Personalized and Energy-Efficient Health Monitoring: A Reinforcement Learning Approach

Batchen Eden, Ilai Bistritz, *Graduate Student Member, IEEE*, Nicholas Bambos, *Member, IEEE*, Irad Ben-Gal, and Evgeni Khmelnitsky

***Abstract*—We consider a network of controlled sensors that monitor the unknown health state of a patient. We assume that the health state process is a Markov chain with a transition matrix that is unknown to the controller. At each timestep, the controller chooses a subset of sensors to activate, which incurs an energy (i.e., battery) cost. Activating more sensors improves the estimation of the unknown state, which introduces an energy-accuracy tradeoff. Our goal is to minimize the combined energy and state misclassification costs over time. Activating sensors now also provides measurements that can be used to learn the model, improving future decisions. Therefore, the learning aspect is intertwined with the energy-accuracy tradeoff. While Reinforcement Learning (RL) is often used when the model is unknown, it cannot be directly applied in health monitoring since the controller does not know the (health) state. Therefore, the monitoring problem is a partially observable Markov decision process (POMDP) where the cost feedback is also only partially available since the misclassification cost is unknown. To overcome this difficulty, we propose a monitoring algorithm that combines RL for POMDPs and online estimation of the expected misclassification cost based on a Hidden Markov Model (HMM). We show empirically that our algorithm achieves comparable performance with a monitoring system that assumes a known transition matrix and quantizes the belief state. It also outperforms the model-based approach where the estimated transition matrix is used for value iteration. Thus, our algorithm can be useful in designing energy-efficient and personalized health monitoring systems.**

***Index Terms*—Reinforcement learning, WBAN, health monitoring, POMDP.**

I. Introduction

WIRELESS Body Area Networks (WBANs) are medical sensor networks that monitor the health state of the patient in real-time [1], [2], [3]. WBANs enable real-time data-driven healthcare, which can provide much more accurate guidance for medical decision-making. Medical decisions, however, need to be based on high-accuracy measurements. On the other hand, wireless wearable sensors are battery-limited and can be hard to replace (e.g., if implanted under the skin). This makes the accuracy-energy tradeoff a key consideration in designing WBANs that enable personalized real-time healthcare.

Patient monitoring is a dynamic decision-making problem where the goal is to estimate an unknown health state. At each timestep, we can choose which subset of sensors to activate. Activating more sensors will improve the estimation accuracy of the current state, but deplete energy needed to monitor future states. By exploiting the structure of the dynamics, we can learn to save energy in states where misclassification is inherently less likely. Hence, the total cost combines the misclassification and energy costs amassed over time.

In cases where the patient's health state model is considered known, it has usually been estimated using past data of other patients. Such a model-based approach is less accurate when dealing with medical conditions that behave considerably differently between patients. One recent example is COVID-19, which exhibited significantly assorted symptoms in diverse patients [4]. Personalized healthcare can lead to more accurate medical decisions even in diseases where the health state model is traditionally assumed to be known [5].

The model-based approach was taken in [1], [6], which assumed that the transition matrix of the health state Markov chain is known or can be initially estimated. Assuming that a group of patients share the same health state model, [7] proposed an algorithm that utilizes the measurements across this group to learn the model online. However, such a homogeneous group of patients is often unrealistic, and assuming its existence precludes personalized healthcare.

In this letter, we relax the assumption that the health model is known, thus enabling a personalized WBAN that can monitor medical conditions where the patients' characteristics vary considerably. This adds the dimension of learning to the already complicated tradeoff between accuracy and energy.

Having to learn how to monitor introduces another tradeoff, between exploration and exploitation. The system needs to learn the dynamics better to save more energy in the future, but this learning itself wastes energy. Moreover, learning the

Manuscript received 7 September 2022; revised 11 November 2022; accepted 27 November 2022. Date of publication 14 December 2022; date of current version 23 December 2022. This work was supported by the Koret Foundation Grant for Smart Cities and Digital Living. Recommended by Senior Editor M. Guay. *(Corresponding author: Ilai Bistritz.)*

Batchen Eden is with the Department of Statistics, Tel Aviv University, Tel Aviv 69978, Israel.

Ilai Bistritz and Nicholas Bambos are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: bistritz@stanford.edu; bambos@stanford.edu).

Irad Ben-Gal and Evgeni Khmelnitsky are with the Department of Industrial Engineering, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: bengal@tauex.tau.ac.il; xmel@tauex.tau.ac.il).

Digital Object Identifier 10.1109/LCSYS.2022.3229074

dynamics means that our estimation of the misclassification cost which drives our decisions, is never perfect.

RL [8] is designed to solve dynamic problems with an unknown model. Implementing RL in health monitoring, however, is challenging for two key reasons: the cost is not given as feedback and the state is only partially observable.

Most RL algorithms assume a Markov decision process (MDP) where the controller can observe the state. In the monitoring problem, the state is only partially observable. To overcome this obstacle, we employ the *Utile Suffix Memory* (USM) algorithm [9], which maintains an inner state-space represented by a suffix tree, based on past observations. The USM algorithm then computes the optimal Q-values ('Utility') for these inner states. However, like any RL algorithm, USM still needs the cost as feedback.

Unique to the monitoring problem, the control decisions do not affect the state but only our knowledge of the state. Since the true health state is unknown, the misclassification cost is also unknown to the controller. When the model is known, we can compute the expected misclassification cost with respect to the random state. When the model is unknown, then the cost feedback, which is necessary for RL, is not available.

We present a novel monitoring system that can deal with both a POMDP and partial cost feedback, which are the main difficulties in introducing RL into WBAN monitoring. Our algorithm combines RL for POMDPs and online estimation of the misclassification cost based on a hidden Markov model (HMM). We show empirically that our algorithm achieves comparable performance with a quantization of the optimal belief-MDP solution that knows the Markov health model. Hence, our work is a step forward toward energy-efficient, adaptive and personalized WBAN monitoring systems.

II. Problem Formulation

Consider J health states a patient can occupy. We typically refer to w^i as a healthier state than w^j if $i < j$, where the superscript denotes the state. We assume that the unknown health state process w_t that we wish to monitor is a Markov chain. As such, given w_t, the transition to w_{t+1} is independent of past states. This would be the case if the current health state summarizes all the necessary medical information. Examples of practical Markovian health models are SIS and SIR in epidemiology [10], which have two and three states, respectively. The transition matrix θ of this Markov chain is unknown to the controller since it depends on the patient's unique profile.

Our WBAN consists of N binary sensors. At each timestep t, the controller picks a subset of n sensors to activate in the next timestep, represented by a binary vector $\boldsymbol{a}_t$ of size N. As a function of $\boldsymbol{a}_t$ and the unknown w_{t+1}, we observe the sensors' output $\boldsymbol{o}_{t+1}$, where $\boldsymbol{o}_{t+1} \in \{\emptyset, 0, 1\}^N$, $o^n = \emptyset$ denotes a deactivated sensor (i.e., $a^n = 0$) and $o^n \in \{0, 1\}$ otherwise. We assume that the sensors have a known accuracy matrix E, where $E_{nj} = p(o^n = 1 | w = w^j)$ for the n-th sensor and the j-th health state. These probabilities are specified by the manufacturer and depend on the sensor technology.

The monitoring problem is an interesting instance of a general POMDP [11] in which the selected action does not impact the true state, but does affect the amount of information we have on this state. In health monitoring, the controller chooses to activate the sensors that would provide the best measurements to estimate the health state, while maintaining efficient power consumption. The controller's goal is to minimize the expected accumulated discounted cost:

$$\mathbb{E}\left\{\sum_{t=1}^{\infty} \gamma^t (C_a(\boldsymbol{a}_{t-1}) + \rho_t(w_t))\right\} \tag{1}$$

for some discount factor γ, $0 < \gamma < 1$.

The instantaneous cost c_t in (1) combines two types of costs. The energy cost $C_a(\boldsymbol{a}_{t-1})$ results from activating the subset of sensors $\boldsymbol{a}_{t-1}$. The misclassification cost at time t $\rho_t(w_t)$ depends on the distribution of w_t, and accounts for wrongly classifying the patient's true health state. In medical applications, miss-detecting is typically more detrimental than false-detecting. Accordingly, we want to assign different costs for these cases:

1) False positive (FP): Occurs when the patient's health state is better (lower) than the predicted state.
2) False negative (FN): Occurs when the patient's state is worse (higher) than predicted.

Let C_{FP}, C_{FN} be the known FP, FN costs. If the estimated state is w^j while the true state is w_t, the misclassification cost is the sum of the FP cost $p(w_t < w^j)C_{FP}$ and FN cost $p(w_t > w^j)C_{FN}$. The magnitudes of C_{FP}, C_{FN} can be tuned to balance the energy and misclassification costs.

Since the probabilities $p(w_t = w^j)$ for all j are unknown, we introduce and calculate the probability distribution of the health states given the past observations and actions. We thus define the misclassification cost of w^j as:

$$\rho_t(w_t = w^j) = C_{FP} \sum_{j'=1}^{j-1} p(w_t = w^{j'} | \boldsymbol{a}_{0:t-1}, \boldsymbol{o}_{1:t}) + C_{FN} \sum_{j'=j+1}^{J} p(w_t = w^{j'} | \boldsymbol{a}_{0:t-1}, \boldsymbol{o}_{1:t}) \tag{2}$$

and the expected misclassification cost at time t as:

$$\bar{\rho}(\boldsymbol{a}_{0:t-1}, \boldsymbol{o}_{1:t}) = \sum_{i=1}^{J} \rho_t(w_t = w^i) p(w_t = w^i | \boldsymbol{a}_{0:t-1}, \boldsymbol{o}_{1:t}). \tag{3}$$

An RL controller learns how to act by observing the cost feedback. While the energy cost is known, the misclassification cost is unavailable to the monitoring system, since it requires knowing the distribution of the patient's health state.

III. Model-Free WBAN Monitoring Algorithm

In this section, we describe the three components of our monitoring algorithm (see Figure 1 and Algorithm 1): RL, cost estimation, and *Baum-Welch* HMM (BW-HMM).

Algorithm 1 is run by the external controller that sends the sensors activation commands. Typically, due to physiological constraints, a medical sensor has to stay active for milliseconds or even seconds to take a reliable measurement (e.g., heartbeat

Algorithm 1 WBAN-RL Algorithm

Initialization: Initiate a tree with depth $d = 1$ and $|\mathcal{O}|$ leaves.
For timesteps $t = 1, 2, \ldots, T$

1) Receive sensor output $\boldsymbol{o}_t$, assign instance I_t to leaf and fringe nodes $L(I_t) = s_t$, $F(I_t) = f_t$.
2) Estimate $\hat{\theta}$ using (11).
3) Estimate $p(w_t|s_t, \boldsymbol{o}_t)$ based on $\hat{\theta}$, using (9), (10).
4) Estimate the cost c_t for I_t using (8).
5) Update the Q-value for inner state s_t using (4), (5).
6) Test for significance of the fringe nodes of s_t using the Kolmogorov-Smirnov test with a given p-value threshold.
7) If one of the fringes is significant: create a leaf from each fringe node that satisfies the minimum leaf weight condition $|\mathcal{I}(s)| > 0.1|\cup_{s\in\mathcal{S}}\mathcal{I}(s)|$.
   a) **For** $I_i \in \mathcal{I}(s_t)$:
      i) Assign I_i to new leaf and fringe nodes $s_t = s_{new}$, $f_t = f_{new}$.
      ii) Calculate the Q-value for all s_{new} using (4), (5).
8) With probability ε, select $\boldsymbol{a}_t$ uniformly at random, and otherwise select $\boldsymbol{a}_t = \boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} Q(L(I_t), \boldsymbol{a})$.
9) Update $U(s_t) = \min_{\boldsymbol{a}} Q\big(L(I_t), \boldsymbol{a}^*\big)$.

End

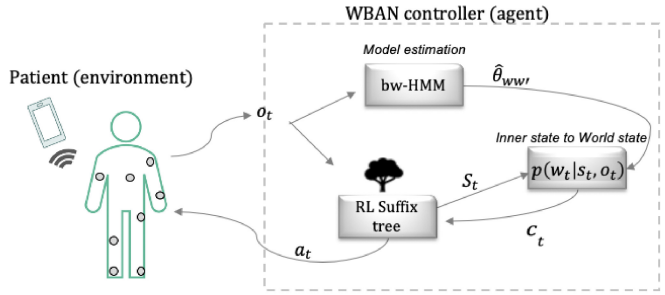

Fig. 1. The observations $\boldsymbol{o}_t$ are used by both the BW-HMM and RL modules. The RL module runs the USM algorithm that assumes the cost feedback is available. However, the misclassification cost is unknown. To overcome that, the cost estimation module estimates this unknown cost based on the coarse model estimation given by BW-HMM. This coarse estimation, however, would not be sufficient for model-based dynamic programming. The controller then chooses which sensors to activate, which determines the next observations.

or blood sugar). Hence, a modest controller (e.g., GHz CPU) has plenty of time to compute the next commands. In applications where a low-complexity algorithm is needed, a greedy version of Algorithm 1 can be used (simulated in Section IV).

A. Inner State-Space and Q-Value

Most RL algorithms assume that the system's state is known, which is not the case in our setting. Hence, we have to treat the monitoring problem as a POMDP. This introduces the challenge of summarizing the observation history.

One way to capture the observation history is with a suffix tree structure. USM is an RL algorithm [9] that offers state representation via an online growing suffix tree, where states are distinguished based on their utility. The algorithm encodes the raw experience history into the tree, where the leaves act as inner states. An instance of the raw experience of time t, $I_t = (\boldsymbol{a}_{t-1}, \boldsymbol{o}_t, c_t)$ is associated with a leaf of depth d whose label is the d-length suffix of the history sequence preceding I_t. That is, I_t belongs to a leaf with the label $[(\boldsymbol{a}_{t-d-1}, \boldsymbol{o}_{t-d}, c_{t-d}), \ldots, (\boldsymbol{a}_{t-2}, \boldsymbol{o}_{t-1}, c_{t-1})]$. Instances of matching history suffixes of a certain length are clustered in leaf nodes, based on how much history is considered significant for each instance, in terms of the utility. As a result, USM can maintain different history lengths for different states. The current inner-state is denoted by s_t, and is a function of the history $\{\boldsymbol{a}_{1:t-2}, \boldsymbol{o}_{1:t-1}\}$. The instance of time t, I_t, is assigned to a leaf $L(I_t) = s_t$ based on its history sequence beginning at time $t-1$ going backward from the root until reaching a leaf. Let the depth of leaf s be $d(s)$, indicating the length of the suffix of the coded history sequence. To solve Bellman's equation, the Q-value $Q(s, \boldsymbol{a})$ of taking action $\boldsymbol{a}$ at inner state s is updated using value iteration as follows:

$$Q(s, \boldsymbol{a}) \leftarrow C(s, \boldsymbol{a}) + \gamma \sum_{s'} \theta^{\boldsymbol{a}}_{ss'} U(s') \tag{4}$$

where $U(s') = \min_{\boldsymbol{a}} Q(s', \boldsymbol{a})$ is the utility of state s'. The instantaneous cost $C(s, \boldsymbol{a})$ and the inner state transition probabilities $\theta^{\boldsymbol{a}}_{ss'}$ are calculated as:

$$C(s, \boldsymbol{a}) = \frac{\sum_{I_i \in \mathcal{I}(s,\boldsymbol{a})} c_i}{|\mathcal{I}(s, \boldsymbol{a})|}$$
$$\theta^{\boldsymbol{a}}_{ss'} = \frac{|\{I_i \in \mathcal{I}(s, \boldsymbol{a}) \mid L(I_{i+1}) = s'\}|}{|\mathcal{I}(s, \boldsymbol{a})|} \tag{5}$$

where $\mathcal{I}(s, \boldsymbol{a})$ is the subset of $\mathcal{I}(s)$ where action $\boldsymbol{a}$ was taken. At each timestep t, the controller selects the action $\boldsymbol{a}_t$ to be taken at the end of the timestep. With some probability $\varepsilon > 0$, $\boldsymbol{a}_t$ is chosen uniformly at random, and otherwise:

$$\boldsymbol{a}_t = \boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} Q(L(I_t), \boldsymbol{a}). \tag{6}$$

B. Cost Estimation

An RL controller expects to observe the cost feedback for its chosen action. In the health monitoring problem, the misclassification cost is unknown and needs to be estimated. To that end, we analyze the relationship of the true health state to the inner state the controller holds. The mapping $O \to A$ of an observation to its generating action is one-to-one, and we denote it by G, so $G(\boldsymbol{o}_t) = \boldsymbol{a}_{t-1}$. For example, if $\boldsymbol{o}_t = (0, 1, \emptyset)$ then the activation vector was $\boldsymbol{a}_{t-1} = (1, 1, 0)$. This is a unique property of the monitoring problem, which does not generally apply to POMDPs. Therefore, given an observation, its associated action provides no additional information:

$$p(w_t|s_t, \boldsymbol{a}_{t-1}, \boldsymbol{o}_t) = p(w_t|s_t, \boldsymbol{o}_t) \tag{7}$$

which implies that the misclassification cost (3) is only a function of $\boldsymbol{o}_{1:t}$. The next proposition provides an expression for the probability of a health state given an inner state, which is used to estimate the expected misclassification cost in (2) and (3).

Proposition 1: The inner-state to health state conditional probability satisfies:

$$p(w_t|s_t, \boldsymbol{o}_t) = \frac{p(\boldsymbol{o}_t|w_t)}{p(\boldsymbol{o}_t|s_t)} \sum_{j'} p(w_t|w_{t-1} = w^{j'}) p(w_{t-1} = w^{j'}|s_t). \tag{8}$$

Proof:

$$p(w_t|s_t,\boldsymbol{o}_t)=\frac{p(w_t,\boldsymbol{o}_t|s_t)}{p(\boldsymbol{o}_t|s_t)}=\frac{p(\boldsymbol{o}_t|w_t,s_t)p(w_t|s_t)}{p(\boldsymbol{o}_t|s_t)}$$
$$=\frac{p(\boldsymbol{o}_t|w_t)}{p(\boldsymbol{o}_t|s_t)}p(w_t|s_t) \tag{9}$$

where we drop s_t in last equality since w_t is Markovian. We obtain (8) using the law of total probability on $p(w_t|s_t)$, dropping s_t from $p(w_t|w_{t-1}=w^{j'})$ using Markovity again. ■

We now discuss how Algorithm 1 estimates (9). First, $p(\boldsymbol{o}|w)$ is the known sensor accuracy E. Then, $p(\boldsymbol{o}|s)$ is estimated as the proportion of instances in leaf s with observation $\boldsymbol{o}$, $\hat{p}(\boldsymbol{o}_t=\boldsymbol{o}|s_t=s)=\frac{|\mathcal{I}(s,\boldsymbol{o})|}{|\mathcal{I}(s)|}$. Similarly, $p(s)$ is estimated as the proportion of the instances occupying s, i.e., $\hat{p}(s)=\frac{|\mathcal{I}(s)|}{|\mathcal{I}|}$ where $\mathcal{I}=\bigcup_{s\in\mathcal{S}}\mathcal{I}(s)$. Then, $p(w_{t-1}=w^j|s_t=s_i)$ is the probability of occupying health state w^j following the observation of the sequence suffix s_{i+1}. The leaf labeled s_{i+1} defines a specific sequence of observations $(\boldsymbol{o}_i,\boldsymbol{o}_{i-2},..,\boldsymbol{o}_{i-d(s_{i+1})})$ where the joint probability for the sequence s_{i+1} and the health state at the end of the sequence, w_i, is:

$$p(w_i,s_{i+1})=p(w_i,\boldsymbol{o}_i,s_i)=p(\boldsymbol{o}_i|w_i)p(w_i,s_i)=$$
$$=p(\boldsymbol{o}_i|w_i)\sum_{w_{i-1}}p(w_i|w_{i-1})p(w_{i-1},s_i) \tag{10}$$

where s_i is the parent node of leaf s_{i+1}. This can be efficiently computed recursively if each node s in the tree holds its joint probability distribution with the health states. For an initial node of $s'=(\boldsymbol{o}')$, the joint probability is $p(w',s')=p(\boldsymbol{o}'|w')\psi(w')$ where $\psi(w')$ is the initial probability of health state w' (assumed to be known).

As (10) suggests, the transition probability between health states is required to estimate the transition probability from the last controller state s_t to the new health state w_t. We once more leverage the unique WBAN structure where at a certain timestep t we only need the sequence of past observations, since given them, the past actions are redundant. This sequence can be used to estimate the transition matrix θ using the sequential Expectation-Maximization (EM) HMM algorithm of *Baum-Welch* (BW) [12]. The BW algorithm (detailed as Algorithm 2) finds the set of parameters of the HMM model $\lambda=(\theta,E,\psi)$ that maximizes $p(\boldsymbol{o}_{1:T}|\lambda)$. In our case, the sensor accuracies and initial health state distribution E,ψ are given and fixed, and we aim to extract θ:

$$\hat{\theta}_{ij}=\hat{p}(w_{t+1}=w^j|w_t=w^i)=\sum_{t=1}^{T}\frac{\xi_t(i,j)}{\zeta_t(i)} \tag{11}$$

where $\xi_t(i,j)\coloneqq p(w_{t+1},w_t|\boldsymbol{o}_{1:T},\lambda)$ and $\zeta_t(i)\coloneqq p(w_t=i|_{1:T},\lambda)$ are updated sequentially in a forward-backward algorithm. In each iteration r, the estimated transition parameter $\theta^{(r)}$ is updated based on $\theta^{(r-1)}$ (the 'M' step), and for each $t=1,\ldots,T$, the components are recursively updated, forward and backward (the 'E' step). We then utilize the BW algorithm to estimate the transition probabilities online, which are used to estimate the expected misclassification cost in (3).

Algorithm 2 `Baum-Welch Algorithm (HMM)`

Initialization: pick $\theta^{(1)}$ at random, and receive E, ψ as input. Set $\alpha_1(i)=\psi_i E_i(\boldsymbol{o}_{t=1})$, $\beta_T(i)=1$ and $r=1$.
While the likelihood has not converged:
1) **For** all $t\in\{1,..,T\}, i\in\{1,...,J\}$, compute:
   a) The forward component using $\theta^{(r)}$: $\alpha_t(i)=(\sum_j\alpha_t(j)\theta_{ji}^{(r)})E_i(\boldsymbol{o}_t)$.
   b) The backward component using $\theta^{(r)}$: $\beta_t(i)=\sum_j\theta_{ij}^{(r)}E_j(\boldsymbol{o}_t)\beta_t(j)$ and $\beta_T(i)=1$.
   c) $\zeta_t(i)=\frac{\alpha_t(i)\beta_t(i)}{\sum_j\alpha_t(j)\beta_t(j)}$.
   d) $\xi_t(i,j)=\frac{\alpha_t(i)\theta_{ij}^{(r)}E_j(\boldsymbol{o}_t)\beta_{t+1}(j)}{\sum_j\alpha_t(j)\beta_t(j)}$, $\forall j\in\{1,\ldots,J\}$.
2) Compute $\theta_{ij}^{(r+1)}=\frac{\sum_{t=1}^{T}\xi_t(i,j)}{\sum_{t=1}^{T}\zeta_t(i)}$. $\forall i,j\in\{1,...,J\}$.
3) $r\leftarrow r+1$.

C. Tree (Inner State-Space) Construction

The inner state-space defined by the suffix tree is built online while learning the policy. This introduces an exploration-exploitation tradeoff in the tree construction: we want to exploit the knowledge of the best actions for the existing inner states, but also to explore who those inner states should be. The main question in building the tree is, therefore, whether additional memory is required to summarize the history.

The USM algorithm defines 'fringe nodes', which are a potential memory extension of length $k-d$ to an "official" leaf of depth d. The fringe node f_j of depth k holds the set of instances $\mathcal{I}(f_{ij})$ of its parent's leaf s_i extended by an additional suffix of length $k-d$. Each leaf node and its fringes hold the Q-value $Q(I_i)=c_i+\gamma U(L(I_{i+1}))$ for each of their instances and actions. The idea is to split leaves if their descendants show statistically different expected future discounted costs for the same action. The Q-value distributions of the parent leaf s_i and each of its fringes are compared using the Kolmogorov-Smirnov (KS) test to check whether they are significantly different. If the test suggests a significant statistical difference, then additional memory is allocated to distinguish the inner states better and the fringe is 'promoted' to an official leaf, which enlarges the inner state space $\mathcal{S}$.

A major shortcoming of the USM algorithm is its scalability, due to the tree expansion mechanism. For example, a WBAN of N sensors has $|\mathcal{O}|=3^N$ possible observations. The USM algorithm determines that if at least one of the fringes of s_i is found significant, the entire level of fringes is promoted to official leaves, which requires $|\mathcal{O}|^d$ possible sequences. This exponential increase in the size of the inner state-space favors exploration but increases the complexity tremendously.

To improve scalability, we include a pruning mechanism following the KS test. This pruning mechanism limits the allowed exploration to states that are frequent enough. We adopt the "partial leaf" concept [13] in which a parent node can still serve as an inner state, holding instances for which additional memory expansion is not beneficial. Specifically, we employ a threshold on the size of the set of instances of state s, $|\mathcal{I}(s)|>0.1|\mathcal{I}|$ where $\mathcal{I}=\cup_{s\in\mathcal{S}}\mathcal{I}(s)$, such that

only leaves holding sufficient information can be extended. As information continues to flow, we re-evaluate the partial leaf's remaining fringes and if the threshold is reached, the withheld expansions are promoted into official leaves.

IV. Simulation Results

In this section, we test the performance of WBAN-RL compared with two alternative algorithms based on a belief-MDP ("bMDP") [12]. BW-bMDP estimates the unknown Markov model θ on the fly using BW-HMM and uses the estimated matrix to compute the solution, in the spirit of [7]. In contrast, bMDP knows the exact θ, and therefore, while unrealistic, provides a lower bound benchmark for WBAN-RL [14]. We also test the greedy policy that selects the action that minimizes the instantaneous cost from the assigned inner state. It represents a low-complexity alternative to value iteration that still uses both our cost estimation and the USM tree.

In our simulations, we run WBAN-RL (Algorithm 1) for T timesteps to learn the control policy. To evaluate the policy against the benchmarks, we run the resulting policy for 50 timesteps and report the accumulated discounted cost (policy value, see (1)) with $\gamma = 0.95$, averaging over 100 such runs.

A. Performance for Different Transition Matrices

Consider a health monitoring system with $J = 3$ health states, monitored by a WBAN with $N = 2$ sensors whose accuracy matrix is $E = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \end{bmatrix}$ and sensors energy costs of $C_a = [13,\ 5]$. We use the following form for θ:

$$\tilde{\theta} = \begin{bmatrix} 1-x-x^2 & x & x^2 \\ y & 1-x-y & x \\ y^2 & y & 1-y-y^2 \end{bmatrix}$$

where x is the transition probability from a lower health state to a worse (higher) health state, and y is the probability to transition into a better (lower) health state. In addition, we impose the constraint that the probability to remain in the same health state is larger than the probability to transition to another health state, i.e., $1-(x+x^2) > x$; $1-(y+y^2) > y$.

The initial health state probabilities are $\psi = [0.6, 0.3, 0.1]$ and the FP and FN misclassification costs are $C_m = [70, 90]$. We quantize the continuous belief space of bMDP into four uniform levels. For the KS test, we used a p-value threshold of 5%. We used a WBAN-RL learning period of $T = 5,000$.

In Figure 2, we compare the WBAN-RL for a range of transition matrices $\tilde{\theta}$ to the model-free 'greedy' policy and the benchmark bMDP that knows θ. The results demonstrate that throughout the defined space, the WBAN-RL algorithm outperforms the greedy approach, and in most areas of the space only slightly underperforms vis-ä-vis the bMDP approach that assumes a known θ. For example, for $\theta_1 = \begin{bmatrix} 0.61 & 0.3 & 0.09 \\ 0.3 & 0.4 & 0.3 \\ 0.09 & 0.3 & 0.61 \end{bmatrix}$, WBAN-RL obtains a policy value very close to that of bMDP – 2,770 vs. 2,732, where the greedy policy only achieves 2,893. This hierarchy between the policies remains throughout the entire space.

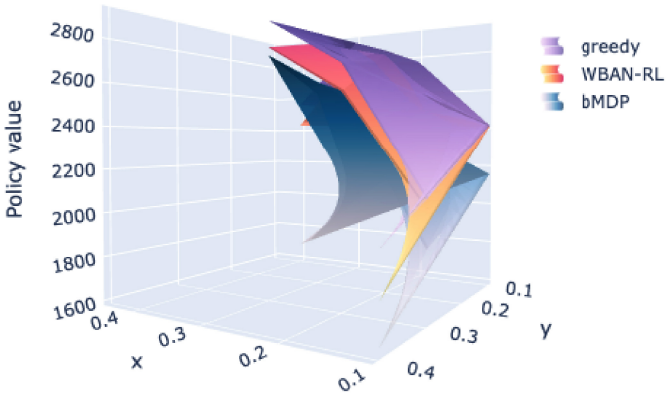

Fig. 2. Performance for a range of transition matrices, for the WBAN-RL model-free algorithm, its greedy low-complexity alternative, and quantization of the bMDP optimal policy that knows the transition matrix.

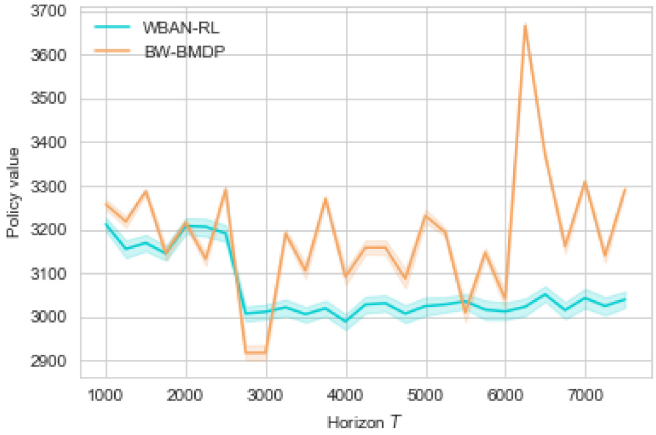

Fig. 3. Comparison of the "model-based" BW-bMDP that estimates the transition matrix online, with the model-free WBAN-RL.

B. Comparison With a Model-Based Alternative

When the transition matrix θ is unknown, an alternative approach is to estimate it given the observations, and then use the estimated matrix in a 'model-based' value iteration (i.e., bMDP). Using the setting in Section IV-A, we compared WBAN-RL with this alternative, termed "BW-bMDP". In each iteration, we first estimate θ using BW-HMM with 100 iterations (Algorithm 2). We then use this estimated matrix to compute the next quantized belief state, select the next action and update the value function. Figure 3 shows, for $\theta_2 = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.1 & 0.7 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$, the accumulated discounted cost for the two methods as a function of the learning period (horizon) T. We observe that WBAN-RL can sustain significantly lower costs compared to BW-bMDP. A fine enough quantization is required to approximate the optimal solution well, but it also increases the convergence time of value iteration. This might explain why BW-bMDP does not converge. In contrast, WBAN-RL quickly converges to its superior performance.

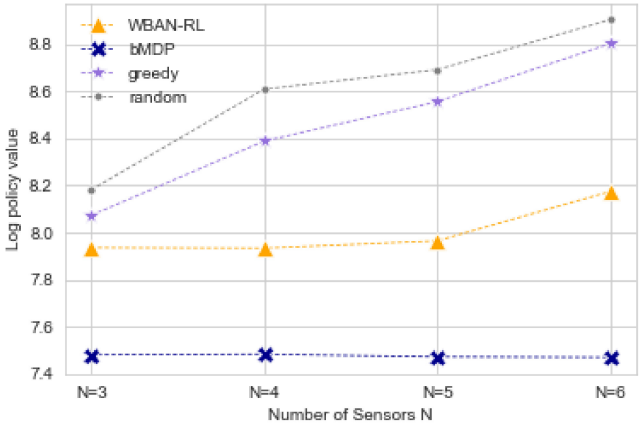

Fig. 4. Performance as a function of the number of sensors N.

C. Performance for Different Numbers of Sensors

Figure 4 shows the accumulated discounted cost after a learning period of $T = 5,000$, as a function of the number of sensors. The accuracy matrix for $N = 6$ sensors was $E = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.8 \\ 0.7 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$ and for smaller N we used the submatrix starting from row one. Similarly, the energy costs were $C_a = [13, 5, 25, 50, 15, 35]$. The transition matrix was θ_1 from Section IV-A. We used 6 uniform quantization levels for the state space of bMDP. The performance of WBAN-RL stays almost constant for $N = 3, 4, 5$ and only degrades for $N = 6$. With more sensors, WBAN-RL needs a bigger USM tree to summarize the past observations, which takes more time to build. Nevertheless, the scalability of the performance is far better than that of the greedy or random policies. The performance of bMDP improves with 6 quantization levels (compared to 4), but its complexity increases exponentially.

V. Conclusion

We studied the problem of monitoring an unknown Markov process (i.e., health model) by activating a subset of a network of sensors every timestep. We focused on the energy–accuracy tradeoff that is intertwined with the exploration-exploitation tradeoff introduced when learning the control policy.

The main goal of this letter is to enable RL for WBAN health monitoring. The two main obstacles to implementing RL in health monitoring are that the state is unknown (i.e., the problem is a POMDP) and that the cost feedback is not fully available (i.e., misclassification cost is unknown). Our algorithm overcomes these obstacles by combining USM-RL for POMDPs with cost estimation using BW-HMM. The BW-HMM module, which estimates the unknown transition matrix θ, is only used to estimate the unknown misclassification cost, while the decision-making itself is based on the USM-RL algorithm. Our simulations suggest that our algorithm outperforms the model-based approach that estimates θ directly and then uses a model-based value iteration. Our algorithm also achieves comparable performance with a four-level quantization of the optimal belief-MDP solution.

Our work lifts the assumption that a Markovian health model is known. However, Markov models are unsuitable for some physiological processes. Therefore an exciting next step is to lift the assumption that the health process is Markovian, and study the performance of the proposed monitoring algorithm in more detailed models [15], [16], [17].

Our monitoring algorithm is applicable beyond health applications. However, in applications where real-time computation is necessary, a scalable version of our algorithm would have to overcome the exponential complexity of the USM tree (e.g., in the number of sensors or states).

References

[1] Y. B. David et al., "Optimal health monitoring via wireless body area networks," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 6800–6805.
[2] D. Miller, Z. Zhou, N. Bambos, and I. Ben-Gal, "Optimal sensing for patient health monitoring," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.
[3] A. Panangadan, S. M. Ali, and A. Talukder, "Markov decision processes for control of a sensor network-based health monitoring system," in *Proc. Nat. Conf. Artif. Intell.*, vol. 20, 2005, pp. 1529–1534.
[4] G. Quer et al., "Wearable sensor data and self-reported symptoms for COVID-19 detection," *Nature Med.*, vol. 27, no. 1, pp. 73–77, 2021.
[5] N. V. Chawla and D. A. Davis, "Bringing big data to personalized healthcare: A patient-centered framework," *J. General Internal Med.*, vol. 28, no. 3, pp. 660–665, 2013.
[6] G. Shani, R. I. Brafman, and S. E. Shimony, "Model-based online learning of POMDPs," in *Proc. Eur. Conf. Mach. Learn.*, 2005, pp. 353–364.
[7] T. Geller "Learning health state transition probabilities via wireless body area networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.
[8] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation Learn. Optim.*, vol. 12, no. 3, p. 729, 2012.
[9] R. A. McCallum, "Instance-based utile distinctions for reinforcement learning with hidden state," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 387–395.
[10] L. J. Allen, "Some discrete-time SI, SIR, and SIS epidemic models," *Math. Biosci.*, vol. 124, no. 1, pp. 83–105, 1994.
[11] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman, "Acting optimally in partially observable stochastic domains," in *Proc. AAAI*, vol. 94, 1994, pp. 1023–1028.
[12] O. Ibe, *Markov Processes for Stochastic Modeling*. London, U.K.: Newnes, 2013.
[13] I. Ben-Gal, G. Morag, and A. Shmilovici, "Context-based statistical process control: A monitoring procedure for state-dependent processes," *Technometrics*, vol. 45, no. 4, pp. 293–311, 2003.
[14] Y. B. David, T. Geller, I. Bistritz, I. Ben-Gal, N. Bambos, and E. Khmelnitsky, "Wireless body area network control policies for energy-efficient health monitoring," *Sensors*, vol. 21, no. 12, p. 4245, 2021.
[15] G. Gupta, S. Pequito, and P. Bogdan, "Dealing with unknown unknowns: Identification and selection of minimal sensing for fractional dynamics with unknown inputs," in *Proc. Annu. Amer. Control Conf. (ACC)*, 2018, pp. 2814–2820.
[16] G. Gupta, S. Pequito, and P. Bogdan, "Learning latent fractional dynamics with unknown unknowns," in *Proc. Amer. Control Conf. (ACC)*, 2019, pp. 217–222.
[17] G. Gupta, C. Yin, J. V. Deshmukh, and P. Bogdan, "Non-Markovian reinforcement learning using fractional dynamics," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 1542–1547.