2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Optimal Detection for Bayesian Attack Graphs under Uncertainty in Monitoring and Reimaging

Armita Kazeminajafabadi, Seyede Fatemeh Ghoreishi, and Mahdi Imani

Abstract— **Bayesian attack graphs (BAGs) are powerful models to capture the time-varying progression of attacks in complex interconnected networks. Network elements are modeled by graph nodes, and connections among components are represented through edges. The nodes take binary values, representing the compromised and uncompromised state of the network components. BAGs also offer a probabilistic representation of the likelihood of external and internal attacks progressing through exploit probabilities. The accuracy and timely detection of attacks are the main objectives in the security analysis of networks modeled by BAGs. This can ensure network safety by identifying network vulnerabilities and designing better defense strategies (e.g., reimaging devices, installing firewalls, changing connections, etc.). Two main challenges in achieving accurate detection in complex networks are 1) the partial monitoring of the network components due to the limited available resources and 2) the uncertainty in identifying and removing some compromises in the network due to the ever-evolving complexity of attacks. For a general class of BAGs, this paper presents an optimal minimum mean square error (MMSE) attack detection technique with arbitrary uncertainty in the monitoring and reimaging process. As with the Kalman filtering approach used for linear Gaussian state-space models, the derived solution exhibits the same optimality. A recursive matrix-form implementation of the proposed detection method is introduced, and its performance is examined through numerical experiments using a synthetic BAG.**

I. Introduction

Many practical systems consist of multiple devices connected through the Internet or communication systems. The growth in network systems has provided the opportunity to achieve superior operating performance while at the same time putting these systems at serious risk of cyber attacks. Examples include computer networks, autonomous cars, traffic lights, power systems, and many more [1]–[5]. If successful, external attacks can enter these systems and spread throughout the network components. Therefore, rapid and efficient detection of attacks is crucial to ensure the security of these complex networks.

Graph-based models have shown significant success in interconnected network security analysis. These models evaluate the network security and the risks associated with network components according to the external and internal threats [6]. Bayesian attack graph (BAG) is a powerful class of models within graph-based models that represent the propagation of the attack in complex interconnected networks [7]–[10]. BAGs are probabilistic graphical models consisting of nodes and edges. Nodes represent the compromised status of the network components, and edges indicate the probabilities of attack progression among connected nodes/components [11]–[13]. The attack progression depends on many factors, such as the type of machine/server, the security installed in the machine (e.g., firewalls), the number of connected devices, and connections to external sources. The use of graph-based models allows Markovian representation of the attack propagation in the network [14], [15]. Monitoring network compromises is often constrained by the availability of resources, and its accuracy is impacted by the complexity of attacks and monitoring systems. This often leads to partial observability of network compromises, which poses a significant challenge in detecting network compromises.

In recent years, several attack detection methods have been developed for BAGs. These include those built on maximum aposteriori (MAP) or maximum likelihood (ML) criteria, as well as those relying on heuristics [16]–[22] or approximations [23]–[25] to scale detection to larger domains. Most existing attack detection methods are built on the assumption that the compromises are directly observable upon routine monitoring. Furthermore, they assume that reimaging can fully clean machines' compromises [26]–[28].

The ever-evolving complexity of attacks and attackers poses uncertainty in monitoring and removing compromises in the network. Meanwhile, the limited available resources often allow partial monitoring or reimaging of the nodes. Therefore, it is critical to develop methods that can effectively detect system compromises while taking into account all sources of uncertainty. For a general form of BAG, this paper derives the optimal minimum mean square error (MMSE) attack detection method with arbitrary uncertainty in the monitoring and reimaging process. MMSE optimality, similar to the Kalman filter for the linear Gaussian state-space model, is achieved by considering the binary structure of the nodes in the graph [29], [30]. Furthermore, the proposed detection method maintains the component-wise maximum aposteriori optimality, unlike the commonly used maximum aposteriori solution for all nodes. The paper presents the exact matrix-form implementation of the optimal detector. Numerical experiments are carried out to examine the performance of the proposed method in terms of its accuracy and robustness to uncertainty and available resources.

A. Kazeminajafabadi and M. Imani are with the Department of Electrical and Computer Engineering, and S. F. Ghoreishi is with the Department of Civil and Environmental Engineering and Khoury College of Computer Sciences at Northeastern University. Emails: kazeminajafabadi.a@northeastern.edu, f.ghoreishi@northeastern.edu, m.imani@northeastern.edu.

979-8-3503-8265-5/$31.00 ©2024 AACC

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

II. Background - Bayesian Attack Graphs (BAGs)

Bayesian attack graphs are well-known models that represent how attacks progress through a network. Essentially, a BAG can be described as a structured tuple [31] denoted as: $\mathcal{G} = (\mathcal{N}, \mathcal{T}, \mathcal{E}, \mathcal{P})$. Here, $\mathcal{N}$ signifies the set of n network components, $\mathcal{T}$ represents the various types of nodes, $\mathcal{E}$ is the set of directed edges connecting these nodes, and $\mathcal{P}$ is the corresponding set of exploit probabilities. The nodes are represented as binary random variables, taking 0 or 1 corresponding to the uncompromised and compromised status of nodes, respectively. The nature of a network component significantly influences its vulnerability to compromise. In this context, we distinguish between two types of nodes, denoted as $\mathcal{T}_i$, which can take values from the set $\{\text{AND}, \text{OR}\}$. Specifically, AND nodes (e.g., administrative servers) are susceptible to compromise only when all of their incoming neighbors are compromised. In contrast, OR nodes (e.g., SQL server) can be compromised even if just one of their incoming neighbors is compromised. The direction of edges in the network graph is indicative of the potential attack path. Regarding the relationships among the nodes, the ith node is considered an in-neighbor of the jth node if the edge $(i, j) \in \mathcal{E}$ exists. Consequently, the in-neighbor set of any given node j can be explicitly denoted as $D_j = \{i \in \mathcal{N} \mid (i, j) \in \mathcal{E}\}$. The presence of an edge $(i, j) \in \mathcal{E}$ indicates the potential for the jth node to be compromised through node i. The set $\mathcal{P}$ includes the exploit probabilities associated with these edges, where $\rho_{ij} \in \mathcal{P}$ means the probability that the jth node is compromised through the ith node when node i is already compromised. Additionally, we identify a subset of nodes within the network, denoted as $\mathcal{N}_L \subset \mathcal{N}$, which are exposed to direct external attacks. For each node l in this subset, its corresponding exploit probability is represented as ρ_l.

III. Optimal Detection for Bayesian Attack Graphs

A. BAG Representation as a Hidden Markov Model (HMM)

1) State Process: The BAG can be viewed as a binary-state Markov process, where binary-state variables represent the compromise status of nodes. The state vector encompasses the compromise states of all (n) nodes within the network and is denoted as $\mathbf{x}_t = [\mathbf{x}_t(1), ..., \mathbf{x}_t(n)]$. Here, each $\mathbf{x}_t(i)$ can get a value of 0 or 1, with $\mathbf{x}_t(i) = 1$ indicating the compromise of the ith component at the time step t and, conversely, $\mathbf{x}_t(i) = 0$ denoting its uncompromised state. Consequently, the state vector $\mathbf{x}_t = (0, 0, \cdots, 0)$ signifies a network entirely devoid of compromises, while $\mathbf{x}_t = (1, 1, \cdots, 1)$ indicates a network in which all nodes have been compromised. As such, the state vector can take any of the 2^n possible state values, collectively represented as $\{\mathbf{x}^1, \cdots, \mathbf{x}^{2^n}\}$.

As the attack spreads across the network, the graph nodes undergo compromised status changes. The propagation of the attack depends on the external attack probabilities, the internal exploit probabilities, the security level of the nodes indicated by the nodes' type, and the mitigation made to remove the compromises in the network. For example, AND nodes resist single threats in their in-neighbor set, as all nodes in the set must be compromised for a successful breach at an AND node. In contrast, OR nodes can be compromised with just one compromised in-neighbor. Similarly, large exploit probabilities and external attacks increase the vulnerability of the network to breaches.

A common way to remove network compromises is by applying automated security patches on selected computers/servers with potential compromises. Despite the simplicity of this approach (resulting from the possibility of being performed remotely in the background), these computer/server compromises may not be certainly removed, especially in domains with new and difficult-to-detect attacks. Reimaging is another common way to remove compromises in network components, which requires reinstalling machines and servers. Despite the cost and potential disruptions in network performance, this approach has a high success rate in removing compromises. The unsuccessful removal occurs in the case that attackers have stolen the server/computer credentials (e.g., domain passwords), in which reinstalling might not fully secure the device. For simplicity, we refer to any type of action for removing compromises as reimaging. Let $\mathbf{a}_{t-1} \subset \mathcal{N}$ be a subset of nodes selected for the reimaging process at the time step t. We assume that $(1-\alpha)$ is the success probability of removing the compromise at any selected node, where $0 \leq \alpha \leq 1$ is the probability of unsuccessful removal of compromise. The value of α depends on the complexity of the reimaging process; a more extensive patching or reimaging process corresponds to a smaller α value.

Depending on the type of node, the conditional probability that the jth node is compromised at the time step t, given the state of the nodes at the time step $t-1$ (denoted as $\mathbf{x}_{t-1}$) and the set of reimaged nodes ($\mathbf{a}_{t-1} = \{i_1, ..., i_r\} \subset \mathcal{N}$), can be written as:

- *AND Nodes:*

$$P(\mathbf{x}_t(j) = 1 \mid \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = \begin{cases} (1_{j \notin \mathbf{a}_{t-1}} + \alpha 1_{j \in \mathbf{a}_{t-1}}) \left[\rho_j + (1 - \rho_j) \prod\limits_{i \in D_j} \rho_{ij} 1_{\mathbf{x}_{t-1}(i)=1} \right] & \text{if } \mathbf{x}_{t-1}(j) = 0 \\ 1_{j \notin \mathbf{a}_{t-1}} + \alpha 1_{j \in \mathbf{a}_{t-1}} & \text{if } \mathbf{x}_{t-1}(j) = 1, \end{cases} \tag{1}$$

- *OR Nodes:*

$$P(\mathbf{x}_t(j) = 1 \mid \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = \begin{cases} (1_{j \notin \mathbf{a}_{t-1}} + \alpha 1_{j \in \mathbf{a}_{t-1}}) \left[\rho_j + (1 - \rho_j) \left[1 - \prod\limits_{i \in D_j} \left(1 - \rho_{ij} 1_{\mathbf{x}_{t-1}(i)=1} \right) \right] \right] & \text{if } \mathbf{x}_{t-1}(j) = 0, \\ 1_{j \notin \mathbf{a}_{t-1}} + \alpha 1_{j \in \mathbf{a}_{t-1}} & \text{if } \mathbf{x}_{t-1}(j) = 1, \end{cases} \tag{2}$$

where $1_{\text{condition}}$ is equal to 1, when the condition holds (i.e. condition = True), and 0 otherwise (i.e. condition = False).

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

With binary-state variables, one can calculate the probability of the jth variable being 0 with $P(\mathbf{x}_t(j) = 0 \mid \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = 1 - P(\mathbf{x}_t(j) = 1 \mid \mathbf{x}_{t-1}, \mathbf{a}_{t-1})$.

2) Measurement Process: This paper considers a realistic scenario where the network components are monitored partially and imperfectly. In practice, network compromises might not be definitively detectable by routine network monitoring. In particular, it can be difficult to identify more sophisticated and newer types of attacks on computers and servers. Meanwhile, monitoring network compromises demands resources, time, and costs, which are often limited in practical settings, and excessive monitoring can also delay network operations. Therefore, a small subset of nodes is often monitored to ensure network security. Consider $\mathbf{s}_{t-1}$, which consists of the indices of m nodes chosen for monitoring at the time step t. These indices, $\{i_1, ..., i_m\} \subset \mathcal{N}$, are selected in advance at time step $t-1$ for the subsequent network monitoring at time step t. The resulting observations from this selection are collectively denoted as $\mathbf{y}_t$, where $\mathbf{y}_t(i)$ represents the observation obtained from node $\mathbf{s}_{t-1}(i)$.

In this paper, we assume the potential compromise is monitored with probability $(1-q)$ where $0 \leq q \leq 1$ denotes the probability of missing the compromise in the monitoring process. The observation process described above can be expressed at time step t as:

$$\mathbf{y}_t(i) = \begin{cases} 1 \text{ if } \mathbf{x}_t(s_{t-1}(i)) = 1 & \text{with probability } 1-q \\ 0 \text{ if } \mathbf{x}_t(s_{t-1}(i)) = 1 & \text{with probability } q \\ 0 \text{ if } \mathbf{x}_t(s_{t-1}(i)) = 0 & \text{with probability } 1 \end{cases}, \tag{3}$$

for $i = 1, ..., m$. Accurate monitoring systems can be represented by small q values, whereas larger q values represent networks with more advanced compromises or less accurate monitoring systems.

B. Optimal MMSE Attack Detection

Let $\mathbf{a}_{0:t-1} = (\mathbf{a}_0, ..., \mathbf{a}_{t-1})$ be the sequence of the reimaged nodes, and $\mathbf{s}_{0:t-1} = (\mathbf{s}_0, ..., \mathbf{s}_{t-1})$ and $\mathbf{y}_{1:t} = (\mathbf{y}_1, ..., \mathbf{y}_t)$ be the selected monitoring nodes and associated observations up to the time step t. The detection objective is to find the best estimate of network compromises at any given time step. This can be formulated as finding the attack detector $\hat{\mathbf{x}}_{t|t}$ of the true state $\mathbf{x}_t$ by minimizing a criterion that measures the difference between the detected and the true unobserved attack. One of the most popular criteria is mean-squared error (MSE), formulated as:

$$C(\mathbf{x}_t, \hat{\mathbf{x}}_{t|t}) = E\left[\|\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\|_2^2 \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right], \tag{4}$$

where $\|.\|_2^2$ is the squared L_2 norm (i.e. MSE). The estimate that minimizes MSE is called the minimum mean square error (MMSE) attack detector and can be obtained as:

$$\hat{\mathbf{x}}_{t|t}^{\text{MS}} = \underset{\hat{\mathbf{x}}_{t|t} \in \Psi}{\operatorname{argmin}} C(\mathbf{x}_t, \hat{\mathbf{x}}_{t|t}), \tag{5}$$

where Ψ is the set of all possible attack detectors. The solution for this optimization problem resembles the exact solution obtained by Kalman Filtering for linear and Gaussian state-space model [32].

We define the thresholding operator $\overline{\mathbf{v}}$, which acts on any vector $\mathbf{v} \in [0,1]^n$ and sets $\overline{\mathbf{v}}(i) = 1$ if $\mathbf{v}(i) > 1/2$, and 0 otherwise, for $i = 1, \ldots, n$. The following theorem characterizes the exact optimization solution in (5).

Theorem 1: *Given $\mathbf{a}_{0:t-1}$, $\mathbf{s}_{0:t-1}$ and $\mathbf{y}_{1:t}$ be the reimaged nodes, the monitored nodes, and the observations, up to time t. The exact optimal MMSE attack detector at time step t can be achieved as:*

$$\hat{\mathbf{x}}_{t|t}^{\text{MS}} = \overline{E\left[\mathbf{x}_t \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right]}, \tag{6}$$

with optimal conditional MSE

$$C_{t|t}^{MS} = \frac{n}{2} - \sum_{i=1}^{n} \left| E\left[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right] - \frac{1}{2} \right|. \tag{7}$$

The error takes the values $0 \leq C_{t|t}^{\text{MS}} \leq n/2$, and the operator $|.|$ represents the absolute value. The smaller values of $C_{t|t}^{\text{MS}}$ indicate a more precise attack detection, while the larger values indicate a higher expected attack detection error. The operator $\overline{E[.]}$ applied to the expected value of the state vector transforms the values of the vector greater than 0.5 to 1 and less than or equal to 0.5 to 0.

Proof: Given the sequence of observation $\mathbf{y}_{1:t}$ at nodes $\mathbf{s}_{0:t-1}$, we seek a detector $\hat{\mathbf{x}}_{t|t}$ of the true compromise $\mathbf{x}_t$ by solving the minimization in (5). This minimization can be expanded as:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t}^{\text{MS}} &= \underset{\hat{\mathbf{x}}_{t|t} \in \Psi}{\operatorname{argmin}} C\left(\mathbf{x}_t, \hat{\mathbf{x}}_{t|t}\right) \\ &= \underset{\hat{\mathbf{x}}_{t|t} \in \Psi}{\operatorname{argmin}} E\left[\|\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\|_2^2 \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right] \\ &= \underset{\hat{\mathbf{x}}_{t|t} \in \Psi}{\operatorname{argmin}} \sum_{i=1}^{n} E\left[|\mathbf{x}_t(i) - \hat{\mathbf{x}}_{t|t}(i)|^2 \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right]. \\ &= \underset{\hat{\mathbf{x}}_{t|t} \in \Psi}{\operatorname{argmin}} \sum_{i=1}^{n} E\left[|\mathbf{x}_t(i) - \hat{\mathbf{x}}_{t|t}(i)| \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right], \end{aligned} \tag{8}$$

where the last line is obtained given that $\|\mathbf{v}\|_2^2 = \|\mathbf{v}\|_1 = \sum_{i=1}^{n} |\mathbf{v}(i)|$ for a Boolean vector $|\mathbf{v}|$.

The minimization in (8) can be achieved by choosing $\hat{\mathbf{x}}_{t|t}(i)$ that minimizes $E[|\mathbf{x}_t(i) - \hat{\mathbf{x}}_{t|t}(i)| \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}]$, for $i = 1, \ldots, n$. Since the state variables are Boolean, the minimizer is given by:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t}^{\text{MS}}(i) &= 1_{E[\mathbf{x}_t(i) | \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}] > 1/2} \\ &= \overline{E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}]}, \end{aligned} \tag{9}$$

In other words,

$$\hat{\mathbf{x}}_{t|t}^{\text{MS}} = \overline{E[\mathbf{x}_t \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}]}. \tag{10}$$

The optimal conditional MSE can also be expressed as:

$$C_{t|t}^{\text{MS}} = \sum_{i=1}^{n} P\left(\hat{\mathbf{x}}_{t|t}^{\text{MS}}(i) \neq \mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right), \tag{11}$$

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

where

$$P\left(\hat{\mathbf{x}}_{t|t}^{\mathrm{MS}}(i) \neq \mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right) = \begin{cases} 1 - E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}] \\ \qquad \text{if } E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}] > 1/2 \\ E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}], \qquad \text{otherwise} \end{cases} \tag{12}$$

$$= \min\{E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}], 1 - E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}]\}.$$

Substituting (12) into (11) leads to

$$C_{t|t}^{\mathrm{MS}} = \frac{n}{2} - \sum_{i=1}^{n} \left| E[\mathbf{x}_t(i) \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}] - \frac{1}{2} \right|. \tag{13}$$

The last expression in (11) is derived from the identity $\min\{a, 1-a\} = 1/2 - |a - 1/2|$, which is true for values of a between 0 and 1.

■

C. Exact Calculation of Optimal Attack Detector

This section presents our recursive algorithm for calculating the optimal MMSE attack detector. Let $A = [\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}]$ be a $n \times 2^n$ matrix consisting of an arbitrary enumeration of possible network compromises. We define the conditional probability distribution of the compromise-status vector at time step t given the information up to time step t as:

$$\mathbf{\Pi}_{t|t}(i) = P\left(\mathbf{x}_t = \mathbf{x}^i \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right), \ i = 1, \ldots, 2^n. \tag{14}$$

Using (6) and (14), we can formulate the optimal MMSE attack detector as follows:

$$\hat{\mathbf{x}}_{t|t}^{\mathrm{MS}} = \overline{E\left[\mathbf{x}_t \mid \mathbf{a}_{0:t-1}, \mathbf{s}_{0:t-1}, \mathbf{y}_{1:t}\right]} = \overline{A\mathbf{\Pi}_{t|t}}. \tag{15}$$

Similarly, using (7) and (14), we can compute the optimal conditional MSE as:

$$C_{t|t}^{\mathrm{MS}} = \frac{n}{2} - \sum_{i=1}^{n} \left| (A\mathbf{\Pi}_{t|t})_i - \frac{1}{2} \right|, \tag{16}$$

where $(A\mathbf{\Pi}_{t|t})_i$ represents the probability of node i being compromised at time step t, given information up to time $t-1$.

Under the reimaged nodes $\mathbf{a}_{t-1}$ let $M_t(\mathbf{a}_{t-1})$ be the $2^n \times 2^n$ *transition matrix* of the Markov chain:

$$(M_t(\mathbf{a}_{t-1}))_{ij} = P(\mathbf{x}_t = \mathbf{x}^i \mid \mathbf{x}_{t-1} = \mathbf{x}^j, \mathbf{a}_{t-1}) = \prod_{l=1}^{n} \left(\eta_l^{ij}(\mathbf{a}_{t-1}) \, 1_{\mathbf{x}^i(l)=1} + (1 - \eta_l^{ij}(\mathbf{a}_{t-1})) \, 1_{\mathbf{x}^i(l)=0} \right), \tag{17}$$

for $i, j = 1, \ldots, 2^n$; where

$$\eta_l^{ij}(\mathbf{a}_{t-1}) = (1_{l \notin \mathbf{a}_{t-1}} + \alpha 1_{l \in \mathbf{a}_{t-1}}) 1_{\mathbf{x}^j(l)=0} \Bigg[\rho_l + (1 - \rho_l) \prod_{r \in D_l} \rho_{rl} 1_{\mathbf{x}^j(r)=1} \Bigg] 1_{\mathcal{N}_l = \mathrm{AND}} + (1_{l \notin \mathbf{a}_{t-1}} + \alpha 1_{l \in \mathbf{a}_{t-1}}) 1_{\mathbf{x}^j(l)=0} \Bigg[\rho_l + (1 - \rho_l) \Bigg(1 - \prod_{r \in D_l} (1 - \rho_{rl} 1_{\mathbf{x}^j(r)=1}) \Bigg) \Bigg] 1_{\mathcal{N}_l = \mathrm{OR}} + (1_{l \notin \mathbf{a}_{t-1}} + \alpha 1_{l \in \mathbf{a}_{t-1}}) 1_{\mathbf{x}^j(l)=1}. \tag{18}$$

Note that $1_{\mathcal{N}_l=\mathrm{AND}}$ is 1 if node l is an AND node and 0 otherwise. Similarly, $1_{\mathcal{N}_l=\mathrm{OR}}$ is 1 if node l is an OR node and 0 otherwise, and the expression in (18) is derived according to (1) and (2).

Furthermore, we define the *update vector*, $T_t(\mathbf{y}_t, \mathbf{s}_{t-1})$ given the observation vector $\mathbf{y}_t$ from nodes $\mathbf{s}_{t-1}$ at time step t:

$$\begin{aligned} (T_t(\mathbf{y}_t, \mathbf{s}_{t-1}))_i &= P\left(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}^i, \mathbf{s}_{t-1}\right) \\ &= \prod_{l=1}^{m} P\left(\mathbf{y}_t(l) \mid \mathbf{x}_t = \mathbf{x}^i, \mathbf{s}_{t-1}\right) \\ &= \prod_{l=1}^{m} P\left(\mathbf{y}_t(l) \mid \mathbf{x}_t\left(\mathbf{s}_{t-1}(l)\right) = \mathbf{x}^i\left(\mathbf{s}_{t-1}(l)\right)\right) \\ &= \prod_{l=1}^{m} \left| (q-1)\mathbf{x}^i\left(\mathbf{s}_{t-1}(l)\right) - \mathbf{y}_t(l) + 1 \right|, \end{aligned} \tag{19}$$

for $i = 1, \ldots, 2^n$, where the last expression in (19) is deduced based on the observation model presented in (3).

Algorithm 1 **Optimal Detection for Bayesian Attack Graphs**

1: Initialization: $\mathbf{\Pi}_{0|0}$.
2: **for** $k = 1, 2, \ldots$ **do**
3: Prediction: $\mathbf{\Pi}_{t|t-1} = M_t(\mathbf{a}_{t-1}) \mathbf{\Pi}_{t-1|t-1}$.
4: Update: $\mathbf{\Pi}_{t|t} = \frac{T_t(\mathbf{y}_t, \mathbf{s}_{t-1}) \circ \mathbf{\Pi}_{t|t-1}}{\|T_t(\mathbf{y}_t, \mathbf{s}_{t-1}) \circ \mathbf{\Pi}_{t|t-1}\|_1}$.
5: MMSE Attack Detector: $\hat{\mathbf{x}}_{t|t}^{\mathrm{MS}} = \overline{A\mathbf{\Pi}_{t|t}}$.
6: Optimal MSE: $C_{t|t}^{\mathrm{MS}} = \frac{n}{2} - \sum_{i=1}^{n} \left| (A\mathbf{\Pi}_{t|t})_i - \frac{1}{2} \right|$.
7: **end for**

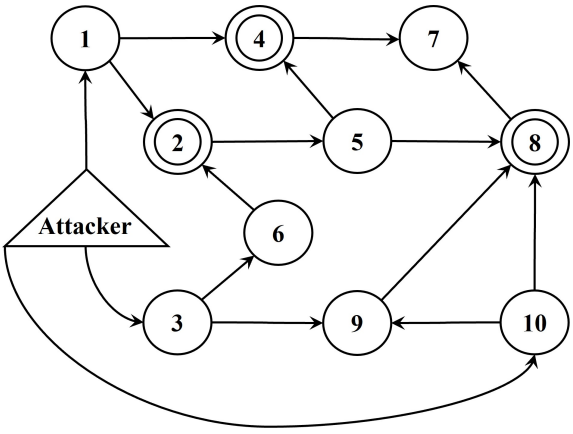

Fig. 1: The 10-node BAG used for our numerical experiments.

We also define $\mathbf{\Pi}_{t|t-1}$ as the predictive posterior of compromise status at time step t as:

$$\mathbf{\Pi}_{t|t-1}(i) = P\left(\mathbf{x}_t = \mathbf{x}^i \mid \mathbf{a}_{0:t-2}, \mathbf{s}_{0:t-2}, \mathbf{y}_{1:t-1}\right), \ i = 1, \ldots, 2^n. \tag{20}$$

A recursive computation of $\mathbf{\Pi}_{t|t-1}$ can be performed as follows:

$$\mathbf{\Pi}_{t|t-1} = M_t(\mathbf{a}_{t-1}) \mathbf{\Pi}_{t-1|t-1}. \tag{21}$$

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

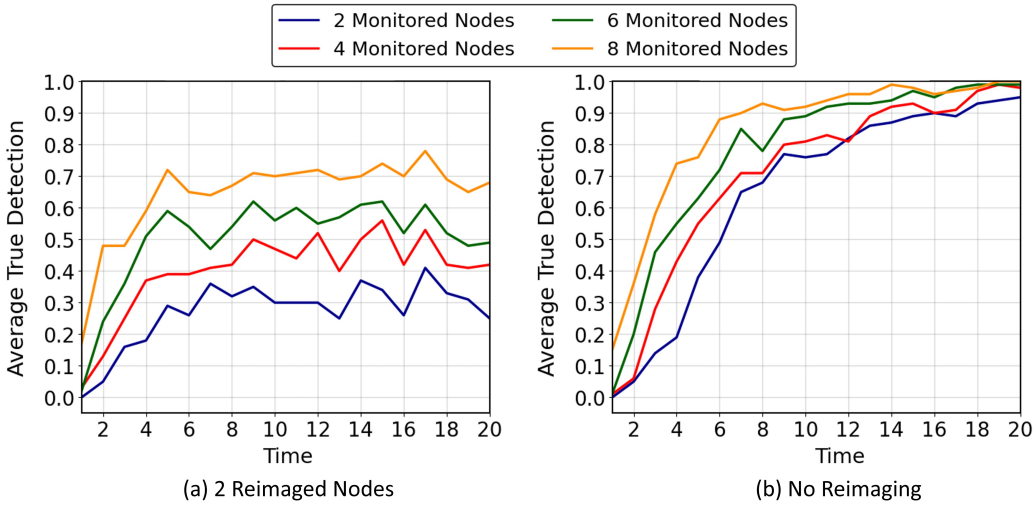

Fig. 2: The average true attack detection under 2, 4, 6, and 8 monitored nodes when (a) 2 random nodes are reimaged at any given time; (b) no reimaging.

With the information available up to time step t, we can express the posterior distribution of attacks at time t as follows [33]–[35]:

$$\mathbf{\Pi}_{t|t} = \frac{T_t(\mathbf{y}_t, \mathbf{s}_{t-1}) \circ \mathbf{\Pi}_{t|t-1}}{\|T_t(\mathbf{y}_t, \mathbf{s}_{t-1}) \circ \mathbf{\Pi}_{t|t-1}\|_1}, \tag{22}$$

where $\circ$ is the component-wise multiplication of two vectors.

The steps of the proposed detection method are provided in Algorithm 1. The computational complexity of the algorithm is of the order $O(2^{2n})$ since it requires the use of a transition matrix to compute the predictive posterior distribution.

IV. Numerical Experiments

The numerical experiments presented in this section assess the performance of our proposed attack detection method. We focus on the network illustrated in Fig. 1, originally presented in [31], [36] (minor modification is made to suit the network to our context). This BAG comprises 10 nodes, resulting in a total of $2^{10} = 1{,}024$ possible network compromise states. We assume a uniform prior for initial network compromises, i.e., $\mathbf{\Pi}_{0|0}(i) = 1/2^{10},\ i = 1, ..., 2^{10}$. All results in these experiments are averaged over 100 independent runs. Network vulnerabilities, denoted by ρ_{ij}, are as follows:

$$\rho_{12}=0.5, \rho_{14}=0.6, \rho_{25}=0.4, \rho_{36}=0.35, \rho_{39}=0.4,$$
$$\rho_{47}=0.7, \rho_{54}=0.6, \rho_{58}=0.5, \rho_{62}=0.5, \rho_{87}=0.7,$$
$$\rho_{98}=0.5, \rho_{10\,8}=0.5, \rho_{10\,9}=0.4.$$

The vulnerability of three nodes to external attacks is denoted by the following parameters: $\rho_1=0.6, \rho_3=0.7, \rho_{10}=0.6$.

In the first experiment, the observation uncertainty q is set to 0.2, indicating the inaccuracy of the monitoring system in capturing the compromise. Likewise, the inaccuracy of the reimaging is set to $\alpha = 0.2$. The average detection accuracy for different numbers of monitoring nodes is shown in Fig. 2. Fig. 2(a) represents the case where 2 random nodes are selected for reimaging at any time step. As can be seen, higher accuracy is achieved when more nodes are used for monitoring network compromises. The accuracy rate increases at early time steps as the uncertainty in network compromises due to the uniform prior fade out upon monitoring more nodes. However, after 6 to 7 time steps, the accuracy becomes, on average, constant (while fluctuating) and larger for cases with more monitoring nodes. Fig. 2(b) represents similar results when there is no reimaging. Compromised nodes, in this case, stay compromised and are likely to spread attacks rapidly to the entire network. Despite being more catastrophic, this case is easier for state detection. This is evident by the larger detection accuracy in all cases compared to 2 reimaged nodes per step in Fig. 2(a). In the event of no reimaging, there are fewer switches from 0 to 1 or from 1 to 0, which ultimately has helped to achieve higher detection accuracy and fewer fluctuations.

The average detection error of each node associated with the result in Fig. 2(a) with 2 reimaging nodes per step is shown in Fig. 3. It can be seen that the error decreases as more observations become available. Furthermore, monitoring more nodes leads to lower errors. Comparing the error for various nodes, one can see that the least detection error is obtained for node 8, which is an AND node with 3 connections from nodes 5, 9, and 10. The compromise in this node requires all neighboring nodes to be compromised and, at the same time, all exploits from neighboring nodes to be successful (see equation (1)). This makes the node less vulnerable than others, which can be seen in fewer possible dynamics and less average detection error. Fig. 4 shows the average compromise rate obtained without reimaging, random reimaging of 1 node and 2 nodes per time step. The number of randomly monitored nodes per step is two. There are 100 trajectories of length 20 with uncertainty parameters $q = 0.2$ and $\alpha = 0.2$. It can be seen that the average true compromise rate decreases as the number of reimaging nodes increases. The reduction is especially evident for AND nodes (nodes 2, 4, and 8) since the removal of compromise at their single in-neighbor can prevent them from being compromised. However, the

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

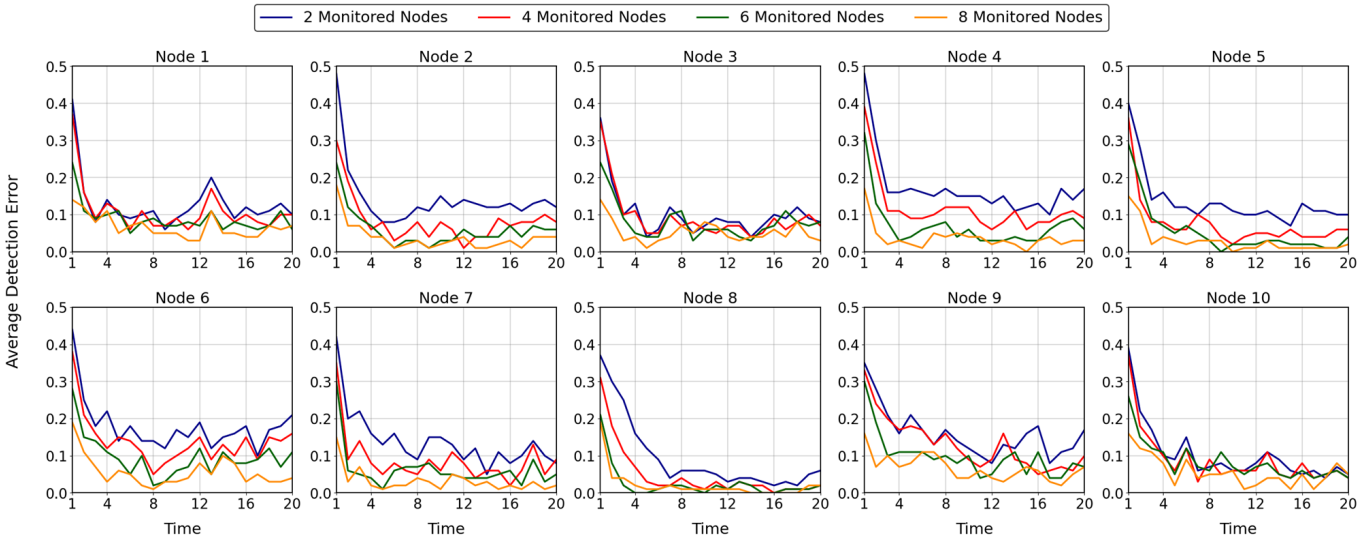

Fig. 3: The average detection error per node under 2, 4, 6, and 8 monitored nodes and 2 randomly selected nodes for reimaging.

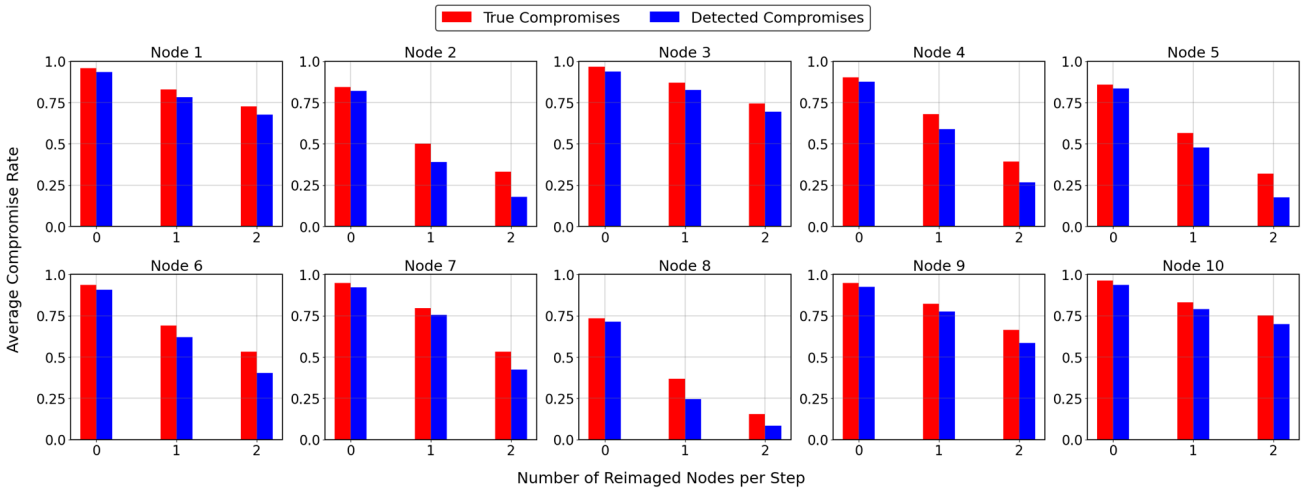

Fig. 4: The average true and detected compromise rate per node under no reimaging, and 1-node and 2-node reimaging.

OR nodes could still be compromised, even if one of their neighboring nodes becomes uncompromised. Comparing the results of the proposed detection policy and the true compromise rates, one can see that the detection accuracy is very high under no reimaging. This comes from the fact that all nodes under no reimaging become compromised and stay at that status, leading to high detection accuracy. However, a larger detection error can be seen with two reimaging nodes. The reason is that the spread of attacks becomes more challenging once reimaging takes place, especially in scenarios where imperfect reimaging poses another layer of uncertainty regarding the compromised status of the nodes. Nodes 1, 3, and 10 are directly susceptible to external attacks, making them more prone to compromise. Therefore, it is expected that they have higher compromise rates than other nodes. However, the fact that they are likely to be and remain compromised makes it easier to detect their state. As we can see, nodes 1, 3, and 10 exhibit a smaller difference between true and detected compromises.

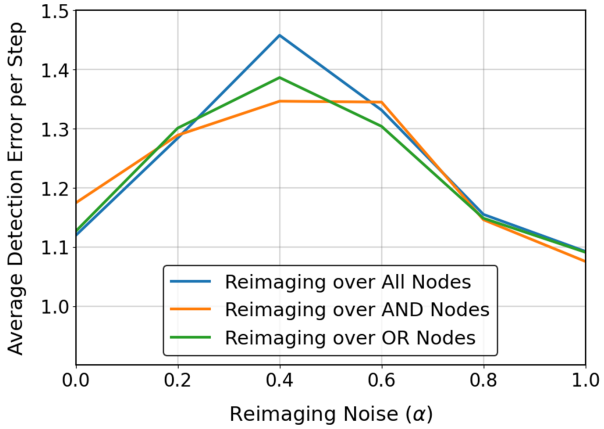

Fig. 5: The average detection error with respect to the reimaging uncertainty (α).

The next part of the numerical experiment analyzes how reimaging uncertainty affects the performance of the proposed detection method. Fig. 5 represents the average attack

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

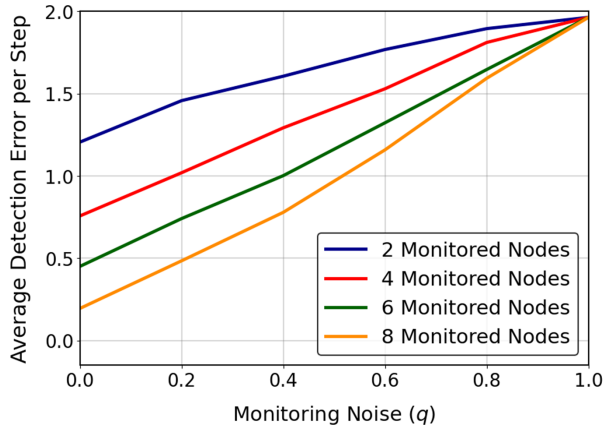

Fig. 6: The average detection error with respect to the monitoring uncertainty (q).

detection error with respect to reimaging uncertainty when a random reimaging node is selected over all, AND, and OR nodes. Two nodes are randomly monitored with the measurement uncertainty of $q = 0.2$ per step. It can be seen that the error is largest under the highest uncertainty in the reimaging process (i.e., values of α close to 0.5). The two ends of $\alpha = 0$ and $\alpha = 1$ represent the perfect reimaging and no reimaging scenarios, respectively. Therefore, the minimum error has been obtained given the minimum uncertainty of cleaning or not cleaning the compromises in these two cases. Comparing the reimagining in AND and OR nodes, one can see that similar trends have been obtained when the reimagining is over all, AND or OR nodes. For $\alpha = 0.5$, which corresponds to the most uncertain reimaging process, reimaging over AND nodes has a less negative impact on detection performance. This comes from the fact that the AND nodes are less likely to be compromised; thus, reimaging them does not significantly affect the detection error.

Fig. 6 represents the average error of attack detection per step with respect to the monitoring uncertainty. A node is randomly reimaged at any given time with $\alpha = 0.4$. q represents the uncertainty in observing the compromise at 2, 4, 6, 8 randomly monitored nodes per step. It can be seen that the error is minimum for $q = 0$ under 8 monitored nodes and is larger for smaller monitored nodes. As monitoring uncertainty increases, the average detection error increases in all cases. In particular, $q = 1$ represents the case in which the monitored node will show uncompromised results, regardless of whether it is compromised. This has led to a relatively larger detection error than the smaller uncertainty values of the monitoring at various monitored nodes. Monitoring will be ineffective in that case ($q = 1$) since it does not provide any information to detect network compromises.

V. Conclusion

Cybersecurity aims to detect attacks in the network accurately and promptly to prevent their spread. The main challenges in detection are partial monitoring due to resource limitations and difficulty in detecting and removing complex attacks. This paper derives the optimal minimum mean square error (MMSE) attack detection for a general class of BAGs under uncertain monitoring and reimaging. The optimal solution resembles the Kalman filtering method derived for linear Gaussian state-space models. The proposed attack detection method is implemented with a recursive and efficient matrix-form algorithm. The performance of the proposed method has been demonstrated for attack detection of a network containing 10 nodes when the network is compromised through external attacks. Our future work will study the security analysis of larger networks, including systematizing detection, monitoring, and defense in large and complex networks.

Acknowledgment

The authors acknowledge the support of the National Science Foundation awards IIS-2311969 and IIS-2202395, ARMY Research Laboratory awards W911NF2320179 and W911NF2410098, and Office of Naval Research award N00014-23-1-2850.

References

[1] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 4, pp. 853–865, 2010.

[2] A. T. Al Ghazo, M. Ibrahim, H. Ren, and R. Kumar, "A2G2V: automatic attack graph generation and visualization and its applications to computer and SCADA networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3488–3498, 2019.

[3] N. Asadi, S. H. Hosseini, M. Imani, D. P. Aldrich, and S. F. Ghoreishi, "Privacy-preserved federated reinforcement learning for autonomy in signalized intersections," in *ASCE International Conference on Transportation and Development (ICTD)*, American Society of Civil Engineers, 2024.

[4] A. Ravari, S. F. Ghoreishi, and M. Imani, "Implicit human perception learning in complex and unknown environments," in *American Control Conference (ACC)*, IEEE, 2024.

[5] M. Alali, A. Kazeminajafabadi, and M. Imani, "Deep reinforcement learning sensor scheduling for effective monitoring of dynamical systems," *Systems Science & Control Engineering*, 2024.

[6] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 Workshop on New Security Paradigms*, NSPW '98, (New York, NY, USA), p. 71–79, Association for Computing Machinery, 1998.

[7] L. Muñoz-González, D. Sgandurra, M. Barrère, and E. C. Lupu, "Exact inference techniques for the analysis of Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 231–244, 2017.

[8] A. Kazeminajafabadi and M. Imani, "Optimal monitoring and attack detection of networks modeled by Bayesian attack graphs," *Cybersecurity*, vol. 6, no. 1, p. 22, 2023.

[9] J. Sembiring, M. Ramadhan, Y. S. Gondokaryono, and A. A. Arman, "Network security risk analysis using improved MulVAL Bayesian attack graphs," *International Journal on Electrical Engineering and Informatics*, vol. 7, no. 4, p. 735, 2015.

[10] J. Liu, B. Liu, R. Zhang, and C. Wang, "Multi-step attack scenarios mining based on neural network and Bayesian network attack graph," in *International Conference on Artificial Intelligence and Security*, pp. 62–74, Springer, 2019.

[11] Z. Hu, M. Zhu, and P. Liu, "Online algorithms for adaptive cyber defense on Bayesian attack graphs," in *Proceedings of the 2017 Workshop on moving target defense*, pp. 99–109, 2017.

[12] I. Matthews, J. Mace, S. Soudjani, and A. van Moorsel, "Cyclic Bayesian attack graphs: a systematic computational approach," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 129–136, IEEE, 2020.

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.

[13] N. Asadi and S. F. Ghoreishi, "Bayesian state estimation in partially-observed dynamic multidisciplinary systems," *Frontiers in Aerospace Engineering*, vol. 1, p. 1036642, 2022.

[14] S. M. Radack *et al.*, "The common vulnerability scoring system (CVSS)," 2007.

[15] K. Durkota, V. Lisy, B. Bo**v**sansky, and C. Kiekintveld, "Optimal network security hardening using attack graph games," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 526–532, AAAI Press, 2015.

[16] Y. Liu and H. Man, "Network vulnerability assessment using Bayesian networks," in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005* (B. V. Dasarathy, ed.), vol. 5812, pp. 61 – 71, International Society for Optics and Photonics, SPIE, 2005.

[17] M. Alhomidi and M. Reed, "Risk assessment and analysis through population-based attack graph modelling," in *World Congress on Internet Security (WorldCIS-2013)*, pp. 19–24, IEEE, 2013.

[18] M. Husák, J. Komárková, E. Bou-Harb, and P. **v**Celeda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2018.

[19] B. Asvija, R. Eswari, and M. Bijoy, "Bayesian attack graphs for platform virtualized infrastructures in clouds," *Journal of Information Security and Applications*, vol. 51, p. 102455, 2020.

[20] K. Wu, H. Qu, and C. Huang, "A network intrusion detection method incorporating Bayesian attack graph and incremental learning part," *Future Internet*, vol. 15, no. 4, p. 128, 2023.

[21] M. Alali and M. Imani, "Inference of regulatory networks through temporally sparse data," *Frontiers in control engineering*, vol. 3, p. 1017256, 2022.

[22] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal recursive expert-enabled inference in regulatory networks," *IEEE control systems letters*, vol. 7, pp. 1027–1032, 2022.

[23] S.-c. Liu and Y. Liu, "Network security risk assessment method based on HMM and attack graph model," in *2016 17th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)*, pp. 517–522, IEEE, 2016.

[24] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: A probabilistic approach," *Computers & security*, vol. 32, pp. 158–169, 2013.

[25] Y. Ma, Y. Wu, D. Yu, L. Ding, and Y. Chen, "Vulnerability association evaluation of internet of thing devices based on attack graph," *International Journal of Distributed Sensor Networks*, vol. 18, no. 5, p. 15501329221097817, 2022.

[26] S. Chockalingam, W. Pieters, A. Teixeira, and P. v. Gelder, "Bayesian network models in cyber security: a systematic review," in *Nordic Conference on Secure IT Systems*, pp. 105–122, Springer, 2017.

[27] A. Sahu and K. Davis, "Structural learning techniques for Bayesian attack graphs in cyber physical power systems," in *2021 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, IEEE, 2021.

[28] M. Frigault, L. Wang, S. Jajodia, and A. Singhal, "Measuring the overall network security by combining CVSS scores based on attack graphs and Bayesian networks," in *Network Security Metrics*, pp. 1–23, Springer, 2017.

[29] C. Liang, F. Wen, and Z. Wang, "Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks," *Information Fusion*, vol. 46, pp. 44–50, 2019.

[30] C.-Z. Bai, V. Gupta, and F. Pasqualetti, "On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6641–6648, 2017.

[31] Z. Hu, M. Zhu, and P. Liu, "Adaptive cyber defense against multi-stage attacks using learning-based POMDP," *ACM Transactions on Privacy and Security (TOPS)*, vol. 24, no. 1, pp. 1–25, 2020.

[32] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Kalman filtering*. Wiley Telecom, 2020.

[33] M. Alali and M. Imani, "Reinforcement learning data-acquiring for causal inference of regulatory networks," in *American Control Conference (ACC)*, IEEE, 2023.

[34] S. H. Hosseini and M. Imani, "An optimal Bayesian intervention policy in response to unknown dynamic cell stimuli," *Information Sciences*, 2024.

[35] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal inference of hidden Markov models through expert-acquired data," *IEEE Transactions on Artificial Intelligence*, 2024.

[36] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2011.

Authorized licensed use limited to: Northeastern University. Downloaded on June 02,2025 at 15:41:41 UTC from IEEE Xplore. Restrictions apply.