Bayesian Two-stage Sequential Change Diagnosis Via Sensor Arrays

Xiaochuan Ma, Lifeng Lai, and Shuguang Cui

Abstract

In this paper, we formulate and solve a two-stage Bayesian sequential change diagnosis (SCD) problem in a multi-sensor setting. In the considered problem, a change first occurs at a sensor and then propagates across the sensor array gradually. After a change is detected, we are allowed to continue observing more samples so that we can identify the distribution after the change more accurately. Our goal is to minimize the total cost including delay, false alarm, and misdiagnosis probabilities. We first characterize the optimal SCD rule. Moreover, to address the high computational complexity issue of the optimal SCD rule, we propose a low-complexity threshold SCD rule. We further analyze the asymptotic optimality of the threshold SCD rule. In addition, we investigate how increasing the number of sensors can improve the performance of the proposed threshold SCD rule. Our analysis holds for different sensor array structures, including linear sensor arrays and 2D lattice sensor arrays.

I. Introduction

SEQUENTIAL change diagnosis (SCD) is a joint problem of the quickest change-point detection (QCD) problem [2]–[13] and sequential multiple hypothesis testing (SMHT) problem [14]–[17]. In SCD problems, the data distribution will change from f_0 to one of the I candidate distributions at an unknown time. The goal of the SCD problem is to detect the change point quickly and identify the post-change distribution accurately. The SCD problem has many applications, including intrusion detection in computer networks [18], outage detection and identification in power system [19], and dynamic spectrum access and allocation [20], etc.

In the one-stage SCD problem studied in [21], [22], the change detection and identification must be made at the same time. For the detection task, the goal is to detect the change quickly. On the other hand, spending more time to collect more data samples can increase the accuracy of identification. Therefore, the requirement that the detection and identification must occur at the same time creates a tension between the detection and identification objectives. In practice, however, after we detect the change, we may still have the opportunity to observe extra data samples, which may help us to make a more accurate identification decision. In other words, the change detection and identification do not have to occur at the same time. This extra degree of freedom provides opportunities to design new detection and identification schemes to achieve better performance. In our recent work [23], [24], we formulate such problems as two-stage SCD problems in Bayesian setting, provide the optimal and asumptotically optimal solutions, and prove that the two-stage SCD rule outperforms the one-stage SCD rule when the detection task is more urgent than the identification task. The two-stage SCD problems may arise in many applications. For example, in the structural health monitoring (SHM) system [25], sensors are used to monitor a building. When the building experiences a sudden damage, the SHM system should detect the damage quickly and identify the type of damage accurately. Typically, the identification task requires more data than damage detection, i.e., more time is needed for damage identification than damage detection. However, the detection task is very urgent because people in the building can be in great danger. Therefore, a smart SHM system

X. Ma, L. Lai are with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616. Email:{xcma, lflai}@ucdavis.edu. S. Cui is currently with the School of Science and Engineering (SSE) and Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, and Shenzhen Research Institute of Big Data, Shenzhen, China, 518172 (e-mail: shuguangcui@cuhk.edu.cn). This work of X. Ma and L. Lai was supported in part by the National Science Foundation under Grants CNS-1824553, ECCS-2000415 and CCF-2112504. The work of S. Cui was supported in part by Shenzhen Outstanding Talents Training Fund 202002, and by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104. This paper has been presented in part at 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), Gold Coast, Australia, Oct. 2021 [1].

should allow the identification decision to be made after the damage detection. In this case, the people can be evacuated from the building immediately once the damage is detected. After that, more data can be collected to make an accurate damage identification. Other examples of such two-stage situations include diagnosis of intrusions in computer networks [26], industrial systems monitoring [27] etc.

In our recent work [23], [24], we focus on the case with a single sensor. To further improve the performance, one could employ multiple sensors that collect information and send it to the fusion center, where the detection and identification decision is made. In this paper, we consider a two-stage SCD problem in the multi-sensor scenario where there are multiple sensors monitoring the environment. A change will happen to the environment at an unknown time. At the change point, the distribution of the observed signal changes from f_0 to one of the I candidate distributions. After the change happens, it will propagate across the sensor array. Using the observed information collected by the sensors, the fusion center needs to decide two stopping times and one identification decision. At the first stopping time, the fusion center raises an alarm that a change has been detected. After that, the sensors can still collect extra observations to facilitate the identification task. At the second stopping time, the fusion center makes the identification decision. It's worth noting that the detection and identification stages are not independent, as the end state of the detection stage is the start state of the identification stage. Hence the proposed problem is not a simple combination of a QCD problem and a SMHT problem.

For the proposed problem, we characterize the structure of the optimal diagnosis rule. The optimal stopping rule is obtained by converting the two-stage SCD problem into a two-ordered optimal stopping time problem, which can be solved using dynamic programming (DP). However, the dimension of the state space grows exponentially with the number of sensors and candidate post-change distributions. Thus the complexity to implement the DP solution is extremely high. To address this issue, we propose a low complexity threshold SCD rule. Furthermore, we analyze the performance of the proposed multi-sensor threshold SCD rule in two different linear array cases depending on whether the sensor first affected by the change is known or not. Concretely, for the general case in which the sensor first being affected by the change is randomly chosen and unknown, we prove the threshold rule is asymptotically optimal under some technical conditions. On the other hand, for the special case in which the sensor first affected by the change is fixed and known, we prove that the threshold rule is asymptotically optimal without additional technical conditions. Moreover, we extend the low-cost SCD rule to a more general 2D sensor array. In this 2D sensor array case, the change can happen to any sensor and then gradually propagate to the surrounding sensors. For this 2D sensor array case, we also prove the asymptotic optimality of the multi-sensor SCD rule. In addition, we investigate how increasing the number of sensors can improve the asymptotic performance of the multi-sensor threshold SCD rule. Our work is related to [11], which studies the QCD problem under a linear multi-sensor setting. QCD problems can be viewed as a special case of SCD problem, which has only one post-change distribution. Therefore, the identification part, which is important in this paper, is not considered in [11].

Compared with the our previous works [1], [24], [28], this paper makes the following contributions. Firstly, [1] focuses only on a simple case that the change propagates across the sensor array following a fixed and known order. In this paper, we focus on a more general sensor array model in which the change can first reach any sensor in the array and then propagate to other sensors. Secondly, we provide detailed proof of the asymptotic optimality of the threshold SCD rule. The main idea of the proof similar to the proof for the single sensor case in [24]. However, the most important step of the proof, the asymptotic analysis of log-likelihood-ratio (LLR) processes of the sensor array case, is different from the single sensor case and more challenging. Thirdly, [1] focuses only on the linear array scenario. In this paper, we also extend the work to the 2D array case. Fourthly, we investigate the relationship between the performance of the threshold rule and the number of sensors in the array. Finally, we provided comprehensive numerical examples to illustrate the analytical results obtained in this paper.

The remainder of the paper is organized as follows. In Section II, we provide our problem formulation with a linear sensor array. In Section III, we study the evolution of the posterior probabilities. In Section IV, we discuss the structure of the optimal SCD rule. Then we introduce the threshold two-stage SCD rule

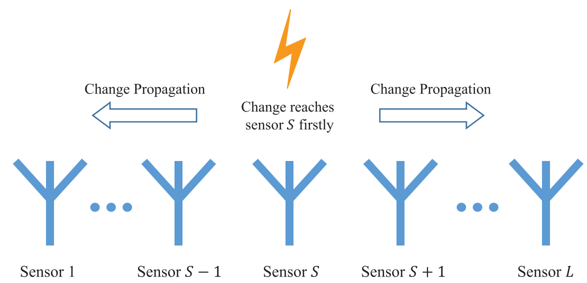


Figure 1: Change propagation model of the linear sensor array

and prove its asymptotic optimality for the linear sensor array case in Section V. Afterward, the threshold rule is extended to the 2D sensor array case in Section VI. In Section VII, we investigate the benefit of increasing the number of sensors. Simulation results are provided in Section VIII. Finally, we provide concluding remarks in Section IX.

II. Problem Formulation

To facilitate the presentation and easiness of understanding, we will first present our work for the linear array case. The more complicated 2D array scenario will be presented in Section VI.

In the linear array scenario, there is a linear array of L sensors monitoring the environment. The L sensors collect data at each time unit and then immediately send data to the fusion center for analysis. The observation of the system is a stochastic process hosted by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. At time k, the observation of the system is $\vec{\boldsymbol{X}}_k = (x_{k,1}, x_{k,2}, ..., x_{k,L})$, where $x_{k,l}$ is the data collected by the lth sensor at time k. Let $\lambda : \Omega \mapsto \{0, 1, \ldots\}$ be the time when an abrupt change happens in the sensing environment and $\theta : \Omega \mapsto \mathcal{I} := \{1, \ldots, I\}$ be the environment state after the change. The prior distribution of the change time is $\mathbb{P}(\lambda = k) = \rho(1-\rho)^k$. In addition, we denote $\mathcal{I} \cup \{0\}$ as $\mathcal{I}_0$. After time λ, the distribution of the data collected by each sensor may experience a change from f_0 to f_θ. f_θ can be one of the candidate distributions $\{f_i\}_{i\in\mathcal{I}}$. In addition, $\mathbb{F} = (\mathcal{F}_k)_{k\geq 0}$ is the filtration generated by the stochastic process $\{\vec{\boldsymbol{X}}_k\}_{k\geq 1}$.

A. Change Propagation Model

The change propagation model is illustrated in Fig. 1, the change will first happen to one sensor in the array and then propagate to other sensors. In the considered model, the change times of different sensors may be different. We denote the time change happen to sensor l as λ_l for all $1 \leq l \leq L$. Let S denote the index of the sensor that the change first reaches. The prior probability $\mathbb{P}(S = l) = \kappa_l$ is known. We denote $(\kappa_1, \kappa_2, \ldots, \kappa_L)$ as $\vec{\boldsymbol{K}}$. As shown in Fig. 1, the change first reaches sensor S at time $\lambda_S = \lambda$, then the change will propagate to sensors on both sides of sensor S following the directions $S \to S+1 \to \cdots \to L$ and $S \to S-1 \to \cdots \to 1$. The propagation of the change in the sensor array follows a geometric distribution, i.e., for $k_2 \geq 0$,

$$\begin{cases} \mathbb{P}\left[\lambda_{j-1} = k_1 + k_2 | \lambda_j = k_1, S = i\right] = \rho_1(1-\rho_1)^{k_2}, i > j \\ \mathbb{P}\left[\lambda_{j+1} = k_1 + k_2 | \lambda_j = k_1, S = i\right] = \rho_2(1-\rho_2)^{k_2}, i < j \end{cases}$$

where ρ_1 and ρ_2 are the probabilities of the change propagate from a sensor to its neighbor at each time step for the two directions.

B. Observation Model

In this paper, we assume that, conditioned on the change information, the observations of different times at every sensor are independent. Concretely, if $k < \lambda_l$, $x_{k,l} \sim f_0$, otherwise $x_{k,l} \sim f_\theta$, where $\theta \in \mathcal{I}$.

The prior probability of the state after change is defined as $v_i = \mathbb{P}\{\theta = i\}, i \in \mathcal{I}$. To simplify the notation, we express the conditional probabilities as:

$$\begin{cases} \mathbb{P}_i\{\cdot\} = \mathbb{P}\{\cdot|\theta = i\}, \\ \mathbb{P}_i^{(t)}\{\cdot\} = \mathbb{P}\{\cdot|\theta = i, \lambda = t\}, t \geq 0. \end{cases}$$

Correspondingly, $\mathbb{E}_i$ and $\mathbb{E}_i^{(t)}$ are the expectations under $\mathbb{P}_i$ and $\mathbb{P}_i^{(t)}$. In addition, We have the following assumption on f_0 and f_θ.

Assumption 1. *For every $i \in \mathcal{I}_0$ and $j \in \mathcal{I}_0 \backslash \{i\}$, we have*
(i) $0 < f_i(x)/f_j(x) < \infty$ a.s.*;*
(ii) $\int_{\{x: f_i(x) \neq f_j(x)\}} f_i(x)(dx) > 0$.

C. Two-stage Multi-sensor SCD Problem

Our goal is to quickly raise an alarm after the change occurs and further accurately determine the state θ, based on all the data samples $\{\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k\}$. Towards this goal, we employ a two-stage SCD rule $\delta = (\tau_1, \tau_2, d)$ that includes two stopping times τ_1, $\tau_1 + \tau_2$, and an identification decision d. Here, τ_1 is the time for the change detection and $\tau_1 + \tau_2$ is the time for the identification. Let $\Delta := \{(\tau_1, \tau_2, d) | \tau_1 \geq 0, \tau_2 \geq 0, d \in \mathcal{I}_0\}$ be the set of all possible two-stage SCD rules. We should note that if a wrong decision is made at τ_1, i.e., $\tau_1 < \lambda$, then $d = 0$ is the correct identification as long as this identification is made before λ, i.e., $\tau_1 + \tau_2 < \lambda$. Besides, the parameters ρ, ρ_1, ρ_2, $\vec{\boldsymbol{K}}$ and $\{v_i\}_{i \in \mathcal{I}}$ are known.

The possible costs of an SCD rule include costs of delay, false alarm, and misdiagnosis. The delay consists of the delays in the change detection stage and the distribution identification stage, i.e. $(\tau_1 - \lambda)_+$ and τ_2. The expected delay costs of them are $\mathbb{E}[c_1(\tau_1 - \lambda)_+]$ and $\mathbb{E}[c_2 \tau_2]$, where c_1 and c_2 are per-unit delay costs associated with each stage and $(z)_+ = \max(0, z)$ for any z. In addition, we define $r := c_2/c_1$ as the ratio between per-unit delay costs. A false alarm is the situation that a change alarm is raised before λ. The expected false alarm cost is $\mathbb{E}[a \mathbf{1}_{\{\tau_1 < \lambda\}}]$, where a is the penalty factor of false alarm and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Misdiagnosis occurs when a wrong identification is made, i.e., $d \neq \theta$. The expected misdiagnosis cost is $\mathbb{E}\Big[\sum_{i \in \mathcal{I}} b_{ij} \mathbf{1}_{\{\infty > \tau_1 + \tau_2 > \lambda, \theta = i, d = j\}} + b_{0j} \mathbf{1}_{\{\tau_1 + \tau_2 < \lambda, d = j\}}\Big]$ for $d = j$, where b_{ij} is the penalty factor for wrong decision $d = j$ when $\theta = i$ and $b_{0,j}$ is the penalty factor of the false alarm of the identification stage. We set $b_{ij} = 0$ when $i = j$. Hence the Bayesian cost function for a two-stage SCD rule $\delta \in \Delta$ is

$$\begin{aligned} &C(\delta) = c_1 \mathbb{E}\left[(\tau_1 - \lambda)_+\right] + c_2 \mathbb{E}[\tau_2] + a \mathbb{E}[\mathbf{1}_{\{\tau_1 < \lambda\}}] + \\ &\sum_{j=0}^{I} \mathbb{E}\Big[\sum_{i=1}^{I} b_{ij} \mathbf{1}_{\{\infty > \tau_1 + \tau_2 > \lambda, \theta = i, d = j\}} + b_{0j} \mathbf{1}_{\{\tau_1 + \tau_2 < \lambda, d = j\}}\Big]. \end{aligned} \tag{1}$$

The goal of the SCD problem is to find an SCD rule (τ_1, τ_2, d) that minimizes the expected cost $C(\delta)$.

III. Posterior Probability Analysis

Following the main idea of [24], we can solve a two-stage SCD problem using posterior probability process, $\Pi_k = (\Pi_k^{(0)}, \ldots, \Pi_k^{(I)})_{k \geq 0} \in \mathcal{Z}$, which is defined as

$$\begin{cases} \Pi_k^{(i)} := \mathbb{P}\{\lambda \leq k, \theta = i | \mathcal{F}_k\}, i \in \mathcal{I}, \\ \Pi_k^{(0)} := \mathbb{P}\{\lambda > k | \mathcal{F}_k\}, \end{cases}$$

and

$$\mathcal{Z} \triangleq \{\Pi \in [0,1]^{I+1} | \sum_{i \in \mathcal{I} \cup \{0\}} \Pi^{(i)} = 1\}.$$

Using Bayesian rule, we know that, at any time $k \geq 1$, each component of Π_k can be computed as

$$\Pi_k^{(i)} = \frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)}{\sum_{j \in \mathcal{I}_0} \alpha_k^{(j)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)}, \tag{2}$$

in which

$$
\begin{cases}
\alpha_k^{(0)} = (1-\rho)^{k+1}\prod_{l=1}^{L}\prod_{n=1}^{k} f_0(x_{n,l}) \\
\alpha_k^{(i)} = \sum_{s=1}^{L}\kappa_s v_i \rho \sum_{n_s=0}^{k}\left[(1-\rho)^{n_s}\left(\prod_{n=1}^{n_s-1} f_0(x_{n,s})\right)\cdot\left(\prod_{n=max(n_s,1)}^{k} f_i(x_{n,s})\right)\Psi_{s-1}^{(i)}(k,n_s)\Phi_{s+1}^{(i)}(k,n_s)\right] \\
\Psi_{l-1}^{(i)}(k,n_l) = (1-\rho_1)^{k-n_l+1}\prod_{t=1}^{l-1}\prod_{n=1}^{k} f_0(x_{n,t}) + \rho_1 \sum_{n_{l-1}=n_l}^{k}\left[(1-\rho_1)^{n_{l-1}-n_l}\left(\prod_{n=1}^{n_{l-1}-1} f_0(x_{n,l-1})\right)\cdot\right. \\
\left.\left(\prod_{n=n_{l-1}}^{k} f_i(x_{n,l-1})\right)\Psi_{l-2}^{(i)}(k,n_{l-1})\right], l>1 \\
\Phi_{l+1}^{(i)}(k,n_l) = (1-\rho_2)^{k-n_l+1}\prod_{t=l+1}^{L}\prod_{n=1}^{k} f_0(x_{n,t}) + \rho_2 \sum_{n_{l+1}=n_l}^{k}\left[(1-\rho_2)^{n_{l+1}-n_l}\left(\prod_{n=1}^{n_{l+1}-1} f_0(x_{n,l+1})\right)\cdot\right. \\
\left.\left(\prod_{n=n_{l+1}}^{k} f_i(x_{n,l+1})\right)\Phi_{l+2}^{(i)}(k,n_{l+1})\right], l<L \\
\Phi_{L+1}^{(i)}(k,n_l) = \Psi_0^{(i)}(k,n_l) = 1
\end{cases}
. \tag{3}
$$

Assumption 1 implies $0 < \Pi_k^{(i)} < 1$ for every finite $k \geq 1$ and $i \in \mathcal{I}_0$. We define the log-likelihood-ratio (LLR) processes as

$$
\Lambda_k(i,j) := \log\frac{\Pi_k^{(i)}}{\Pi_k^{(j)}} = \log\frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,\ldots,\vec{\boldsymbol{X}}_k)}{\alpha_k^{(j)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,\ldots,\vec{\boldsymbol{X}}_k)}. \tag{4}
$$

Directly calculating Π_k based on (2) requires us to remember all past samples, which requires large storage and is not easy for implementation. Hence it is desirable to compute Π_k recursively once a new sample $\vec{\boldsymbol{X}}_k$ arrives. To achieve this, we further define the event

$$
T_{i,k,s,l_1,l_2} = \{S=s, \lambda_{l_1-1} > k, \lambda_{l_1} \leq k, \lambda_{l_2+1} > k, \lambda_{l_2} \leq k, \theta = i\}
$$

for $1 < s < L$, $l_1 \leq s$ and $l_2 \geq s$. Specially, $T_{i,k,1,1,l_2} = \{S=1, \lambda_{l_2+1} > k, \lambda_{l_2} \leq k, \theta = i\}$ and $T_{i,k,L,l_1,L} = \{S=L, \lambda_{l_1-1} > k, \lambda_{l_1} \leq k, \theta = i\}$. From the definition, we know that event T_{i,k,s,l_1,l_2} denotes the event that the change with post-change distribution f_i firstly reaches sensor s and already propogates to sensors l_1 and l_2 at time k. In addition, we define the event that change has not happened yet as $T_{0,k} = \{\lambda > k\}$. In this change process setting, we can see that the underlying probability space Ω can be partitioned as

$$
\Omega = \left(\bigcup_{s=1}^{L}\bigcup_{l_1=1}^{s}\bigcup_{l_2=s}^{L}\bigcup_{i\in\mathcal{I}} T_{i,k,s,l_1,l_2}\right)\bigcup T_{0,k}.
$$

Then, we denote the posterior probability as $p_{i,k,s,l_1,l_2} := \mathbb{P}\{T_{i,k,s,l_1,l_2}|\mathcal{F}_k\}$ and $p_{0,k} = \mathbb{P}\{T_{0,k}|\mathcal{F}_k\}$. Using Bayesian rule, we can derive the updating rule for these posterior probabilities as

$$
\begin{cases}
p_{i,k,s,l_1,l_2} = \frac{N_{i,k,s,l_1,l_2}}{\sum_{s=1}^{L}\sum_{l_1=1}^{s}\sum_{l_2=s}^{L}\sum_{i\in I} N_{i,k,s,l_1,l_2} + N_{0,k}} \\
1 < s < L, 1 \leq l_1 \leq s, s \leq l_2 \leq L, i \in \mathcal{I} \\
p_{0,k} = \frac{N_{0,k}}{\sum_{s=1}^{L}\sum_{l_1=1}^{s}\sum_{l_2=s}^{L}\sum_{i\in I} N_{i,k,s,l_1,l_2} + N_{0,k}}
\end{cases}, \tag{5}
$$

where N_{i,k,s,l_1,l_2} denotes the probability density

$$
\begin{aligned}
&d\mathbb{P}((\vec{\boldsymbol{X}}_1,\ldots,\vec{\boldsymbol{X}}_k),T_{i,k,s,l_1,l_2})\\
&=\left(\prod_{n=l_1}^{l_2} f_i(x_{k,n})\right)\left(\prod_{n=1}^{l_1-1} f_0(x_{k,n})\right)\left(\prod_{n=l_2+1}^{L} f_0(x_{k,n})\right)\cdot\Bigg[p_{0,k-1}\kappa_s\rho(1-\rho_1)^{\mathbf{1}_{\{l_1\neq 1\}}}(1-\rho_2)^{\mathbf{1}_{\{l_2\neq L\}}}\rho_1^{s-l_1}\rho_2^{l_2-s}+\\
&\left(\sum_{n_1=l_1}^{s}\sum_{n_2=s}^{l_2} p_{i,k-1,s,n_1,n_2}(1-\rho_1)^{\mathbf{1}_{\{l_1\neq 1\}}}(1-\rho_2)^{\mathbf{1}_{\{l_2\neq L\}}}\rho_1^{n_1-l_1}\rho_2^{l_2-n_2}\right)\Bigg]
\end{aligned}
\tag{6}
$$

and $N_{0,k}$ denotes the probability density

$$
d\mathbb{P}((\vec{\boldsymbol{X}}_1,\ldots,\vec{\boldsymbol{X}}_k),T_{0,k})=p_{0,k-1}(1-\rho)\prod_{n=1}^{L} f_0(x_{k,n}). \tag{7}
$$

For $k=0$, we have $p_{0,0}=1-\rho$. For $l_1\le s\le l_2$, we have

$$
p_{i,0,s,l_1,l_2}=\kappa_s v_i\rho(1-\rho_1)^{\mathbf{1}_{\{l_1\neq 1\}}}(1-\rho_2)^{\mathbf{1}_{\{l_2\neq L\}}}\rho_1^{s-l_1}\rho_2^{l_2-s}.
$$

Let $\mathbf{P}_k$ denote the 4-dimensional posterior probabilities tensor in which its elements are p_{i,k,s,l_1,l_2}. In $\mathbf{P}_k$, only elements satisfying $l_1\le s$ and $l_2\ge s$ can be non-zero values. From (5) (6) and (7), we see that $\mathbf{P}_k$ can be computed from $\mathbf{P}_{k-1}$ and observation $\vec{\boldsymbol{X}}_k$ at time k. Hence, we have the recursive update formula for the posterior probabilities $\{\mathbf{P}_k,p_{0,k}\}$. More importantly, by the relationship between $\{\mathbf{P}_k,p_{0,k}\}$ and Π_k,

$$
\begin{cases}
\Pi_k^{(i)}=\sum_{s=1}^{L}\sum_{l_1=1}^{s}\sum_{l_2=s}^{L} p_{i,k,s,l_1,l_2},\ i\in\mathcal{I}\\
\Pi_k^{(0)}=p_{0,k}
\end{cases},
\tag{8}
$$

we can update Π_k recursively.

IV. Optimal Multi-sensor Two-stage SCD rule

Given the updating rule of Π_k, (5) and (8), the optimal rule (τ_1^*,τ_2^*,d^*) that minimizes (1) can be obtained by following similar steps as those in our recent work [24]. In particular, by converting the two-stage problem into two optimal single stopping time problems and solving them in reversed order, we can obtain the optimal SCD rule for the proposed two-stage sensor array SCD problem. Here, for completeness, we introduce the main steps of obtaining the optimal rule (τ_1^*,τ_2^*,d^*).

To start, using Π_k, we can express the Bayesian cost (1) as

$$
C(\delta)=\mathbb{E}\left[\sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right)+c_2\tau_2+\mathbf{1}_{\{\tau_1<\infty\}}a\Pi_{\tau_1}^{(0)}+\mathbf{1}_{\{\tau_1+\tau_2<\infty\}}\sum_{j=0}^{I}\mathbf{1}_{\{d=j\}}B_j(\Pi_{\tau_1+\tau_2})\right],
$$

where $B_j(\Pi)=\sum_{i\in\mathcal{I}_0} b_{ij}\Pi^{(i)}$ is the misdiagnosis cost associated with the decision $d=j$. Therefore, $B(\Pi)=\min\limits_{j\in\mathcal{I}_0}B_j(\Pi)$ is the smallest misdiagnosis cost can be achieved at time k. As a result, the optimal identification decision is $d^*=\arg\min_{j\in\mathcal{I}_0}B_j(\Pi)$. Using this result, we have $C(\tau_1,\tau_2,d^*)=\mathbb{E}[C_1(\tau_1)+C_2(\Pi_{\tau_1},\tau_2)]$, where

$$
C_1(\tau_1)=\sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right)+\mathbf{1}_{\{\tau_1<\infty\}}a\Pi_{\tau_1}^{(0)}
$$

and $C_2(\Pi_{\tau_1},\tau_2)=c_2\tau_2+\mathbf{1}_{\{\tau_1+\tau_2<\infty\}}B\left(\Pi_{\tau_1+\tau_2}\right)$ are the cost functions of the change detection stage and distribution identification stage respectively. Then we have the minimal expected cost for the SCD process,

$$\begin{aligned}C(\tau_1^*,\tau_2^*,d^*) &= \min_{\tau_1,\tau_1+\tau_2\in\mathbb{F}}\mathbb{E}\left[C_1(\tau_1)+C_2(\tau_1,\tau_2)\right]\\ &= \min_{\tau_1,\tau_1+\tau_2\in\mathbb{F}}\mathbb{E}\Big[C_1(\tau_1)+\mathbb{E}\left[C_2(\tau_2)|\mathbf{P}_{\tau_1},p_{0,\tau_1}\right]\Big]\\ &= \min_{\tau_1\in\mathbb{F}}\mathbb{E}\left[C_1(\tau_1)+\min_{\tau_1+\tau_2\in\mathbb{F}}\mathbb{E}\left[C_2(\tau_2)|\mathbf{P}_{\tau_1},p_{0,\tau_1}\right]\right].\end{aligned}\tag{9}$$

By (9), the two-stage stopping time problem becomes two ordered optimal single stopping time problems. The first one is for the identification stage, its goal is finding the optimal τ_2 which minimizes $\mathbb{E}[C_2(\tau_2)|\mathbf{P}_{\tau_1},p_{0,\tau_1}]$ for any given τ_1, $\mathbf{P}_{\tau_1}$ and p_{0,τ_1}. The second single stopping time problem is to find the best stopping rule for the detection stage. From the last line of (9), we can find an optimal τ_1 to minimize the expected cost for the whole SCD process if the optimal rule for τ_2 is known. Therefore, we will firstly find the optimal rule for the identification stage, then select the optimal stopping time for the detection stage. DP is a good way to solve optimal single stopping time problems. With the expression C_1 and C_2, we can built the cost-to-go functions of the two optimal single stopping time problems. In particular, for the identification stage, let $\{\tilde{\boldsymbol{P}},\tilde{p}_0\}$ be the posterior probabilities at time next to the time of $\{\boldsymbol{P},p_0\}$. The infinite-horizon cost-to-go function for the DP process of the identification stage can be obtained by solving $V(\boldsymbol{P},p_0)=\min(B(\boldsymbol{P},p_0),c_2+\mathbb{E}[V(\tilde{\boldsymbol{P}},\tilde{p}_0)|\boldsymbol{P},p_0])$. This implies that we should make an identification when the expected cost for keep observing exceeds the cost of making identification immediately. In addition, the optimal identification decision is $d=\arg\min_{j\in\mathcal{I}_0}B_j(\boldsymbol{P})$. Similarly, in the change detection stage, for any $\{\boldsymbol{P},p_0\}$, the infinite-horizon cost-to-go function for the detection stage satisfies the following Bellman equation $W(\boldsymbol{P},p_0)=\min(ap_0+V(\boldsymbol{P},p_0),c_1(1-p_0)+\mathbb{E}[W(\tilde{\boldsymbol{P}},\tilde{p}_0)|\boldsymbol{P},p_0])$. From this, we know that we should raise a change alarm when the expected cost of observing more data exceeds the cost of declaring a change has happened.

The cost-to-go functions $V(\boldsymbol{P})$ and $W(\boldsymbol{P})$ and the optimal stopping times can be calculated using DP. However, the size of the state space increases exponentially with L and I. With such a high complexity, the optimal solution is hard to implement.

V. Low-complexity Rule

Same as other DP-based methods, the complexity of the optimal solution is very high, even with an array with only two sensors and two post-change distributions. To address this issue, we propose a threshold SCD rule that is easy to implement. Moreover, we will prove this threshold SCD rule is asymptotically optimal as c_1 and c_2 go to zero. The main idea of the proof is similar to the proof for the single sensor case considered in [24]. However, the most important step of the proof, i.e., analyzing the convergence of the LLR process, becomes much more complicated in the sensor array case. In this section, we will introduce the main steps of the asymptotic optimality analysis and underline the proof details of the LLR convergence (Proposition 1).

A. Threshold SCD Rule

Here, we introduce the proposed low complexity two-stage SCD rule. The low complexity rule is a threshold rule. In particular, it is characterized by a set of thresholds $\{A,\vec{\boldsymbol{B}}\}$ where $\vec{\boldsymbol{B}}=(B_0,B_1,B_2,...,B_I)$. A and all elements in $\vec{\boldsymbol{B}}$ are strictly positive constants. Using these thresholds, the proposed threshold rule $\delta_T=(\tau_A,\tau_{\vec{\boldsymbol{B}}},d_{\vec{\boldsymbol{B}}})$ is defined as

$$\begin{cases}\tau_A:=\inf\{k\geq 1,\Pi_k^{(0)}<1/(1+A)\},\\ \tau_{\vec{\boldsymbol{B}}}:=\min\limits_{i\in\mathcal{I}_0}\tau_{\vec{\boldsymbol{B}}}^{(i)},\\ \tau_{\vec{\boldsymbol{B}}}^{(i)}:=\inf\{k\geq 1,\Pi_k^{(i)}>1/(1+B_i)\}-\tau_A,\\ d_{\vec{\boldsymbol{B}}}:=\arg\min\limits_{i\in\mathcal{I}_0}\tau_{\vec{\boldsymbol{B}}}^{(i)}.\end{cases}\tag{10}$$

In this threshold SCD rule, the first stopping time τ_A is the first time $\Pi_k^{(0)}$ falls below the threshold $1/(1+A)$. After τ_A, the rule turns to check the posterior probabilities $\Pi_k^{(i)}$ for all $i \in \mathcal{I}_0$. It will stop immediately if any threshold $1/(1+B_i)$ is exceeded. The identification decision depends on which threshold is passed. In order to guarantee that this rule is in the two-stage SCD rule space Δ, it must satisfy $\tau_{\vec{\boldsymbol{B}}} \geq 0$. This condition can be satisfied by choosing appropriate A and $\vec{\boldsymbol{B}}$, as will be introduced in Section V-C.

For $i \in \mathcal{I}_0$ and $k \geq 1$, define the logarithm of the odds-ratio process as

$$\pi_k^{(i)} := \log \frac{\Pi_k^{(i)}}{1-\Pi_k^{(i)}} = -\log \left[\sum_{j \in \mathcal{I}_0 \backslash \{i\}} \exp(-\Lambda_k(i,j)) \right].$$

Using $\pi_k^{(i)}$, δ_T can be expressed as:

$$\begin{cases} \tau_A = \inf \left\{ k \geq 1, \dfrac{1-\Pi_k^{(0)}}{\Pi_k^{(0)}} > A \right\} = \inf\{k \geq 1, \pi_k^{(0)} < -\log A\}, \\ \tau_{\vec{\boldsymbol{B}}} = \min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}, \\ \tau_{\vec{\boldsymbol{B}}}^{(i)} = \inf \left\{ k \geq 1, \dfrac{1-\Pi_k^{(i)}}{\Pi_k^{(i)}} < B_i \right\} - \tau_A = \inf\{k \geq 1, \pi_k^{(i)} > -\log B_i\} - \tau_A, \\ d_{\vec{\boldsymbol{B}}} = \arg\min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}. \end{cases} \tag{11}$$

The complexity of the threshold rule (10) is very low. After obtaining a new sample, we only need to update the posterior probabilities using the recursive formula (5), and then compare them with the thresholds. In the following parts, we will show that this rule is asymptotically optimal as c_1 and c_2 go to zero.

B. Convergence of LLR Process

By (2) and (4), we can see that

$$\Lambda_k(i,j) = \log \alpha_k^{(i)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k) - \log \alpha_k^{(j)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k).$$

For $i \in \mathcal{I}$ and time $k > 0$, we define

$$H_k^{(i)} = \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \left[\left(\prod_{n=1}^{n_s-1} \left(\frac{(1-\rho) f_0(x_{n,s})}{(1-\rho_1)(1-\rho_2) f_i(x_{n,s})} \right) \right) \cdot \psi_{s-1}^{(i)}(k, n_s) \phi_{s+1}^{(i)}(k, n_s) \right]$$

where

$$\begin{cases} \psi_l^{(i)}(k, n_{l+1}) = \prod\limits_{n=1}^{k} \left[(1-\rho_1) \prod\limits_{t=1}^{l} \dfrac{f_0(x_{n,t})}{f_i(x_{n,t})} \right] + \rho_1 \sum\limits_{n_l = n_{l+1}}^{k} \prod\limits_{n=1}^{n_l - 1} \dfrac{f_0(x_{n,l})}{f_i(x_{n,l})} \psi_{l-1}^{(i)}(k, n_l), l \geq 1 \\ \phi_l^{(i)}(k, n_{l-1}) = \prod\limits_{n=1}^{k} \left[(1-\rho_2) \prod\limits_{t=l}^{L} \dfrac{f_0(x_{n,t})}{f_i(x_{n,t})} \right] + \rho_2 \sum\limits_{n_l = n_{l-1}}^{k} \prod\limits_{n=1}^{n_l - 1} \dfrac{f_0(x_{n,l})}{f_i(x_{n,l})} \phi_{l+1}^{(i)}(k, n_l), l \leq L \end{cases}.$$

In addition, $\phi_{L+1}^{(i)}(k, n_L) = (1-\rho_2)^{n_L - 1}$ and $\psi_0^{(i)}(k, n_1) = (1-\rho_1)^{n_1 - 1}$. Therefore, we can express $\log \alpha_k^{(i)}$ as

$$\begin{cases} \log \alpha_k^{(i)} = \log[v_i \rho (1-\rho)] + \log \left(\prod\limits_{l=1}^{L} \prod\limits_{m=1}^{k} f_i(x_{m,l}) \right) + \log H_k^{(i)}, \text{for } i \in \mathcal{I} \\ \log \alpha_k^{(0)} = (k+1)\log(1-\rho) + \log \left(\prod\limits_{l=1}^{L} \prod\limits_{m=1}^{k} f_0(x_{m,l}) \right). \end{cases}$$

Let $q(j,i)$ be the KL divergence from f_i to f_j. We define the following condition for $i,j \in \mathcal{I}$.

Condition 1. $\log(1-\rho) + q(j,i) - q(j,0) \geq 0$ *or* $q(j,i) - q(j,0) \leq 0$.

The next proposition describes the limit of $\log H_k^{(i)}/k$ as $k \to \infty$.

Proposition 1. *For any $i,j \in \mathcal{I}$, if Condition 1 is satisfied,*

$$\frac{1}{k}\log H_k^{(i)} \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} h(i,j) \tag{12}$$

where $h(i,j) = (\log(1-\rho) + L(q(j,i) - q(j,0)))_+$.

Proof. Please see Appendix A. □

C. Asymptotic Optimality

Once we show the convergence of $\log H_k^{(i)}/k$, we can proceed to show the asymptotic optimality of the threshold rule. The main steps on this proof are: (1) Obtain approximations of the delay, false alarm probability and misdiagnosis probability, which leads to the expression of the Bayesian cost of the threshold rule, $C(\delta_T)$, w.r.t. A and $\boldsymbol{B}$; (2) Select the optimal A and $\boldsymbol{B}$ that can minimize $C(\delta_T)$; (3) Prove that $C(\delta_T, A_{opt}, \boldsymbol{B}_{opt})$ achieves the lower bound of the Bayesian cost for arbitrary two-stage SCD rule when c_1 and c_2 go to 0.

For any $i \in \mathcal{I}$, define

$$w(i,j) = \begin{cases} Lq(i,j) - h(j,i), j \in \mathcal{I} \\ Lq(i,0) - \log(1-\rho), j = 0 \end{cases}. \tag{13}$$

If the first affected sensor is unknown, and Condition 1 is satisfied for $i,j \in \mathcal{I}$, $h(i,j)$ can be calculated as in Proposition 1. As introduced in [24], the approximation of delay can be expressed as

$$\begin{cases} \mathbb{E}_i\left[(\tau_{\vec{\boldsymbol{B}}} + \tau_A - \lambda)_+\right] \xrightarrow[B_i\to 0]{\mathbb{P}_i-a.s.} \frac{-\log B_i}{w(i)}, \text{ for } i \in \mathcal{I} \\ \mathbb{E}_i\left[(\tau_A - \lambda)_+\right] \xrightarrow[A\to\infty]{\mathbb{P}_i-a.s.} \frac{\log A}{w(i,0)}, \text{ for } i \in \mathcal{I} \end{cases}, \tag{14}$$

where $w(i) = w(i, j(i))$, $j(i) = \arg\min\limits_{j\in\mathcal{I}_0\backslash\{i\}} w(i,j)$. In addition, the false alarm and misdiagnosis probability can be approximated as $\frac{k_a}{1+A}$ and $\sum\limits_{i\in\mathcal{I}} v_i B_i k_i$, respectively. Here $k_a = a$ and $k_i = \max_{j\in\mathcal{I}_0\backslash\{i\}} b_{ji}$. Therefore, the Bayesian cost of the threshold rule can be approximated as

$$C^{(c_2)}(\delta_T) = c_2 \sum_{i\in\mathcal{I}} v_i \left(\frac{-\log(B_i)}{w(i)}\right) + \sum_{i\in\mathcal{I}} v_i B_i k_i + c_2\left(\frac{1}{r} - 1\right)\sum_{i\in\mathcal{I}} \frac{v_i \log A}{w(i,0)} + \frac{k_a}{1+A}. \tag{15}$$

By minimizing (15) w.r.t A and $\vec{\boldsymbol{B}}$, we get the optimal A and $\vec{\boldsymbol{B}}$ as

$$\begin{cases} A_{opt} \approx \frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}} \frac{v_i}{w(i,0)}} - 2, \\ B_{i,opt} = \frac{c_2}{k_i w(i)}, i \in \mathcal{I}. \end{cases} \tag{16}$$

The Bayesian cost for the optimal threshold SCD rule is

$$\begin{aligned} C^{(c_2)}(\delta_T^*) = c_2 \sum_{i\in\mathcal{I}} \frac{-v_i}{w(i)} \log\left(\frac{c_2}{k_i w(i)}\right) + \sum_{i\in\mathcal{I}} \frac{v_i c_2}{w(i)} + c_2\left(\frac{1}{r}-1\right)\sum_{i\in\mathcal{I}} \frac{v_i}{w(i,0)} \log\left(\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}}\frac{v_i}{w(i,0)}} - 2\right) + \\ k_a \frac{1}{\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}}\frac{v_i}{w(i,0)}} - 1}. \end{aligned} \tag{17}$$

Now we need to check if the condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is satisfied. By the threshold rule (10), we know that $\tau_{A_{opt}}$ is the first time $\sum_{i\in\mathcal{I}} \Pi_n^{(i)} = 1 - \Pi_n^{(0)}$ exceeds the threshold $1 - 1/(1 + A_{opt})$. Also, $\tau^{(i)}_{\vec{\boldsymbol{B}}_{opt}} + \tau_{A_{opt}}$ is the first time for $\Pi_n^{(i)}$ exceeds the threshold $1/(1 + B_{i,opt})$. So if

$$1 - \frac{1}{1 + A_{opt}} < \frac{1}{1 + B_{i,opt}} \tag{18}$$

for all $i \in \mathcal{I}$, it is guaranteed that the threshold $\vec{\boldsymbol{B}}$ can not be reached before threshold A, namely, $\tau_{\vec{\boldsymbol{B}}} \geq 0$. After plugging the explicit expressions of the optimal thresholds (16) in inequality (18), we know that a sufficient condition of $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is

$$0 < r \leq \min_{i\in\mathcal{I}} \frac{1}{1 + \frac{k_a}{k_i w(i) \sum\limits_{i\in\mathcal{I}} \frac{v_i}{w(i,0)}}}. \tag{19}$$

If the value of r satisfies (19), condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is satisfied. However, for the case (19) is not satisfied, we need to change the threshold accordingly as

$$\begin{cases} A' = A_{opt}, \\ B'_i = B_{i,opt}\frac{k_i}{\eta}, i \in \mathcal{I} \end{cases} \tag{20}$$

where η is a constant such that

$$r = \min_{i\in\mathcal{I}} \frac{1}{1 + \frac{k_a}{\eta w(i) \sum\limits_{i\in\mathcal{I}} \frac{v_i}{w(i,0)}}}.$$

We can see that with A' and $\vec{\boldsymbol{B}}'_{opt}$, condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ still is satisfied even if (19) is not satisfied. In this case, the Bayesian cost of the rule $\delta'_T = (\tau_{A'}, \tau_{\vec{\boldsymbol{B}}'}, d')$ is

$$\begin{aligned} C^{(c_2)}(\delta'_T) = C^{(c_2)}(\delta^*_T) - c_2 \sum_{i\in\mathcal{I}} \log\left(\frac{k_i}{\eta}\right)\frac{v_i}{w(i)} \\ + \sum_{i\in\mathcal{I}} v_i B_{i,opt}\left(\frac{k_i^2}{\eta} - k_i\right). \end{aligned} \tag{21}$$

Since k_i, $w(i)$ and η are constants, the last two terms in (21) decay much faster than $C^{(c_2)}(\delta^*_T)$ as $c_2 \to 0$. This implies that the difference between the cost calculated by (17) and (21) is negligible as $c_2 \to 0$.

Finally, in the following proposition, we prove that (17) (also true for (21) if (19) is not satisfied) is the lowest Bayesian cost any two-stage SCD rule can achieve when c_1 and c_2 go to 0. In other words, the proposed threshold rule is asymptotically optimal.

Proposition 2. *If $\delta_T = (\tau_{A_T}, \tau_{\vec{\boldsymbol{B}}_T}, d_T)$ is a threshold two-stage SCD rule with thresholds as* (16)*, then for any given fixed $r := c_2/c_1$ we have*

$$\lim_{c_2\to 0} \frac{\inf_{\delta\in\Delta} C^{(c_2)}(\delta)}{C^{(c_2)}(\delta_T)} \geq 1.$$

The main steps to prove Proposition 2 are as follows: (1). Derive a lower bound of the Baysian cost for any possible SCD rule; (2) Prove the proposed threshold SCD rule can achieve the lower bound as c_1 and c_2 go to zero. For more details of the proof, please refer to [24]. Note that, since Proposition 2 is proved based on Proposition 1, Condition 1 is also necessary for Proposition 2.

From the results of asymptotic analysis in Proposition 1 and equation (14) and (16), we can see that the prior probabilities of first affected sensor $\{\kappa_s\}_{1\leq s\leq L}$ do not affect the asymptotic behaviors of the threshold rule. Therefore, in the case when $\{\kappa_s\}_{1\leq s\leq L}$ are unknown, we can just set $\kappa_s = 1/L$ for all $1 \leq s \leq L$. Even if the true prior probabilities are not $1/L$, it will not affect the asymptotic optimality of the threshold SCD rule. In addition, the Condition 1 is not a strong condition because it just rules out the case $0 < q(j,i) - q(j,0) < -\log(1-\rho)$ in which the change is very hard to detect and identify. Considering the change is typically rare, i.e. ρ is small and the range $[0, -\log(1-\rho)]$ is narrow, Condition 1 can be satisfied in most cases.

D. Special Case: When the First Affected Sensor is Known

As discussed above, when the first affected sensor S is an unknown random variable, Condition 1 is necessary for the asymptotic optimality of the multi-sensor threshold SCD rule. In this section, we will show that, when the first affected sensor is fixed and known, the multi-sensor threshold SCD rule is asymptotically optimal with no additional condition.

When the first affected sensor is fixed and known, one element of $\vec{\boldsymbol{K}}$ is 1 and all other elements are 0. Without loss of generality, we assume that the first affected sensor is the sth sensor, i.e., $\kappa_s = 1$. With this additional assumption, the computations in the previous section can be further simplified and we can prove stronger asymptotic optimality results. In particular, for any time $k \geq 1$, Π_k can be directly calculated as

$$\Pi_k^{(i)} = \frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)}{\sum_{j \in \mathcal{I}_0} \alpha_k^{(j)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)} \tag{22}$$

where

$$\begin{cases} \alpha_k^{(0)} = (1-\rho)^{k+1} \prod\limits_{l=1}^{L} \prod\limits_{n=1}^{k} f_0(x_{n,l}) \\ \alpha_k^{(i)} = v_i \rho \sum\limits_{n_s=0}^{k} \left[(1-\rho)^{n_s} \left(\prod\limits_{n=1}^{n_s-1} f_0(x_{n,s}) \right) \cdot \left(\prod\limits_{n=max(n_s,1)}^{k} f_i(x_{n,s}) \right) \Psi_{s-1}^{(i)}(k, n_s) \Phi_{s+1}^{(i)}(k, n_s) \right]. \end{cases} \tag{23}$$

For $i \in \mathcal{I}$, we define

$$H_k^{(i)} = \sum_{n_s=0}^{k} \left[\left(\prod_{n=1}^{n_s-1} \left(\frac{(1-\rho) f_0(x_{n,s})}{(1-\rho_1)(1-\rho_2) f_i(x_{n,s})} \right) \right) \cdot \psi_{s-1}^{(i)}(k, n_s) \phi_{s+1}^{(i)}(k, n_s) \right]. \tag{24}$$

Define

$$\eta_l(i,j) = \begin{cases} \log \left[\frac{1-\rho}{(1-\rho_1)^{\mathbb{1}\{s \neq 1\}} (1-\rho_2)^{\mathbb{1}\{s \neq L\}}} \right] + q(j,i) - q(j,0), l = s \\ \log(1-\rho_1) + q(j,i) - q(j,0), l = 1 \text{ and } s \neq 1 \\ \log(1-\rho_2) + q(j,i) - q(j,0), l = L \text{ and } s \neq L \\ q(j,i) - q(j,0), \text{otherwise.} \end{cases} \tag{25}$$

For any $i, j \in \mathcal{I}$, according to the value of $\eta_l(i,j)$, we divide the sensor labels $1 \leq l \leq L$ into several consecutive groups (the labels in each group are consecutive). The grouping rule is described in Algorithm 1. After implementing Algorithm 1 for the case $i, j \in \mathcal{I}$, we will have $M(i,j) + N(i,j) + 1$ consecutive groups

$$\begin{aligned} &\{a_1^m(i,j), a_1^m(i,j)+1, \ldots, a_2^m(i,j)\}_{1 \leq m \leq M(i,j)}, \\ &\{a_2^{M(i,j)}(i,j)+1, a_2^{M(i,j)}(i,j)+2, \ldots, b_2^{N(i,j)}(i,j)-1\}, \\ &\{b_2^n(i,j), \ldots, b_1^n(i,j)-1, b_1^n(i,j)\}_{N(i,j) \geq n \geq 1}. \end{aligned}$$

The next proposition describes the limit of $\log H_k^{(i)}/k$ as $k \to \infty$.

Proposition 3. *For any $i, j \in \mathcal{I}$,*

$$\frac{\log H_k^{(i)}}{k} \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} h(i,j) \tag{26}$$

where

$$h(i,j) = \sum_{l=1}^{a_2^{M(i,j)}(i,j)} \eta_l(i,j) + \sum_{l=b_2^{N(i,j)}(i,j)}^{L} \eta_l(i,j) + \left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \eta_l(i,j) \right)_+ . \tag{27}$$

Algorithm 1: Grouping the sensors

Initialize $a_1^1(i,j)=1$, $a_2^0(i,j)=0$, $b_1^1(i,j)=L$, $b_2^0(i,j)=L+1$, $m=1$, $n=1$;
for *l=1,2,...,s-2,s-1* **do**
  if $\sum_{k=a_1^m(i,j)}^{l} \eta_k(i,j) \geq 0$ **then**
    $a_2^m(i,j)=l$, $a_1^{m+1}(i,j)=l+1$;
    $m+=1$;
end
for *l=L,L-1,...,s+2, s+1* **do**
  if $\sum_{k=b_1^m(i,j)}^{l} \eta_k(i,j) \geq 0$ **then**
    $b_2^n(i,j)=l$, $b_1^{n+1}(i,j)=l-1$;
    $n+=1$;
  $M(i,j)=m-1, N(i,j)=n-1$
end

Proof. Please see Appendix B. □

Then following the same steps of Section V-C, we can prove that the multi-sensor threshold SCD rule is asymptotically optimal as c_1 and c_2 go to zero. Plugging (27) in (13), (16) and (17), we will have the optimal threshold and the corresponding Bayesian cost. Different from the asymptotic optimality for the general case in Section V-C, in this special case when the first affected sensor is known, the asymptotic optimality does not need any additional condition. This is because knowing first affected sensor makes the structure of $H(i,j)$ easier and thus we can prove Proposition 3 true in general. Moreover, if Condition 1 is true for any $i,j \in \mathcal{I}$, we can easily check that the $h(i,j)$ in Proposition 1 and equation (3) are equivalent following Algorithm 1. With equivalent $h(i,j)$, $w(i,j)$ and the limit of cost function in (17) will also be equivalent. This indicates that the performances of the general case and special case will tend to be the same as c_1 and c_2 go to zero.

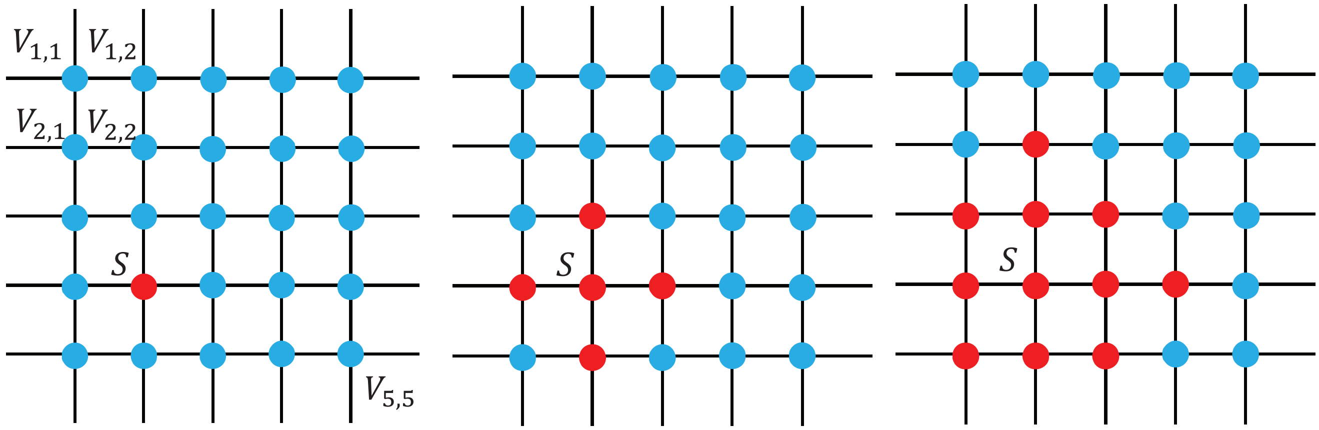
VI. Extension of the proposed SCD rules to 2D sensor array Case

In the Section V, we studied the SCD problem in a linear sensor array. In this section, we extend our study to a 2D lattice array scenario.

A. Change Propagation Model on 2D Lattice

Consider an 2D lattice with vertices $\{V_{a,b}\}_{1\leq a\leq H, 1\leq b\leq W}$, where $V_{a,b}$ denotes the vertex at the ith row and the jth column of the lattice. An edge exists between vertex pair $(V_{a,b}, V_{c,d})$ if $|a-c|+|b-d|=1$. A change could happen at any single vertex first and then start to diffuse outward via the edges. At time k, the sensors collect the data samples $\vec{\boldsymbol{X}}_k=(x_{k,1,1}, x_{k,1,1}, ..., x_{k,H,W})$. Let $S=(S_1,S_2)$ be index of the sensor where the change happens first, the prior probability $\mathcal{P}(S=(a,b))=\kappa_{a,b}$ is known. We denote $(\kappa_{1,1},\kappa_{1,2},\ldots,\kappa_{H,W})$ as $\boldsymbol{K}$. The change propagation process is characterized by the distance between the target sensor and the first affected sensor and follows a geometric distribution. let $\mathcal{V}(a,b,r)$ be vertex layer whose distance to $V_{a,b}$ is exactly r, i.e., $\mathcal{V}(a,b,r)=\{V_{m,n}||a-m|+|b-n|=r\}$. The change will first propagate from V_{S_1,S_2} to all the vertex in $\mathcal{V}(S_1,S_2,1)$ at time $\lambda_{\mathcal{V}(S_1,S_2,1)}$, then to all vertices in $\mathcal{V}(S_1,S_1,2)$ at time $\lambda_{\mathcal{V}(S_1,S_2,2)}$ and so on. The propagation of the change in the 2D lattice follows a geometric distribution as

$$\mathbb{P}\left[\lambda_{\mathcal{V}(a,b,r+1)}=k_1+k_2|\lambda_{\mathcal{V}(a,b,r)}=k_1, S=(a,b)\right]=\rho_1(1-\rho_1)^{k_2}, k_2\geq 0 \tag{28}$$



(a) Sensor array status at λ (b) Sensor array status at $\lambda_{\mathcal{V}(S_1,S_2,1)}$ (c) Sensor array status at $\lambda_{\mathcal{V}(S_1,S_2,2)}$

Figure 2: Change propagation model of the 2D lattice sensor array

where ρ_1 is the probability of the change propagates outward the next layer. As an example of the 2D lattice sensor array, we illustrate the change propagation process in a 5×5 lattice sensor array in Fig. 2. On each vertex, a sensor is implemented to collect data. $x_{k,a,b}$ denotes the data sample collected by the sensor at $V_{a,b}$ at time k. For the convenience of expression, we denote

$$\begin{cases} \mathcal{C}(S_1,S_2,r) = \{(a,b)||a-S_1|+|b-S_2|=r, 1 \leq a \leq H, 1 \leq b \leq W\}, \\ \mathcal{I}(S_1,S_2,r) = \{(a,b)||a-S_1|+|b-S_2|\leq r, 1 \leq a \leq H, 1 \leq b \leq W\}, \\ \mathcal{O}(S_1,S_2,r) = \{(a,b)||a-S_1|+|b-S_2|> r, 1 \leq a \leq H, 1 \leq b \leq W\}, \\ R(S_1,S_2) = max_{1\leq a\leq H, 1\leq b\leq W}(|a-S_1|+|b-S_2|). \end{cases}$$

Now we have a new 2D lattice sensor array and a corresponding change propagation model. The other parts in the SCD problem formulation, such as the prior distribution of the change time λ, observation model and etc., are the same as in Section II.

B. Posterior Probability Analysis

In the SCD problem with the 2D lattice sensor array, the posterior probability Π_k defined in (2) still plays a key role. However, the calculation of $\alpha_k^{(i)}$ in (3) will be replaced as

$$\begin{cases} \alpha_k^{(0)} = (1-\rho)^{k+1} \prod_{a=1}^{H} \prod_{b=1}^{W} \prod_{n=1}^{k} f_0(x_{n,a,b}) \\ \alpha_k^{(i)} = \sum_{S_1=1}^{H} \sum_{S_2=1}^{W} \kappa_{S_1,S_2} v_i \rho \sum_{n_0=0}^{k} \left[(1-\rho)^{n_0} \left(\prod_{n=1}^{n_0-1} f_0(x_{n,S_1,S_2}) \right) \cdot \left(\prod_{n=max(n_0,1)}^{k} f_i(x_{n,S_1,S_2}) \right) \Psi_1^{(i)}(k,n_0,S_1,S_2) \right] \\ \Psi_{l+1}^{(i)}(k,n_l,S_1,S_2) = (1-\rho_1)^{k-n_l+1} \prod_{(a,b)\in\mathcal{O}(S_1,S_2,l)} \prod_{n=1}^{k} f_0(x_{n,a,b}) + \\ \rho_1 \sum_{n_{l+1}=n_l}^{k} \left[(1-\rho_1)^{n_{l+1}-n_l} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,l+1)} \left(\left(\prod_{n=1}^{n_{l+1}-1} f_0(x_{n,a,b}) \right) \cdot \right. \right. \\ \left. \left. \left(\prod_{n=n_{l+1}}^{k} f_i(x_{n,a,b}) \right) \right) \Psi_{l+2}^{(i)}(k,n_{l+1},S_1,S_2) \right], R(S_1,S_2) > l \geq 0 \\ \Psi_{R(S_1,S_2)+1}^{(i)}(k,n_l) = 1 \end{cases} .$$

Similar to Section III, we want to compute Π_k recursively once a new sample $\vec{\boldsymbol{X}}_k$ arrives rather than remembering all historical data samples. To this end, we further define the event

$$T_{i,k,a,b,r} = \{S_1 = a, S_2 = b, \lambda_{\mathcal{V}(a,b,r+1)} > k, \lambda_{\mathcal{V}(a,b,r)} \leq k, \theta = i\}$$

for $1 \leq a \leq H, 1 \leq b \leq W, 1 \leq r \leq R(a,b), i \in \mathcal{I}$. From the definition, we know that event T_{i,k,s,l_1,l_2} denotes the event that the change with post-change distribution f_i firstly reaches vertex $V_{a,b}$ and already propogates to vertices $\mathcal{V}(a,b,r)$ at time k. In addition, we define the event that change has not happened yet as $T_{0,k} = \{\lambda > k\}$. In this change process setting, we can see that the underlying probability space Ω can be partitioned as

$$\Omega = \left(\bigcup_{a=1}^{H} \bigcup_{b=1}^{W} \bigcup_{r=0}^{R(a,b)} \bigcup_{i \in \mathcal{I}} T_{i,k,a,b,r} \right) \bigcup T_{0,k}.$$

Then, we denote the posterior probability as $p_{i,k,a,b,r} := \mathbb{P}\{T_{i,k,a,b,r} | \mathcal{F}_k\}$ and $p_{0,k} = \mathbb{P}\{T_{0,k} | \mathcal{F}_k\}$. Using Bayesian rule, we can derive the updating rule for these posterior probabilities as

$$\begin{cases} p_{i,k,a,b,r} = \frac{N_{i,k,a,b,r}}{\sum\limits_{a=1}^{H} \sum\limits_{b=1}^{W} \sum\limits_{r=0}^{R(a,b)} \sum\limits_{i \in I} N_{i,k,a,b,r} + N_{0,k}}, 1 \leq a \leq H, 1 \leq b \leq W, 0 \leq r \leq R(a,b), i \in \mathcal{I} \\ p_{0,k} = \frac{N_{0,k}}{\sum\limits_{a=1}^{H} \sum\limits_{b=1}^{W} \sum\limits_{r=1}^{R(a,b)} \sum\limits_{i \in I} N_{i,k,a,b,r} + N_{0,k}} \end{cases} \tag{29}$$

where $N_{i,k,a,b,r}$ denotes the probability density

$$\begin{aligned} d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{i,k,a,b,r}) = \left(\prod_{(m,n) \in \mathcal{I}(a,b,r)} f_i(x_{m,n,k}) \right) \left(\prod_{(a,b) \in \mathcal{O}(a,b,r)} f_0(x_{m,n,k}) \right) \cdot \\ \left(p_{0,k-1} \kappa_{a,b} \rho (1-\rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \rho_1^r + \sum_{r_{k-1}=1}^{r} p_{i,k,a,b,r_{k-1}} \rho_1^{r - r_{k-1}} (1-\rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \right) \end{aligned} \tag{30}$$

and $N_{0,k}$ denotes the probability density

$$d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{0,k}) = p_{0,k-1}(1-\rho) \prod_{a=1}^{H} \prod_{b=1}^{W} f_0(x_{k,a,b}). \tag{31}$$

For $k = 0$, we have $p_{0,0} = 1 - \rho$. For $1 \leq a \leq H, 1 \leq b \leq W, 0 \leq r \leq R(a,b), i \in \mathcal{I}$, we have

$$p_{i,k,a,b,r} = \kappa_{a,b} v_i \rho (1-\rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \rho_1^r.$$

Let $\mathbf{P}_k$ denote the 4-dimensional posterior probabilities tensor in which its elements are $p_{i,k,a,b,r}$. In $\mathbf{P}_k$, only elements satisfying $1 \leq r \leq R(a,b)$ can be non-zero values. From (29) (30) and (31), we see that $\mathbf{P}_k$ can be computed from $\mathbf{P}_{k-1}$ and observation $\vec{\boldsymbol{X}}_k$ at time k. Hence, we have the recursive update formula for the posterior probabilities $\{\mathbf{P}_k, p_{0,k}\}$. More importantly, by the relationship between $\{\mathbf{P}_k, p_{0,k}\}$ and Π_k,

$$\begin{cases} \Pi_k^{(i)} = \sum\limits_{a=1}^{H} \sum\limits_{b=1}^{W} \sum\limits_{1=1}^{R(a,b)} p_{i,k,a,b,r}, i \in \mathcal{I} \\ \Pi_k^{(0)} = p_{0,k} \end{cases} \tag{32}$$

we can update Π_k recursively. Afterwards, we can follow the same steps described in Section IV and obtain the optimal SCD rule of the 2D lattice case. Similar to the linear sensor array case, since the state space increase exponentially with H, W and I, the extreme high complexity make the optimal method hard to implement.

C. Low-complexity rule

The low-complexity threshold given in (11) works for te 2D lattice sensor array case and the asymptotic optimality also preserves. The only difference between the threshold rules of the linear sensor case and the 2D lattice case is the proof of the convergence of the LLR process. Therefore, we only provide the proof the convergence of the LLR process for the 2D lattice case.

For $i \in \mathcal{I}$ and time $k > 0$, we define

$$H_k^{(i)} = \sum_{S_1=1}^{H} \sum_{S_2=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k} \left[\left(\prod_{n=1}^{n_0-1} \left(\frac{(1-\rho) f_0(x_{n,S_1,S_2})}{(1-\rho_1) f_i(x_{n,S_1,S_2})} \right) \right) \cdot \psi_1^{(i)}(k, n_0, S_1, S_2) \right]$$

where

$$\begin{aligned} &\psi_{r+1}^{(i)}\left(k, n_r, S_1, S_2\right) = \prod_{n=1}^{k} \left[(1-\rho_1) \prod_{(a,b)\in\mathcal{O}(S_1,S_2,r)} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right] + \\ &\rho_1 \sum_{n_{r+1}=n_r}^{k} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_{r+1}-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \psi_{r+2}^{(i)}\left(k, n_{r+1}, S_1, S_2\right), R(S_1,S_2) > r \geq 0. \end{aligned} \tag{33}$$

In addition, $\psi_{R(S_1,S_2)+1}^{(i)}\left(k, n_{R(S_1,S_2)}, S_1, S_2\right) = (1-\rho_1)^{n_{R(S_1,S_2)}-1}$. Therefore, we can express $\log \alpha_k^{(i)}$ as

$$\begin{cases} \log \alpha_k^{(i)} = \log[v_i \rho(1-\rho)] + \log\left(\prod_{a=1}^{H} \prod_{b=1}^{W} \prod_{m=1}^{k} f_i(x_{m,a,b}) \right) + \log H_k^{(i)}, \text{for } i \in \mathcal{I} \\ \log \alpha_k^{(0)} = (k+1)\log(1-\rho) + \log\left(\prod_{a=1}^{H} \prod_{b=1}^{W} \prod_{m=1}^{k} f_0(x_{m,a,b}) \right). \end{cases}$$

The next proposition describes the limit of $\log H_k^{(i)}/k$ as $k \to \infty$.

Proposition 4. *For any $i, j \in \mathcal{I}$, if Condition 1 is satisfied,*

$$\frac{1}{k} \log H_k^{(i)} \xrightarrow[k\to\infty]{\mathbb{P}_j - a.s.} h(i,j) \tag{34}$$

where $h(i,j) = (\log(1-\rho) + L(q(j,i) - q(j,0)))_+$ and $L = HW$.

Proof. Please see Appendix C. □

After proving the convergence of the LLR process, the asymptotic optimality of the threshold rule (11) in the 2D sensor array case can be proved following the same steps introduced in Section V.

VII. Benefits of Increasing Number of Sensors

In this section we will prove that adding more sensors to the sensor array will always improve the performance of the multi-sensor threshold SCD rule when c_1 and c_2 are sufficiently small. From the Bayesian cost of the optimal threshold rule in (17), we can see that if constants $w(i)$ and $w(i,0)$ increase, the cost will decrease. Although we know that $C^{(c_2)}(\delta_T^*) \to 0$ as $c_1, c_2 \to 0$, greater constants $w(i)$ and $w(i,0)$ can make $C^{(c_2)}(\delta_T^*)$ converge to 0 faster. Next, we will analyze how $w(i)$ and $w(i,0)$ change as more sensors are added to different sensor array structures.

A. Case 1: The first affected sensor is unknown

When Condition 1 is satisfied for $i, j \in \mathcal{I}$, and the first affected sensor is randomly chosen and unknown (as in Section V and VI). By (13) and Proposition 1, we have

$$w(i,j) = \begin{cases} Lq(i,j), \text{if } \log(1-\rho) \geq q(i,0) - q(i,j), i \in \mathcal{I} \\ Lq(i,0) - \log(1-\rho), j = 0 \text{ or } q(i,j) \leq q(i,0) \end{cases}.$$

By Assumption 1 and the fact $q(i,j)$ is the KL divergence, $q(i,j)$ is positive for $i, j \in \mathcal{I}$. Therefore, $w(i)$ and $w(i,j)$ will increase with the number of sensors. This implies that, with more sensors in the sensor array, the performance of the multi-sensor threshold SCD rule will be improved when Condition 1 is satisfied for all $i, j \in \mathcal{I}$ in the general case.

B. Case 2: The first affected sensor is known

As we introduced in Section V-D, when Condition 1 does not hold and the first affected sensor is fixed and known, the calculation of constant w is more complicated. The reason is that adding one more sensor to the array may change the grouping result of Algorithm 1. Without of generality, we assume the sensor is added to the right of the first affected sensor s, i.e., we added the $l = (L+1)th$ sensor to the array. Then $\eta_L(i,j)$ change from $\log(1-\rho_2) + q(j,i) - q(j,0)$ to $q(j,i) - q(j,0)$. The new added $\eta_{L+1}(i,j) = \log(1-\rho_2) + q(j,i) - q(j,0)$. Based on the value of $\eta_L(i,j)$, the increment of $h(i,j)$ could be different. However, it's easy to check that, the increment of $h(i,j)$ is upper bounded by $(q(j,i) - q(j,0))_+$. Based on this observation and (13), we can see that by adding one sensor, $w(i,j)$ will always increase. Therefore, the performance of the multi-sensor threshold SCD rule can always be improved by adding sensors to the sensor array.

It is worth noting that the benefit introduced in this section is for the asymptotic case, i.e. $c_1, c_2 \to 0$. In other words, adding more sensors will improve the performance when c_1 and c_2 are sufficiently small. However, such property many not hold when c_1 and c_2 is relative large.

VIII. Numerical results

Since the optimal SCD rule is too complex to implement in the multi-sensor case, obtaining the optimal solution is extremely time-consuming, even for a simple case with $L = 2$ and $I = 2$. Therefore, we will not carry out experiments to directly compare the performance of the optimal SCD rule and the threshold SCD rule. However, we still can validate that the multi-sensor threshold SCD rule has a considerable improvement over a single sensor threshold rule (all sensors except the first one are ignored) and a mismatched threshold rule (changes of all sensors are falsely assumed to happen at the same time). Particularly, we will investigate the performance of the multi-sensor threshold SCD rule in a general case (first affected sensor is a random variable) and a special case (first affected sensor is fixed and known). In this section, we provide 4 numerical examples to illustrate the performance of the threshold SCD rule. In all following examples, the results are estimated by Monte-Carlo simulations. Concretely, we generate data samples following the underlying SCD process and apply the SCD rules to the generated sequence. An episode ends when the SCD rule makes the final detection and identification decisions. Then we calculate the Bayesian cost and start another episode. The Bayesian cost $C(\tau_1, \tau_2, d)$ is approximated using the average value of 10,000 episodes of Monte-Carlo simulation.

In the first example, the observed data samples are generated by a two-dimensional normal distribution, $\mathcal{N}(\vec{\boldsymbol{\mu}}, \boldsymbol{I}_2)$. The mean vector $\vec{\boldsymbol{\mu}}$ changes at the change point. In the first example, we consider the case with two possible post-change mean vectors $\vec{\boldsymbol{\mu}}_1 = (0,1)$ and $\vec{\boldsymbol{\mu}}_2 = (0,-1)$ and the pre-change mean vector $\vec{\boldsymbol{\mu}}_0 = (0,0)$. In addition, we set $\rho_1 = 0.2$, $\rho_2 = 0.2$, $\rho = 0.01$, $(v_1, v_2) = (0.3, 0.7)$ and $c_2/c_1 = 0.1$. All the penalty factors of the false alarm and misdiagnosis are set to be 1. For this problem formulation, we study 7 different cases: (1). $L = 5$ with $\vec{\boldsymbol{K}} = [0.2, 0.2, 0.2, 0.2, 0.2]$ (General case); (2). $L = 5$ with $\vec{\boldsymbol{K}} = [0, 0, 1, 0, 0]$ (Special case); (3) $L = 5$ with $\vec{\boldsymbol{K}} = [0, 0, 1, 0, 0]$ (Mismatch case); (4). $L = 2$ with

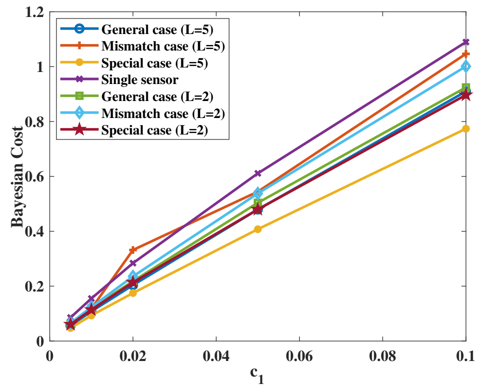
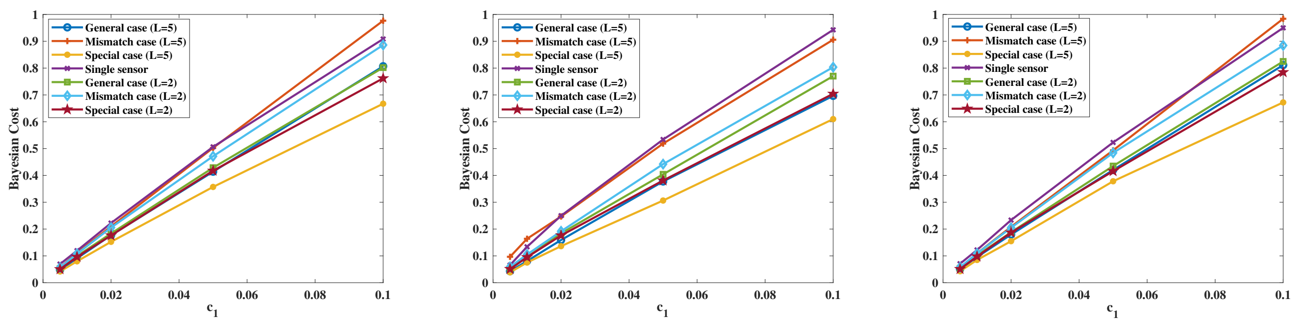


Figure 3: Performance of the multi-sensor threshold SCD rule in 7 different cases for the change on the mean of 2-D Gaussian distribution



(a) Change in the mean vector of $F_L(\mu_1, \mu_2)$

(b) Change in the covariance matrix of 2-D Gaussian distributions

(c) Change in the type of 2-D distributions

Figure 4: Performance of the multi-sensor threshold SCD rule in 7 different cases for different types of change

$\vec{\boldsymbol{K}} = [0.5, 0.5]$ (General case); (5). $L = 2$ with $\vec{\boldsymbol{K}} = [0, 1]$ (Special case); (6) $L = 2$ with $\vec{\boldsymbol{K}} = [0, 1]$ (Mismatch case); (7) Single sensor case. The result of these 7 cases are shown in Fig. 3. In addition, Table I presents the performance of the two-stage SCD rule with different sensor arrays. In Table I, we have the following columns: FAP (false alarm probability), MISDP (misdiagnosis probability), delay1 (expected delay time in the detection stage), delay2 (expected delay time in the identification stage), wrong decision costs (FAP + MISDP), total delay cost (c_1*delay1+c_2*delay2), Bayesian cost (FAP+ MISDP+total delay cost). From these results, we can see the general trends of the performance of the threshold rule are: (1) Special case> General case> Mismatch case and single sensor case; (2) $L = 5 > L = 2$ for the general and the special case. The advantage of the special case over the general case is due to the additional information that the first sensor affected by the change is known in the special case. In conclusion, the results of this example indicate that with more sensors and the correct information about the problem formulation, the proposed multi-sensor threshold SCD rule can efficiently improve the performance.

In the second example, we illustrate our results using pre-change and post-change distributions that are more complex than the one used in the first example. Firstly, we define a 2-D distribution, $F_L(\mu_1, \mu_2)$. With $F_L(\mu_1, \mu_2)$, the two elements in each data sample are independent and follow the Laplace distributions, $L(\mu_1, 1/\sqrt{2})$ and $L(\mu_2, 1/\sqrt{2})$, respectively. In this example, we implement three experiments: (1) Change in the mean vector of $F_L(\mu_1, \mu_2)$. The pre-change distribution is $F_L(0, 0)$, the post-change distributions are $F_L(0, 1)$ and $F_L(0, -1)$; (2) Change in the covariance matrix of 2-D Gaussian distribution. The pre-change distribution is 2-D Gaussian distribution, $\mathcal{N}(\vec{\mathbf{0}}, 0.5\boldsymbol{I}_2)$, the post-change distributions are $\mathcal{N}(\vec{\mathbf{0}}, \boldsymbol{I}_2)$ and $\mathcal{N}(\vec{\mathbf{0}}, 2\boldsymbol{I}_2)$; (3) Change in the type of the distribution. The pre-change distribution is a 2-D Gaussian

	c_1	FAP	MISDP	Delay1	Delay2	Wrong decision cost	Total delay cost	Bayesian cost
Single sensor	0.1	0.1334	0.0045	7.6437	18.729	0.1379	0.9517	1.08956
	0.05	0.058	0.0023	10.0065	10.0966	0.0603	0.5508	0.6111
	0.02	0.0215	0.0013	12.4123	6.4296	0.0228	0.2611	0.2839
	0.01	0.0099	0.0006	13.926	5.2957	0.0105	0.1446	0.1551
	0.005	0.006	0.0004	15.3682	4.9344	0.0064	0.07931	0.08571
General case (L=2)	0.1	0.0887	0.0044	7.2048	11.0701	0.0931	0.8312	0.9243
	0.05	0.0459	0.0023	8.4144	6.9719	0.0482	0.4556	0.5038
	0.02	0.0161	0.0004	9.6692	4.0748	0.0165	0.2015	0.218
	0.01	0.0097	0.0002	10.5496	3.2829	0.0099	0.1088	0.1187
	0.005	0.0043	0.0002	11.387	2.7081	0.0045	0.05829	0.06279
General case (L=5)	0.1	0.0286	0.0016	8.3372	4.5829	0.0302	0.8795	0.9097
	0.05	0.0134	0.0006	9.0294	2.7989	0.014	0.4655	0.4795
	0.02	0.0049	0.0003	9.7337	1.9988	0.0052	0.1987	0.2039
	0.01	0.0025	0.0003	10.3545	1.7942	0.0028	0.1053	0.10814
	0.005	0.0015	0.0002	10.771	1.5232	0.0017	0.05461	0.05632
Special case (L=2)	0.1	0.0885	0.0041	6.9261	11.1448	0.0926	0.8041	0.8967
	0.05	0.0386	0.0022	8.1569	6.2093	0.0408	0.4389	0.4797
	0.02	0.0166	0.0006	9.4394	3.9092	0.0172	0.1966	0.2138
	0.01	0.0083	0.0004	10.2133	3.1732	0.0087	0.1053	0.114
	0.005	0.0038	0.0002	11.0598	2.6632	0.004	0.05663	0.06063
Special case (L=5)	0.1	0.0265	0.0018	7.0316	4.2327	0.0283	0.7455	0.7738
	0.05	0.0127	0.0005	7.6022	2.8422	0.0132	0.3943	0.4075
	0.02	0.005	0.0003	8.3083	1.7477	0.0053	0.1697	0.175
	0.01	0.0038	0.0002	8.7582	1.6554	0.004	0.08924	0.09323
	0.005	0.0008	0.0001	9.1803	1.2985	0.0009	0.04655	0.04745
Mismatch case (L=2)	0.1	0.045	0.0017	8.8483	6.957	0.0467	0.9544	1.0011
	0.05	0.0226	0.0009	9.8357	4.5418	0.0235	0.5145	0.538
	0.02	0.009	0.0005	10.9599	3.3959	0.0095	0.226	0.2355
	0.01	0.0035	0.0001	11.7296	2.5338	0.0036	0.1198	0.1234
	0.005	0.0017	0.0001	12.5006	2.3469	0.0018	0.06368	0.06548
Mismatch case (L=5)	0.1	0.0115	0.0003	10.0698	2.7791	0.0118	1.03477	1.04657
	0.05	0.0064	0.0003	10.5553	1.9905	0.0067	0.5377	0.5444
	0.02	0.0021	0	16.3119	1.7122	0.0021	0.3297	0.3318
	0.01	0.001	0	11.5815	1.2742	0.001	0.1171	0.1181
	0.005	0.0005	0	12.0188	1.229	0.0005	0.0607	0.0612

Table I: Performance of the multi-sensor threshold SCD rule in 7 different cases for the change on the mean of 2-D Gaussian distribution

distribution, $F_L(0,0)$, the post-change distributions are $\mathcal{N}((0,1),\boldsymbol{I}_2)$ and $\mathcal{N}((0,-1),\boldsymbol{I}_2)$. All the other parameters in this example are the same as the first example. The simulation results of the three settings are shown in Figure 4. These results are very similar to the results in the first example. It indicates that the proposed multi-sensor threshold SCD rule (general case and special case) works well for various settings of pre-change and post-change distributions.

In the first two examples, we know that the additional information about the first sensor affected by the change makes the special case has better performance than the general case. However, from the analysis in Section V-D, the limit of the cost function of the two cases should be the same. In the third example, we implement an experiment to validate this analysis result. Assume $L=5$, for the general case, we assume $\vec{\boldsymbol{K}}=[0.2,0.2,0.2,0.2,0.2]$. For the special case, we assume $\vec{\boldsymbol{K}}=[0,0,1,0,0]$. Following similar setting of the first example, we only change the mean vector to $\vec{\boldsymbol{\mu}}_1=(0,0.2)$ and $\vec{\boldsymbol{\mu}}_2=(0,-0.2)$. It is easy to check that Condition 1 is satisfied for all $i,j\in\mathcal{I}$. The cost functions of the two cases and the ratio between them are given in Table II. From that table, we can see that, with smaller c_1 (and smaller c_2 since c_2/c_1 is set to be 0.1), the ratio between the cost of the special case and the general case is getting closer to 1. From the experiments we did in the first three examples, we can see that the prior information about the first affected sensor can help to improve the performance of the multi-sensor threshold SCD rule, especially when c_1 and c_2 is not very small. However, this improvement will get smaller as c_1 and c_2 approach zero.

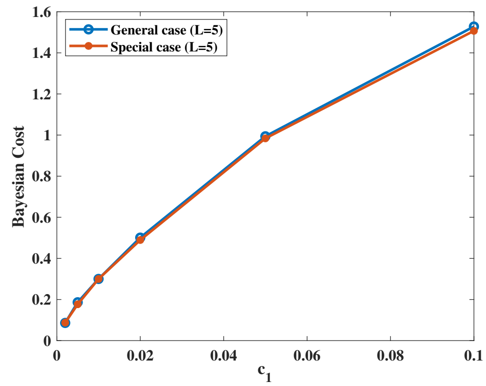


Figure 5: Performance of the multi-sensor threshold SCD rule in general case and special case when Condition 1 is not satisfied

As we introduced in Section V-C, the threshold SCD rule is asymptotically optimal when the Condition 1 is satisfied for all $i, j \in \mathcal{I}$. If the condition is not satisfied, currently we are not able to prove the asymptotic optimality of the threshold SCD rule for the general case. In the fourth example, we numerically study the performance of the multi-sensor SCD rule in the general case when Condition 1 is not satisfied. We still use the same 2-D Gaussian setting of the first example except for the mean vector. We set $\vec{\boldsymbol{\mu}}_1 = (0, 0.1)$ and $\vec{\boldsymbol{\mu}}_2 = (0, -0.1)$ in order to make the Condition 1 unsatisfied. In this setting, we compare the performance of the general case and the special case. The result is shown in Fig. 5. From this figure, we can see that the performance of the multi-sensor threshold SCD rule in the general case is very close to that in the special case. According to our analysis in Section V, we know the multi-sensor threshold SCD rule is always asymptotically optimal in the special case. Therefore, we know that without the asymptotic optimal guarantee, the multi-sensor threshold SCD rule can still have good performance.

Finally, we provide a numerical experiment for the 2D sensor array described in Section VI. In this experiment, the propagation probability of the 2D lattice sensor array is $\rho_1 = 0.2$. The change can happen to any sensor in the array following a uniform distribution, i.e., $\mathcal{P}(S = (a, b)) = 1/(HW)$ for any $1 \le a \le H$ and $1 \le b \le W$. All other settings of this experiment are the same as the first experiment. The Bayesian costs of the multi-sensor threshold SCD rule with three different 2D lattice arrays are presented in Table. III. The performance of the single sensor case is also given as a reference. From this table, we can see that the performance of the threshold SCD rule in the sensor array case is generally better than in the single sensor case. We also notice that the performance of a large sensor array can be worse than a smaller sensor array when the unit delay cost is relatively big. For example, the Bayesian costs of 10×10 and 5×5 sensor array are larger than that of the 2×2 sensor array when $c_1 = 0.1$. This result indicates that the Bayesian cost of the multi-sensor threshold SCD rule does not strictly decrease as the number of sensors increases when the unit delay cost is not very small. However, the results in Table. III also validate that, when the unit delay costs are sufficiently small, e.g. $c_1 = 1 \times 10^{-6}, 1 \times 10^{-8}$ or 1×10^{-10}, the performance of the multi-sensor threshold SCD rule with a large sensor array is always better than that with a smaller sensor array. This result is consistent with the conclusion we obtained in Section VII.

IX. Conclusion and Future Work

In this paper, we have formulated the Bayesian two-stage sequential change diagnosis over a linear sensor array problem. By analyzing the posterior probability, we have converted the multi-sensor version SCD problem to a normal SCD problem and characterized the optimal solution. However, the complexity of the proposed optimal solution is high due to the DP steps. To reduce the computational complexity, we have designed a threshold multi-sensor two-stage SCD rule. For the general case in which the first

Table II: Performances of the two-stage multi-sensor threshold SCD rules with different c_1

c_1	General Case	Special Case	Bayesian Cost Ratio
10^{-2}	0.5291	0.4956	0.937
10^{-4}	1.03e-2	9.83e-3	0.955
10^{-6}	1.26e-4	1.23e-4	0.980
10^{-8}	1.69e-6	1.66e-6	0.988
10^{-10}	2.09e-8	2.08e-8	0.993

Table III: Performances of the two-stage multi-sensor threshold SCD rules in 2D lattice sensor array case

	c_1=0.1	c_1=0.05	c_1=0.01	c_1=0.05	c_1=0.001	c_1=1e-6	c_1=1e-8	c_1=1e-10
Single Sensor	1.08956	0.6111	0.1551	0.08571	0.01958	3.2514e-05	4.1536e-07	5.08306e-09
2×2 Sensor Array	0.8477	0.4504	0.09892	0.05267	0.01121	1.46e-05	1.73129e-07	1.96511e-09
5×5 Sensor Array	0.8534	0.4606	0.09743	0.05018	0.01049	1.2134e-05	1.3309e-07	1.3984e-09
10×10 Sensor Array	0.9512	0.4803	0.1017	0.05129	0.01062	1.1824e-05	1.2765e-07	1.3422e-09

sensor affected by the change is randomly chosen and unknown, we have proved that the threshold SCD rule is asymptotically optimal under Condition 1. For the special case that the first affected sensor is fixed and known, we have proved that the threshold rule is generally asymptotically optimal. Furthermore, we have extended the threshold SCD rule to a more general 2D sensor array case and proved its asymptotic optimality. Finally we have analyzed how increasing the number of sensors can improve the performance of the threshold SCD rule.

In terms of future work, it is of interest to investigate the proposed two-stage change diagnosis model in more general scenarios, for example, the more general change propagation models and more complex sensor arrays, the case with unknown parameters (such as v_i) and the case that the post-change distributions are unknown, etc. It is also of interest to carry out the asymptotic analysis when the prior probabilities of the change and change propagation, such as ρ, ρ_1 and ρ_2, go to zero.

Appendix A
Proof of Proposition 1

Before we prove Proposition 1, we introduce some helpful results.

Lemma 1. *Let $\{\xi_k\}_{k\geq 1}$ be a positive stochastic process and T be an a.s. finite random time defined on the same probability space $(\Omega, \varepsilon, \mathbb{P})$. Given T, the random variables $\{\xi_k\}_{k\geq 1}$ are conditionally independent, and $\{\xi_k\}_{1\leq k\leq T-1}$ and $\{\xi_k\}_{k\geq T}$ have common conditional probability distributions $\mathbb{P}^{(\infty)}$ and $\mathbb{P}^{(0)}$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, the expectations with respect to which are denoted by $\mathbb{E}^{(\infty)}$ and $\mathbb{E}^{(0)}$, respectively. Suppose that $\mathbb{E}^{(\infty)}[log\xi_1]$ and $\mathbb{E}^{(0)}[log\xi_1]$ exist, and define $\eta := \mathbb{E}^{(0)}[log\xi_1]$. Then for any fixed constant $c > 0$*

$$\frac{1}{k}\log\left(c + \sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} \eta_+. \tag{35}$$

This lemma is the first part of Lemma 5.5 in the paper [22]. Here we further extend this lemma so that it can be applied to our sensor array problem.

Lemma 2. *Let $\{\xi_k\}_{k\geq 1}$ be a positive stochastic process and $T_{L-1} \leq T_L$ are two a.s. finite random times defined on the same probability space $(\Omega, \varepsilon, \mathbb{P})$. Given T_{L-1} and T_L, the random variables $\{\xi_k\}_{k\geq 1}$ are conditionally independent, and $\{\xi_k\}_{T_{L-1}\leq k\leq T_L-1}$ and $\{\xi_k\}_{k\geq T_L}$ have common conditional probability distributions $\mathbb{P}^{(\infty)}$ and $\mathbb{P}^{(0)}$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, the expectations with respect to which are denoted by $\mathbb{E}^{(\infty)}$ and $\mathbb{E}^{(0)}$, respectively. Suppose that $\mathbb{E}^{(\infty)}[log\xi_1]$ and $\mathbb{E}^{(0)}[log\xi_1]$ exist, $0 < \xi_k < \infty$ for all $k \geq 1$ and define $\eta := \mathbb{E}^{(0)}[log\xi_1]$. Then for any fixed constant $c > 0$*

$$\frac{1}{k}\log\left(c + \sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} \eta_+. \tag{36}$$

Proof.

$$
\begin{aligned}
&\tfrac{1}{k}\log(c+\sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n)\\
&=\tfrac{1}{k}\log\left(c+\sum_{l=1}^{T_{L-1}-1}\prod_{n=1}^{l}\xi_n+\sum_{l=T_{L-1}}^{k}\prod_{n=1}^{l}\xi_n\right)\\
&=\tfrac{1}{k}\log\left(\left(c+\sum_{l=1}^{T_L-1}\prod_{n=1}^{l}\xi_n\right)\prod_{n=1}^{T_L-1}\xi_n^{-1}+\sum_{l=T_{L-1}}^{k}\prod_{n=T_{L-1}}^{l}\xi_n\right)+\tfrac{1}{k}\log\left(\prod_{n=1}^{T_{L-1}-1}\xi_n\right).
\end{aligned}
$$

The last equality holds by setting

$$c'=\left(c+\sum_{l=1}^{T_L-1}\prod_{n=1}^{l}\xi_n\right)\prod_{n=1}^{T_L-1}\xi_n^{-1}>0.$$

By Lemma 1, we can see that

$$\tfrac{1}{k}\log\left(c'+\sum_{l=T_{L-1}}^{k}\prod_{n=T_{L-1}}^{l}\xi_n\right)\xrightarrow[k\to\infty]{\mathbb{P}-a.s.}\eta_+.$$

On the other hand, since T_{L-1} is a.s. finite, we have

$$\tfrac{1}{k}\log\left(\prod_{n=1}^{T_{L-1}-1}\xi_n\right)\xrightarrow[k\to\infty]{\mathbb{P}-a.s.}0.$$

Then the lemma is proved. $\square$

Now, we first prove that, for any $i,j\in\mathcal{I}$, we have

$$\liminf_{k\to\infty}\frac{1}{k}\log H_k^{(i)}\geq\left(\log(1-\rho)+Lq(j,i)-Lq(j,0)\right)_+ \tag{37}$$

$\mathbb{P}_j$ almost surely.

For any $i\in\mathcal{I}$, define

$$\begin{cases}\xi_k^{(l)}(i)=\frac{f_0(x_{k,l})}{f_i(x_{k,l})},2\leq l\leq L-1,\\ \xi_k^{(1)}(i)=\frac{f_0(x_{k,l})}{f_i(x_{k,l})}(1-\rho_1),\\ \xi_k^{(L)}(i)=\frac{f_0(x_{k,l})}{f_i(x_{k,l})}(1-\rho_2).\end{cases} \tag{38}$$

With this definition, we can have

$$H_k^{(i)}=\sum_{s=1}^{L}\kappa_s\sum_{n_s=0}^{k}\left(\prod_{n=1}^{n_s-1}\left(\frac{(1-\rho)\xi_n^{(s)}(i)}{(1-\rho_1)(1-\rho_2)}\right)\right)\cdot\psi_{s-1}^{(i)}(k,n_s)\phi_{s+1}^{(i)}(k,n_s) \tag{39}$$

where

$$\begin{cases}\psi_l^{(i)}(k,n_{l+1})=\prod_{n=1}^{k}(1-\rho_1)\prod_{t=1}^{l}\xi_n^{(t)}(i)+\rho_1\sum_{n_l=n_{l+1}}^{k}\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\psi_{l-1}^{(i)}(k,n_l),L-1\geq l\geq 1\\ \psi_0^{(i)}(k,n_1)=1\end{cases}, \tag{40}$$

and

$$\begin{cases}\phi_l^{(i)}(k,n_{l-1})=\prod_{n=1}^{k}(1-\rho_2)\prod_{t=l}^{L}\xi_n^{(t)}(i)\\ +\rho_2\sum_{n_l=n_{l-1}}^{k}\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\phi_{l+1}^{(i)}(k,n_l),2\leq l\leq L\\ \phi_{L+1}^{(i)}(k,n_L)=1.\end{cases} \tag{41}$$

Then, we can see that

$$\begin{cases} \psi_l^{(i)}(k, n_{l+1}) \geq \rho_1 \left(\prod_{n=1}^{n_{l+1}-1} \xi_n^{(l)}(i) \psi_{l-1}^{(i)}(k, n_{l+1}) \right), L-1 \geq l \geq 1 \\ \psi_l^{(i)}(k, n_{l+1}) \geq \rho_1 \left(\prod_{n=1}^{k-1} \xi_n^{(l)}(i) \psi_{l-1}^{(i)}(k, k) \right), \\ \phi_l^{(i)}(k, n_{l-1}) \geq \rho_2 \left(\prod_{n=1}^{n_{l-1}-1} \xi_n^{(l)}(i) \phi_{l+1}^{(i)}(k, n_{l-1}) \right), 2 \leq l \leq L \\ \phi_l^{(i)}(k, n_{l-1}) \geq \rho_2 \left(\prod_{n=1}^{k-1} \xi_n^{(l)}(i) \phi_{l+1}^{(i)}(k, k) \right). \end{cases} \tag{42}$$

Applying equation (42) repeatedly, we have

$$H_k^{(i)} \geq \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} \frac{(1-\rho) f_0(x_{s,n})}{(1-\rho_1)(1-\rho_2) f_i(x_{s,n})} \cdot \left[\rho_1^{s-1} \left(\prod_{t=1}^{s-1} \prod_{n=1}^{n_s-1} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right) (1-\rho_1)^{n_s-1} \right] \cdot$$
$$\left[\rho_2^{L-s-1} \left(\prod_{t=s+1}^{L} \prod_{n=1}^{n_s-1} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right) (1-\rho_2)^{n_s-1} \right] = \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1} \right) \left(\sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} (1-\rho) \prod_{t=1}^{L} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right).$$

Then we have

$$\frac{1}{k} \log H_k^{(i)} \geq \frac{1}{k} \log \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1} \right) + \frac{1}{k} \log \left(\sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} (1-\rho) \prod_{t=1}^{L} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right). \tag{43}$$

Since the parameters ρ_{s-1} are all positive for all $1 \leq s \leq L$, we have

$$\frac{1}{k} \log \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1} \right) \xrightarrow[k \to \infty]{} 0. \tag{44}$$

Since the change will happen at all sensors at an almost surely finite time T, then by applying Lemma 2, we have

$$\begin{aligned} &\frac{1}{k} \log \left(\sum_{n_l=0}^{k} \prod_{n=1}^{n_l-1} (1-\rho) \prod_{t=1}^{L} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right) = \frac{1}{k} \log \left(2 + \sum_{n_l=2}^{k} \prod_{n=1}^{n_l-1} (1-\rho) \prod_{t=1}^{L} \frac{f_0(x_{t,n})}{f_i(x_{t,n})} \right) \\ &\xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} (\log(1-\rho) + Lq(j,i) - Lq(j,0))_+. \end{aligned} \tag{45}$$

Combining (43), (44) and (45), we can see that (37) is proved. Next we need to prove the other direction, i.e., for any $i, j \in \mathcal{I}$,

$$\limsup_{k \to \infty} \frac{1}{k} \log H_k^{(i)} \leq (\log(1-\rho) + Lq(j,i) - Lq(j,0))_+, \tag{46}$$

$\mathbb{P}_j$ almost surely.

For any integer $n_x \geq 0$, we can see that for $L-1 \geq l \geq 1$

$$\prod_{n=1}^{k} (1-\rho_1) \prod_{t=1}^{l} \xi_n^{(t)}(i) = \left(\prod_{n=1}^{k} \xi_n^{(l)}(i) \right) \left(\prod_{n=1}^{k} (1-\rho_1) \prod_{t=1}^{l-1} \xi_n^{(t)}(i) \right) \tag{47}$$

$$\leq \left(\prod_{n=1}^{k} \xi_n^{(l)}(i) \right) \psi_{l-1}^{(i)}(k, n_x). \tag{48}$$

Similarly, for $2 \leq l \leq L$ we have

$$\prod_{n=1}^{k} (1-\rho_2) \prod_{t=l}^{L} \xi_n^{(t)}(i) \leq \left(\prod_{n=1}^{k} \xi_n^{(l)}(i) \right) \phi_{l+1}^{(i)}(k, n_x). \tag{49}$$

From (41), using (48) and (49) with $n_x = k+1$ and the fact that $\rho_1 < 1$ and $\rho_2 < 1$, we can see that

$$\begin{cases} \psi_l^{(i)}(k, n_{l+1}) \leq \sum\limits_{n_l = n_{l+1}}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \psi_{l-1}^{(i)}(k, n_l), L-1 \geq l \geq 1 \\ \phi_l^{(i)}(k, n_{l-1}) \leq \sum\limits_{n_l = n_{l-1}}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \phi_{l+1}^{(i)}(k, n_l), 2 \leq l \leq L. \end{cases} \tag{50}$$

Applying these two inequalities in (50) recursively, we have

$$H_k^{(i)} \leq \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \left(\prod_{n=1}^{n_s - 1} \left(\frac{(1-\rho)\xi_k^{(s)}(i)}{(1-\rho_1)(1-\rho_2)} \right) \right) \cdot \tilde{\psi}_{s-1}^{(i)}(k, n_s) \tilde{\phi}_{s+1}^{(i)}(k, n_s) \tag{51}$$

where

$$\begin{cases} \tilde{\psi}_l^{(i)}(k, n_{l+1}) = \sum\limits_{n_l = n_{l+1}}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \tilde{\psi}_{l-1}^{(i)}(k, n_l), L-1 \geq l \geq 1 \\ \tilde{\psi}_0^{(i)}(k, n_1) = 1, \\ \tilde{\phi}_l^{(i)}(k, n_{l-1}) = \sum\limits_{n_l = n_{l-1}}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \tilde{\phi}_{l+1}^{(i)}(k, n_l), 2 \leq l \leq L \\ \tilde{\phi}_{L+1}^{(i)}(k, n_L) = 1. \end{cases} \tag{52}$$

Since n_l in (51) is no larger than n_L and n_1 in (52), so the right hand side of (51) will become larger if we cancel all $(1-\rho_1)$ and $(1-\rho_2)$ in (51). Furthermore, we know that

$$\begin{cases} \tilde{\psi}_l^{(i)}(k, n_{l+1}) \leq \sum\limits_{n_l = 0}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \tilde{\psi}_{l-1}^{(i)}(k, n_l), L-1 \geq l \geq 1 \\ \tilde{\phi}_l^{(i)}(k, n_{l-1}) \leq \sum\limits_{n_l = 0}^{k+1} \left(\prod\limits_{n=1}^{n_l - 1} \xi_n^{(l)}(i) \right) \tilde{\phi}_{l+1}^{(i)}(k, n_l), 2 \leq l \leq L. \end{cases} \tag{53}$$

By canceling all $(1-\rho_1)$ and $(1-\rho_2)$ in (51) and applying (53) reversely, we have that $H_k^{(i)} \leq \sum\limits_{s=1}^{L} \kappa_s \gamma_s$ where

$$\gamma_s = \left(\sum_{n_s=0}^{k} \left(\prod_{n=1}^{n_s-1} \left((1-\rho)\xi_k^{(s)}(i) \right) \right) \right) \cdot \left(\prod_{l_1=1}^{s-1} \left(\sum_{t=0}^{k+1} \left(\prod_{n=1}^{t-1} \xi_n^{(l_1)}(i) \right) \right) \right) \cdot \left(\prod_{l_2=s+1}^{L} \left(\sum_{t=0}^{k+1} \left(\prod_{n=1}^{t-1} \xi_n^{(l_2)}(i) \right) \right) \right). \tag{54}$$

By Lemma 2, for any $1 \leq s \leq L$ we have

$$\frac{1}{k} \log(\gamma_s) \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} (L-1)\left(q(j,i) - q(j,0)\right)_+ + \left(\log(1-\rho) + q(j,i) - q(j,0)\right)_+ .$$

Since $\kappa_s \geq 0$ and $\sum\limits_{s=1}^{L} \kappa_s = 1$, we have

$$\min\left(\log\left(\tfrac{\gamma_1}{k}\right), \log\left(\tfrac{\gamma_2}{k}\right), \ldots, \log\left(\tfrac{\gamma_L}{k}\right)\right) \leq \frac{\log\left(\sum\limits_{s=1}^{L} \kappa_s \gamma_s\right)}{k} \leq \max\left(\log\left(\tfrac{\gamma_1}{k}\right), \log\left(\tfrac{\gamma_2}{k}\right), \ldots, \log\left(\tfrac{\gamma_L}{k}\right)\right).$$

We can have

$$\frac{1}{k} \log\left(\sum_{s=1}^{L} \kappa_s \gamma_s\right) \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} (L-1)\left(q(j,i) - q(j,0)\right)_+ + \left(\log(1-\rho) + q(j,i) - q(j,0)\right)_+ .$$

When Condition 1 is satisfied, we have

$$\left(\log(1-\rho) + Lq(j,i) - Lq(j,0)\right)_+ = \left(\log(1-\rho) + q(j,i) - q(j,0)\right)_+ + (L-1)\left(q(j,i) - q(j,0)\right)_+ . \tag{55}$$

Hence (46) is proved. Therefore, Proposition 1 is true.

Appendix B
Proof of Proposition 3

Now, we first prove that, for any $i, j \in \mathcal{I}$,

$$\liminf_{k\to\infty} \frac{1}{k}\log H_k^{(i)} \geq h(i,j) \tag{56}$$

$\mathbb{P}_j$ almost surely. Please note that the $h(i,j)$ in this section is defined as (27) since we are studying the special case.

In (42), we have four inequalities about $\psi_l^{(i)}(k, n_{l+1})$ and $\phi_l^{(i)}(k, n_{l-1})$. For (24), we apply the first inequality of (42) to $\{\psi_l^{(i)}\}_{s-1\geq l\geq a_2^{M(i,j)}(i,j)+1}$, the second inequality of (42) to $\{\psi_l^{(i)}\}_{1\leq l\leq a_2^{M(i,j)}(i,j)}$, the third inequality of (42) to $\{\phi_l^{(i)}\}_{s+1\leq l\leq b_2^{N(i,j)}(i,j)-1}$ and the fourth inequality of (42) to $\{\phi_l^{(i)}\}_{L\geq l\geq b_2^{N(i,j)}(i,j)+1}$. Then we have

$$H_k^{(i)} \geq \sum_{n_s=0}^{k}\prod_{n=1}^{n_s-1}\left(\tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_n^{(l)}(i)\right)\cdot\left(\prod_{l=1}^{a_2^{M(i,j)}(i,j)}\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)\left(\prod_{l=b_2^{N(i,j)}(i,j)}^{L}\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)\cdot$$
$$\rho_1{}^{s-1}\rho_2{}^{L-s}(1-\rho_1)^{k-1}(1-\rho_2)^{k-1}.$$

Therefore,

$$\tfrac{\log H_k^{(i)}}{k} \geq \tfrac{1}{k}\sum_{l=1}^{a_2^{M(i,j)}(i,j)}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\sum_{l=b_2^{N(i,j)}(i,j)}^{L}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\log\left(\rho_1{}^{s-1}\rho_2{}^{L-s}\right)+$$
$$\tfrac{1}{k}\log\left(\sum_{n_s=0}^{k}\prod_{n=1}^{n_s-1}\tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\log\left((1-\rho_1)^{k-1}\right)+\tfrac{1}{k}\log\left((1-\rho_2)^{k-1}\right).$$

Since L is a finite integer, we have

$$\tfrac{1}{k}\log\left(\rho_1{}^{s-1}\rho_2{}^{L-s}\right)\xrightarrow[k\to\infty]{}0. \tag{57}$$

By Lemma 2 and the definition of η_l in (25), we can see that

$$\tfrac{1}{k}\log\left(\sum_{n_s=0}^{k}\prod_{n=1}^{n_s-1}\tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_n^{(l)}(i)\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\eta_l(i,j)\right)_+. \tag{58}$$

In addition, by the definition of η_l and Algorithm 1, we can see that

$$\tfrac{1}{k}\sum_{l=1}^{a_2^{M(i,j)}(i,j)}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\log\left((1-\rho_1)^{k-1}\right)+\tfrac{1}{k}\sum_{l=b_2^{N(i,j)}(i,j)}^{L}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\log\left((1-\rho_2)^{k-1}\right)$$
$$\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\sum_{l=1}^{a_2^{M(i,j)}(i,j)}\eta_l(i,j)+\sum_{l=b_2^{N(i,j)}(i,j)}^{L}\eta_l(i,j). \tag{59}$$

Combining (57), (58) and (59), (56) is proved. Next, we need to prove the other direction, i.e., for any $i, j \in \mathcal{I}$,

$$\liminf_{k\to\infty}\frac{1}{k}\log H_k^{(i)} \leq h(i,j) \tag{60}$$

$\mathbb{P}_j$ almost surely.

Applying (50) recursively, we have

$$H_k^{(i)} \leq \sum_{n_l=0}^{k}\left(\prod_{n=1}^{n_l-1}\left(\frac{(1-\rho)\xi_k^{(l)}(i)}{(1-\rho_1)(1-\rho_2)}\right)\right)\cdot\tilde{\psi}_{l-1}^{(i)}(k,n_l)\tilde{\phi}_{l+1}^{(i)}(k,n_l). \tag{61}$$

Here $\tilde{\phi}^{(i)}_{l+1}(k,n_l)$ and $\tilde{\psi}^{(i)}_{l-1}(k,n_l)$ are given (52). We apply the first inequality in (53) to $\tilde{\psi}^{(i)}_{l+1}(k,n_l)$ for $l=a_2^m(i,j)$ and $1\le m\le M(i,j)$, following the order from $m=1$ to $m=M(i,j)$. Then we also apply the second inequality in (53) to $\tilde{\phi}^{(i)}_{l+1}(k,n_l)$ for $l=b_2^n(i,j)$ and $1\le n\le N(i,j)$, following the order from $n=1$ to $n=N(i,j)$. We define

$$\begin{cases} \Omega_m = \sum\limits_{n_{a_2^m(i,j)}=0}^{k+1}\left[\left(\prod\limits_{n=1}^{n_{a_2^m(i,j)}-1}\xi_n^{(a_2^m(i,j))}(i)\right)\zeta^{(i)}_{m,a_2^m(i,j)-1}(n_{a_2^m(i,j)})\right], 1\le m\le M(i,j),\\ \Theta_m = \sum\limits_{n_{b_2^n(i,j)}=0}^{k+1}\left[\left(\prod\limits_{m=1}^{n_{b_2^n(i,j)}-1}\xi_m^{(b_2^n(i,j))}(i)\right)\varepsilon^{(i)}_{n,b_2^n(i,j)+1}(n_{b_2^n(i,j)})\right], N(i,j)\ge n\ge 1,\\ \Gamma = \sum\limits_{n_s=0}^{k+1}\left[\left(\prod\limits_{n=1}^{n_s-1}\xi_n^{(s)}(i)\right)\zeta^{(i)}_{M(i,j)+1,s-1}(n_s)\varepsilon^{(i)}_{N(i,j)+1,s+1}(n_s)\right] \end{cases} \tag{62}$$

where

$$\begin{cases} \zeta^{(i)}_{m,t}(n_{t+1}) = \sum\limits_{n_t=n_{t+1}}^{k+1}\left[\left(\prod\limits_{n=1}^{n_t-1}\xi_n^{(t)}(i)\right)\zeta^{(i)}_{m,t-1}(n_t)\right], a_2^m(i,j)-1\ge t\ge a_1^m(i,j), 1\le m\le M(i,j)+1,\\ \zeta^{(i)}_{m,t}(n_{t+1}) = 1, t=a_1^m(i,j)-1,\\ \varepsilon^{(i)}_{n,t}(n_{t-1}) = \sum\limits_{n_t=n_{t-1}}^{k+1}\left[\left(\prod\limits_{m=1}^{n_t-1}\xi_m^{(t)}(i)\right)\varepsilon^{(i)}_{n,t+1}(n_t)\right], b_1^n(i,j)\ge t\ge b_2^n(i,j)+1, N(i,j)+1\ge n\ge 1. \end{cases} \tag{63}$$

With the definitions in (62), we have

$$H_k^{(i)} \le \left(\prod_{m=1}^{M(i,j)}\Omega_m\right)\left(\prod_{n=1}^{N(i,j)}\Theta_n\right)\Gamma. \tag{64}$$

In (62), we denote

$$\begin{cases} a_1^{M(i,j)+1}(i,j) = a_2^{M(i,j)}(i,j)+1,\\ a_2^{M(i,j)+1}(i,j) = s-1,\\ b_1^{N(i,j)+1}(i,j) = b_2^{N(i,j)}(i,j)-1,\\ b_2^{N(i,j)+1}(i,j) = s+1 \end{cases}. \tag{65}$$

Now, it suffices to show that

$$\begin{cases} \limsup\limits_{k\to\infty}\frac{1}{k}\log\Omega_m = \sum\limits_{l=a_1^m(i,j)}^{a_2^m(i,j)}\eta_l(i,j), 1\le m\le M(i,j)\\ \limsup\limits_{k\to\infty}\frac{1}{k}\log\Theta_m = \sum\limits_{l=b_2^n(i,j)}^{b_1^n(i,j)}\eta_l(i,j), 1\le n\le N(i,j)\\ \limsup\limits_{k\to\infty}\frac{1}{k}\log\Gamma = \sum\limits_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\eta_l(i,j) \end{cases} \tag{66}$$

$\mathbb{P}_j$ almost surely. The proof of the three inequalities are similar, and the third one is more complicated. So here we only provide the proof of the third one. For any $1\le l\le L$,

$$\begin{aligned} &\sum_{n_l=n_{l-1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi^{(l)}_{m_l}\left(\sum_{n_{l+1}=n_l}^{k+1}\prod_{m_{l+1}=1}^{n_{l+1}-1}\xi^{(l+1)}_{m_{l+1}}\right) = \sum_{n_l=n_{l-1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi^{(l)}_{m_l}\xi^{(l+1)}_{m_l}\left(\sum_{n_{l+1}=n_l}^{k+1}\prod_{m_{l+1}=n_l}^{n_{l+1}-1}\xi^{(l+1)}_{m_{l+1}}\right)\\ &\le \sum_{n_l=n_{l-1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi^{(l)}_{m_l}\xi^{(l+1)}_{m_l}\left(\sum_{n_{l+1}=n_l}^{k+n_l+1}\prod_{m_{l+1}=n_l}^{n_{l+1}-1}\xi^{(l+1)}_{m_{l+1}}\right) \le \sum_{n_l=n_{l-1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi^{(l)}_{m_l}\xi^{(l+1)}_{m_l}\left(\max_{n_l\le k+1}\sum_{n_{l+1}=n_l}^{k+n_l+1}\prod_{m_{l+1}=n_l}^{n_{l+1}-1}\xi^{(l+1)}_{m_{l+1}}\right). \end{aligned} \tag{67}$$

Similarly, we have

$$\sum_{n_l=n_{l+1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi_{m_l}^{(l)}\left(\sum_{n_{l-1}=n_l}^{k+1}\prod_{m_{l-1}=1}^{n_{l-1}-1}\xi_{m_{l-1}}^{(l-1)}\right)\leq\sum_{n_l=n_{l+1}}^{k+1}\prod_{m_l=1}^{n_l-1}\xi_{m_l}^{(l)}\xi_{m_l}^{(l-1)}\left(\max_{n_l\leq k+1}\sum_{n_{l-1}=n_l}^{k+n_l+1}\prod_{m_{l-1}=n_l}^{n_{l-1}-1}\xi_{m_{l-1}}^{(l-1)}\right). \tag{68}$$

For Γ, apply (68) from $l=a_2^{M(i,j)}(i,j)+2$ to $l=s$, then apply (67) from to $l=b_2^{N(i,j)}(i,j)-2$ to $l=s$, we have

$$\Gamma\leq\sum_{n_s=0}^{k+1}\left(\prod_{q=1}^{n_s-1}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_q^{(l)}(i)\right)\prod_{l=a_2^{M(i,j)}(i,j)+1}^{s}C_l\prod_{l=s}^{b_2^{N(i,j)}(i,j)-1}D_l, \tag{69}$$

where

$$\begin{cases} C_l=\max\limits_{n_l\leq k+1}\left(1+\sum\limits_{n_{l-1}=n_l}^{k+n_l}\prod\limits_{m_{l-1}=n_l}^{n_{l-1}}\prod\limits_{t=a_2^{M(i,j)}(i,j)}^{l}\xi_{m_{l-1}}^{(t)}\right), \\ D_l=\max\limits_{n_l\leq k+1}\left(1+\sum\limits_{n_{l+1}=n_l}^{k+n_l}\prod\limits_{m_{l+1}=n_l}^{n_{l+1}}\prod\limits_{t=l}^{b_2^{N(i,j)}(i,j)}\xi_{m_{l+1}}^{(t)}\right). \end{cases}$$

By Lemma 2, for $s\leq l\leq b_2^{N(i,j)}(i,j)-1$, we have

$$\frac{1}{k}\log\left(1+\sum_{n_{l+1}=n_l}^{k+n_l}\prod_{m_{l+1}=n_l}^{n_{l+1}}\prod_{t=l}^{b_2^{N(i,j)}(i,j)}\xi_{m_{l+1}}^{(l)}\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\left(\sum_{t=l}^{b_2^{N(i,j)}(i,j)}\eta_t(i,j)\right)_+=0.$$

And for $a_2^{M(i,j)}(i,j)+1\leq l\leq s$, we have

$$\frac{1}{k}\log\left(1+\sum_{n_{l-1}=n_l}^{k+n_l}\prod_{m_{l-1}=n_l}^{n_{l-1}}\prod_{t=a_2^{M(i,j)}(i,j)}^{l}\xi_{m_{l-1}}^{(t)}\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\left(\sum_{t=a_2^{M(i,j)}(i,j)}^{l}\eta_t(i,j)\right)_+=0.$$

Therefore, we have

$$\begin{cases} \frac{1}{k}\log C_l\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}0, \\ \frac{1}{k}\log D_l\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}0. \end{cases}$$

Similarly, by lemma 2, we can see that,

$$\sum_{n_s=0}^{k+1}\left(\prod_{q=1}^{n_s-1}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_q^{(l)}(i)\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\eta_l(i,j)\right)_+. \tag{70}$$

By (69), (B) and (70), we know that the third inequality in (66) is true. Using similar steps, we can prove the other two inequalities in (66). Hence (56) is proved. Finally, by (56) and (60), the proof of Proposition 3 is complete.

APPENDIX C
PROOF OF PROPOSITION 4

we first prove that, for any $i, j \in \mathcal{I}$,

$$\liminf_{k\to\infty} \frac{1}{k} \log H_k^{(i)} \geq h(i,j) \tag{71}$$

$\mathbb{P}_j$ almost surely. Please note that the $h(i,j)$ in this section is defined in Proposition 4 since we are studying the 2D case.

Then, we can see that

$$\psi_{r+1}^{(i)}\left(k, n_r, S_1, S_2\right) \geq \rho_1 \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \psi_{r+2}^{(i)}\left(k, n_r, S_1, S_2\right), R(S_1,S_2) > r \geq 0 \;. \tag{72}$$

Applying equation (72) repeatedly, we have

$$\begin{aligned} H_k^{(i)} &\geq \sum_{S_1=1}^{H}\sum_{S_1=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k}\prod_{n=1}^{n_0-1} \frac{(1-\rho)f_0(x_{n,S_1,S_2})}{(1-\rho_1)f_i(x_{n,S_1,S_2})} \cdot \left[\rho_1^{R(S_1,S_2)-1}\left(\prod_{(a,b)\in\mathcal{O}(S_1,S_2,0)}\prod_{n=1}^{n_0-1}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right)(1-\rho_1)^{n_0-1}\right] \\ &= \left(\sum_{S_1=1}^{H}\sum_{S_1=1}^{W}\kappa_{S_1,S_2}\rho_1^{R(S_1,S_2)-1}\right)\left(\sum_{n_0=0}^{k}\prod_{n=1}^{n_0-1}(1-\rho)\prod_{a=1}^{H}\prod_{b=1}^{W}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right). \end{aligned} \tag{73}$$

Since $R(S_1,S_2)$ is finite for $1 \leq S_1 \leq H, 1 \leq S_2 \leq W$, we have

$$\frac{1}{k}\log\left(\sum_{S_1=1}^{H}\sum_{S_1=1}^{W}\kappa_{S_1,S_2}\rho_1^{R(S_1,S_2)-1}\right) \xrightarrow[k\to\infty]{} 0. \tag{74}$$

Since the change will happen at all sensors at an almost surely finite time T, then by applying Lemma 2, we have

$$\begin{aligned} &\frac{1}{k}\log\left(\sum_{n_0=0}^{k}\prod_{n=1}^{n_0-1}(1-\rho)\prod_{a=1}^{H}\prod_{b=1}^{W}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right) = \frac{1}{k}\log\left(2+\sum_{n_0=2}^{k}\prod_{n=1}^{n_0-1}(1-\rho)\prod_{a=1}^{H}\prod_{b=1}^{W}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right) \\ &\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} \left(\log(1-\rho)+HWq(j,i)-HWq(j,0)\right)_+. \end{aligned} \tag{75}$$

Combining (73), (74) and (75), we can see that (71) is proved. Next we need to prove the other direction, i.e., for any $i, j \in \mathcal{I}$,

$$\limsup_{k\to\infty}\frac{1}{k}\log H_k^{(i)} \leq h(i,j). \tag{76}$$

For any integer $n_x \geq 0$, we can see that

$$\begin{aligned} &\prod_{n=1}^{k}(1-\rho_1)\prod_{(a,b)\in\mathcal{O}(S_1,S_2,r)}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \\ &= \left(\prod_{n=1}^{k}\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right)\left(\prod_{n=1}^{k}(1-\rho_1)\prod_{(a,b)\in\mathcal{O}(S_1,S_2,r+1)}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right) \\ &\leq \left(\prod_{n=1}^{k}\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right)\psi_{r+2}^{(i)}\left(k, n_x, S_1, S_2\right). \end{aligned} \tag{77}$$

From (33), using (77) with $n_x = k+1$ and the fact that $\rho_1 < 1$, we can see that

$$\psi_{r+1}^{(i)}\left(k, n_r, S_1, S_2\right) \leq \sum_{n_{r+1}=n_r}^{k+1}\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)}\prod_{n=1}^{n_r-1}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\psi_{r+2}^{(i)}\left(k, n_{r+1}, S_1, S_2\right), R(S_1,S_2) > r \geq 0. \tag{78}$$

Applying these two inequalities in (78) recursively, we have

$$H_k^{(i)} \leq \sum_{S_1=1}^{H} \sum_{S_1=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} \frac{(1-\rho) f_0(x_{n,S_1,S_2})}{(1-\rho_1) f_i(x_{n,S_1,S_2})} \cdot \tilde{\psi}_1^{(i)}(k, n_0, S_1, S_2) \tag{79}$$

where

$$\begin{cases} \tilde{\psi}_{r+1}^{(i)}(k, n_r, S_1, S_2) = \sum_{n_{r+1}=n_r}^{k+1} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_{r+1}-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \tilde{\psi}_{r+2}^{(i)}(k, n_{r+1}, S_1, S_2), R(S_1,S_2) > r \geq 0 \\ \tilde{\psi}_{R(S_1,S_2)+1}^{(i)}(k, n_r, S_1, S_2) = (1-\rho_1)^{n_{R(S_1,S_2)}-1}. \end{cases} \tag{80}$$

Since n_0 in (79) is no larger than $n_{R(S_1,S_2)}$ in (80), so the right hand side of (79) will become larger if we cancel all $(1-\rho_1)$ in (79).

Furthermore, we know that

$$\left\{ \tilde{\psi}_{r+1}^{(i)}(k, n_r, S_1, S_2) \leq \sum_{n_{r+1}=0}^{k+1} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_{r+1}-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \tilde{\psi}_{r+2}^{(i)}(k, n_{r+1}, S_1, S_2), R(S_1,S_2) > r \geq 0. \right. \tag{81}$$

By canceling all $(1-\rho_1)$ in (79) and applying (81) reversely, we have that $H_k^{(i)} \leq \sum\limits_{S_1=1}^{H} \sum\limits_{S_2=1}^{W} \kappa_{a,b}\gamma_{S_1,S_2}$ where

$$\gamma_{S_1,S_2} = \left(\sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} \frac{(1-\rho) f_0(x_{n,S_1,S_2})}{f_i(x_{n,S_1,S_2})} \right) \cdot \left(\prod_{r=1}^{R(S_1,S_2)} \left(\sum_{t=0}^{k+1} \left(\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{t-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) \right) \right).$$

By Lemma 2, for any $1 \leq S_1 \leq H$ and $1 \leq S_2 \leq W$, we have

$$\frac{1}{k} \log(\gamma_{S_1,S_2}) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} (HW-1)(q(j,i)-q(j,0))_+ + (\log(1-\rho)+q(j,i)-q(j,0))_+.$$

Since $\kappa_{a,b} \geq 0$ and $\sum\limits_{1\leq a\leq H, 1\leq b\leq W} \kappa_{a,b} = 1$, we have

$$\begin{aligned} &\min\left(\log\left(\tfrac{\gamma_{1,1}}{k}\right), \log\left(\tfrac{\gamma_{1,2}}{k}\right), \ldots, \log\left(\tfrac{\gamma_{H,W}}{k}\right)\right) \leq \frac{\log\left(\sum\limits_{S_1=1}^{H}\sum\limits_{S_2=1}^{W} \kappa_{S_1,S_2}\gamma_{S_1,S_2}\right)}{k} \\ &\leq \max\left(\log\left(\tfrac{\gamma_{1,1}}{k}\right), \log\left(\tfrac{\gamma_{1,2}}{k}\right), \ldots, \log\left(\tfrac{\gamma_{H,W}}{k}\right)\right). \end{aligned}$$

We can have

$$\frac{1}{k}\log\left(\sum_{S_1=1}^{H}\sum_{S_2=1}^{W} \kappa_{S_1,S_2}\gamma_{S_1,S_2}\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} (HW-1)(q(j,i)-q(j,0))_+ + (\log(1-\rho)+q(j,i)-q(j,0))_+.$$

When Condition 1 is satisfied, we have

$$\begin{aligned} &(\log(1-\rho) + HWq(j,i) - HWq(j,0))_+ = (\log(1-\rho)+ \\ &q(j,i)-q(j,0))_+ + (HW-1)(q(j,i)-q(j,0))_+. \end{aligned} \tag{82}$$

Hence (76) is proved. Therefore, Proposition 4 is true.

REFERENCES

[1] X. Ma, L. Lai, and S. Cui, “Bayesian two-stage sequential change diagnosis via multi-sensor array,” in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. Gold Coast, Australia, Oct. 2021, pp. 1–6.

[2] T. Lau, W. Tay, and V. V. Veeravalli, “Quickest change detection with unknown post-change distribution,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA, Mar. 2017, pp. 3924–3928.

[3] J. Geng and L. Lai, “Quickest change-point detection over multiple data streams via sequential observations,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, Sept. 2018, pp. 4404–4408.

[4] T. Banerjee and V. V. Veeravalli, “Data-efficient minimax quickest change detection,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan, Mar. 2012, pp. 3937–3940.

[5] Y. Xie and D. Siegmund, “Sequential multi-sensor change-point detection,” in *Proc. Information Theory and Applications Workshop*. San Diego, CA, May. 2013, pp. 1–20.

[6] H. V. Poor and O. Hadjiliadis, *Quickest detection*. Cambridge University Press, 2008.

[7] Y. Wang and Y. Mei, “Large-scale multi-stream quickest change detection via shrinkage post-change estimation,” *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6926–6938, Dec. 2015.

[8] J. Geng and L. Lai, “Bayesian quickest detection with unknown post-change parameter,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China, May 2016, pp. 4169–4173.

[9] L. Lai, “Quickest change point identification across a sensor array,” in *Proc. IEEE Military Communications Conference (MILCOM)*. Orlando, FL, Jan. 2012, pp. 1–5.

[10] D. Li, L. Lai, and S. Cui, “Quickest change detection and identification across a sensor array,” in *Proc. IEEE Global Conference on Signal and Information Processing*. Austin, TX, Dec. 2013, pp. 145–148.

[11] V. Raghavan and V. V. Veeravalli, “Quickest change detection of a markov process across a sensor array,” *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1961–1981, Mar. 2010.

[12] S. Zou, V. V. Veeravalli, J. Li, and D. Towsley, “Quickest detection of dynamic events in networks,” *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2280–2295, Apr. 2020.

[13] A. N. Shiryaev, “On optimum methods in quickest detection problems,” *Theory of Probability & Its Applications*, vol. 8, no. 1, pp. 22–46, 1963.

[14] A. Novikov, “Optimal sequential multiple hypothesis tests,” *Kybernetika*, vol. 45, no. 2, pp. 309–330, 2009.

[15] S. Nitinawarat, G. K. Atia, and V. V. Veeravalli, “Controlled sensing for multihypothesis testing,” *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2451–2464, May. 2013.

[16] J. Heydari and A. Tajer, “Controlled sensing for multi-hypothesis testing with co-dependent actions,” in *Proc. IEEE International Symposium on Information Theory*. Vail, CO, Jun. 2018, pp. 2321–2325.

[17] M. R. Leonard, M. Stiefel, M. Fauβ, and A. M. Zoubir, “Robust sequential testing of multiple hypotheses in distributed sensor networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, Sept. 2018, pp. 4394–4398.

[18] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and H. Kim, “A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods,” *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3372–3382, Sept. 2006.

[19] Y. C. Chen, T. Banerjee, A. D. Domínguez-García, and V. V. Veeravalli, “Quickest line outage detection and identification,” *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 749–758, Feb. 2015.

[20] H. Li, C. Li, and H. Dai, “Quickest spectrum sensing in cognitive radio,” in *Proc. IEEE Annual Conference on Information Sciences and Systems*. Princeton, NJ, Mar. 2008, pp. 203–208.

[21] S. Dayanik, C. Goulding, and H. V. Poor, “Bayesian sequential change diagnosis,” *Mathematics of Operations Research*, vol. 33, no. 2, pp. 475–496, May. 2008.

[22] S. Dayanik, W. B. Powell, and K. Yamazaki, “Asymptotically optimal Bayesian sequential change detection and identification rules,” *Annals of Operations Research*, vol. 208, no. 1, pp. 337–370, Sept. 2013.

[23] X. Ma, L. Lai, and S. Cui, “Optimal two-stage Bayesian sequential change diagnosis,” in *Proc. IEEE International Symposium on Information Theory (ISIT)*. Los Angles, CA, Jun. 2020, pp. 1130–1135.

[24] ——, “Two-stage Bayesian sequential change diagnosis,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 6131–6147, Sept. 2021.

[25] D. Tibaduiza Burgos, R. Gomez Vargas, C. Pedraza, D. Agis, and F. Pozo, “Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications,” *Sensors*, vol. 20, no. 3, p. 733, Jan. 2020.

[26] L. Fillatre and I. Nikiforov, “Asymptotically uniformly minimax detection and isolation in network monitoring,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3357–3371, Apr. 2012.

[27] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[28] X. Ma, L. Lai, and S. Cui, “Optimal two-stage Bayesian sequential change diagnosis,” in *Proc. IEEE International Symposium on Information Theory*. Los Angles, CA, June 2020, pp. 1130–1135.