thukkaraju, ashrithreddyt@gmail.com; Jin-Hee Cho, jicho@vt.edu, Virginia Tech, Arlington, Virginia, USA; Shou Matsumoto, smatsum2@c5i.gmu.edu; Jair Ferrari, jfeldens@gmu.edu; Paulo Costa, pcosta@gmu.edu, George Mason University, Fairfax, USA; Donghwan Lee, dlee@add.re.kr; Myung Kil Ahn, happyahn@add.re.kr, Agency for Defense Development, Seoul, Republic of Korea.

ACM 2157-6912/2026/1-ART
https://doi.org/10.1145/3779065

Table 1. Advancements in Strategic MIA Modeling: From Our Prior iMIA [42] to the Current Work

Component	**Our Prior iMIA Work** [42]	**This Work**
Attack–Defense Dynamics	Random attacker selection only	SL-based hypergame reasoning against traditional hypergame, random, and path-based attackers
Software Development Process	Waterfall model [32]	Incremental development [33] with anytime evaluator feedback
Mission-Centric Metrics	Security, Trust	Security, Trust, Resilience, and Agility
Scope of MIA Evaluation	Inference accuracy under uncertainty; impact of attack severity on TSA and P_{ACC}	Expanded analysis encompassing noisy data and multiple attacker–defender strategies; includes metrics such as ASR, MTBF, TSA, and P_{ACC}; identifies critical node types contributing to success/failure; enables 3D visualization of pre/post asset reallocation effects

Existing approaches often assume full observability, overlook sparse evidence, and lack adaptability to historical data. These limitations hinder realistic analysis of complex, uncertain AI systems. This work addresses the gap by integrating Subjective Logic with Hypergame Theory, enabling reasoning under both epistemic uncertainty and perceptual asymmetries. Subjective Logic models belief under limited observations, while Hypergame Theory captures misperceptions in evolving, adversarial environments, supporting adaptive and realistic decision-making.

Our earlier study [42] laid the groundwork. Table 1 highlights key differences from the present work. This paper makes the following **key contributions**:

- We propose a Subjective Logic (SL)-based Hypergame (HG) model to capture strategic interactions under uncertainty, jointly modeling epistemic uncertainty and perceptual asymmetries, extending prior hypergame formulations.
- We design `iMIA` with an incremental model [33], supporting adaptive refinement and continuous feedback for evolving, AI-driven mission systems.
- We evaluate mission outcomes across random, path-based, traditional, and SL-based HG strategies, offering comparative insights into diverse attacker-defender dynamics.
- We adopt multidimensional metrics, including security, trust, resilience, and agility, and identify critical node types to guide asset capacity enhancement for mission resilience.
- We demonstrate `iMIA` in a traffic sign mapping use case that combines visual recognition and geospatial data to enhance situational awareness in uncertain environments.

Table 1 summarizes the key advancements of our approach over the prior iMIA framework [42], emphasizing methodological, architectural, and analytical improvements. To the best of our knowledge, existing MIA approaches (see Section 2.1) do not incorporate these contributions.

2 Related Work & Background

This section reviews the foundational concepts underpinning our framework, including prior MIA approaches, the Subjective Logic (SL) formalism for reasoning under uncertainty, and Hypergame Theory (HGT) for modeling perceptual asymmetries in adversarial settings.

2.1 Mission Impact Assessment

For mission-critical applications, several efforts have been made to develop MIA tools capable of evaluating the effects of cyberattacks on system assets and overall mission performance. Carnegie Mellon University pioneered this field with the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) framework [1],

Table 2. Comparison of Existing MIA Frameworks and the Proposed SL-based Hypergame iMIA

Aspect	Existing MIA Frameworks	Proposed SL-based Hypergame iMIA
Examples	OCTAVE [1], AMICA [30], CMIA [26], Cyber-ARGUS [5]	This work
Uncertainty Modeling	Limited or none; primarily deterministic assessments	Epistemic uncertainty captured via Subjective Logic (SL) binomial opinions
Perceptual Asymmetry	Not explicitly modeled	Captured using SL-based hypergame theory to model attacker and defender misperceptions
Attack–Defense Dynamics	Often static or predefined scenarios	Dynamic attacker-defender interactions across random, path-based, and SL-based hypergames
Adaptivity and Feedback	Typically static, waterfall-style development [32]	Incremental development with anytime evaluator feedback [33]
Metric Scope	Focus on basic mission impact (e.g., asset loss, risk)	Multidimensional: Security, Trust, Resilience, and Agility
Validation and Analysis Depth	Proof-of-concept or qualitative validation	Empirical evaluation with 3D visualization, node-type analysis, and attacker–defender comparisons

designed to identify organizational risks. Subsequently, MITRE introduced several MIA frameworks, including Analyzing Mission Impacts of Cyber Actions (AMICA) [30], Cyber Command System (CyCS) [37], Cyber Defense Situational Awareness (CDSA) [25], and Cyber Mission Impact Assessment (CMIA) [26, 27]. However, these earlier frameworks did not account for interdependencies between asset vulnerabilities and the effects of cyberattacks and defense strategies.

More recent research has sought to improve mission impact and situational awareness assessment, leading to the development of frameworks such as Cyber Situational Awareness (CRUSOE) [20], the Framework for Evaluating the Impact of a Cyber Attack on a Physical Mission (FEICAPM) [4], and Cyber-ARGUS [5]. However, CRUSOE [20] primarily focused on defining ontological relationships between key assessment components but lacked an empirical proof of concept to validate its approach. FEICAPM [4] laid the groundwork for concepts that were later refined and tested in Cyber-ARGUS [5]. Despite this, both Cyber-ARGUS and its experimental implementation served as proof-of-concept demonstrations, aimed at illustrating the feasibility and core characteristics of the framework rather than providing a thorough experimental validation.

However, most existing approaches have limitations in validating and analyzing the interdependent relationships among key components of MIA, including attack-defense dynamics, system asset capacity, vulnerabilities, and overall mission effectiveness. Table 2 contrasts representative existing MIA frameworks with our proposed SL-based hypergame iMIA, highlighting key innovations in uncertainty modeling, strategic interaction, adaptability, and analytical depth.

2.2 Subjective Logic

Subjective Logic (SL) is a probabilistic logic framework designed for reasoning under uncertainty, particularly when available information is incomplete, imprecise, or derived from subjective sources [16]. Unlike classical probability theory, SL explicitly models degrees of belief, disbelief, and uncertainty, making it well-suited for adversarial and dynamic decision-making contexts. A key construct in SL is the *binomial opinion*, which expresses belief about a binary proposition as a tuple $\omega = \{b, d, u, a\}$, representing belief mass b, disbelief mass d, uncertainty mass u, and a base rate a, with the constraint $b + d + u = 1$.

In this work, we leverage SL's binomial opinion to model both the availability of mission components and the inferred strategy of an adversary under limited observations. This allows our framework to quantify epistemic uncertainty arising from sparse evidence and to reason adaptively under such conditions. The binomial opinion has a direct mapping to the Beta distribution, allowing SL to integrate with probabilistic inference mechanisms. Section A of the Supplementary Material provides further mathematical details and mapping functions in SL.

2.3 Hypergame Theory

Hypergame Theory (HGT) [6] extends traditional game theory by modeling subjective perceptions, including misperceptions among players. It is particularly relevant in cyber attacker-defender scenarios, where each player's strategy depends on both their own view and their beliefs about the opponent's perception [13, 40]. HGT thus enables more realistic modeling of strategic interactions under perceptual asymmetries. Details on HGT and utility computations are provided in Section B of the Supplementary Material.

However, classical hypergames lack the ability to quantify uncertainty from limited observations, a common condition in cyber environments. They do not support probabilistic updates based on incomplete or conflicting evidence. To address this, we adopt SL-based hypergame theory, integrating SL [16] to reason under epistemic uncertainty. This hybrid approach supports more adaptive and resilient decision-making in information-scarce, adversarial settings.

3 Problem Statement

The proposed `iMIA` framework is designed to accurately infer mission outcomes (i.e., success or failure) for a given mission system by reasoning about the interdependencies between key nodes (i.e., assets, services, tasks) using Subjective Bayesian Networks (SBNs). Thus, `iMIA` seeks to:

$$\text{Maximize} \quad \sum_{c_i \in C} \Big[\mathcal{GS}(c_i) == \mathcal{ES}(c_i) \Big], \tag{1}$$

where $\mathcal{GS}(c_i)$ and $\mathcal{ES}(c_i)$ represent binary decisions, indicating either success or failure. These correspond to the system's ground truth mission outcomes and the `iMIA` inferred mission outcomes for a given mission case c_i. Each mission outcome is measured as a binary decision since the inference process is conducted through Subjective Bayesian Networks (SBNs), which operate on binomial subjective opinions (see Section 5.3 for details). The performance of the given mission system will also be assessed using multiple metrics that quantify the Measures of Performance and Effectivenes (MPE) within the mission system, as detailed in Section 6. The objective of `iMIA` in Eq. 1 is well aligned with the mission of the V2I application. Specifically, the V2I mission aims to enable accurate and timely traffic sign classification and mapping, evaluated by classification accuracy and service delivery timeliness, as well as metrics for security, trust, resilience, and agility.

The mission outcome is classified as either *success* or *failure*. A mission is considered successful within the given mission system if it satisfies the following two conditions:

- **Accuracy Condition (AC)**: To ensure sufficient performance for mission-critical decision-making, the final global model must attain a prediction accuracy denoted by $\mathcal{P}_{ACC}$ that meets or exceeds a predefined threshold ρ, i.e., $\mathcal{P}_{ACC} \geq \rho$.
- **Timeliness Condition (TC)**: To maintain mission readiness and operational continuity, mission assets are required to successfully deliver their designated services within the specified federated learning (FL) iteration window at least $\gamma\%$ of the time.

Based on the above criteria, we define the mission outcome $\mathcal{O}_M$ as:

$$\mathcal{O}_M = \begin{cases} 1 \text{ (success)}, & \text{if both AC and TC are satisfied,} \\ 0 \text{ (failure)}, & \text{otherwise.} \end{cases} \tag{2}$$

Prior to deriving the expected success score $\mathcal{ES}(c_i)$ for each component, we evaluate mission success by examining the interdependencies among nodes within the System Behavior Networks (SBNs), enabling a holistic assessment of the mission's viability under given conditions.

4 System Model

This section describes the system architecture and operational context of our proposed FL-based mission system, including its network and node structure, threat and defense assumptions, and the underlying federated learning process.

4.1 Network Model

We consider a vehicle-to-infrastructure (V2I) system in which a set of vehicles communicate with Roadside Units (RSUs). To enable the mission system to learn an accurate traffic sign classification model, we adopt Federated Learning (FL) [41], a decentralized approach that preserves data locality. Upon successful mission completion, the resulting models are integrated into a traffic sign prediction and mapping application developed as part of our system.

In this framework, RSUs serve as FL clients, training deep learning (DL) models, such as Convolutional Neural Networks (CNNs), using data received from nearby vehicles (e.g., images of traffic signs). After local training, each RSU transmits its model parameters to a Central Cloud Server (CCS), which performs global aggregation. Unless compromised, all RSUs are assumed to participate faithfully in the FL process. A detailed threat model describing adversarial behavior is provided in Section 4.3. The network architecture for the FL-enabled mission system is illustrated in Fig. 1.

4.2 Node Model

We define the following node types and infrastructure components that comprise the proposed mission system:

- *Vehicle*: Each vehicle voluntarily transmits traffic sign data to its associated Roadside Unit (RSU). The behavior of vehicles in forwarding data is modeled probabilistically, with a forwarding probability denoted as P_{vf}. Due to their mobility and potential to exit the mission area, vehicle contributions cannot be deterministically guaranteed. Instead, their behavior is approximated using the statistical mean of P_{vf}. Vehicles are treated as external information providers rather than mission assets, as they do not persist within the mission environment.
- *Roadside Unit (RSU)*: Multiple stationary RSUs are deployed throughout the mission area to collect traffic sign data from nearby vehicles. As FL clients, RSUs locally train deep learning models using their collected datasets and participate in the federated learning process by transmitting local model updates to the Central Cloud Server (CCS) and receiving aggregated global models. RSUs may be subject to compromise and can exhibit adversarial behaviors such as data poisoning, model poisoning, or denial-of-service (DoS) attacks.
- *Central Cloud Server (CCS)*: The CCS functions as the FL aggregator, synthesizing the global model by averaging the local model parameters received from all participating RSUs. During each FL iteration, the CCS distributes updated global models to RSUs based on the received contributions.
- *Mission Control Center (MCC)*: The MCC continuously monitors system activity during mission execution. It uses a Network-Based Intrusion Detection System (NIDS) to detect potential threats and evaluates the health of mission-critical assets (e.g., RSUs and CCS) by tracking system metrics such as CPU load, memory utilization, and vulnerability status. The MCC is assumed to be a fully trusted entity.

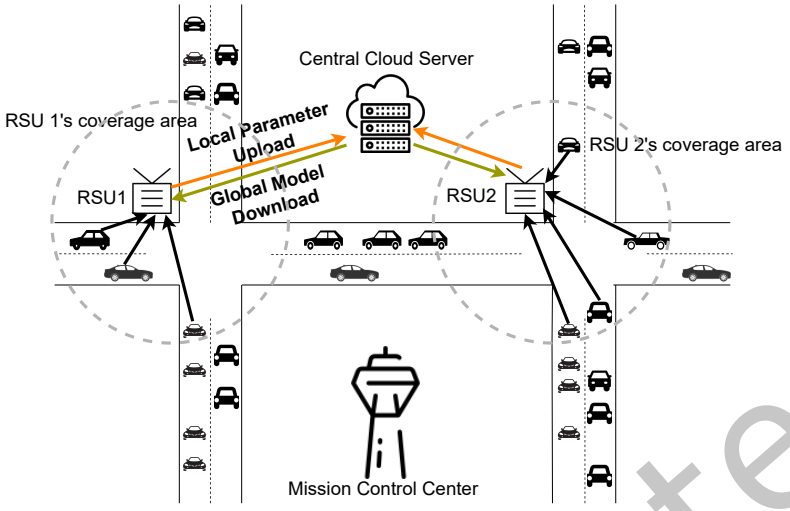


Fig. 1. The overall Vehicle-to-Infrastructure (V-to-I) architecture using Federated Learning (FL).

Since the MCC is assumed to be uncompromised, its potential failure or compromise is excluded from the mission performance analysis and omitted from Figs. 3 and 6. All node types, except vehicles, which serve as voluntary data providers, are considered mission assets within our system.

4.3 Threat Model

We adopt the five-stage Modified Cyber Kill Chain (MCKC) model to characterize adversarial behavior by Advanced Persistent Threat (APT) attackers [17]. The MCKC framework consists of the following sequential attack stages:

(1) **Reconnaissance (R)**: The attacker identifies potential targets by scanning for system vulnerabilities.
(2) **Weaponization and Delivery (WD)**: The attacker gains initial system access by delivering malicious payloads to vulnerable components.
(3) **Exploitation and Installation (EI)**: Access privileges are escalated within the system through exploitation, allowing the attacker to install malicious tools.
(4) **Command and Control (C2)**: The attacker establishes control over compromised nodes and introduces additional vulnerabilities to maintain persistence.
(5) **Action (A)**: The attacker executes final objectives, such as degrading or disrupting services provided by the mission system.

Throughout the five MCKC stages (R to A), APT attackers may launch a variety of cyberattacks aimed at compromising the mission system. Below, we summarize the representative attack strategies considered in our threat model:

- **Vulnerability Assessment** (AS_1): Attackers use scanning tools to identify system weaknesses and exploitable components.
- **Phishing** (AS_2): Victims are deceived into revealing sensitive information, such as access credentials, through social engineering techniques like phishing emails.
- **Exploit Public-Facing Application** (AS_3): Publicly accessible applications are targeted to inject malicious code by exploiting known or zero-day vulnerabilities.

- **Password Guessing** (AS_4): In the absence of valid credentials, attackers attempt brute-force or dictionary attacks to gain unauthorized access and escalate system privileges.
- **Process Injection** (AS_5): Exploits are used to inject malicious processes into legitimate software or operating system components, facilitating stealthy system control.
- **Domain Name System (DNS) Abuse** (AS_6): Compromised nodes are controlled via DNS channels, enabling covert command execution and detection evasion. Successful DNS-based attacks can cascade, increasing vulnerabilities in connected nodes.
- **Denial-of-Service (DoS)** (AS_7): Attackers flood mission assets with excessive requests or traffic to degrade or disable essential services.
- **Data Poisoning** (AS_8): Adversaries tamper with input data labels to corrupt AI-based mission services, such as traffic sign classification or detection.

In our framework, phishing is viewed as part of the broader threat landscape affecting RSUs. Attackers may exploit human operators managing RSUs or cloud services to gain unauthorized access or inject malicious data into the federated learning process. Thus, while RSUs are not direct targets, social engineering attacks on their operators can compromise overall security and performance.

All eight attack types described above are cataloged in MITRE's ATT&CK framework [38], providing a standardized reference for real-world adversarial techniques. An attack is deemed successful when it results in the exploitation of a node's existing vulnerabilities. Upon a successful attack, the affected node's vulnerability levels, specifically those related to encryption, software, or unknown weaknesses, are increased by $\epsilon\%$ relative to their prior states. The mechanisms for adjusting these vulnerability levels are detailed in Section 5.1.3.

To further strengthen the cybersecurity background and reduce hypothetical assumptions, we associated each representative attack type with real-world vulnerabilities documented in the National Vulnerability Database (NVD) and cataloged in MITRE ATT&CK. Specifically, each attack strategy is mapped to a corresponding Common Vulnerabilities and Exposures (CVE) entry (see Supplementary Material Table 2). This mapping demonstrates that the modeled threats (e.g., phishing, brute-force, process injection, denial-of-service, and data poisoning) align with well-documented exploits that can realistically compromise RSUs, CCS nodes, or federated learning processes in V2I systems.

4.4 Defense Model

We adopt seven standard defense mechanisms from MITRE's D3FEND [39] to mitigate cyber threats:

- **Dynamic Analysis** (DS_1): Executes a file within a synthetic sandbox environment to detect malicious code. Reduces unknown vulnerabilities across all nodes in the network.
- **Software Update** (DS_2): Replaces outdated software on system components (e.g., IoT devices, edge nodes, MCC, CCS) by applying patches and updates. Reduces software vulnerabilities across the network by addressing known Common Vulnerability Scoring System (CVSS) weaknesses.
- **Rekeying Cryptographic Keys** (DS_3): Nodes that use encryption periodically regenerate their cryptographic keys, thereby reducing encryption vulnerabilities and strengthening secure communications.
- **Local File Permission** (DS_4): Limits local file access to a restricted set of accounts, preventing unauthorized access to sensitive files. Reduces unknown vulnerabilities throughout the network.
- **Network-Based Intrusion Detection System (NIDS)** (DS_5): Detects potential intrusions based on network traffic analysis. Characterized by false positive (P_{fp}) and false negative (P_{fn}) rates. All detected intrusions, including false positives, are quarantened to minimize risk.
- **DNS Allowlisting** (DS_6): Permits access only to pre-approved domains and subdomains, restricting malicious communications. Reduces unknown vulnerabilities across all nodes.

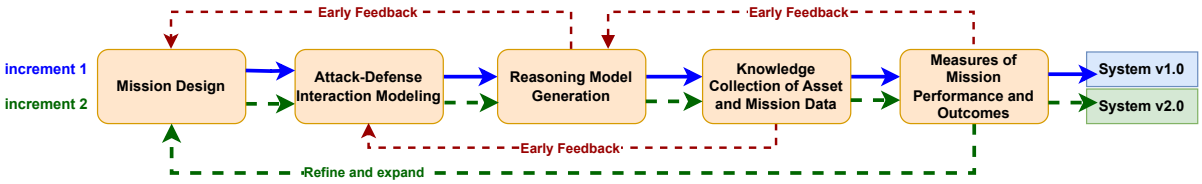


Fig. 2. The incremental design of the proposed iMIA framework enabling an anytime feedback loop for continuous refinement and adaptation.

- **Network Traffic Filtering** (DS_7): Restricts inbound and outbound traffic from unauthorized sources. Mitigates packet flooding attacks such as DoS and DDoS, reducing externally exploitable vulnerabilities.

We model the reduction of software vulnerabilities (sv_i), encryption vulnerabilities (ev_i), and unknown vulnerabilities (uv_i) by decreasing each by $\epsilon\%$ relative to their previous levels. Further details on vulnerability estimation are provided in Section 5.1.3. Section E of the Supplementary Material summarizes the attack and defense strategies aligned with each stage of the Modified Cyber Kill Chain (MCKC).

4.5 FL-based Mission System

We consider an FL-based system that aims to provide accurate classification of road traffic signs on the roads. In this work, we use FederatedAveraging (FedAvg) algorithm which combines each client that employs the stochastic gradient descent (SGD) with a server that averages all collected local models [23]. We assume the local dataset to be independent and identically distributed (IID). In every FL iteration, each RSU downloads the global convolutional neural network (CNN) model parameters from the CCS and trains the model with its local data. After each of the RSUs sends its local model parameters to the CCS. The CCS then aggregates all received local models, resulting in a new global model. The stochastic gradient descent (i.e., SDG) algorithm is used to update the local model parameters based on the gradient of the loss function, $\nabla F_k(\mathbf{w}_t)$ and learning rate, η accordingly. In t-th FL iteration, all RSUs first download the global model parameter $\mathbf{w}_t$ from the CCS, and each RSU client k calculates the gradient of its local loss function, $\nabla F_k(\mathbf{w}_t)$ on its local data. For each local epoch, each client k updates its local model for every batch $b \in B$,

$$\mathbf{w^k}_{t+1} = \mathbf{w}_t - \eta \nabla F_k(\mathbf{w}_t). \tag{3}$$

After receiving all updated local models from RSU clients, then the CCS performs aggregation as follows:

$$\mathbf{w}_{t+1} = \frac{1}{K} \sum_{k \in K} \mathbf{w^k}_{t+1}, \tag{4}$$

where $\mathbf{w^k}_{t+1}$ is the local model uploaded by RSU client k. CCS aggregates all collected local model parameters from a set of RSU clients and then aggregates them into a newly updated global model, $\mathbf{w}_{t+1}$ that is downloaded to clients for the next training round.

We suppose B is the local mini-batch size, and the number of updates on RSU client k in each round can be expressed as $u_k = E\frac{n_k}{B}$ where n_k is the number of data samples on a client k and E is the number of local epochs. We train our FL model until it reaches convergence when additional training will not improve the model. In other words, when accuracy on the test set doesn't improve for a certain number of times, that's when we say our model has converged for good accuracy both for training and unseen data. It is important to note that the focus of our work is to do mission impact assessment, not to find the optimal ground truth hyper-parameters for learning the optimal model possible.

5 Interdependent Mission Impact Assessment (iMIA) Framework

The iMIA framework is composed of five key phases: *Mission Design*, *Attack-Defense Interaction*, *Reasoning Model Generation*, *Knowledge Collection of Asset and Mission Data*, and *Measures of Mission Performance and Outcomes*. Each of these phases is detailed below. The generalized abstract pseudocode for the iMIA framework is provided in Algorithm 3 of the Supplementary Material.

The iMIA framework adopts an *incremental model*, also known as the *iterative waterfall model* [33]. Unlike the traditional waterfall model's strictly sequential approach, where each phase must be completed before the next and delivery occurs only at the end, the iterative variant applies the process repeatedly, delivering small, incremental updates with each cycle. For instance, a mission analyst may begin with a simplified Mission Description based on a narrow subdomain, then proceed through subsequent iMIA phases to produce an initial reasoning model. In the next iteration, feedback from the first cycle informs small refinements to previously developed components. An overview of this incremental approach in the iMIA framework is shown in Fig. 2.

5.1 Mission Design

The mission design phase includes three components: mission description, topology, and vulnerability discovery. Each is described below.

5.1.1 Mission Description. Fig. D.2 in the Supplementary Material (Section D) illustrates a generic concept model for representing integrated mission data via an ontology of tasks, services, and assets. Analysts use this model to define mission scenarios and node interactions. Following Sowa's conceptual graph [34], concepts are shown as nodes and semantic relations as directed arrows. The model aligns with the DoD Architectural Framework (DoDAF) [10], leveraging its terminology and structure for consistency and reuse in defense applications.

This model enables attacker-defender interaction modeling and stores mission-specific information in a Knowledge Base (KB). We use Vaticle TypeDB, which outperforms MySQL for reasoning over complex graph structures. As shown in Fig. 2 of the Supplementary Material, the *performer* includes attackers, defenders, and services. Assets provide services to support mission tasks (e.g., *service*, *send*, *user*, *manual*, or *script* tasks). Cyber attacks are modeled using MITRE ATT&CK [38], and corresponding defenses follow MITRE D3FEND [39], as detailed in Section 4.4.

The mission system includes assets such as RSUs and CCS, whose capacity can be compromised by cyber attacks. Asset vulnerabilities, particularly software-related, are assessed using the CVSS framework [9] (Section 5.1.3). Mission performance is then evaluated using the mSTRA framework, which measures security, trust, resilience, and agility (Section 5.5).

We use Business Process Modeling Notation (BPMN) [31] to collect mission-related information. With its simplicity and growing adoption [12], BPMN has become a standard for modeling business processes and services. Using BPMN 2.0, Fig. 3 describes mission systems through elements such as Events, Activities, Gateways, Connectors, Pools, and Artifacts. BPMN helps system designers standardize mission implementation. Once the BPMN and concept model are complete, we parse and store the BPMN file in the KB. We developed BPMN2TypeDB and TypeDB2BPMN scripts to convert between BPMN and Type-Theoretic and Polymorphic Query Language (TypeQL). Since BPMN lacks the vocabulary of the concept model, we apply *semantic mapping* to bridge the two. For example, an activity tag in BPMN mapped to a task under a lane (performer) is linked via the *isPerformedBy* relation. This enables persistent KB integration and supports automated translation between BPMN and TypeQL.

5.1.2 Topology Discovery. After parsing the concept model and BPMN file into the knowledge base (KB), our framework performs a series of *insert* and *match* queries on the database. The *insert* operation is used to instantiate entities (e.g., multiple RSUs instantiated as system assets), while the *match* operation retrieves the network topology to verify correct connectivity between cyber assets and their associated activities.

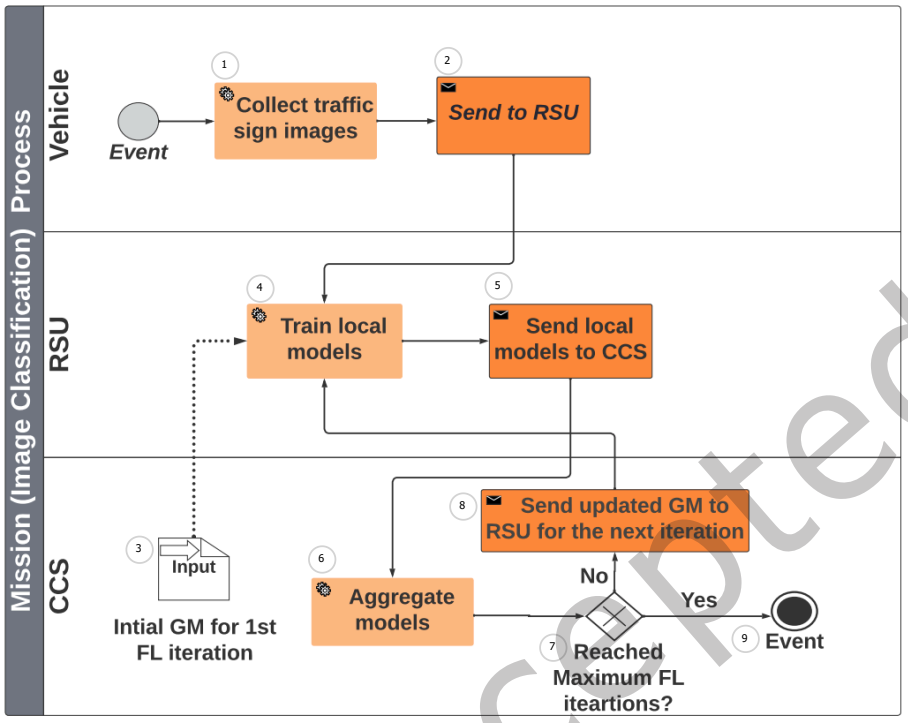


Fig. 3. BPMN of V-to-I mission.

In addition, the framework extracts relevant asset-specific information, including name, address, description, value, and hosted services, by querying the KB. This information may be encoded using BPMN-supported semantics or manually defined as attributes in TypeDB during the mission description phase following BPMN parsing.

Fig. 4 illustrates the static network topology of the V-to-I mission, where each asset is linked to its corresponding service tasks. In our mission framework, each service task is uniquely mapped to a single service, ensuring a strict one-to-one correspondence between services and their execution responsibilities. Note that Fig. 4 uses an undirected topology to highlight structural dependencies, while causal order and execution flow are captured in the Impact Dependency Graph (IDG) through directed edges that propagate asset capacities and task states to mission outcomes.

5.1.3 Vulnerability Discovery. This step identifies security vulnerabilities within the mission system. In `iMIA`, each node may be susceptible to various types of attacks. We consider three types of vulnerabilities, software, encryption, and unknown, that are associated with mission assets and store their values in the knowledge base (KB).

We define the vulnerabilities as follows:

- **Software Vulnerability** (sv_i): Refers to exploitable flaws in the software installed on node *i*. These vulnerabilities are obtained using the Common Vulnerability Scoring System (CVSS) [9] and sourced from the National Vulnerability Database (NVD), based on scans conducted by an open vulnerability assessment tool.

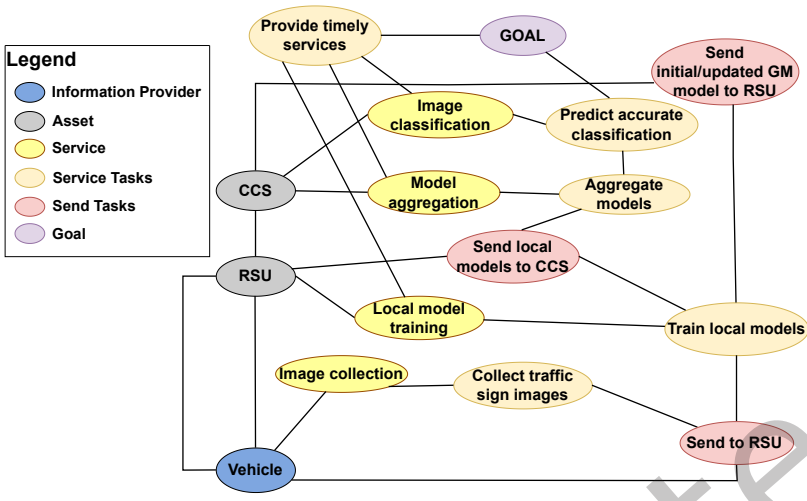


Fig. 4. Static network topology of V-to-I mission.

- **Encryption Vulnerability** (ev_i): Increases with prolonged reuse of the same cryptographic key. We estimate ev*i* using the following equation:

$$\hat{\text{ev}}_i = \text{ev}_i \times e^{-1/T_{\text{rekey}}}, \tag{5}$$

  where T_{rekey} denotes the elapsed time (measured as the number of attacker–defender interactions) since the last rekeying operation. Rekeying is performed periodically or whenever T_{rekey} exceeds a predefined threshold, after which it is reset to 1. If a key is reused for ζ consecutive periods without rekeying, we update $T_{\text{rekey}} = T_{\text{rekey}} + \zeta$. Eq. (5) models how prolonged key reuse increases encryption-related vulnerability. ev_i denotes the encryption vulnerability of each mission asset, while Eq. (6) normalizes and aggregates it with software and unknown vulnerabilities on a [0,10] scale for consistent impact assessment.
- **Unknown Vulnerability** (uv_i): Represents unquantified or latent vulnerabilities due to uncertainty. Although the ground truth value of uv_i is unknown, it is assigned a predefined value to analyze its impact. The value of uv_i may reflect different risk attitudes, e.g., risk-averse analysts may assign higher values, while risk-neutral or risk-seeking perspectives may justify lower values.

The average vulnerability of node *i* is computed as:

$$\text{vulnerability}_i = \sum_{v_j \in V_i} \frac{v_j}{|V_i|}, \tag{6}$$

where $V_i = \{\text{sv}_i, \text{ev}_i, \text{uv}_i\}$ is the set of vulnerabilities associated with node *i*. The probability that an attacker successfully exploits node *i* is modeled as:

$$P_i^v = \frac{\text{vulnerability}_i}{10}. \tag{7}$$

5.2 Attack-Defense Interaction: SL-based Hypergame

Traditional hypergame theory [7] extends conventional game theory by incorporating perceived uncertainty, allowing each player to form a subjective understanding of the game. Moreover, hypergame theory improves

resilience to asymmetric information, enabling each player to recognize that their opponent's perception of the game may differ. This reduces the risk of committing to suboptimal strategies based on incomplete or incorrect assumptions. It is instrumental in dynamic attack-defense interactions, such as Advanced Persistent Threat (APT) attacks.

Although traditional hypergame theory represents players' divergent views of the game, it does not provide a structured approach for explicitly quantifying uncertainty in the opponent's historical strategy selection. In contrast, SL-Based Hypergame Theory (SLHG) enhances traditional hypergame modeling by incorporating subjective logic, representing uncertainty through belief, disbelief, and uncertainty masses. This improvement is particularly significant when data regarding the opponent's historical strategy selection is scarce or limited, as SLHG provides a mathematical framework for reasoning under limited observation. In contrast, traditional hypergame theory relies on binary, discrete perceptions that do not dynamically adjust to evolving uncertainties.

In this work, we propose SLHG to model attack-defense interactions more effectively. Unlike traditional hypergame theory, which assumes that players form binary perceptions towards the opponent's strategy selection (i.e., either a certain opponent's strategy was picked or not picked), SL allows for uncertainty modeling, allowing each player to make probabilistic assessments of their opponent's strategy selection. This adaptive uncertainty-aware approach of observing the opponent's strategy history significantly enhances strategic responses in adversarial scenarios, particularly in cybersecurity and mission-critical systems.

In this work, we consider two types of uncertainties: 1) epistemic uncertainty due to a lack of observation or data; 2) perceptual uncertainty due to divergent views of the games. The proposed SLHG captures both epistemic uncertainty through SL's representation of limited evidence using binomial opinions and perceptual uncertainty through hypergame theory's modeling of asymmetric awareness and misperceptions between players. By integrating SL into hypergame theory, the bijective mapping in SLHG enables players to make optimal decisions even in low-data environments, where traditional hypergame theory struggles due to its reliance on fixed knowledge states. Therefore, SLHG provides a more robust and adaptive approach to modeling uncertain attack-defense interactions.

5.2.1 Opponent Strategy Observation Using SL. Traditional hypergame-theoretic models often assume that players possess static or probabilistic beliefs about their opponents' strategies without explicitly modeling epistemic uncertainty arising from limited observations. This can lead to suboptimal decision-making, especially in dynamic and adversarial environments. To address this limitation and effectively model an opponent's behavior in hypergame-theoretic interactions, we employ SL to estimate expected probabilities regarding the opponent's choice of strategies. Given the historical data of the opponent's strategy usage, `Algorithm 1` combines what is known (b) with what is not known (uncertainty), using a prior (a) guess to fill in the gaps. The input consists of a strategy history matrix (H), where each entry represents the number of times a specific strategy was observed to occur in a particular subgame. For each subgame/strategy pair, it computes belief from observed frequency, disbelief from how often the strategy was not chosen, and uncertainty from lack of evidence. It then computes the expected probability, representing the best estimate of the opponent playing each strategy, even under limited or uncertain information. The resulting output is a matrix of expected probabilities across all subgames and strategies, which serves as a key input for decision-making in the hypergame framework, supporting adaptive responses to dynamic and uncertain adversarial behavior.

5.2.2 Action Selection. A game round is a one-time interaction between the attacker and the defender. Via SL, the defender observes the attacker's move, perceives the subgame played, and estimates its Hypergame Expected Utility (HEU). It then selects the defense strategy that maximizes HEU (See Eq. (11)). The system updates vulnerabilities in the knowledge base (KB) based on the defense measures taken.

Similarly, attackers in the network observe the defender's move (i.e., the selected defense strategy) and choose the optimal attack strategy at the current MCKC stage (i.e., subgame) based on HEU, selecting the attack strategy

Algorithm 1: Modeling Opponent Strategy History via Subjective Logic in Hypergames

Input: $\mathbf{H} \in \mathbb{R}^{n \times m}$: Strategy history, where H_{ij} is the count for subgame i, strategy j
Input: $W \in \mathbb{R}$: Uncertainty weight (default $W = 2$)
Output: $\mathbf{P} \in \mathbb{R}^{n \times m}$: Expected strategy probabilities

$n \leftarrow$ number of subgames; $m \leftarrow$ number of strategies;
Initialize $\mathbf{b}, \mathbf{d}, \mathbf{u}, \mathbf{P} \in \mathbb{R}^{n \times m}$ as zero matrices;
Set uniform base rate: $a_j \leftarrow 1/m$ for all j;
Compute total evidence per subgame: $E_i \leftarrow \sum_{j=1}^{m} H_{ij}$;
for $i \leftarrow 1$ **to** n **do**
    $\mathbf{r} \leftarrow \mathbf{H}_i$;
    $\mathbf{s} \leftarrow E_i - \mathbf{r}$;
    for $j \leftarrow 1$ **to** m **do**
        sum $\leftarrow r_j + s_j$;
        if sum > 0 **then**
            total_evidence $\leftarrow$ sum $+ W$;
            $b_{ij} \leftarrow r_j$/total_evidence;
            $d_{ij} \leftarrow s_j$/total_evidence;
            $u_{ij} \leftarrow W$/total_evidence;
        else
            $b_{ij} \leftarrow 0$; $d_{ij} \leftarrow 0$; $u_{ij} \leftarrow 1$;
        $P_{ij} \leftarrow b_{ij} + a_j \cdot u_{ij}$;
return P

that yields the highest HEU. Upon a successful attack (i.e., exploitation of a target node's vulnerability), the attacker advances to the next stage of the MCKC.

In the SL-based hypergame modeling for the V2I mission, the information set available to each player is intentionally asymmetric and uncertain. The attacker observes the defender's most recent move and maintains a distribution over the defender's possible strategies using Subjective Logic, which encodes belief, disbelief, and uncertainty. This enables reasoning under incomplete observations while also accounting for perceptual uncertainty (i.e., misperceptions about the defender's actual understanding of the game). Conversely, the defender observes the attacker's move and uses Subjective Logic to estimate the attacker's likely strategy distribution, combining epistemic uncertainty due to limited or noisy evidence with perceptual uncertainty arising from divergent game views. The interaction is modeled as a sequential game rather than a simultaneous move game. Each round corresponds to a subgame in the Modified Cyber Kill Chain, where the attacker first selects an attack strategy against an asset based on observed defender behavior and expected utility. The defender then chooses a counter-strategy after observing the attack and updating the asset's vulnerabilities. This interleaving structure captures adaptive, round-by-round adversarial dynamics under uncertainty, rather than assuming simultaneous or fully observable decisions.

5.2.3 Strategies. The attacker and defender have defined action spaces: $AS = AS_1 - AS_8$ and $DS = DS_1 - DS_7$. The perceptual uncertainty estimates due to misperception for each player, g_t^A (attacker) and g_t^D (defender), are modeled by:

$$g_t^A = e^{-\lambda t}, \quad g_t^D = e^{-\mu t}, \tag{8}$$

where λ and μ control how perceptual uncertainty decreases over time. To incorporate perception errors in decision-making, we use hypergame theory to model uncertainty in action selection. The HEU quantifies these effects by estimating perceived utilities based on uncertainty and perception errors.

5.2.4 Utility Estimation. When an attacker employs strategy i on asset k while the defender adopts strategy j, the attacker's utility, u_{ij}^{Ak}, is computed as follows:

$$u_{ij}^{Ak} = G_{ij}^{Ak} - L_{ij}^{Ak}, \tag{9}$$
$$G_{ij}^{Ak} = \mathrm{ai}_{ik} + \mathrm{dc}_j, \quad L_{ij}^{Ak} = \mathrm{ac}_{ik} + \mathrm{di}_{jk}.$$

Similarly, the defender's utility, u_{ji}^{Dk}, is given by:

$$u_{ji}^{Dk} = G_{ji}^{Dk} - L_{ji}^{Dk}, \tag{10}$$
$$G_{ji}^{Dk} = \mathrm{di}_{jk} + \mathrm{ac}_{ik}, \quad L_{ji}^{Dk} = \mathrm{dc}_j + \mathrm{ai}_{ik}.$$

Using $u^{Ak}ij$ and $u^{Dk}ji$, we derive each player's HEU, applying Eqs. (11)–(13) to determine the optimal strategy. Here, *ai* and *ac* denote the attack impact and attack cost, while *di* and *dc* represent the defense impact and defense cost, respectively. Additional details are available in Section E of the Supplementary Material.

Hypergame Expected Utilities (HEUs). Based on the HEU, each player selects their action. The HEU reflects the player's level of uncertainty in perceiving the game and their opponent's strategies. It consists of the expected utility (*EU*) and the perceived uncertainty (g):

$$\mathrm{HEU}(rs_i, g) = (1 - g) \cdot EU(rs_i, C_{\sum}) + g \cdot EU(rs_i, CMS_w), \tag{11}$$

where $EU(rs_i, C_{\sum})$ represents the expected utility of the row player when the column player selects any of its available strategies. The Column-Mixed Strategy (CMS) refers to a strategy derived from the column player's perception of the row player's potential actions. $EU(rs_i, CMS_w)$ represents the row player's expected utility in the scenario where the column player selects a strategy that results in the minimum utility for the row player. The expected utility of the row player under both certain and uncertain conditions is expressed as:

$$\mathrm{EU}(rs_i, C_{\sum}) = \sum_{j=1}^{n} S_j \cdot u_{ij}, \tag{12}$$
$$\mathrm{EU}(rs_i, \mathrm{CMS}_w) = n \cdot S_w \cdot u_{iw}, \tag{13}$$

where S_j represents the row player's belief that the column player will select strategy j, while S_w denotes the row player's belief that the column player will opt for the worst-case strategy w, resulting in the lowest possible utility for the row player. Min-max normalization [11] is applied to compute these beliefs, transforming the Hypergame Expected Utility (HEU) values into a positive range, typically between 1 and 10. This normalization facilitates an assessment of the likelihood of various strategies, accounting for perceived uncertainties.

For additional details on belief estimation and the theoretical background of hypergame theory, including the derivation of S_j, refer to Section B (Hypergame Theory) of the Supplementary Material. Furthermore, Fig. 5 summarizes the SL-based hypergame framework, highlighting how each player perceives the game differently and makes strategic decisions using HEU derived from SL-based observations.

AHEU (Attacker's Hypergame Expected Utility) denotes the attacker's perceived payoff, combining expected utility with subjective logic–based uncertainty about the defender's move. The attacker selects the action with the highest AHEU.

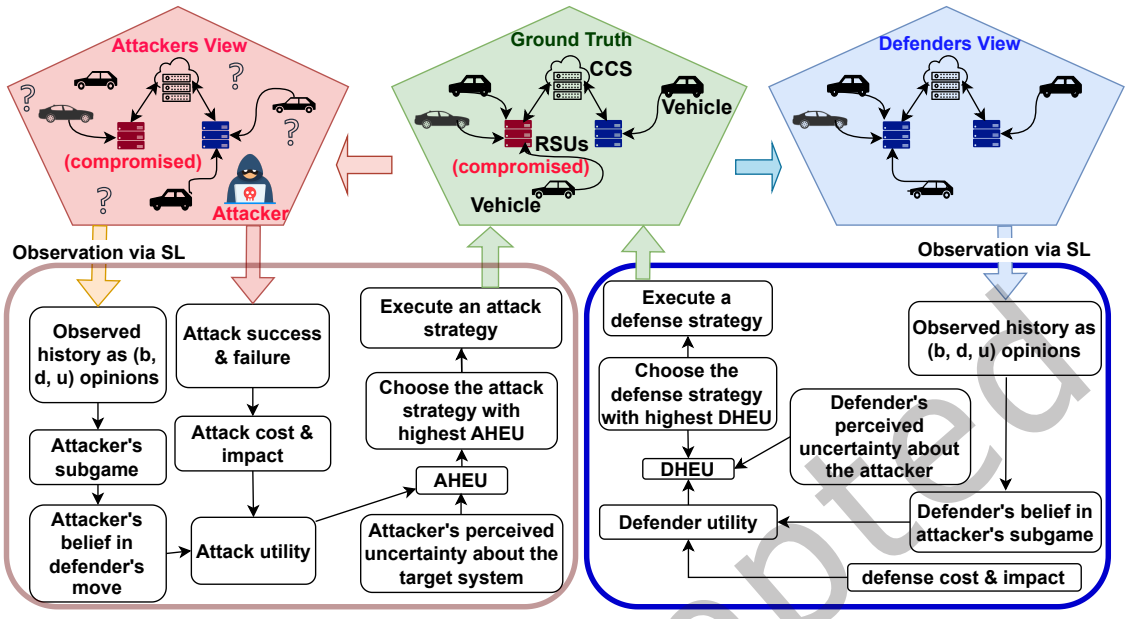


Fig. 5. SL-based hypergame framework modeling epistemic and perceptual uncertainties in attacker-defender interactions.

5.3 Reasoning Model Generation

This section outlines our approach for reasoning about mission outcomes using an *Impact Dependency Graph* (IDG) [14], which models dependencies within the mission system. For instance, the unavailability of an asset can cause service failure, potentially cascading into task and mission failure.

We leverage opinion parameters from SL to construct an SBN [18] using information encoded in the IDG. Unlike existing MIA frameworks such as Cyber-Argus [1, 4, 5, 20, 25–27, 30, 37], which primarily rely on traditional Bayesian Networks (BNs), we adopt SBNs to explicitly model and reason with second-order uncertainty, which BNs cannot capture.

5.3.1 Impact Dependency Graph (IDG). We describe our mission scenario based on an IDG in Fig. 6. In Fig. 6, we have vehicles (i.e., V_i's) that are information providers, and images collected by the vehicles are transmitted to RSUs. The mission system's assets include RSUs and CCS. We have four different service tasks (i.e., ST_i's) that are *collect images*, *train (initial) local model*, *aggregate models*, and *predict image classification*. Their associated services are *image collection*, *(initial) local model training*, *model aggregation*, and *image classification*. We have three different send tasks: *send directly to RSU* and *send local models to CCS*. In this work, "send tasks" also implicitly has its own service to transmit packets via some wireless protocol. Depending on the asset capacity, the assets provide services to perform tasks, which compound into attaining the given mission. Hence, each node's state can impact the connected nodes' states. This means propagating the impact of the information providers, assets, services, and tasks on the mission.

It is important to note that although the federated learning (FL) process involves multiple training iterations, a new Instance Dependency Graph (IDG) is only generated when a significant change occurs in the mission environment—such as the arrival of a new set of vehicles or an event that alters node values or the network

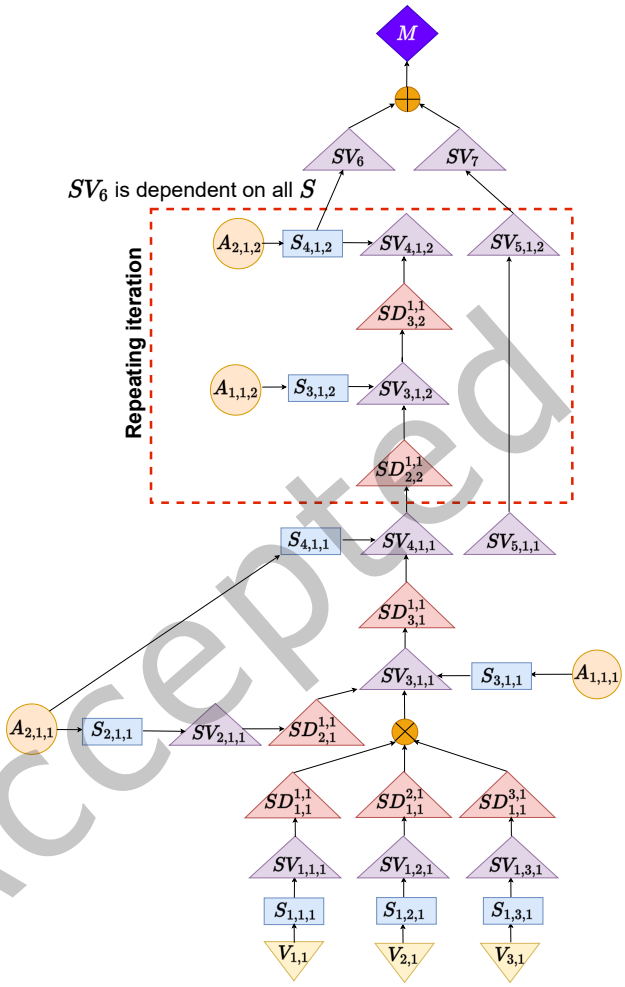


Fig. 6. Simplified Impact Dependency Graph (IDG) of the mission system. Node types are represented as follows: `send task` ($SD^{y,z}_{x,u}$), `vehicle` ($V_{q,u}$), and `service` / `service task` ($S_{x,q,u}$, $SV_{x,q,u}$), where x, y, z, q, and u denote type, sender, recipient, unique ID, and FL iteration, respectively. For instance, $A_{x,q,u}$ indicates asset type x, unique identifier q, and current federated learning (FL) iteration u. See Table 3 for definitions.

structure (e.g., a cyberattack). If no such event occurs, or if an event does not affect the structure or state of the IDG, a new graph is not instantiated.

In our mission system, a node is considered *compromised* under two conditions: (1) it becomes unavailable due to a Denial-of-Service (DoS) attack, or (2) it is transformed into a malicious node. In the first case, the affected node is marked inactive for one time step, after which it is re-integrated into the system to continue contributing to the mission. In the second case, the malicious node—typically an RSU—remains in the environment and continues to provide degraded service. It is reconfigured to its non-compromised state at the beginning of the next FL iteration.

5.3.2 Asset Capacity Calculation. We measure each asset's capacity based on CPU and memory load [29]. We use the exponential reliability function [19] to model each asset's capacity by:

$$\mathcal{R}_{AC} = \min[a_0 e^{-\lambda t}, 1], \tag{14}$$

where a_0 refers to the initial, distinct capacity of each asset (e.g., CCS's capacity is different from an RSU's capacity) and λ means a failure rate which is a function of memory and CPU load. Considering the short mission

Table 3. Definitions of Variables in IDG
(Note: ST: Service Task; SET: Send Task)

Var.	Type	Definition
V	Vehicle	Does a given vehicle appear in the mission service area?
A_1	Asset	Does a given RSU participate as an FL client to train?
A_2	Asset	Does a given CCS operate?
S_1	Service	Does a given vehicle have enough capacity to collect images before entering the service area?
S_2	Service	Does a given CCS have enough capacity to create an initial global model parameter properly?
S_3	Service	Does a given RSU have enough capacity to train its local model with the requested specifics (i.e., announced hyperparameters)?
S_4	Service	Does a given CCS have enough capacity to aggregate collected local models?
SV_1	ST	Does a given vehicle collect images before entering the service area?
SV_2	ST	Does a given CCS create an initial global model?
SV_3	ST	Does a given RSU train its local model based on requested specifics?
SV_4	ST	Does a given CCS produce an enhanced global model?
SV_5	ST	Does the given *updated* global model at least provide the required prediction accuracy, ρ%?
SV_6	ST	Does a given mission system provide timely services throughout the mission with minimum γ%?
SV_7	ST	Does the given *final* global model at least provide the required prediction accuracy, ρ%?
SD_1	SET	Does a given vehicle send its collected images?
SD_2	SET	Does a given CCS send the initial or updated global model to RSU clients?
SD_3	SET	Does an RSU client send its local model parameters to the CCS?
M	Mission	Does a given mission succeed or fail?

duration, we omit the effect of time (i.e., $t = 1$) and define λ by:

$$\lambda = w_1 \times L_{\text{CPU}} + w_2 \times L_{\text{memory}}, \tag{15}$$

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. L_{CPU} and L_{memory} denote CPU and memory loads, which are used to measure each asset's capacity. Each asset will deliver its service based on its capacity value. When attacks by inside (compromised nodes) or outside attackers (e.g., DoS attacks directly on asset nodes) are performed, they impact the CPU and memory load of the targeted asset by introducing 50% additional load compared to the current load.

Each system asset (e.g., RSUs, CCS) and vehicle has an AC value that propagates its impact. Based on asset attributes under attack, the IDG propagates the capability to perform required tasks, with impact inferred via SBNs.

5.3.3 SBN-based Reasoning Model. Reasoning under uncertainty is essential in mission scenarios where probabilistic information is often imprecise. Prior work [1, 4, 5, 20, 25–27, 30, 37] has used Bayesian Networks (BNs), which require precise conditional probabilities, a limitation when dealing with uncertain data. To address this, we extend BNs to handle second-order uncertainty using Subjective Bayesian Networks (SBNs).

We build on the UnBBayes framework [22], enhancing it to support SBN-based inference by incorporating mappings between binomial opinions and Beta distributions. Our approach integrates stochastic simulation (SS), junction tree (JT) transformation, and belief propagation (BP) to construct a representative BN, convert it into an undirected form, and propagate/update beliefs, respectively (Fig. 7). The input is an IDG-based graph structure

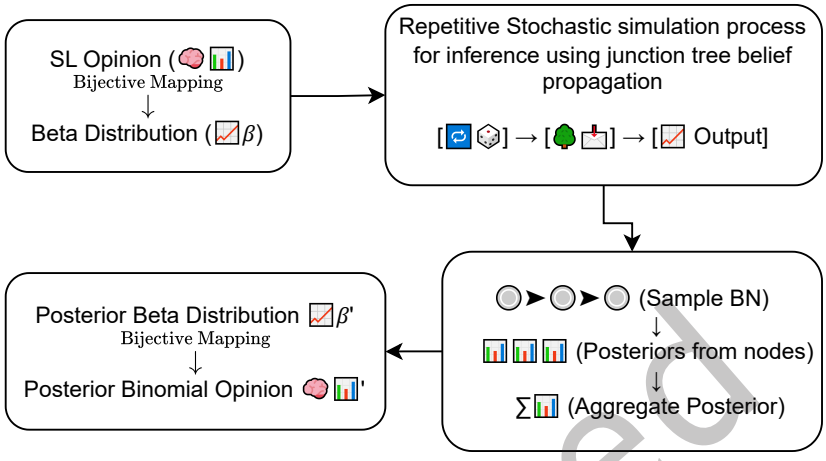


Fig. 7. SBN-based Reasoning Algorithm: This figure depicts the end-to-end pipeline of SL-based probabilistic reasoning, including (1) repetitive stochastic simulation for inference via junction tree belief propagation, (2) aggregation of node-level posteriors from a sample Bayesian network, and (3) bijective mapping from posterior Beta distributions to posterior binomial opinions.

(Fig. 6), where conditional probabilities are generalized to binomial opinions using Subjective Logic [16]. Details of JT and BP are provided in Section C (SBN-based Impact Inference) of the Supplementary Material.

5.4 Knowledge Collection of Mission Data

This phase collects current cyber assets and mission data within the operational environment. The `iMIA` framework gathers asset-level security status (e.g., compromised or not) via Network Intrusion Detection Systems (NIDS), and evaluates asset capacity for task execution.

System specifications, including node identifiers and connectivity, are obtained via the Simple Network Management Protocol (SNMP) [35], which supports asset discovery, network monitoring, and assessment of cyber attributes such as CPU and memory loads. All devices are assumed to support SNMP-based monitoring.

5.5 Measures of Mission Performance and Outcomes

Mission system performance is evaluated as follows:

(1) Construct an SBN-based reasoning model using the IDG (Section 5.3.3).
(2) Train the model with IDG-generated datasets, treating each node as a random variable.
(3) Test the model to infer mission outcomes based on timely and accurate service delivery.
(4) Identify high-impact nodes and consider enhancing their asset capacities to improve mission performance and resilience.

6 Experimental Setup

This section outlines the dataset, evaluation metrics, simulation environment, and comparison schemes used to assess the performance of different `iMIA` variants.

6.1 Datasets

We use the Belgian Traffic Sign Dataset (BTSD) for a multi-class, single-image classification task. BTSD contains over 7,000 traffic sign images across 60 classes, captured in urban areas of Belgium, making it well-suited to our mission environment. For this study, we select 13 of the 62 classes relevant to urban road conditions in the mission simulation. This subset is used to train and evaluate the traffic sign classification model.

We used an IID partition of the BTSD dataset across RSUs to ensure controlled evaluation, reproducibility, and avoidance of confounding effects when demonstrating the core iMIA framework. While this design supports tractable simulation, we acknowledge that real-world V2I data are typically non-IID and large-scale. Extending our framework to such settings is an important direction for future work to assess scalability and robustness.

6.2 Metrics

Building on [8], we incorporate four key system quality metrics, security, trust, resilience, and agility, each evaluated using specific subattributes:

- **Security**: Measured by the system's effectiveness in defending against cyber attacks.
  - *Attack Success Ratio* ($\mathcal{ASR}$): The proportion of successful attacks to total attack attempts.
- **Trust**: Evaluated by the system's ability to maintain reliable and error-free operation.
  - *Prediction Accuracy* ($\mathcal{P}_{\mathcal{ACC}}$): Classification accuracy of the global model in the system.
- **Resilience**: Reflects the system's capacity to adapt and continue delivering services under changing conditions.
  - *Mean Time Between Failures* ($\mathcal{MTBF}$): Ratio of service uptime to total session duration.
- **Agility**: Assessed by the system's responsiveness and timeliness in service delivery.
  - *Service Availability* ($\mathcal{SA}$): Ratio of timely services delivered by assets to total mission service requests.

Beyond MPE metrics, we also evaluate:

- *Inference Accuracy* ($\mathcal{I}_{\text{ACC}}$): Accuracy of mission outcome assessment.
- *Algorithmic Complexity* ($\mathcal{A}_{\text{COM}}$): Asymptotic complexity of the MIA algorithm.

6.3 Environmental Setting

To simulate the mission environment involving vehicles, RSUs, and CCS nodes, we use the Veins framework, which integrates SUMO (for urban traffic simulation) and OMNeT++ (for network simulation). This setup enables a realistic V-to-I mission simulation. In SUMO, we model four intersection roads with various traffic signs, including signals, right-of-way, direction indicators, yield, and stop signs. Vehicles operate in compliance with traffic rules under dynamic conditions.

We implement our system model using FL protocols to train a classification model across multiple iterations. This model supports real-time traffic sign prediction via an online tracking application, enhancing vehicle situational awareness.

6.4 Comparing Schemes

We evaluate the performance of four `iMIA` variants:

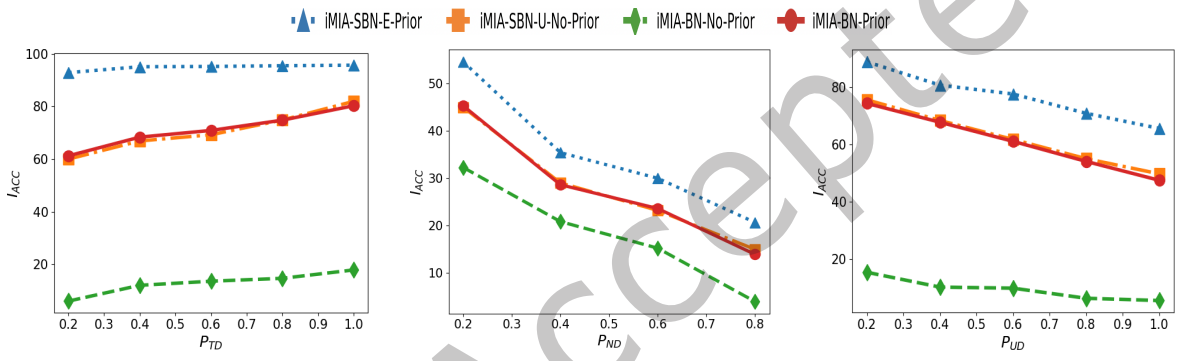
- `iMIA-SBN-E-Prior`: Uses SBN with expert-provided prior knowledge, aligned with ground truth (i.e., uncertainty is correctly interpreted).
- `iMIA-SBN-No-Prior`: Uses SBN without prior knowledge, assuming equal base rates from a uniform distribution.
- `iMIA-BN-Prior`: Uses BN with prior knowledge modeled as Beta$(1, 1)$, representing conventional MIA approaches [5, 15, 24, 36].
- `iMIA-BN-No-Prior`: Uses BN without prior knowledge, modeled as Beta$(0, 0)$.

For mission performance evaluation under varying attack-defense interactions, we assess `iMIA` where both attacker and defender adopt one of three strategies: random (R), traditional hypergame (HG), or subjective logic-based hypergame (SLHG). We also examine a path attacker that sequentially traverses subgames, choosing strategies randomly at each step.

We consider two categories of baselines. For prediction accuracy, we compare our Subjective Bayesian Network (SBN) reasoning model with the conventional Bayesian Network (BN) model, the standard in prior MIA frameworks. For attacker–defender dynamics, we evaluate mission outcomes under random, path-based, traditional hypergame, and SL-based hypergame strategies. Together, these baselines provide both methodological and strategic points of comparison for assessing the contributions of the proposed framework.

Direct comparison with frameworks like [1, 5, 28, 30] is infeasible due to differences in scope, metrics, data assumptions, and lack of public implementations. Instead, we benchmarked our Subjective Bayesian Network (SBN) model against traditional Bayesian Networks (BNs), the core inference method in most MIA approaches. This provides a fair, interpretable baseline and highlights the benefits of explicitly modeling epistemic uncertainty, emphasizing comparative insights between SBN and BN reasoning rather than reproducing heterogeneous frameworks.

7 Numerical Results & Analyses



(a) Varying P_{TD} (b) Varying P_{ND} (c) Varying P_{UD}

Fig. 8. Effect of different ratios of training (P_{TD}), noisy (P_{ND}), uncertain (P_{UD}) dataset on the inference accuracy ($\mathcal{J}_{ACC}$) of mission outcomes under various MIA schemes.

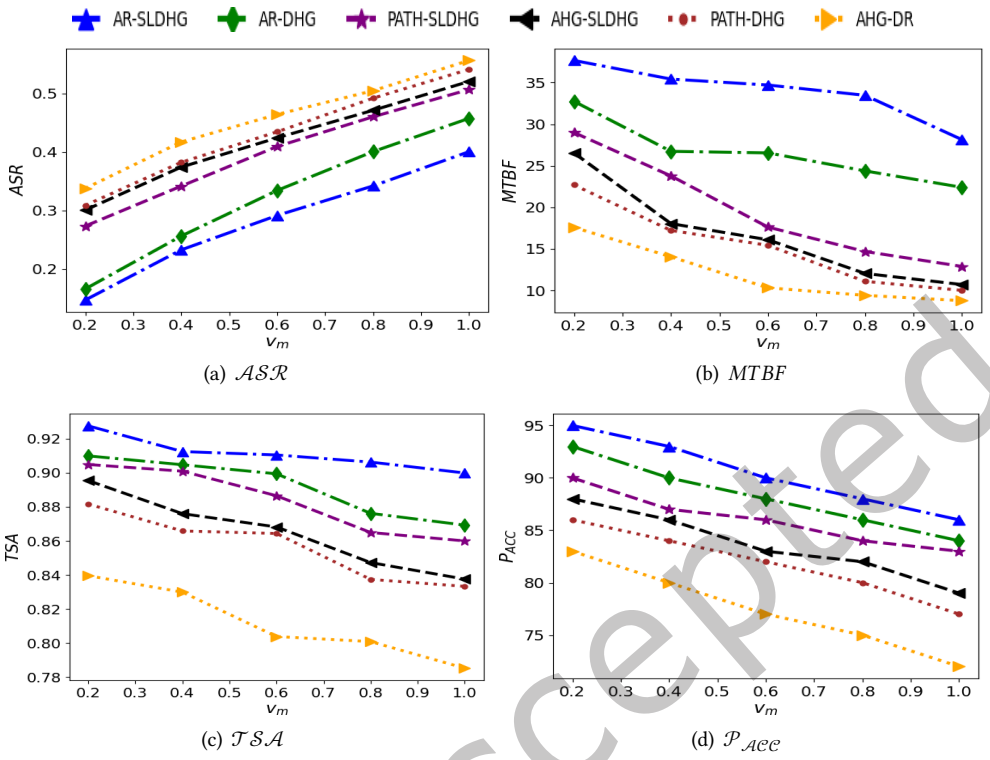
7.1 Comparative Accuracy Validation of MIA Schemes

Fig. 8 shows how varying proportions of training, noisy, and uncertain datasets affect mission outcome inference accuracy across MIA schemes. The training data ratio (P_{TD}) was adjusted by changing the number of training instances. Noise in the training data (P_{ND}) was introduced by flipping observed values (i.e., switching True and False). The uncertainty level in the testing data (P_{UD}) was varied by increasing the proportion of cases with balanced outcomes or limited evidence.

Experimental results indicate the performance ranking: `iMIA-SBN-E-Prior` $\gg$ `iMIA-BN-Prior` $\approx$ `iMIA-SBN-No-Prior` $\gg$ `iMIA-BN-No-Prior`. Accurate prior knowledge significantly improves inference accuracy for both SBN and BN. Notably, even without prior knowledge, SBN, by interpreting uncertainty via a uniform distribution that supports both success and failure, performs comparably to BN with prior. Thus, integrating expert opinions as prior knowledge in SBN-based MIA yields substantial gains in inference accuracy.

7.2 Algorithmic Complexity

Utilizing prior knowledge does not affect algorithmic complexity. Thus, we compare the complexity of `iMIA` using Subjective Bayesian Networks (SBN) and Bayesian Networks (BN). The complexity of `iMIA-SBN` is $\mathcal{O}(v^2+n\times v\times s^w)$,



(a) $\mathcal{ASR}$ (b) $MTBF$

(c) $\mathcal{TSA}$ (d) $\mathcal{P}_{\mathcal{ACC}}$

Fig. 9. Analysis of Different Attack-Defense Interactions: Effect of varying mean vulnerability (v_m) on the metrics including attack success ratio ($\mathcal{ASR}$), mean time between failure ($\mathcal{MTBF}$), timely service availability ($\mathcal{TSA}$), and prediction accuracy ($\mathcal{P}_{\mathcal{ACC}}$).

while `iMIA-BN` has a complexity of $\mathcal{O}(v^2 + v \times s^w)$. Here, v is the number of nodes, s the number of states (e.g., true or false), w the treewidth, and n the number of stochastic simulations. The v^2 term accounts for junction tree construction, and $v \times s^w$ reflects the cost of belief propagation.

To generate and aggregate multiple BN samples into a representative BN, we apply Hugin's Junction Tree (JT) belief propagation algorithm [2] n times. This yields sufficient statistics (mean and variance) used to compute α and β. Since binomial opinions follow a Beta distribution, the belief (b), disbelief (d), and uncertainty (u) masses are derived via bijective mapping [16].

In our case study, the number of nodes was $v = 156$ and the edge/interdependency count ranged from $w = 300$–400. Under these values, the quadratic components of the algorithm were tractable and did not pose a bottleneck. Reporting these values clarifies the tested computational scale and improves reproducibility. We acknowledge, however, that quadratic complexity may become unsatisfactory in large-scale or real-time deployments. To address scalability, strategies such as pruning, approximation, sampling-based inference, and parallelization can be applied, balancing accuracy and efficiency to ensure iMIA's reasoning remains practical for real-world systems.

7.3 Effect of Varying Attack-Defense Interactions

Fig. 9(a) shows that AR-SLDHG achieves the lowest $\mathcal{ASR}$ among all interactions. This is due to the defender's use of subjective-logic-based hypergame-theoretic (SLHG) strategies, while the attacker selects actions randomly. SLHG enables strategic decision-making under limited observability by incorporating beliefs and observations, effectively reducing attack success. In contrast, DHG schemes perform worse, as defenders use conventional hypergame-theoretic (HG) strategies that lack epistemic uncertainty modeling. Unlike HG, which assumes binary beliefs about the opponent's strategy, SLHG models belief, disbelief, and uncertainty, allowing more adaptive decisions when data is limited. While HG captures only perceptual uncertainty, SLHG explicitly accounts for both epistemic and perceptual uncertainties.

Fig. 9(b) shows that AR-SLDHG achieves the highest MTBF, due to its lower $\mathcal{ASR}$ and improved mission performance. Similarly, in Figs. 9(c) and 9(d), AR-SLDHG outperforms other schemes in both $\mathcal{TSA}$ and $\mathcal{P}_{\mathcal{ACC}}$, as the SLHG-based defender effectively interprets the opponent's behavior, while the random attacker ignores observations and beliefs.

Overall, SLDHG shows the most consistent positive impact across all four metrics, followed by DHG and DR. As v_m increases, system vulnerability rises, leading to more successful attacks and degraded performance in all metrics except $\mathcal{ASR}$.

7.4 Critical Node Types Affecting Mission Outcomes

Fig. 10(a) shows the top 10 most probable failure nodes in the IDG when a mission fails, identified through the SBN-based diagnostic reasoning in `iMIA`. For instance, if the mission fails, node SV_5 is highly likely to have also failed. Node types (i.e., random variables) are ranked in descending order of their failure probability given mission failure, as listed in Table 3. This analysis highlights key contributors to mission failure and provides actionable insights for decision-makers.

In SBNs, node types most likely to fail are analyzed by estimating the masses of disbelief (d), belief (b), and uncertainty (u), representing failure, success, and indeterminate outcomes, respectively. The uncertainty mass u is further weighted by a base rate a, which reflects prior knowledge. If no prior is available, a uniform distribution is assumed.

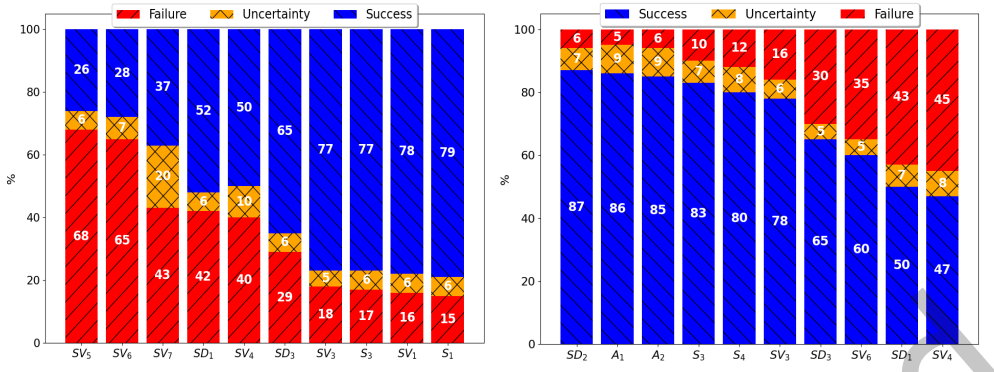
For example, SV_5 has $(\mathrm{b}, \mathrm{d}, \mathrm{u}) = (0.68, 0.26, 0.06)$; with a uniform base rate ($\mathrm{a} = 0.5$), the expected success and failure probabilities are $0.68 + 0.5 \times 0.06$ and $0.26 + 0.5 \times 0.06$, respectively. Incorporating prior knowledge allows more accurate assessments, as shown in Fig. 8.

As indicated in Fig. 10(a), SV_4–SV_7 and SD_1 are major failure sources, significantly impacting service quality. Strengthening these components can enhance the resilience and performance of the global model, especially under adversarial conditions (see Section 4.3).

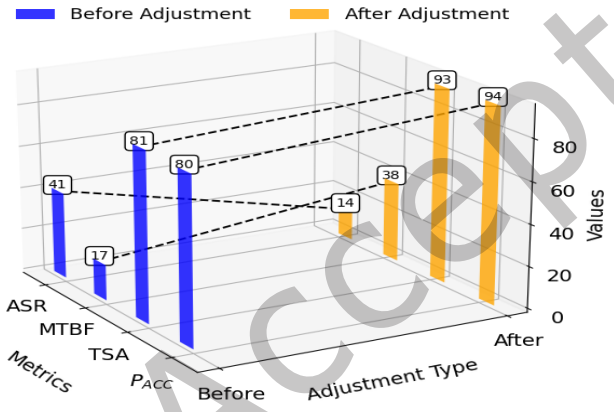
Fig. 10(b) presents the top 10 most probable success nodes when the mission succeeds, also based on `iMIA`'s SBN reasoning. For example, SD_2 is highly likely to succeed when the mission is successful. Interestingly, some nodes appear in both success and failure lists, indicating they may drive either outcome depending on the context. Recognizing these dual-role nodes enables targeted resource allocation to improve mission performance.

7.5 Adjusting the Asset Capacity and Vulnerability

Figs. 10(a) and 10(b) highlight nodes with a strong positive correlation to mission outcomes, i.e., nodes likely to fail or succeed when the mission fails or succeeds, respectively. To enhance mission effectiveness, we reinforced the asset capacity and mitigated the asset vulnerability of these nodes, resulting in measurable improvements across all four MPE metrics. As shown in Fig. 10(c), this adjustment led to a lower $\mathcal{ASR}$, indicating increased resilience to adversarial threats, and higher MTBF, TSA, and P_{ACC}. These results underscore the importance of strengthening nodes most correlated with mission outcomes to significantly improve overall system performance.



(a) Top failure-contributing node types (b) Top success-contributing node types



(c) 3D comparison of metrics pre/post-asset adjustment

Fig. 10. Analysis of key node types contributing to mission failure or success, and 3D comparison of performance metrics before and after asset capacity/vulnerability reinforcement.

8 Conclusions & Future Work

In this work, we introduced a Subjective Logic (SL)-based Hypergame (HG) model that captures strategic interactions under uncertainty by jointly modeling epistemic uncertainty and perceptual asymmetries, advancing prior hypergame approaches. We developed `iMIA`, an incremental mission impact assessment framework grounded in iterative refinement and continuous feedback, enabling adaptability in evolving AI-driven mission systems. Through comprehensive evaluation across random, path-based, traditional, and SL-based HG strategies, we provided comparative insights into attacker-defender dynamics. Our approach integrates multidimensional metrics, security, trust, resilience, and agility, and identifies critical node types to inform targeted asset adjustments for improved mission resilience. Finally, we demonstrated the applicability of `iMIA` in a traffic sign mapping scenario that fuses visual recognition with geospatial data to support situational awareness under uncertainty.

8.1 Summary of Key Findings

- Incorporating expert prior knowledge in SBN-based MIA significantly improves inference accuracy; even without prior knowledge, SBN matches or outperforms BN by modeling uncertainty through uniform base rates.
- While both SBN and BN share the same order of complexity for junction tree construction, SBN incurs additional overhead from stochastic simulations, enabling richer uncertainty modeling via belief, disbelief, and uncertainty masses.
- SLHG-based defenders outperform traditional and random strategies across all MPE metrics by accounting for both epistemic and perceptual uncertainty, resulting in lower attack success and greater mission resilience.
- SBN-based diagnosis reveals specific node types that strongly influence mission outcomes, providing actionable insights to guide targeted asset reinforcement.
- Reinforcing high-impact nodes identified by SBN analysis improves all MPE metrics, demonstrating the value of asset adjustment for mission robustness in adversarial environments.

8.2 Future Research Directions

As future work, we plan to apply the `iMIA` framework to other mission systems that involve complex cyber-physical interactions. All experiments are simulation-driven, which limits realism, may not capture operational noise, and reduces external validity. Future work will incorporate empirical data and real-world validation to enhance fidelity and applicability. Beyond the V2I use case, the proposed iMIA framework can be extended to other domains such as IoT, healthcare, smart grids, or industrial automation by reconstructing the Impact Dependency Graph (IDG) with new assets, services, and tasks, while reusing the general SL-based hypergame model and reasoning engine. However, reconstructing the IDG alone is not always sufficient. Domain-specific threat and defense models, realistic datasets for evaluation, and calibrated mission success criteria must also be incorporated to ensure meaningful analysis. Limitations of applying iMIA to other scenarios include the difficulty of quantifying hidden or unknown vulnerabilities and scalability challenges as the system size and interdependencies increase. These considerations highlight that iMIA provides a generalizable foundation for mission impact assessment, but its practical adaptation requires careful integration of domain-specific knowledge and empirical validation. While the current study focuses on cyber threats, `iMIA` is also applicable to scenarios involving cyber-physical threats. These include physical attacks, such as projectiles (e.g., missiles, bullets, or high-speed drones) targeting servers, antennas, or other critical infrastructure, as well as defensive measures like nets, wire traps, or entangling mechanisms to disable drones. Such interactions highlight the importance of modeling physical disruptions as both offensive and defensive tactics in mission systems like industrial automation networks, where cyber and physical components are tightly integrated.

Acknowledgments

This research was partially funded by the Agency for Defense Development, Republic of Korea, under grant number U22051XF.

References

[1] Christopher J Alberts, Sandra G Behrens, Richard D Pethia, and William R Wilson. 1999. *Operationally critical threat, asset, and vulnerability evaluation framework, Version 1.0.* Technical Report. Carnegie Mellon University, Pittsburg, PA, Software Engineering Institute.

[2] Stig K Andersen, Kristian G Olesen, Finn Verner Jensen, and Frank Jensen. 1989. HUGIN-A Shell for Building Bayesian Belief Universes for Expert Systems.. In *IJCAI*, Vol. 89. Morgan Kaufmann, Detroit, Michigan, USA, 1080–1085.

[3] Georgios Bakirtzis, Bryan T Carter, Cody H Fleming, and Carl R Elks. 2017. MISSION AWARE: evidence-based, mission-centric cybersecurity analysis.

[4] Alexandre Barreto, Paulo Cesar G Costa, and Edgar T Yano. [n. d.]. A semantic approach to evaluate the impact of cyber actions on the physical domain. In *7th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2012)*. Fairfax, VA.

[5] Alexandre B. Barreto and Paulo C.G. Costa. 2019. Cyber-ARGUS - A mission assurance framework. *Journal of Network and Computer Applications* 133 (2019), 86–108. doi:10.1016/j.jnca.2019.02.001

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

[7] P. G. Bennett and M. R. Dando. 1979. Complex strategic analysis: a hypergame study of the fall of France. *The Journal of the Operational Research Society* 30, 1 (1979), 23–32. doi:10.2307/3009663 Publisher: Palgrave Macmillan Journals.

[8] Jin-Hee Cho, Shouhuai Xu, Patrick M Hurley, Matthew Mackay, Trevor Benjamin, and Mark Beaumont. 2019. Stram: Measuring the trustworthiness of computer-based systems. *ACM Computing Surveys (CSUR)* 51, 6 (2019), 1–47.

[9] CVSS. 2015. *Common Vulnerability Scoring System.* https://www.first.org/cvss/specification-document

[10] CIO DoD. 2009. DoD Architecture Framework Version 2.0: Volume 1: Introduction, Overview and Concepts.

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

[12] Paul Harmon and Celia Wolf. 2016. The state of business process management. *BP Trends* (2016).

[13] James Thomas House and George Cybenko. 2010. 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*, Vol. 7666. SPIE, 39–49.

[14] Gabriel Jakobson. 2011. Mission cyber security situation assessment using impact dependency graphs. In *14th International Conference on Information Fusion*. IEEE, 1–8.

[15] Michal Javorník and Martin Husák. 2022. Mission-centric decision support in cybersecurity via Bayesian Privilege Attack Graph. *Engineering Reports* 4, 12 (2022), e12538.

[16] Audun Jøsang. 2016. *Subjective Logic: a formalism for reasoning under uncertainty.* Springer.

[17] Ankang Ju, Yuanbo Guo, and Tao Li. 2020. MCKC: a modified cyber kill chain model for cognitive APTs analysis within Enterprise multimedia network. *Multimedia Tools and Applications* 79, 39-40 (2020), 29923–29949.

[18] Lance Kaplan and Magdalena Ivanovska. 2016. Efficient Subjective Bayesian network belief propagation for trees. In *2016 19th International Conference on Information Fusion (FUSION)*. 1300–1307.

[19] Kailash Chander Kapur and Leonard R Lamberson. 1977. Reliability in engineering design. *New York* (1977).

[20] Jana Komárková, Martin Husák, Martin Laštovička, and Daniel Tovarňák. 2018. CRUSOE: data model for cyber situational awareness. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, New York, NY, 1–10.

[21] Alexander Kott, Jackson Ludwig, and Mona Lange. 2017. Assessing mission impact of cyberattacks: toward a model-driven paradigm. *IEEE Security & Privacy* 15, 5 (2017), 65–74.

[22] Shou Matsumoto, Rommel Novaes Carvalho, Marcelo Ladeira, Paulo CG Costa, Laecio Lima Santos, Danilo Silva, Michael Onishi, Emerson Machado, and Ke Cai. 2011. UnBBayes: a java framework for probabilistic models in AI. *Java in academia and research* (2011), 34.

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[24] Alexander Motzek and Ralf Möller. 2017. Context-and bias-free probabilistic mission impact assessment. *Computers & Security* 65 (2017), 166–186.

[25] Tamsin Moye, Reginald Sawilla, Rodney Sullivan, and Philippe Lagadec. 2015. Cyber Defence Situational Awareness Demonstration/Request for Information (RFI) from Industry and Government (CO-14068-MNCD2). *NCI Agency Acquisition* (2015).

[26] Scott Musman, Mike Tanner, Aaron Temin, Evan Elsaesser, and Lewis Loren. [n. d.]. Computing the impact of cyber attacks on complex missions. In *2011 IEEE International Systems Conference*. New York, NY, 46–51.

[27] Scott Musman and Aaron Temin. [n. d.]. A Cyber Mission Impact Assessment tool. In *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*. Waltham, MA, USA, 1–7. doi:10.1109/THS.2015.7225283

[28] Scott Musman, Aaron Temin, Mike Tanner, Dick Fox, and Brian Pridemore. 2010. *Evaluating the Impact of Cyber Attacks on Missions.* Technical Report. MITRE Corp.

[29] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 1–7.

[30] Steven Noel, Jackson Ludwig, Prem Jain, Dale Johnson, Roshan K Thomas, Jenny McFarland, Ben King, Seth Webster, and Brady Tello. 2015. Analyzing mission impacts of cyber actions (AMICA). In *NATO IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact.*

[31] Martin Owen and Jog Raj. 2003. BPMN and business process management. *Introduction to the new business process modeling standard* (2003), 1–27.

[32] Kai Petersen, Claes Wohlin, and Dejan Baca. 2009. The waterfall model in large-scale development. In *Product-Focused Software Process Improvement: 10th International Conference, PROFES 2009, Oulu, Finland, June 15-17, 2009. Proceedings 10*. Springer, 386–400.

[33] Nayan B Ruparelia. 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes* 35, 3 (2010), 8–13.
[34] John F Sowa. 1999. *Knowledge representation: logical, philosophical and computational foundations.* Brooks/Cole Publishing Co.
[35] William Stallings. 1998. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2.* Addison-Wesley Longman Publishing Co., Inc.
[36] Xiaoyan Sun, Anoop Singhal, and Peng Liu. [n. d.]. Who touched my mission: towards probabilistic mission impact assessment. In *Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense.* 21–26.
[37] The MITRE Corporation. 2015. *Cyber command system.* http://www.mitre.org/research/technology-transfer/technology-licensing/cyber-command-system-cycs
[38] The MITRE Corporation. 2022. ATT&CK Matrix for Enterprise. https://attack.mitre.org/.
[39] The MITRE Corporation. 2022. MITRE D3FEND: A knowledge graph of cybersecurity countermeasures 0.10.1-BETA-1. https://d3fend.mitre.org/.
[40] Zelin Wan, Jin-Hee Cho, Mu Zhu, Ahmed H Anwar, Charles A Kamhoua, and Munindar P Singh. 2021. Foureye: Defensive deception against advanced persistent threats via hypergame theory. *IEEE Trans. Network and Service Management* 19, 1 (2021), 112–129.
[41] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12, 19 pages. doi:10.1145/3298981
[42] Han Jun Yoon, Ashrith Reddy Thukkaraju, Shou Matsumoto, Jair Feldens Ferrari, Donghwan Lee, Myung Kil Ahn, Paulo Costa, and Jin-Hee Cho. [n. d.]. iMIA: interdependent Mission Impact Assessment Using Subjective Bayesian Networks. In *NOMS 2024 IEEE Network Operations and Management Symposium.* 1–7. doi:10.1109/NOMS59830.2024.10575273