Manuscript submitted to *Security and Safety*, For Peer Review Only, Vol. x, xxxxxxx (2025)
https://doi.org/10.1051/sands/xxxxxxx
sands.edpsciences.org

Research Article OPEN ACCESS

Section Category

Optimal DoS attack scheduling against multi-systems remote state estimation: A reinforcement learning approach

Shunpeng Zhang[1,2,*], Zhitao Fan[1,2], and Xingquan Fu[3]

[1] *School of Automation, Beijing Institute of Technology, Beijing 100081, China*
[2] *Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314000, China*
[3] *School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China*

Received: xx xxxxx 2022 / Revised: xx xxxxx 2022 / Accepted: xx xxxxx 2022 / Published online: xx xxxxx 2022

Abstract This paper investigates the optimal denial-of-service (DoS) attack scheduling problem in multi-systems operating over multiple Markovian fading channels. At each time instant, each smart sensor obtains the local state estimate of its associated subsystem and transmits it to a remote estimator via its corresponding wireless communication channel. During this process, malicious DoS attackers inject interference into the channels, thereby increasing the packet-drop probability. However, because the attackers have limited energy resources, they cannot attack all transmissions simultaneously. Consequently, an attack scheduling strategy must be designed to determine which channels to target under the energy constraint. To address this issue, a Markov decision process (MDP) model is established. Moreover, the existence of an optimal stationary policy for the formulated MDP model is rigorously established, and its structural properties are analyzed. Furthermore, a reinforcement learning based DoS attack scheduling algorithm is developed to approximate the optimal policy. Finally, simulation results are provided to demonstrate the superiority of the proposed method.

Keywords DoS attack, remote state estimation, markov decision process, reinforcement learning.

Citation Shunpeng Zhang, Zhitao Fan and Xingquan Fu. Optimal DoS attack scheduling against multi-systems remote state estimation: A reinforcement learning approach. Security and Safety 2025; **x**: xxxxxxx. https://doi.org/10.1051/sands/xxxxxxx

1 Introduction

Remote state estimation serves as a core enabling technology in cyber–physical systems (CPSs) [1]. It has been extensively applied in unmanned vessel target tracking, real-time monitoring of smart grids, and feedback control in industrial manufacturing processes [2]. However, the remote state estimation process is inherently vulnerable to malicious network attacks due to its dependence on wireless communication links [3, 4]. Among various attack types, denial-of-service (DoS) attacks are particularly common, as they increase packet loss probability by injecting interference energy into the communication channels during data transmission [5, 6]. When a packet is dropped, the remote estimator must solely rely on model-based prediction, leading to a progressive growth of the estimation error covariance. If packet losses occur over consecutive transmission attempts, model uncertainties and noise accumulation can substantially amplify the estimation error and may even drive the estimation process into instability.

Under normal operating conditions, the power available to DoS attackers is inherently limited. Therefore, the attackers must strategically allocate their limited jamming resources to maximize the degradation of the remote estimator's performance. This requirement naturally leads to the problem

* Corresponding author (email: 3220241255@bit.edu.cn)

of optimal DoS attack scheduling, which has consequently become an active and extensively studied research topic [7–10]. To tackle the fundamental question of determining the optimal power expenditure for energy-limited DoS attacks, [7] introduced power allocation algorithms tailored for static environments and developed a Markov decision process (MDP) framework for dynamic settings to derive the optimal attack strategy. [8] examined energy-constrained DoS attacks with the objective of minimizing the performance of remote estimation in wireless networks by employing an optimal energy allocation strategy. Within a framework that accounts for wireless transmission losses, the research established sufficient conditions for the optimal solution in the context of static attacks. Furthermore, it developed an algorithm grounded in MDP theory to address dynamic attacks, effectively balancing the trade-off between attack energy consumption and the resulting degradation in system performance. The issue of energy scheduling in the context of DoS attacks aimed at remote state estimation within multi-hop networks was examined in [9]. This problem was modeled as a MDP, and the threshold structure characterizing the optimal attack strategy was rigorously established. In the presence of attackers, achieving optimal outcomes for the attack allocation problem across multiple systems proved challenging when employing conventional algorithms [10].

Reinforcement learning, as an effective framework for addressing large-scale MDP, is capable of adaptively learning near-optimal policies through interactions with the environment, even when the state–action spaces are extremely high-dimensional and the underlying model is challenging to specify explicitly [11–16]. To address the energy scheduling problem for DoS attacks, which was formulated as a MDP, [9] utilized a deep reinforcement learning (DRL) algorithm, specifically the dueling double Q-network (D3QN), to approximate the optimal deterministic and stationary policy. [17] employed a parameterized deep Q-network (P-DQN) algorithm to approximate the optimal stationary policy for the unconstrained Markov decision process (UMDP), which was derived from the original constrained MDP (CMDP) formulation of the co-design problem, effectively addressing the discrete-continuous hybrid action space. To obtain a near-optimal policy for the transmission scheduling problem, which was formulated as a modified MDP incorporating historical actions into the state, [18, 19] introduced a DRL algorithm, specifically the D3QN. [20] employed reinforcement learning, specifically the D3QN algorithm, to solve the MDP formulated for optimal redundant transmission scheduling, yielding a near-optimal policy with a threshold structure that minimized estimation errors and reduced computational complexity.

Based on the above summary and analysis, this article employs reinforcement learning to solve the optimal DoS attack scheduling problem for multi-systems remote state estimation [21]. To this end, a MDP model is formulated to characterize the attacker's decision-making mechanism. The existence of an optimal stationary policy for the proposed MDP is rigorously established, and its structural properties are thoroughly analyzed. Building upon these theoretical results, a reinforcement learning–based DoS attack scheduling algorithm is subsequently developed to approximate the optimal policy in scenarios where model information is unavailable [22, 23]. The main contributions of this paper comprise:

(1) Using a MDP formulation, this paper establishes the existence and structural properties of a deterministic stationary optimal strategy for the attack scheduling problem in multi-systems remote state estimation, thereby providing a solid theoretical foundation for reinforcement learning–based approaches.

(2) To overcome the dimensionality explosion inherent in the MDP formulation, the D3QN algorithm from reinforcement learning is employed. This approach enables the effective derivation of the optimal attack policy and provides insights into its structural characteristics. Experimental results further demonstrate that the learned optimal attack strategy significantly outperforms conventional attack scheduling methods.

The remainder of this paper is organized as follows: Section 2 is devoted to the problem formulation; Section 3 presents the main results; Section 4 shows illustrative examples; Section 5 summarizes the overall content.

Notations: $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{m\times n}$ denote the sets of real numbers, n-dimensional real vectors and $m \times n$ real-valued matrices, respectively. $\mathbb{U}$ denote a complete set. $\mathbb{N}$, $\mathbb{N}^*$ denote the set of natural numbers, the set of positive integers, respectively. For a symmetric matrix X, $X \succ 0$ ($X \succeq 0$) signifies positive definiteness (semi-definiteness). For a matrix Y, $\rho(Y)$, Y^{T} and $\mathrm{Tr}(Y)$ represent its spectral radius, transpose and trace. The operators $\mathbb{E}[\cdot]$ and $\mathbb{E}[\cdot \mid \cdot]$ correspond to expectation and conditional expectation, while $\mathcal{P}(\cdot)$ and $\mathcal{P}(\cdot \mid \cdot)$ denote probability and conditional probability measures. For two items $s = (s_1, s_2)$, and $j = (j_1, j_2)$, the relation $s \preceq (\succeq) j$ represents that $s_1 \leq (\geq) j_1$, and $s_2 \leq (\geq) j_2$. Let $s \uparrow (\downarrow) j$ denote the

join (meet), where $s \uparrow j = (\min(s_1,j_1),\min(s_2,j_2))$, $s \downarrow j = (\max(s_1,j_1),\max(s_2,j_2))$. $\otimes$ denotes Cartesian product. For a function $h(\cdot)$, $h^n(X) = h(h^{n-1}(X))$ is the nth composition function.

2 Problem formulation

2.1 System Description

There are M linear time-invariant systems with the following dynamics:

$$x_{i,k+1} = A_i x_{i,k} + \omega_{i,k},\ i \in \{1,\dots,M\} \tag{1}$$

where $x_{i,k} \in \mathbb{R}^{n_i}$ is the ith system state at time instant k, and $A_i \in \mathbb{R}^{n_i \times n_i}$ is the system matrix. The noises $\omega_{i,k} \in \mathbb{R}^{n_i}$ are i.i.d. white Gaussian random variables with zero mean and covariance $Q_i (\succeq 0) \in \mathbb{R}^{n_i \times n_i}$. To avoid trivial problems, assume systems are unstable, i.e., $\rho(A_i) > 1$, for all $i \in \{1,\dots,M\}$. Each process is measured by a smart sensor as

$$y_{i,k} = C_i x_{i,k} + v_{i,k}, \tag{2}$$

where $y_{i,k} \in \mathbb{R}^{m_i}$ is the measurement of process i at time k, and $C_i \in \mathbb{R}^{m_i \times n_i}$ is the ith sensor's observation matrix. The measurement noise $v_{i,k} \in \mathbb{R}^{m_i}$ are i.i.d. Gaussian random variables with zero mean and covariance matrix $R_i (\succ 0) \in \mathbb{R}^{m_i \times m_i}$. The noise processes $\{w_{i,k}\}$ and $\{v_{i,k}\}$ are assumed to be mutually independent for all i. The systems across different sensors are assumed to be mutually independent, with each sensor i possessing the computational capacity to run a Kalman filter, thereby enabling the local computation of state estimates and their corresponding error covariance matrices. The definitions are shown as follows:

$$\begin{aligned}
\hat{x}^s_{i,k|k-1} &\triangleq \mathbb{E}[x_{i,k} \mid y_{i,0},\dots,y_{i,k-1}],\\
\hat{x}^s_{i,k} &\triangleq \mathbb{E}[x_{i,k} \mid y_{i,0},\dots,y_{i,k}],\\
P^s_{i,k|k-1} &\triangleq \mathbb{E}[(x_{i,k}-\hat{x}^s_{i,k|k-1})(x_{i,k}-\hat{x}^s_{i,k|k-1})^{\mathrm{T}} \mid y_{i,0},\dots,y_{i,k-1}],\\
P^s_{i,k} &\triangleq \mathbb{E}[(x_{i,k}-\hat{x}^s_{i,k})(x_{i,k}-\hat{x}^s_{i,k})^{\mathrm{T}} \mid y_{i,0},\dots,y_{i,k}].
\end{aligned}$$

Assumption 1. *The pair (A_i, C_i) is observable and $(A_i, Q_i^{1/2})$ is controllable.*

As a consequence of Assumption 1, the local Kalman filter converges to a steady state exponentially fast. Without loss of generality, assume that the local Kalman filter has been in its steady state $P^s_{i,k}$ at the initial time instant, that is, $P^s_{i,k} = \overline{P}_i$, $\forall i \in \{1,\dots,M\}$, $k \geq 0$. Kalman filter process is presented as follows:

$$\begin{aligned}
\hat{x}^s_{i,k|k-1} &= A_i \hat{x}^s_{i,k-1},\\
P^s_{i,k|k-1} &= A_i P^s_{i,k-1} A_i^{\mathrm{T}} + Q_i,\\
K_i &= P^s_{i,k|k-1} C_i^{\mathrm{T}} (C_i P^s_{i,k|k-1} C_i^{\mathrm{T}} + R_i)^{-1},\\
\hat{x}^s_{i,k} &= A_i \hat{x}^s_{i,k-1} + K_i (y_{i,k} - C_i A_i \hat{x}^s_{i,k-1}),\\
P^s_{i,k} &= P^s_{i,k|k-1} - K_i C_i P^s_{i,k|k-1}.
\end{aligned}$$

At each time k, sensor i sends the output of its local Kalman filter (i.e. a posterior minimum mean square error (MMSE) estimate) $\hat{x}^s_{i,k}$ to the remote estimator over a lossy communication channel [24]. Let $\gamma_{i,k} \in \{0,1\}$ denote whether or not the packet is received error-free by the remote estimator. If it arrives successfully, $\gamma_{i,k} = 1$; $\gamma_{i,k} = 0$ otherwise. Since the smart sensor sends the MMSE estimates of the local state rather than the original measurement values $y_{i,k}$, when $k \geq 1$, the correlation error covariance

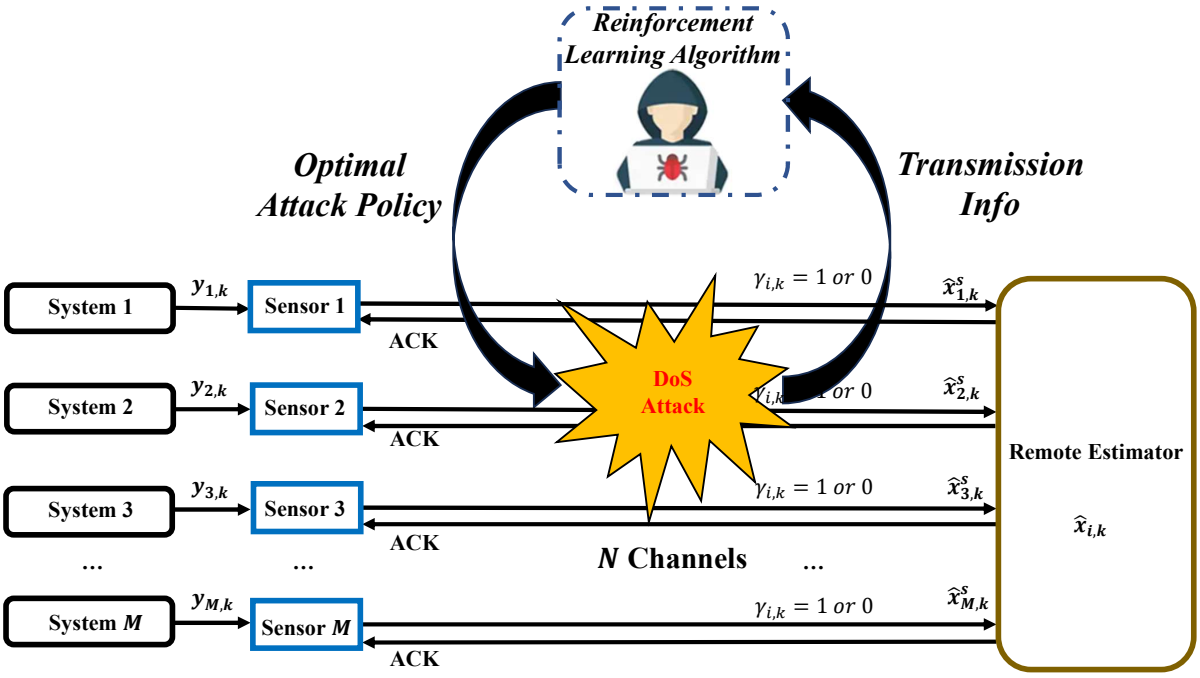


Figure 1: Systems' architecture under wireless packet-drop channels with external attackers.

of the MMSE estimates and the remote estimator is as follows:

$$\hat{x}_{i,k} = \begin{cases} \hat{x}^s_{i,k}, & \text{if } \gamma_{i,k} = 1, \\ A_i \hat{x}_{i,k-1}, & \text{if } \gamma_{i,k} = 0. \end{cases}$$

$$P_{i,k} = \begin{cases} \overline{P}_i, & \text{if } \gamma_{i,k} = 1, \\ h_i(P_{i,k-1}), & \text{if } \gamma_{i,k} = 0, \end{cases}$$

where the function $h_i(\cdot)$ is defined as follows:

$$h_i(X) = A_i X A_i^{\mathrm{T}} + Q_i, \text{ for } X \succeq 0.$$

Specifically, at each discrete time instant and for each system, upon the successful reception of the local estimate, the remote estimator updates its estimate to align with the received local estimate, resetting the corresponding estimation error covariance to the local steady-state estimation error covariance. In contrast, in the event of a packet dropout, the remote estimator produces its current estimate through prediction, utilizing the prior estimate and the system's dynamic model, with the current estimation error covariance being determined as a function of the preceding covariance.

2.2 Attack model

The whole system model is shown in Figure 1. Suppose there exists a DoS attacker who can generate channel noise to affect the communication channel between the sensor and the remote estimator. Due to the attacker's capability limitations, the attacker can at most select $N \in \mathbb{N}$ channels from $M (N < M)$ channels for attack, but the attack can be sustained. Assume that the external attacker has acquired all the knowledge about the system state model and can obtain information from the remote estimator regarding whether the transmission is successful, thereby knowing the effectiveness of the attack. However, this assumption about the attacker is overly absolute and unrealistic, and cannot be fully satisfied. This strong assumption conforms to Kerckhoff's principle [25], which states that the safety of a system should not rely on its obscurity. Let $\lambda_{i,k} \in \{0, 1\}$ indicate whether or not the ith channel is under attack at time k.

Assumption 2. *When considering the possibility of attacks, the process of packet loss in the transmission channel is non-memory-based and conforms to the Markov property, i.e., the following equality holds for any* $k \geq 1$*:*

$$\mathcal{P}(\gamma_{i,1}, \ldots, \gamma_{i,k} \mid \lambda_{i,1:k}) = \prod_{j=1}^{k} \mathcal{P}(\gamma_{i,j} \mid \lambda_{i,j}),$$

where $\lambda_{i,1:k} \triangleq (\lambda_{i,1}, \ldots, \lambda_{i,k})$*. Let* $\mathcal{P}(\gamma_{i,k} = 1 \mid \lambda_{i,k} = 0) = \iota_i$ *and* $\mathcal{P}(\gamma_{i,k} = 1 \mid \lambda_{i,k} = 1) = \underline{\iota}_i$*. Evidently, there is* $0 < \underline{\iota}_i < \iota_i \leq 1$*.*

Remark 1. Knowledge about Assumption 2 based on the relevant knowledge of stochastic process theory [26]. To facilitate subsequent analysis, the effect of channel attacks is simplified here.

At each moment, the attacker makes decisions based on the weights of all the information collected through eavesdropping, and determines the subset of transmission channels to be attacked. Let $\gamma_k = (\gamma_{1,k}, \ldots, \gamma_{M,k})$ and $\gamma_{1:k} = (\gamma_1, \ldots, \gamma_k)$, λ_k and $\lambda_{1:k}$ are defined in the same way. Define a feasible attack attention allocation decision rule at time k as a stochastic kernel π_k from $\gamma_{1:k-1}$ and $\lambda_{1:k-1}$ to Ω, where Ω is the set of all feasible $\lambda_{i,k}$ for all $k \geq 1$, as defined below:

$$\Omega \triangleq \left\{ \lambda_{1,k} \otimes \lambda_{2,k} \otimes \cdots \otimes \lambda_{M,k} \mid \lambda_{i,k} \in \{0,1\} \cap i \in \{1, \ldots, M\} \cap \sum_{i=1}^{M} \lambda_{i,k} \leq N \right\}.$$

Let $\pi = (\pi_1, \ldots, \pi_k, \ldots)$ be the infinite-horizon attack policy. A policy π is feasible only if π_k are feasible. Let Π be the set of all feasible policies.

The reward (from the perspective of the attacker) associated with an attack policy π is the averaged infinite-horizon estimation error at the remote estimator defined as

$$\varlimsup_{T\to\infty} \inf \frac{1}{T} \mathbb{E}\left[\sum_{k=0}^{T-1} \sum_{i=1}^{M} \mathbf{Tr}(P_{i,k}) \right]. \tag{3}$$

Problem 1. The attacker's goal is to seek a feasible attack strategy that maximizes the function (3), which is shown as follows:

$$\sup_{\pi\in\Pi} \varlimsup_{T\to\infty} \inf \frac{1}{T} \mathbb{E}\left[\sum_{k=0}^{T-1} \sum_{i=1}^{M} \mathbf{Tr}(P_{i,k}) \right]. \tag{4}$$

3 Main Results

In this section, we transform Problem 1 into a MDP problem for solution. It is proved that the optimal attack strategy is deterministic (i.e., the stochastic kernel π_k is reduced to a measurable function), stationary (independent of time index k), and in line with the Markov principle, without any performance loss. Additionally, structural results of the optimal attack strategy are presented.

3.1 MDP formulation

Define an equivalent stopping time $\tau_{i,k}$ as

$$\tau_{i,k} \triangleq k - \max\{t \mid 1 \leq t \leq k \cap \gamma_{i,t} = 1\}, \tag{5}$$

which indicates the time duration from the last successful transmission time to time k. Hence, the error covariance $P_{i,k}$ is equivalently computed by $P_{i,k} = h_i^{\tau_{i,k}}(\overline{P}_i)$.

Based on relevant research [4, 5, 17], the following adopts a unified approach. To facilitate the subsequent proof and analysis, in the remaining part, we assume that the number of system models is 2 (i.e., $M = 2$) and the maximum number of attack channels for the attacker is 1 (i.e., $N = 1$). The MDP is described below and the conclusions regarding the existence of deterministic and stable optimal strategies can be simply extended to the general value case. Now we describe the formulated infinite-horizon

discrete-time MDP by a quadruplet $(\mathcal{S}, \mathcal{A}, \mathcal{P}(\cdot \mid \cdot,\cdot), \mathcal{R}(\cdot,\cdot))$. Each item in the tuple is elaborated as follows:

State space: Let $s = (\tau_1, \tau_2) \in \mathcal{S}$. The state at time step $k \geq 1$ is defined as $\mathsf{s}_k \triangleq (\tau_{1,k-1}, \tau_{2,k-1})$, which implies the system's estimation error covariance at last time instant.

Action space: $\mathcal{A} \triangleq \{a_0, a_1, a_2\}$, where $a_0 = (0,0)$ means that none of the systems is attacked, $a_1 = (1,0)$ and $a_2 = (0,1)$ mean that only the first and only the second are attacked, respectively. Let $\mathsf{a}_k \in \mathcal{A}$ denote the choice of action at time k ($k \geq 1$).

Transition probability: The transition probability is stationary. Let $s = (\tau_1, \tau_2)$, $s' = (\tau_1{}', \tau_2{}') \in \mathcal{S}$ with $\tau_i, \tau_i' \in \mathbb{N}$, and $a_{[i]} \in \mathcal{A}$, then for all $k \geq 1$, $i \in \{1,2\}$, we have

$$\begin{aligned}\mathcal{P}(s' \mid s, a) &\triangleq \mathbb{P}(\mathsf{s}_{k+1} = s' \mid \mathsf{s}_k = s, \mathsf{a}_k = a_{[i]}) \\ &= \mathcal{P}_1(\tau_1' \mid \tau_1, a_{[1]})\mathcal{P}_2(\tau_2' \mid \tau_2, a_{[2]}),\end{aligned}$$

where

$$\mathcal{P}_i(\tau_i' \mid \tau_i, a_{[i]}) = \begin{cases} \iota_i, & \text{if } \tau_i' = 0, a_{[i]} = 0, \\ \underline{\iota}_i, & \text{if } \tau_i' = 0, a_{[i]} = 1, \\ 1-\iota_i, & \text{if } \tau_i' = \tau_i + 1, a_{[i]} = 0, \\ 1-\underline{\iota}_i, & \text{if } \tau_i' = \tau_i + 1, a_{[i]} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Cost function: The one-stage reward is independent of the action and defined as

$$\mathcal{R}(s,a) = \mathbb{E}\left[\sum_{i=1}^{2} \mathbf{Tr}(P_{i,k})\right] = \sum_{i=1}^{2} \mathbf{Tr}(\mathbb{E}[h_i^{\tau_{i,k}}(\overline{P}_i)]).$$

Lemma 1 ([5]). *The function $h_i^{t_i}(X)$ is non-decreasing in $t_i \in \mathbb{N}^*$ for any positive semi-definite matrix X, i.e.,*

$$h_i^{t_2}(X) \geq h_i^{t_1}(X),\ \forall t_2 \geq t_1.$$

Let $\mathbb{H}_k \triangleq (\mathsf{s}_0, \mathsf{a}_0, \ldots, \mathsf{s}_k)$ be the history of states and actions up to time k, and $\theta = (\theta_1, \ldots, \theta_k, \ldots)$ be an admissible policy with θ_k as a stochastic kernel from $\mathbb{H}_k$ to $\mathcal{A}$. Let Θ be the class of all such admissible policies. Define the reward associated with initial state $\mathsf{s}_0 = s$, initial action $\mathsf{a}_0 = a$ and policy θ by

$$\mathcal{J}(s,\theta) = \lim_{T\to\infty} \inf \frac{1}{T}\mathbb{E}_s^{\theta}\left[\sum_{k=0}^{T-1} \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k)\right]. \tag{6}$$

Let $\mathsf{s}_{0:k-1} \triangleq (\mathsf{s}_0, \ldots, \mathsf{s}_{k-1})$. It is evident that $\mathsf{s}_{0:k-1}$ is equivalent to $\gamma_{0:k-1}$, and thus θ is also equivalent to π. Problem 1 can be equally transformed to the following problem:

Problem 2. The attacker aim to find the optimal policy $\theta^* \in \Theta$ such that

$$\mathcal{J}(s,\theta^*) = \sup_{\theta\in\Theta} \mathcal{J}(s,\theta). \tag{7}$$

Remark 2. The following discussion and explanation in this article will focus on the case where $M = 2$ and $N = 1$. The relevant lemmas and theorems can naturally be extended to more general situations. Due to the complexity of the proof process, only the simpler cases will be analyzed and explained subsequently.

3.2 Existence of an optimal stationary policy

To find an optimal attack strategy, one must first prove its existence. We then put forward the main theorem in this section. Define a value function as

$$\mathbb{V}_\theta(s) = \lim_{T\to\infty} \mathbb{E}_\theta\left[\sum_{k=0}^{T-1} \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k) \mid \mathsf{s}_0 = s\right],$$

where s_0 is the initial state of the process, s_k is the state after the kth transition, and a_k is the decision made in that state under the policy θ. The long-term average cost for policy θ is:

$$g_\theta(s) = \liminf_{T\to\infty} \frac{1}{T}\mathbb{V}_\theta(s) = \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}_\theta\left[\sum_{k=0}^{T-1} \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k) \mid \mathsf{s}_0 = s\right].$$

The optimal average cost g^* is the supremum over all policies, i.e., $g^* = \sup g_\theta(s)$.

By introducing a discount factor $\alpha \in (0,1)$ in $\mathbb{V}_\theta(s)$, the infinite-horizon discounted cost $\mathbb{V}_\theta^\alpha(s)$ for a policy θ can be described as

$$\mathbb{V}_\theta^\alpha(s) = \lim_{T\to\infty} \mathbb{E}_\theta\left[\sum_{k=0}^{T-1} \alpha^k \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k) \mid \mathsf{s}_0 = s\right] = \mathbb{E}_\theta\left[\sum_{k=0}^{\infty} \alpha^k \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k) \mid \mathsf{s}_0 = s\right].$$

Define the optimal discounted-cost value function is $\mathbb{V}_\alpha^*(s) = \sup_{\theta\in\Theta} \mathbb{V}_\theta^\alpha(s)$.

Remark 3. According to Proposition 1 in [27]. The discounted-cost value function $\mathbb{V}_\alpha^*(s)$ satisfies the Bellman optimality equation and can be found via value iteration.

$$\mathbb{V}_\alpha^*(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \alpha \sum_{s'\in\mathcal{S}} \mathcal{P}(s' \mid s,a) \mathbb{V}_\alpha^*(s') \right\}, \tag{8}$$

where $\alpha \in (0,1)$ is the discount factor. For any iteration $n \geq 0$ of the value iteration algorithm, the value iteration algorithm converges as follows:

$$\mathbb{V}_{\alpha,n+1}^*(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \alpha \sum_{s'\in\mathcal{S}} \mathcal{P}(s' \mid s,a) \mathbb{V}_{\alpha,n}^*(s') \right\},$$
$$\mathbb{V}_{\alpha,0}^*(s) = 0, \lim_{n\to\infty} \mathbb{V}_{\alpha,n}^*(s) = \mathbb{V}_\alpha^*(s).$$

Next, we will introduce some interesting properties of function $\mathbb{V}_\alpha^*(s)$. Before presenting the following lemma, we first define a function $\Phi(\cdot)$. Let $s_1 = (\sigma_1, \sigma_2)$, $s_2 = (\sigma_1', \sigma_2')$, the function $\Phi(\cdot)$ is said to be non-decreasing with respect to s, if $\Phi(s_1) \geq \Phi(s_2)$ holds for all $\sigma_1 \geq \sigma_1'$ and $\sigma_2 \geq \sigma_2'$. Then, define a partial order on $\mathcal{S}$, we say that $s_2 \preceq s_1$ if $\sigma_1' \leq \sigma_1$ and $\sigma_2' \leq \sigma_2$, this partially ordered set is a lattice.

Lemma 2. *The optimal discounted value function $\mathbb{V}_\alpha^*(s)$ is non-decreasing with respect to s, if $\mathbb{V}_\alpha^*(s) \leq \mathbb{V}_\alpha^*(s')$ holds for all $s \preceq s'$, where $s, s' \in \mathcal{S}$.*

Proof. According to Remark 3, we use mathematical induction to prove Lemma 2 [17]. Assume $\mathbb{V}_{\alpha,0}^*(s) = 0$, and $\mathbb{V}_{\alpha,n}^*(s)$ is non-decreasing in s. Based on the above assumption and analysis, we next discuss about $\mathbb{V}_{\alpha,n+1}^*(s)$, it has the following four scenarios:

$$\begin{aligned}
&\text{Case 1: } \hat{s}_1 = (\sigma_1+1, \sigma_2),\ \check{s}_2 = (\sigma_1, \sigma_2),\ a_1 = (1,0),\ a_2 = (0,1)\\
&\text{Case 2: } \hat{s}_1 = (\sigma_1+1, \sigma_2),\ \check{s}_2 = (\sigma_1, \sigma_2),\ a_1 = (0,1),\ a_2 = (1,0)\\
&\text{Case 3: } \hat{s}_1 = (\sigma_1, \sigma_2+1),\ \check{s}_2 = (\sigma_1, \sigma_2),\ a_1 = (1,0),\ a_2 = (0,1)\\
&\text{Case 4: } \hat{s}_1 = (\sigma_1, \sigma_2+1),\ \check{s}_2 = (\sigma_1, \sigma_2),\ a_1 = (0,1),\ a_2 = (1,0)
\end{aligned}$$

Here, consider Case 1, the optimal action for $\hat{s}$ be $a_1 = (1,0)$ and for $\check{s}$ be $a_2 = (0,1)$, we now prove $\mathbb{V}_{\alpha,n+1}^*(\hat{s}) \geq \mathbb{V}_{\alpha,n+1}^*(\check{s})$.

$$\begin{aligned}
\mathbb{V}_{\alpha,n+1}^*(\check{s}) &= \mathcal{R}(\check{s}, a_2) + \alpha\mathbb{E}\left[\mathbb{V}_{\alpha,n}^*(\check{s}') \mid \check{s}, a_2\right]\\
&= \alpha\mathbb{E}\left[\mathbb{V}_{\alpha,n}^*(\check{s}') \mid \check{s}, a_2\right] + \iota_1 \mathbf{Tr}(\overline{P}_1) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) + \underline{\iota}_2 \mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2)).
\end{aligned}$$

Now, we evaluate the value of state $\hat{s}$ under the sub-optimal action a_2. Since a_1 is optimal for $\check{s}$, we have

$$\begin{aligned}
\mathbb{V}_{\alpha,n+1}^*(\hat{s}) &\geq \mathcal{R}(\hat{s}, a_2) + \alpha\mathbb{E}\left[\mathbb{V}_{\alpha,n}^*(\hat{s}') \mid \hat{s}, a_2\right]\\
&= \alpha\mathbb{E}\left[\mathbb{V}_{\alpha,n}^*(\hat{s}') \mid \hat{s}, a_2\right] + \iota_1 \mathbf{Tr}(\overline{P}_1) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1+2}(\overline{P}_1)) + \underline{\iota}_2 \mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2)).
\end{aligned}$$

Let's expand the expectation terms,

$$\begin{aligned}\mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\hat{s}') \mid \hat{s}, a_2\right] &= \mathbb{V}^*_{\alpha,n}(\sigma_1+2,\sigma_2+1)(1-\iota_1)(1-\underline{\iota}_2) + \mathbb{V}^*_{\alpha,n}(0,\sigma_2+1)\iota_1(1-\underline{\iota}_2)\\ &\quad + \mathbb{V}^*_{\alpha,n}(\sigma_1+2,0)(1-\iota_1)\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(0,0)\iota_1\underline{\iota}_2.\\ \mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\check{s}') \mid \check{s}, a_2\right] &= \mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2+1)(1-\iota_1)(1-\underline{\iota}_2) + \mathbb{V}^*_{\alpha,n}(0,\sigma_2+1)\iota_1(1-\underline{\iota}_2)\\ &\quad + \mathbb{V}^*_{\alpha,n}(\sigma_1+1,0)(1-\iota_1)\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(0,0)\iota_1\underline{\iota}_2.\end{aligned}$$

Based on the initial assumption, $\mathbb{V}^*_{\alpha,n}(s)$ is non-decreasing, we have

$$\mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\hat{s}') \mid \hat{s}, a_2\right] \geq \mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\check{s}') \mid \check{s}, a_2\right]. \tag{9}$$

Furthermore, from Lemma 1 on the properties of function h, we know it is non-decreasing in its exponent:

$$\mathbf{Tr}(h_1^{\sigma_1+2}(\overline{P}_1)) \geq \mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) \text{ and } \mathbf{Tr}(h_2^{\sigma_2+2}(\overline{P}_2)) \geq \mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2)).$$

Combining these inequalities, we conclude,

$$\begin{aligned}\mathbb{V}^*_{\alpha,n+1}(\hat{s}) &\geq \alpha\mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\hat{s}') \mid \hat{s}, a_2\right] + \iota_1\mathbf{Tr}(\overline{P}_1) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1+2}(\overline{P}_1)) + \underline{\iota}_2\mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2))\\ &\geq \alpha\mathbb{E}\left[\mathbb{V}^*_{\alpha,n}(\check{s}') \mid \check{s}, a_2\right] + \iota_1\mathbf{Tr}(\overline{P}_1) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) + \underline{\iota}_2\mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2))\\ &= \mathbb{V}^*_{\alpha,n+1}(\check{s}).\end{aligned}$$

The proofs for Case 2, Case 3, and Case 4 are analogous, thus inequality holds for $\mathbb{V}^*_{\alpha,n+1}(s) \leq \mathbb{V}^*_{\alpha,n+1}(s')$ for all $s \preceq s'$. The proof is complete. ■

Lemma 3. *There exists a non-negative Lyapunov function $\Psi(s)$ and a non-negative vector $p = (p_1, p_2)$ that satisfy the following drift conditions:*

For $0 \preceq s \preceq p$:

$$\sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a)\Psi(s') < \infty. \tag{10}$$

For $s \succeq p$:

$$\sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a)\Psi(s') - \Psi(s) + \mathcal{R}(s, a) \leq 0, \tag{11}$$

with the reward function $\mathcal{R}(s, a)$ is strictly positive (i.e., positive and non-zero).

Proof. We define a non-negative Lyapunov function $\Psi(s)$ from [5]:

$$\Psi(s) = \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} + \eta_2 \sum_{k=0}^{\sigma_2} (\rho(A_2))^{2k}, \tag{12}$$

where $\eta_1, \eta_2 \geq 0$ are weighting constants and $\rho(A_i)$ is the spectral radius of matrix A_i. Let $s = (\sigma_1, \sigma_2)$, we analyze the system under a stationary policy $a_1 = (1, 0) \in \mathcal{A}$ and consider two cases for the state s.

Case 1: $0 \preceq s \preceq p$.

$$\begin{aligned}\mathbb{E}[\Psi(s) \mid s, a] &= \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a)\Psi(s')\\ &= (1-\underline{\iota}_1)(1-\iota_2)\Psi(\sigma_1+1,\sigma_2+1) + (1-\underline{\iota}_1)\iota_2\Psi(\sigma_1+1,0) + \underline{\iota}_1(1-\iota_2)\Psi(0,\sigma_2+1) + \underline{\iota}_1\iota_2\Psi(0,0)\\ &= (1-\underline{\iota}_1)(1-\iota_2)[\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + \eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k}] + (1-\underline{\iota}_1)\iota_2[\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + \eta_2]\\ &+ \underline{\iota}_1(1-\iota_2)[\eta_1 + \eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k}] + \underline{\iota}_1\iota_2[\eta_1 + \eta_2] < \infty.\end{aligned}$$

Each term in the expression is a finite sum. Therefore, the expectation is a finite weighted average of finite values, which is itself finite and bounded.

Case 2: $s \succeq p$.

The function is $\mathbb{E}[\Psi(s') \mid s,a] - \Psi(s) + \mathcal{R}(s,a)$. First, It is quite easy to know that the function $\mathcal{R}(s,a)$ is positive. According to [4], assume the system noise covariance satisfy $\overline{P}_i \leq \varphi_i I$, and $Q_i \leq \varphi_i I$ with $i \in \{1,2\}$. Define a function $g_i(X) \triangleq A_i X A_i^T + \varphi_i I$, $i \in \{1,2\}$. One obtains

$$h_i^{\tau_i}(\overline{P}_i) \leq g_i^{\tau_i}(\varphi_i I) \leq \varphi_i \sum_{k=0}^{\tau_i} A_i^k (A_i^T)^k, \ \mathbf{Tr}(h_i^{\tau_i}(\overline{P}_i)) \leq \varphi_i \sum_{k=0}^{\tau_i} (\rho(A_i))^{2k}.$$

Then, we have

$$\begin{aligned}
&\mathbb{E}[\Psi(s') \mid s,a] - \Psi(s) + \mathcal{R}(s,a) = (1-\underline{\iota}_1)(1-\iota_2)\left[\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + \eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k}\right] - \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} \\
&- \eta_2 \sum_{k=0}^{\sigma_2} (\rho(A_2))^{2k} + (1-\underline{\iota}_1)\iota_2 \left[\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + \eta_2\right] + \underline{\iota}_1(1-\iota_2)\left[\eta_1 + \eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k}\right] + \underline{\iota}_1\iota_2[\eta_1+\eta_2] \\
&+ (1-\underline{\iota}_1)\mathbf{Tr}\left(h_1^{\sigma_1}(\overline{P}_1)\right) + \underline{\iota}_1 \mathbf{Tr}(\overline{P}_1) + (1-\iota_2)\mathbf{Tr}\left(h_2^{\sigma_2}(\overline{P}_2)\right) + \iota_2 \mathbf{Tr}(\overline{P}_2) \\
&= (1-\underline{\iota}_1-\iota_2+\underline{\iota}_1\iota_2)\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + (1-\underline{\iota}_1-\iota_2+\underline{\iota}_1\iota_2)\eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k} - \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} - \eta_2 \sum_{k=0}^{\sigma_2} (\rho(A_2))^{2k} \\
&+ (\iota_2 - \underline{\iota}_1\iota_2)\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + (\underline{\iota}_1 - \underline{\iota}_1\iota_2)\eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k} + \iota_2(1-\underline{\iota}_1)\eta_2 + \underline{\iota}_1(1-\iota_2)\eta_1 + \underline{\iota}_1\iota_2[\eta_1+\eta_2] \\
&+ (1-\underline{\iota}_1)\mathbf{Tr}\left(h_1^{\sigma_1}(\overline{P}_1)\right) + \underline{\iota}_1 \mathbf{Tr}(\overline{P}_1) + (1-\iota_2)\mathbf{Tr}\left(h_2^{\sigma_2}(\overline{P}_2)\right) + \iota_2 \mathbf{Tr}(\overline{P}_2) \\
&= (1-\underline{\iota}_1)\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} + (1-\iota_2)\eta_2 \sum_{k=0}^{\sigma_2+1} (\rho(A_2))^{2k} + \underline{\iota}_1\eta_1 + \iota_2\eta_2 - \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} - \eta_2 \sum_{k=0}^{\sigma_2} (\rho(A_2))^{2k} \\
&+ (1-\underline{\iota}_1)\mathbf{Tr}\left(h_1^{\sigma_1}(\overline{P}_1)\right) + \underline{\iota}_1 \mathbf{Tr}(\overline{P}_1) + (1-\iota_2)\mathbf{Tr}\left(h_2^{\sigma_2}(\overline{P}_2)\right) + \iota_2 \mathbf{Tr}(\overline{P}_2).
\end{aligned}$$

After sorting out, only the discussion of σ_1 is considered. The discussion of σ_2 can be derived in the same way [5]. Define $\mathcal{G}$, we have

$$\mathcal{G} = (1-\underline{\iota}_1)\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} - \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} + \underline{\iota}_1\eta_1 + \underline{\iota}_1 \mathbf{Tr}(\overline{P}_1) + (1-\underline{\iota}_1)\mathbf{Tr}\left(h_1^{\sigma_1+1}(\overline{P}_1)\right). \tag{13}$$

The following inequality can be obtained, which is convenient for scaling,

$$\mathbf{Tr}(\overline{P}_1) \leq \varphi_1, \ \mathbf{Tr}\left(h_1^{\sigma_1}(\overline{P}_1)\right) \leq \varphi_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k}.$$

$$\begin{aligned}
\mathcal{G} &\leq \underline{\iota}_1\eta_1 + (1-\underline{\iota}_1)\eta_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} - \eta_1 \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} + \underline{\iota}_1\varphi_1 + (1-\underline{\iota}_1)\varphi_1 \sum_{k=0}^{\sigma_1+1} (\rho(A_1))^{2k} \\
&= \underline{\iota}_1(\eta_1+\varphi_1) + \left[(1-\underline{\iota}_1)(\eta_1+\varphi_1)\rho^2(A_1) - \eta_1\right] \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} + (1-\underline{\iota}_1)(\eta_1+\varphi_1) \\
&= (\eta_1+\varphi_1) + \left[(1-\underline{\iota}_1)(\eta_1+\varphi_1)\rho^2(A_1) - \eta_1\right] \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k}.
\end{aligned}$$

According to [5], we have

$$\rho^{z_1}(A_1) \geq \frac{\eta_1 + (1-\underline{\iota}_1)\varphi_1}{\eta_1 - (1-\underline{\iota}_1)(\eta_1+\varphi_1)\rho^2(A_1)}.$$

Then, one can be derived

$$\mathcal{G} \leq \eta_1 + (1-\underline{\iota}_1)\varphi_1 + \left[(1-\underline{\iota}_1)(\eta_1+\varphi_1)\rho^2(A_1) - \eta_1\right] \sum_{k=0}^{\sigma_1} (\rho(A_1))^{2k} \leq 0.$$

The proof has been completed. ■

Theorem 1. An optimal stationary policy $\theta^* \in \Theta$ exists such that $\mathcal{J}(s,\theta^*) \geq \mathcal{J}(s,\theta)$ for all $s \in \mathcal{S}$ and $\theta \in \Theta$. In addition, the optimal action is determined by

$$a^*(s) = \arg\max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \sum_{s'\in\mathcal{S}} \mathbb{V}^*(s')\mathcal{P}(s' \mid s,a) - g^* \right\},$$

with the optimal average cost $g^* = \mathcal{J}(s,\theta^*)$. The function $\mathbb{V}(\cdot)$ and the optimal cost g^* satisfy the following Bellman equation:

$$\mathbb{V}^*(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \sum_{s'\in\mathcal{S}} \mathbb{V}^*(s')\mathcal{P}(s' \mid s,a) - g^* \right\}. \tag{14}$$

Proof. The proof follows a line of reasoning similar to that adopted in [5, 9, 20, 28]. Because the space $\mathcal{S}$ is countable, and the space $\mathcal{A} = \{a_0, a_1, a_2\}$ is finite, Theorem 1 can be proved by verifying the above Lemma 1 and 2. The arguments in [27] are applied to prove the theorem, we know Lemma 1 and Lemma 2 are satisfied, it can be concluded that the optimal stationary strategy does exist. ■

3.3 Structural results

Lemma 4 ([4]). *Sub-modularity. A function $\varphi(\cdot)$ is sub-modular on $\mathcal{S}$ if*

$$\varphi(\acute{s}) + \varphi(\grave{s}) \geq \varphi(\acute{s} \downarrow \grave{s}) + \varphi(\acute{s} \uparrow \grave{s}), \text{ where } \acute{s},\ \grave{s} \in \mathcal{S}. \tag{15}$$

Theorem 2. Define the function $z_c(\sigma_m, \sigma_n) = l_m(\sigma_n) - \sigma_m$, where $m, n \in \{1,2\}$ and $m \neq n$. There exists a critical curve $z_c(\sigma_1,\sigma_2) = 0$, of which the function $z_c(\sigma_1,\sigma_2)$ is non-decreasing (and non-increasing) with respect to $\sigma_1(\sigma_2)$, dividing $\mathbb{N}^2$ into disjoint regions such that

Given a σ_2, there is a curve $l_1(\sigma_2)$ satisfying

$$a^*(s=(\sigma_1,\sigma_2)) = \begin{cases} a_1, & \text{if } \sigma_1 \geq l_1(\sigma_2), \\ a_2, & \text{if } \sigma_1 < l_1(\sigma_2). \end{cases} \tag{16}$$

Given a σ_1, there is a curve $l_2(\sigma_1)$ satisfying

$$a^*(s=(\sigma_1,\sigma_2)) = \begin{cases} a_2, & \text{if } \sigma_2 \geq l_2(\sigma_1), \\ a_1, & \text{if } \sigma_2 < l_2(\sigma_1). \end{cases} \tag{17}$$

As a consequence, for a fixed σ_1, the optimal action $a^*(s=(\sigma_1,\sigma_2))$ is non-decreasing in σ_2. In other words, as σ_2 increases, the optimal action $a^*(s=(\sigma_1,\sigma_2))$ with a fixed σ_1 switches from a_1 to a_2. Analogously, it can also derive that the optimal action $a^*(s=(\sigma_1,\sigma_2))$ with a fixed σ_2 is non-decreasing in σ_1.

Proof. Based on Theorem 1, there exists an optimal stationary policy, we have

$$\mathbb{V}^*(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \sum_{s'\in\mathcal{S}} \mathbb{V}^*(s')\mathcal{P}(s' \mid s,a) - g^* \right\} = \lim_{\alpha\to 1} [\mathbb{V}^*_\alpha(s) - \mathbb{V}^*_\alpha(s_0)],$$

if $\mathbb{V}^*_\alpha(s_0) = 0$, $\lim_{\alpha\to1}[\mathbb{V}^*_\alpha(s)] = \mathbb{V}^*(s)$. So the nature of $\mathbb{V}^*(s)$ depends on $\mathbb{V}^*_\alpha(s)$, we also know $\lim_{n\to\infty}[\mathbb{V}^*_{\alpha,n}(s) - \mathbb{V}^*_{\alpha,0}(s)] = \mathbb{V}^*_\alpha(s)$, where

$$\mathbb{V}^*_{\alpha,n+1}(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \alpha \sum_{s'\in\mathcal{S}} \mathcal{P}(s' \mid s,a)\mathbb{V}^*_{\alpha,n}(s') \right\}.$$

The structure (monotonicity or submodularity) of $\mathbb{V}^*(s)$ can be proved by showing that $\mathbb{V}^*_\alpha(s)$ has the same structure. By above equation, it suffices to prove that the structure is preserved by the dynamic operator $\mathbb{V}^*_{\alpha,n}(s)$. We consider that the optimal attack strategy $a^*(s)$ maximizes $\mathbb{Q}(s,a)$, $\mathbb{Q}(s,a)$ and $a^*(s)$ are as follows:

$$\mathbb{Q}(s,a) = \mathcal{R}(s,a) + \alpha\mathbb{E}[\mathbb{V}^*_{\alpha,n}(s') \mid s,a],\ a^*(s) = \arg\max_a \mathbb{Q}(s,a).$$

That is, to prove that the function $\mathbb{Q}(s,a)$ is submodular and monotonic. Based on the previous analysis, the monotonicity of function $\mathbb{Q}(s,a)$ can be easily determined. Therefore, no detailed explanation is provided here.

Assume $\overline{s} = (\sigma_1', \sigma_2')$, $\underline{s} = (\sigma_1, \sigma_2)$, $a_2 = (0,1)$, $a_1 = (1,0)$, where $\sigma_1 \le \sigma_1'$,$\sigma_2 \ge \sigma_2'$, we have

$$\begin{aligned}
&\mathbb{Q}(\overline{s},a_1) + \mathbb{Q}(\underline{s},a_2) - \mathbb{Q}((\overline{s}\downarrow\underline{s}),a_1) - \mathbb{Q}((\overline{s}\uparrow\underline{s}),a_2) = \mathcal{R}(\overline{s},a_1) + \mathcal{R}(\underline{s},a_2) - \mathcal{R}((\overline{s}\downarrow\underline{s}),a_1) - \mathcal{R}((\overline{s}\uparrow\underline{s}),a_2)\\
&+ \alpha\mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}') \mid \overline{s},a_1] + \alpha\mathbb{E}[\mathbb{V}^*_{\alpha,n}(\underline{s}') \mid \underline{s},a_2] - \alpha\mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}\downarrow\underline{s})' \mid (\overline{s}\downarrow\underline{s}),a_1] - \alpha\mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}\uparrow\underline{s})' \mid (\overline{s}\uparrow\underline{s}),a_2] \ge 0.
\end{aligned}$$

First part,

$$\begin{aligned}
&\mathcal{R}(\overline{s},a_1) + \mathcal{R}(\underline{s},a_2) - \mathcal{R}((\overline{s}\downarrow\underline{s}),a_1) - \mathcal{R}((\overline{s}\uparrow\underline{s}),a_2)\\
&= (\underline{\iota}_1\mathbf{Tr}(\overline{P}_1) + \iota_2\mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_1)\mathbf{Tr}(h_1^{\sigma_1'+1}(\overline{P}_1)) + (1-\iota_2)\mathbf{Tr}(h_2^{\sigma_2'+1}(\overline{P}_2)))\\
&+ (\iota_1\mathbf{Tr}(\overline{P}_1) + \underline{\iota}_2\mathbf{Tr}(\overline{P}_2) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2)))\\
&- (\underline{\iota}_1\mathbf{Tr}(\overline{P}_1) + \iota_2\mathbf{Tr}(\overline{P}_2) + (1-\underline{\iota}_1)\mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) + (1-\iota_2)\mathbf{Tr}(h_2^{\sigma_2'+1}(\overline{P}_2)))\\
&- (\iota_1\mathbf{Tr}(\overline{P}_1) + \underline{\iota}_2\mathbf{Tr}(\overline{P}_2) + (1-\iota_1)\mathbf{Tr}(h_1^{\sigma_1'+1}(\overline{P}_1)) + (1-\underline{\iota}_2)\mathbf{Tr}(h_2^{\sigma_2+1}(\overline{P}_2)))\\
&= (1-\underline{\iota}_1)(\mathbf{Tr}(h_1^{\sigma_1'+1}(\overline{P}_1)) - \mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1))) + (1-\iota_1)(\mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1)) - \mathbf{Tr}(h_1^{\sigma_1'+1}(\overline{P}_1)))\\
&= (\iota_1-\underline{\iota}_1)(\mathbf{Tr}(h_1^{\sigma_1'+1}(\overline{P}_1)) - \mathbf{Tr}(h_1^{\sigma_1+1}(\overline{P}_1))) \ge 0.
\end{aligned}$$

Second part, let $\epsilon_1 = (1-\underline{\iota}_1)(1-\iota_2)$, $\epsilon_2 = (1-\iota_1)(1-\underline{\iota}_2)$ and $\epsilon_3 = (1-\underline{\iota}_1)\iota_2 - (1-\iota_1)\underline{\iota}_2$, assume $\epsilon_1 \ge \epsilon_2$, we have

$$\begin{aligned}
&\mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}') \mid \overline{s},a_1] + \mathbb{E}[\mathbb{V}^*_{\alpha,n}(\underline{s}') \mid \underline{s},a_2] - \mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}\downarrow\underline{s})' \mid (\overline{s}\downarrow\underline{s}),a_1] - \mathbb{E}[\mathbb{V}^*_{\alpha,n}(\overline{s}\uparrow\underline{s})' \mid (\overline{s}\uparrow\underline{s}),a_2]\\
&= (\mathbb{V}^*_{\alpha,n}(0,0)\underline{\iota}_1\iota_2 + \mathbb{V}^*_{\alpha,n}(0,\sigma_2'+1)\underline{\iota}_1(1-\iota_2) + \mathbb{V}^*_{\alpha,n}(\sigma_1'+1,0)(1-\underline{\iota}_1)\iota_2 + \mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2'+1)(1-\underline{\iota}_1)(1-\iota_2))\\
&+ (\mathbb{V}^*_{\alpha,n}(0,0)\iota_1\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(0,\sigma_2+1)\iota_1(1-\underline{\iota}_2) + \mathbb{V}^*_{\alpha,n}(\sigma_1+1,0)(1-\iota_1)\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2+1)(1-\iota_1)(1-\underline{\iota}_2))\\
&- (\mathbb{V}^*_{\alpha,n}(0,0)\underline{\iota}_1\iota_2 + \mathbb{V}^*_{\alpha,n}(0,\sigma_2'+1)\underline{\iota}_1(1-\iota_2) + \mathbb{V}_{\alpha,n}(\sigma_1+1,0)(1-\underline{\iota}_1)\iota_2 + \mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2'+1)(1-\underline{\iota}_1)(1-\iota_2))\\
&- (\mathbb{V}^*_{\alpha,n}(0,0)\iota_1\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(0,\sigma_2+1)\iota_1(1-\underline{\iota}_2) + \mathbb{V}^*_{\alpha,n}(\sigma_1'+1,0)(1-\iota_1)\underline{\iota}_2 + \mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2+1)(1-\iota_1)(1-\underline{\iota}_2))\\
&= \epsilon_2(\mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2+1) - \mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2+1)) + \epsilon_1(\mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2'+1) - \mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2'+1))\\
&+ \epsilon_3(\mathbb{V}^*_{\alpha,n}(\sigma_1'+1,0) - \mathbb{V}^*_{\alpha,n}(\sigma_1+1,0))\\
&\ge \epsilon_1\mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2+1) + \epsilon_1\mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2'+1) - \epsilon_1\mathbb{V}^*_{\alpha,n}(\sigma_1+1,\sigma_2'+1) - \epsilon_1\mathbb{V}^*_{\alpha,n}(\sigma_1'+1,\sigma_2+1) \ge 0.
\end{aligned}$$

The proof demonstrating that the function $\mathbb{Q}(s,a)$ is submodular has been completed. Due to the influence of the marginal effect, it has been proven that function $\mathbb{Q}(s,a)$ is submodular, when σ_1 (σ_2) is fixed, it is very easy to test the monotonicity of the other structure by using the submodularity property. The proof process for other situations is the same as above and will not be elaborated further. Finally, the structure of the optimal attack strategy conforms to the form of Theorem 2. ■

Remark 4. Based on the threshold structure and symmetry discovered in the homogeneous model, this strategy is the optimal deterministic steady-state strategy for the general (M, N) problem. Because it greedily maximizes the one-step reward, which in this case is equivalent to maximizing the growth of the estimated error. The threshold structure of the strategy for the heterogeneous model is also shown in the subsequent simulation.

The threshold structure can be extended to cases with general M and N [4]. For $1 \le i \le M$, define $\tau_i^- \triangleq (\tau_1, \ldots, \tau_{i-1}, \tau_{i+1}, \ldots, \tau_M)$ as the state of the whole system except for the ith system. Then the

optimal policy has the following threshold structure. Let state $s = (\tau_1, \ldots, \tau_M)$, there exist measurable functions l_i such that for any $1 \leq i \leq M$, the optimal attack policy a^* has the form:

$$\begin{cases} a^*(s) \in \Xi_i, \ if \ \tau_i \geq l_i(\tau_i^-), \\ a^*(s) \in \mathbb{U} \setminus \Xi_i, \ if \ \tau_i < l_i(\tau_i^-). \end{cases}$$

where Ξ_i represents the feasible attack attention allocation subset such that the ith system is under attack:

$$\Xi_i \triangleq \left\{ \lambda \in \{0,1\}^M : \sum_{i=1}^{M} \lambda_i \leq N, \lambda_i = 1 \right\}.$$

3.4 DRL for attack scheduling

Algorithm 1 Dueling Double DQN-based Optimal Attack Scheduling

Initialize:
- Dueling Q-network $\mathbb{Q}(s, a; \theta)$ with value stream $\mathbb{V}(s)$ and advantage stream $\mathbb{A}(s, a)$
- Target Q-network $\hat{\mathbb{Q}}(s, a; \theta^-)$ with weights $\theta^- \leftarrow \theta$
- Experience replay buffer $\mathcal{D}$
- Set hyperparameters: learning rate ϑ, discount factor α, batch size B, exploration rate ϵ, target update frequency C

for episode $m = 1, 2, \ldots, M$ **do**
  Initialize state $s_1 \leftarrow \{\tau_1 = 0, \tau_2 = 0\}$
  for $t = 1, 2, \ldots, T$ **do**
    Select action a_t using an ϵ-greedy policy based on $\mathbb{Q}(s_t, a; \theta)$
    Execute action a_t, observe reward r_t and next state s_{t+1}
    Store transition $(s_t, a_t, r_t, s_{t+1}, \text{done})$ in $\mathcal{D}$
    Sample a minibatch of B transitions $(s_j, a_j, r_j, s_{j+1}, \text{done}_j)$ from $\mathcal{D}$
    for each transition j in the minibatch **do**
      Double DQN target calculation
      Find best action in the next state using the main network:
      $a'_{\max} \leftarrow \arg\max_{a'} \mathbb{Q}(s_{j+1}, a'; \theta)$
      Evaluate this action using the target network:
      $y_j \leftarrow r_j + \alpha \cdot \hat{\mathbb{Q}}(s_{j+1}, a'_{\max}; \theta^-) \cdot (1 - \text{done}_j)$
    end for
    Compute loss: $L = \frac{1}{B} \sum_j (y_j - \mathbb{Q}(s_j, a_j; \theta))^2$
    Perform a gradient descent step on L with respect to θ
    Clip gradients: $\|\nabla_\theta L\| \leq 10$
    Update state: $s_t \leftarrow s_{t+1}$
  end for
  Decay exploration rate ϵ
  if $m \pmod C = 0$ **then**
    Update target network weights: $\theta^- \leftarrow \theta$
  end if
end for

Commonly, reinforcement learning algorithms based on models update the Q-table continuously, eventually learning an optimal strategy. However, due to the curse of dimensionality, updating only on the Q-table is difficult to handle large-dimensional and complex state spaces. DRL methods, especially DQN, have addressed this limitation by using a neural network-based function approximator. In this section, we employ the improved DQN algorithm(i.e., D3QN), to solve the optimal attack scheduling problem. It only makes minor modifications to the traditional DQN and yet can significantly enhance the performance of DQN. The specific design of the neural network structure is shown in Figure 2. In order to apply the DRL framework, introduce a discount factor into Problem 2 and consider the problem of maximizing the total discounted reward.

$$\mathcal{J}_\alpha(s, \theta^*) = \max_{\theta \in \Theta} \mathcal{J}_\alpha(s, \theta), \tag{18}$$

where $\alpha \in (0,1)$ is a discount factor, and

$$\mathcal{J}_\alpha(s,\theta) \triangleq \liminf_{T\to\infty} \mathbb{E}_\theta \left[\sum_{k=0}^{T-1} \alpha^k \mathcal{R}(\mathsf{s}_k, \mathsf{a}_k) \mid \mathsf{s}_0 = s \right],$$

with $\mathcal{J}^*_\alpha(s) \triangleq \mathcal{J}_\alpha(s,\theta^*)$, which satisfies the following Bellman optimality equation:

$$\mathcal{J}^*_\alpha(s) = \max_{a\in\mathcal{A}} \left\{ \mathcal{R}(s,a) + \alpha \sum_{s'\in\mathcal{S}} \mathcal{J}^*_\alpha(s')\mathcal{P}(s' \mid s,a) \right\}. \tag{19}$$

The action-value function (Q-factor) is defined as:

$$\mathbb{Q}(s,a) \triangleq \mathbb{E}[\mathcal{R}(s,a) + \alpha \max_{a'} \mathbb{Q}^*(s',a') \mid s,a],$$

where s' and a' represent the next state and action respectively, with $\mathbb{Q}^*(s,a)$ denoting the optimal Q-factor. The optimal action policy is derived as $a^*(s) = \arg\max_a \mathbb{Q}^*(s,a)$. DRL aims to estimate the optimal Q-function $\mathbb{Q}^*(s,a)$ utilizing a dueling deep neural network $\mathbb{Q}(s,a;\theta)$. The parameters θ are updated by the gradient descent algorithm, and the loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{s,a,\mathcal{R},s'} \left[(y - \mathbb{Q}(s,a;\theta))^2 \right],$$

where the target value is computed as $y = \mathcal{R} + \alpha \max_{a'} \mathbb{Q}(s',a';\tilde{\theta})$, with $\mathbb{Q}(s',a';\tilde{\theta})$ representing the target Q-network. The parameters $\tilde{\theta}$ of this target network are periodically synchronized with the main network parameters θ at regular intervals. By using a separate target Q-network to compute y, the algorithm reduces the correlation between the target values and the training Q-network $\mathbb{Q}(s,a;\theta)$, thereby improving training stability. Furthermore, experience replay is employed to minimize correlations within the training data. Transition tuples $(s,a,\mathcal{R},s')$ are stored in a replay memory buffer, and at each iteration, a mini-batch of transitions is sampled uniformly from this buffer to compute the loss function. The complete D3QN algorithm for attack scheduling is shown in Algorithm 1.

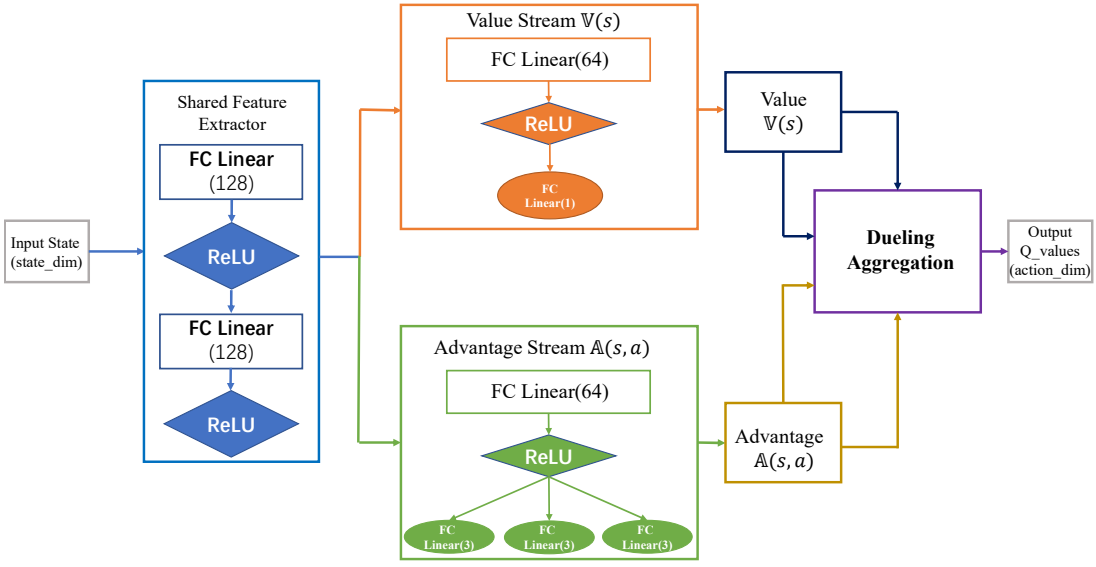


Figure 2: Internal structure diagram of neural network.

4 Illustrative Examples

In this section, we use the D3QN algorithm based on the structure of Figure 2 on several specific numerical examples to illustrate the threshold structure of the optimal attack strategy and prove the feasibility and

effectiveness of the optimal attack strategy. Then, we compare the impact of D3QN with that of other attack strategies to demonstrate that the learned attack strategy is optimal.

4.1 Demonstration of Attack Architectures for IIoT-Based Chemical Reactor Monitoring Systems

To demonstrate the feasibility and effectiveness of the proposed D3QN-based attack scheduling strategy, we simulate an Industrial Internet of Things (IIoT) scenario consisting of M spatially distributed chemical reaction processes. Specifically, each subsystem represents the linearized discrete-time dynamics of a Continuous Stirred Tank Reactor (CSTR) operating near an unstable equilibrium point. In this physical setup:

The state vector $x_{i,k} = [x_{i,k}^{(1)}, x_{i,k}^{(2)}]^T \in \mathbb{R}^2$ denotes the deviations of the reactor temperature and reactant concentration, respectively, from their nominal operating values at time step k. The system matrix parameters reflect the chemical reaction kinetics and heat transfer characteristics. The parameter ξ in matrix A_i represents the instability factor caused by the exothermic nature of the reaction. As shown in the parameters below, $\xi > 1$ indicates that the process is open-loop unstable, necessitating reliable remote estimation to detect potential thermal runaways. Each smart sensor i makes a scalar observation $y_{i,k}$, where C_i representing a fused sensor reading, and transmits the local estimate to the remote estimator via a wireless fading channel. The DoS attacker, limited by energy resources (e.g., a battery-powered jamming device), can only target N channels at any given time instant to disrupt the remote monitoring capabilities.

4.1.1 *Scenario One* : $M = 2, N = 1$

The specific system parameters are defined as follows:

$$A_1 = \begin{bmatrix} \xi & 0.7 \\ 0 & 0.2 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_1 = [1, 1], \quad R_1 = 1,$$
$$A_2 = \begin{bmatrix} \xi & 0.7 \\ 0 & 0.2 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_2 = [1, 1], \quad R_2 = 1.$$

We impose restrictions on the state space such that $0 \leq \tau_1 \leq 20$ and $0 \leq \tau_2 \leq 20$. The action space is $\mathcal{A} = \{(0, 0), (1, 0), (0, 1)\}$. The hyperparameters set before the training of D3QN are shown in Table 1. The optimal attack policies of the MDP with various system parameters are depicted in Figure 3.

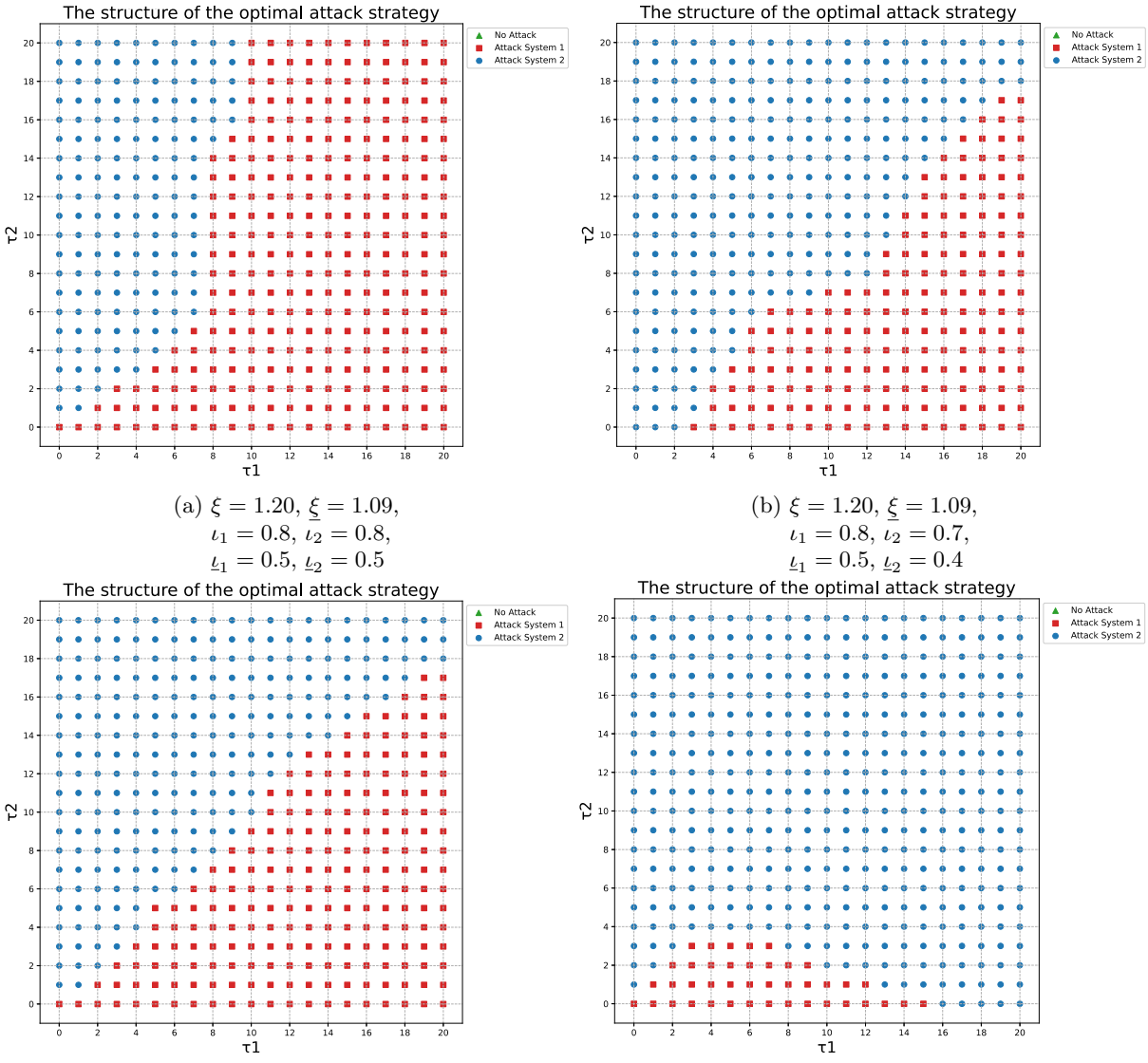
It can be clearly seen that there is a dividing line for the optimal attack strategy, which conforms to the strategy structure proposed in Theorem 2. Moreover, due to the changes in the environmental transmission probability, it will also have an impact on the strategy structure, causing "bias" in the learning process. However, because the size of the state space grows exponentially with respect to M, the required memory will exceed our capacity.

Table 1: D3QN agent settings.

Parameter	**Value**
Agent Initialization	
Learning rate	5×10^{-4}
Discount factor	0.99
Initial expl. rate	1.0
Final expl. rate	0.01
Expl. rate decay	0.9975
Replay buffer	10000
Training Process	
Episodes	2000
Batch size	128
Update freq.	10
Max steps	150

Table 2: D3QN agent settings.

Parameter	**Value**
Agent Initialization	
Learning rate	5×10^{-4}
Discount factor	0.99
Initial expl. rate	1.0
Final expl. rate	0.01
Expl. rate decay	0.9975
Replay buffer	10000
Training Process	
Episodes	3000
Batch size	128
Update freq.	10
Max steps	100



(a) $\xi = 1.20$, $\underline{\xi} = 1.09$, $\iota_1 = 0.8$, $\iota_2 = 0.8$, $\underline{\iota}_1 = 0.5$, $\underline{\iota}_2 = 0.5$

(b) $\xi = 1.20$, $\underline{\xi} = 1.09$, $\iota_1 = 0.8$, $\iota_2 = 0.7$, $\underline{\iota}_1 = 0.5$, $\underline{\iota}_2 = 0.4$

(c) $\xi = 1.20$, $\underline{\xi} = 1.20$, $\iota_1 = 0.8$, $\iota_2 = 0.8$, $\underline{\iota}_1 = 0.5$, $\underline{\iota}_2 = 0.5$

(d) $\xi = 1.20$, $\underline{\xi} = 1.20$, $\iota_1 = 0.8$, $\iota_2 = 0.7$, $\underline{\iota}_1 = 0.5$, $\underline{\iota}_2 = 0.5$

Figure 3: Optimal attack policies with different system parameters. Blue circles, red squares and green triangles correspond to the action (0,1), (1,0), and (0,0), respectively.

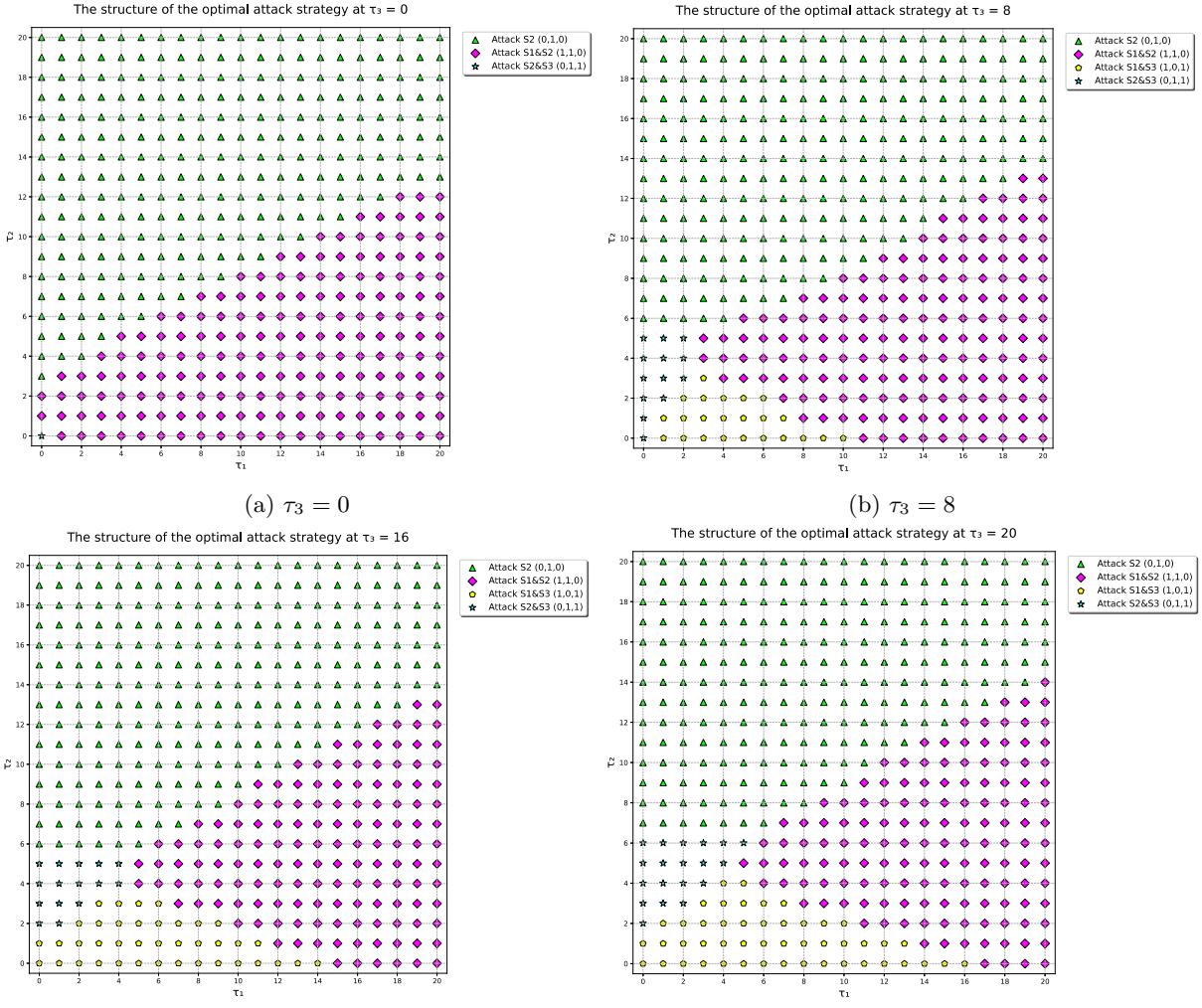
4.1.2 *Scenario Two : $M = 3, N = 2$*

The specific system parameters are defined as follows:

$$A_1 = \begin{bmatrix} 1.09 & 0.7 \\ 0 & 0.2 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_1 = [1, 1], \quad R_1 = 1,$$
$$A_2 = \begin{bmatrix} 1.09 & 0.7 \\ 0 & 0.2 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_2 = [1, 1], \quad R_2 = 1,$$
$$A_3 = \begin{bmatrix} 1.09 & 0.7 \\ 0 & 0.2 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_3 = [1, 1], \quad R_3 = 1.$$

Due to the limitations of the dimensional space and the complexity of the results, certain restrictions will be imposed on the presentation of the subsequent attack strategy structure. With $0 \leq$

$\tau_1 \leq 20$, $0 \leq \tau_2 \leq 20$, τ_3 will be assigned the values of 0, 8, 16, and 20. The action space is $\mathcal{A} = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1)\}$. The hyperparameters set before the training of D3QN are shown in Table 2. The optimal attack policies of the MDP with various system stopping time τ_3 are depicted in Figure 4.



(a) $\tau_3 = 0$ (b) $\tau_3 = 8$

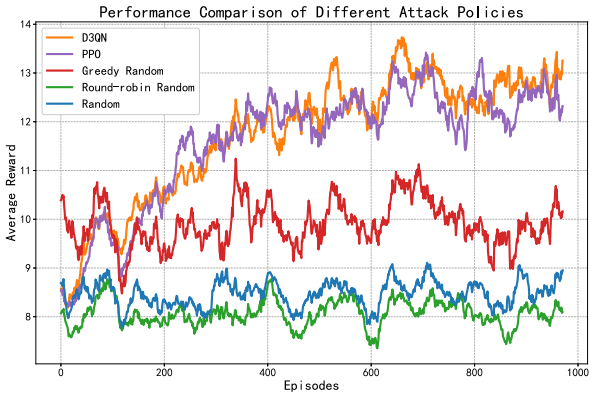
(c) $\tau_3 = 16$ (d) $\tau_3 = 20$

Figure 4: Optimal attack policies with different τ_3, different symbols represent different types of attacks respectively.

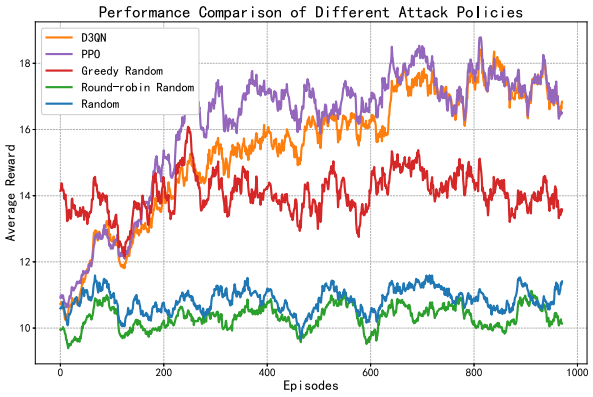
4.2 Performance Comparison of Different Attack Policies

We will consider the average reward under random algorithms, greedy algorithms, round-robin scheduling algorithms, D3QN algorithms and proximal policy optimization (PPO) algorithms.

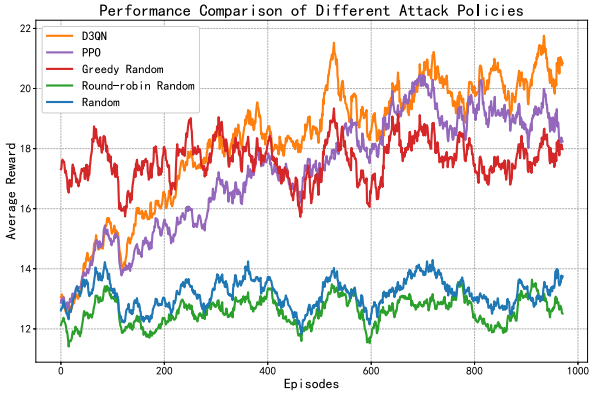
Due to the specific nature of the environment setup and the problems being considered, only the average rewards obtained by different algorithms under 1000 episodes are considered. As shown in Figure 5, due to the packet loss setting of the channel, the training process is constantly fluctuating. It can be clearly seen that the random algorithm, the round-robin algorithm, and the greedy algorithm fluctuate within a certain range, while the two algorithms using reinforcement learning, PPO and D3QN, cause the average rewards to continuously increase and ultimately do not converge. It can be concluded that using reinforcement learning algorithms is more feasible and effective compared to traditional algorithms.



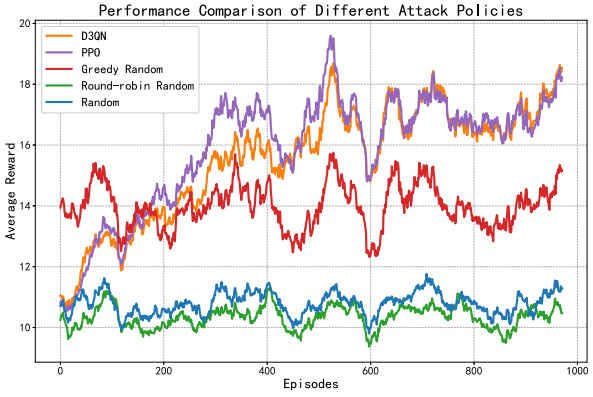
(a) $\iota_1 = \iota_2 = 0.8$, $\underline{\iota}_1 = \underline{\iota}_2 = 0.5$



(b) $\iota_1 = 0.8$, $\iota_2 = 0.7$, $\underline{\iota}_1 = 0.5$, $\underline{\iota}_2 = 0.4$



(c) $\iota_1 = \iota_2 = 0.7$, $\underline{\iota}_1 = \underline{\iota}_2 = 0.4$



(d) $\iota_1 = 0.7$, $\iota_2 = 0.8$, $\underline{\iota}_1 = 0.4$, $\underline{\iota}_2 = 0.5$

Figure 5: The experimental results of the average rewards under different attack strategies, with the entire process consisting of 1000 episodes.

Algorithm	Case 1	Case 2	Case 3	Case 4
D3QN	13.02	17.01	20.59	17.82
PPO	12.27	17.03	18.78	17.47
Greedy Random	9.99	13.85	17.94	14.59
Round-robin Random	8.10	10.47	12.86	10.49
Random	8.73	11.07	13.41	11.07

Table 3: Performance comparison of algorithms under four different scenarios.

Case 1:	$\iota_1 = \iota_2 = 0.8,$	$\underline{\iota}_1 = \underline{\iota}_2 = 0.5$
Case 2:	$\iota_1 = 0.8, \iota_2 = 0.7,$	$\underline{\iota}_1 = 0.5, \underline{\iota}_2 = 0.4$
Case 3:	$\iota_1 = \iota_2 = 0.7,$	$\underline{\iota}_1 = \underline{\iota}_2 = 0.4$
Case 4:	$\iota_1 = 0.7, \iota_2 = 0.8,$	$\underline{\iota}_1 = 0.4, \underline{\iota}_2 = 0.5$

The total average rewards are shown in Table 3. It can be clearly seen that the average reward obtained by using the reinforcement learning algorithm is greater than that obtained by using traditional algorithms such as random, greedy, and round-robin algorithms. Moreover, due to the variation of transmission probability, the average reward under different transmission probabilities will also fluctuate. Finally, the average rewards obtained by the PPO algorithm and the D3QN algorithm are very close.

5 Conclusion

The problem of external attack allocation in multi-systems state estimation transmission has been investigated, and the problem has been formulated as an MDP and solved through an optimal deterministic stationary policy. The threshold structure of the optimal policy has been proved, by which the computational complexity of the optimization problem is significantly reduced. To address the curse of dimensionality, the D3QN algorithm has been employed, allowing the complex computation process to be greatly simplified. Numerical simulations have been conducted and the computational results have been validated. Future work will focus on how the optimal strategy can be better identified based on the strategic structure to make the search process more convenient and efficient, and will further examine the factors influencing the optimal strategy structure in heterogeneous models while exploring attack scheduling strategies in unknown environments, particularly in scenarios more closely aligned with practical applications.

Acknowledgments

This article expresses our sincere gratitude to all concerned parties.

Funding

This article is not supported by any related funds.

Conflicts of interest

The authors declare no conflicts of interest.

Data availability statement

No data are associated with this article.

Author contribution statement

Shunpeng Zhang:	Conceptualisation(primary), Methodology(major), Validation(lead), Writing - Original Draft(lead), Writing - Review & Editing(major).
Zhitao Fan:	Resources(lead), Supervision(supporting).
Xingquan Fu:	Conceptualisation(supporting), Investigation(lead), Formal Analysis(lead), Data Curation(major), Writing - Review & Editing(major).

References

[1] Alex S Leong, Arunselvan Ramaswamy, Daniel E Quevedo, Holger Karl, and Ling Shi. Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems. *Automatica*, 113:108759, 2020.

[2] Sheng Liu and HongMei Zhang. Theory and application of optimal state estimation, 2011.

[3] Kemi Ding, Yuzhe Li, Daniel E Quevedo, Subhrakanti Dey, and Ling Shi. A multi-channel transmission schedule for remote state estimation under dos attacks. *Automatica*, 78:194–201, 2017.

[4] Xiaoqiang Ren, Junfeng Wu, Subhrakanti Dey, and Ling Shi. Attack allocation on remote state estimation in multi-systems: Structural results and asymptotic solution. *Automatica*, 87:184–194, 2018.

[5] Lixin Yang, Weijun Lv, Yong Xu, Jie Tao, and Daniel E Quevedo. Structure-aware reinforcement learning for optimal transmission scheduling over packet length-dependent lossy networks. *IEEE Transactions on Automatic Control*, 2024.

[6] Dong Wang, Peilin Jia, Jie Lian, and Xinyu Pei. An optimal dos attack strategy with pause and restart rules under energy constraints. *IEEE Transactions on Control of Network Systems*, 10(3):1291–1302, 2022.

[7] Heng Zhang, Yifei Qi, Junfeng Wu, Lingkun Fu, and Lidong He. Dos attack energy management against remote state estimation. *IEEE Transactions on Control of Network Systems*, 5(1):383–394, 2016.

[8] Jiahu Qin, Menglin Li, Jie Wang, Ling Shi, Yu Kang, and Wei Xing Zheng. Optimal denial-of-service attack energy management against state estimation over an sinr-based network. *Automatica*, 119:109090, 2020.

[9] Lixin Yang, Jie Tao, Yong-Hua Liu, Yong Xu, and Chun-Yi Su. Energy scheduling for dos attack over multi-hop networks: Deep reinforcement learning approach. *Neural Networks*, 161:735–745, 2023.

[10] Junhui Zhang, Jitao Sun, and Hai Lin. Optimal dos attack schedules on remote state estimation under multi-sensor round-robin protocol. *Automatica*, 127:109517, 2021.

[11] Xianping Guo and Quanxin Zhu. Average optimality for markov decision processes in borel spaces: a new condition and approach. *Journal of Applied Probability*, 43(2):318–334, 2006.

[12] Lixin Yang, Yong Xu, Zenghong Huang, Hongxia Rao, and Daniel E Quevedo. Learning optimal stochastic sensor scheduling for remote estimation with channel capacity constraint. *IEEE Transactions on Industrial Informatics*, 19(3):2565–2573, 2022.

[13] Shuang Wu, Xiaoqiang Ren, Subhrakanti Dey, and Ling Shi. Optimal scheduling of multiple sensors over shared channels with packet transmission constraint. *Automatica*, 96:22–31, 2018.

[14] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
[16] Guannan Qu. Structure-exploiting reinforcement learning for networked systems. In *2025 American Control Conference (ACC)*, pages 4776–4784. IEEE, 2025.
[17] Yong Xu, Haoxiang Xiang, Lixin Yang, Renquan Lu, and Daniel E Quevedo. Optimal transmission strategy for multiple markovian fading channels: Existence, structure, and approximation. *Automatica*, 158:111312, 2023.
[18] Zhitao Fan, Xingquan Fu, and Guanghui Wen. Deep reinforcement learning-based energy-efficient sensor scheduling for remote state estimation in islanded microgrids. In *2025 IEEE 34th International Symposium on Industrial Electronics (ISIE)*, pages 1–7. IEEE, 2025.
[19] Lixin Yang, Yong Xu, Weijun Lv, Jun-Yi Li, and Ling Shi. Optimal transmission scheduling over multihop networks: Structural results and reinforcement learning. *IEEE Transactions on Automatic Control*, 69(3):1826–1833, 2023.
[20] Yijin Jia, Lixin Yang, Yao Zhao, Jun-Yi Li, and Weijun Lv. Optimal redundant transmission scheduling for remote state estimation via reinforcement learning approach. *Neurocomputing*, 576:127337, 2024.
[21] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
[22] Guanghui Wen, Wenwu Yu, Xinghuo Yu, and Jinhu Lü. Complex cyber-physical networks: From cybersecurity to security control. *Journal of Systems Science and Complexity*, 30(1):46–67, 2017.
[23] Guanghui Wen, Wenwu Yu, Yuezu Lv, and Peijun Wang. *Cooperative control of complex network systems with dynamic topologies*. CRC Press, 2021.
[24] B Anderson and J Moore. Optimal filtering (dover books on electrical engineering), 2012.
[25] Auguste Kerckhoffs. La cryptographie militaire. *J. Sci. Militaires*, 9(4):5–38, 1883.
[26] Zeev Schuss. *Theory and applications of stochastic processes: an analytical approach*, volume 170. Springer Science & Business Media, 2009.
[27] Linn I Sennott. Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs. *Operations Research*, 37(4):626–633, 1989.
[28] Shuang Wu, Xiaoqiang Ren, Subhrakanti Dey, and Ling Shi. Optimal scheduling of multiple sensors over shared channels with packet transmission constraint. *Automatica*, 96:22–31, 2018.