DOI:
10.1109/JSAC.2023.3310093

Document Version
Peer reviewed version

A Bayesian Framework for Digital Twin-Based Control, Monitoring, and Data Collection in Wireless Systems

Clement Ruah, *Student Member, IEEE,* Osvaldo Simeone, *Fellow, IEEE,* and Bashir Al-Hashimi, *Fellow, IEEE*
Department of Engineering, King's College London, London, UK

***Abstract*—Commonly adopted in the manufacturing and aerospace sectors, digital twin (DT) platforms are increasingly seen as a promising paradigm to control, monitor, and analyze software-based, "open", communication systems that are expected to dominate 6G deployments. Notably, DT platforms provide a sandbox in which to test artificial intelligence (AI) solutions for communication systems, potentially reducing the need to collect data and test algorithms in the field, i.e., on the physical twin (PT). A key challenge in the deployment of DT systems is to ensure that virtual control optimization, monitoring, and analysis at the DT are safe and reliable, avoiding incorrect decisions caused by "model exploitation". To address this challenge, this paper presents a general Bayesian framework with the aim of quantifying and accounting for model uncertainty at the DT that is caused by limitations in the amount and quality of data available at the DT from the PT. In the proposed framework, the DT builds a Bayesian model of the communication system, which is leveraged to enable core DT functionalities such as control via multi-agent reinforcement learning (MARL), monitoring of the PT for anomaly detection, prediction, data-collection optimization, and counterfactual analysis. To exemplify the application of the proposed framework, we specifically investigate a case-study system encompassing multiple sensing devices that report to a common receiver. Experimental results validate the effectiveness of the proposed Bayesian framework as compared to standard frequentist model-based solutions.**

***Index Terms*—Digital Twin, 6G, Reinforcement Learning, Bayesian Learning, Model-based Learning**

I. Introduction

A. Context, Motivation, and Overview

A digital twin (DT) platform is a cyberphysical system in which a physical entity, referred to as the physical twin (PT), and a virtual model, known as the DT, interact based on an automatized bi-directional flow of information [1], [2]. Leveraging data received from the PT, the DT maintains an up-to-date model of the PT [3], which is used to control, monitor, and analyze the operation of the PT [4]. DT platforms are increasingly regarded as an enabling technology for wireless cellular systems built on the open networking principles of disaggregation and virtualization [5], which are expected to be central to 6G [6]. Notably, through the available PT model, DT platforms provide a sandbox in which to test algorithms, protocols, and artificial intelligence (AI) solutions for communication systems, potentially reducing the need to collect data and carry out testing in the field, i.e., directly on the PT [4], [7].

In this regard, a key challenge in the deployment of DT systems is to ensure that virtual control optimization, monitoring, and analysis at the DT are safe and reliable, avoiding incorrect decisions caused by *model exploitation* [8]. To address this challenge, this paper presents a general Bayesian framework with the aim of quantifying and accounting for model uncertainty at the DT that is caused by limitations in the amount and quality of data available at the DT from the PT (see Fig. 1).

In the proposed framework, the DT builds a *Bayesian model* of the communication system dynamics based on data received from the PT. Unlike conventional *frequentist* parametric models, Bayesian models can quantify model uncertainty by maintaining a distribution over the model parameters [9], [10]. This enables *ensembling*-based control, prediction, and analysis methods, whereby policies, predictions, and recommendations are obtained by accounting for the agreements and disagreements among several models that are consistent with the available information. Intuitively, when different models tend to disagree significantly on an output, this can be taken as quantifiable evidence of model uncertainty. While ensembling is routinely used in fields such as weather prediction [11], its application to DT platforms is still largely unexplored, even outside the field of telecommunications [12], [13].

The Bayesian model at the DT can naturally incorporate domain knowledge about the communication systems, including traffic and channel models, while enabling data-driven exploration of the system dynamics. With the available Bayesian model, the DT can carry out the core functionalities of control, monitoring, prediction, data-collection optimization, and counterfactual analysis, while providing uncertainty-aware outputs. We specifically investigate and detail control via model-based Bayesian multi-agent reinforcement learning (MARL), monitoring for anomaly detection, prediction of unobserved dynamics with uncertainty quantification, and data-collection optimization via directed model-based exploration.

As a possible embodiment of the proposed approach, the DT platform may be implemented as an xApp, or as a collection of

C. Ruah and O. Simeone are with King's Communications, Learning & Information Processing (KCLIP) Lab. The work of O. Simeone was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant 725732), an Open Fellowship of the EPSRC (reference EP/W024101/1), and the CENTRIC project (Grant 101096379). The work of C. Ruah was supported by the Faculty of Natural, Mathematical, and Engineering Sciences at King's College London.

Manuscript received December 1, 2022; revised May 15, 2023.

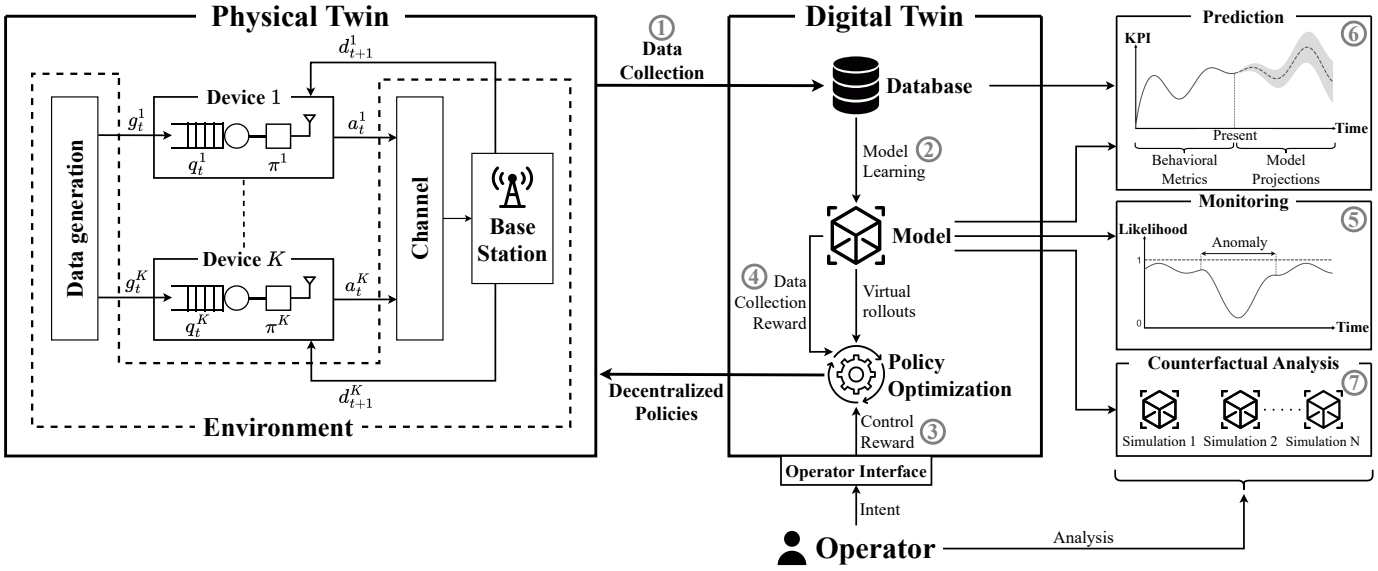


Fig. 1: A digital twin (DT) platform controlling, monitoring, and analyzing the operation of a communication system operates along the phases of data collection ① (Sec. III-A), model learning ② (Sec. III), policy optimization ③ (Sec. IV), and data-collection policy optimization ④ (Sec. IV-C); while also enabling core functionalities such as monitoring ⑤ (Sec. V-A), prediction ⑥ (Sec. V-B), and counterfactual analysis ⑦ (Sec. V-C). Each phase is marked by its corresponding circled number in the figure. In the proposed Bayesian framework, the DT maintains a Bayesian model of the communication system, which serves as the physical twin (PT). The Bayesian model quantifies model uncertainty, and enables safe and reliable control, monitoring, and analysis via ensembling and model-disagreement metrics.

connected xApps, that run in the near-real-time RAN Intelligent Controller (RIC) of an Open-RAN (O-RAN) architecture [14]. As an exemplifying case study, we consider a multi-access PT system consisting of a radio access network (RAN) similar to that studied in [15]–[17]. It is emphasized that, unlike [15]–[17], our goal here is not to address a particular task via MARL, but rather to introduce a general framework supporting the implementation of multiple functionalities at the DT, including control via MARL, monitoring, prediction, and data-collection optimization, despite the limited data transfer from the PT to the DT.

B. Related Work

This section provides a short review of related papers.

1) DT platforms for communication systems: Position papers advocating for the use of DT platforms for the management of next-generation wireless systems include [6], [18]–[21]. Specific contributions to the design of DT platforms for wireless systems have investigated mechanisms for DT-PT synchronization [22], [23], DT-aided network optimization and monitoring [7], DT-based control for computation offloading via model-based RL [24]–[26], user association [27], as well as the design of intelligent reflecting surfaces [28]. A layered deployment strategy for DTs from edge to cloud in 5G networks is studied in [29]; while the optimization of DT deployment subject to resource and latency constrains in edge servers is investigated in [30]. For general reviews on DT systems, we refer the reader to [2], [13], [31]. To the best of our knowledge, the adoption of a general Bayesian framework for the development of DT platforms implementing control, monitoring, and analysis functionalities is yet to be proposed.

2) Model-based reinforcement learning: Reinforcement learning (RL) algorithms fall into two categories: *model-free* algorithms, in which the policy is optimized through trial and error interactions with the ground-truth environment, and *model-based* algorithms, where a model of the environment dynamics is first learned, and then used to optimize the policy in a simulated environment [8]. In the context of DT platforms, model-based algorithms are the natural choice [24], [26], [32], [33]. In fact, they allow the DT to optimize policies to be run at the PT, while bypassing the additional communication overhead and potential safety hazards caused by the interactions with the environment required by model-free methods [34]. That said, DT-aided control can also benefit from model-free RL, e.g., to refine a policy trained based on an inaccurate model [34]. Conversely, model-free RL can benefit from the DT model by exploring alternative actions inside the DT simulation in-between training steps in the ground-truth environment [26]. In addition, the learned dynamics also serve other core DT functionalities, such as monitoring, prediction and counterfactual analysis [4].

3) MARL for communication systems: In MARL, each agent is given a partial observation of the global system state [35], and the actions of one agent can influence the state of another, rendering the dynamics non-stationary from the single-agent perspective [36]. Thus, optimizing each agent independently often proves sub-optimal. State-of-the-art MARL algorithms include *centralized training with decentralized ex-*

ecution (CTDE) methods [37], in which training is done at a central location that optimizes a set of single-agent policies to be deployed at the individual agents. CTDE algorithms can be implemented using value-based methods, often relying on value-decomposition networks [38]; using actor-critic methods, typically based on the *centralized critic with decentralized actors* (CCDA) paradigm [39]; or using both methods [40]. Identifying which agents contribute to the team's success in cooperative settings is not a trivial task, and is known as the *credit assignment problem*. To tackle this problem, the COMA algorithm in [41] proposes a counterfactual baseline to reflect how the reward would have changed had the agent taken a different action. Application of MARL in telecommunications can be found in medium access control signaling protocols [42], dynamic spectrum access [15] and network routing [43].

4) Uncertainty quantification in DT platforms: Non-stationary dynamics and limited PT-to-DT communication in real-world scenarios may cause the DT to "desynchronize" with the ground-truth dynamics of the PT [22]. In turn, model errors can result in *model exploitation* during policy optimization, whereby the optimized policy takes advantage of inaccuracies in the DT model and behaves sub-optimally with respect to the ground-truth environment. Therefore, it is critical that the DT reasons explicitly about its epistemic uncertainty regarding its model of the PT [13] to avoid over-confident and potentially biased decisions. To this end, references [44] and [45] propose to use Bayesian models at the DT. Unlike our work, the focus of these references is on monitoring and predicting the health status of the PT components.

Model-based Bayesian RL, in which the Bayesian model of the environment dynamics reflects the partial observability of the transition probabilities, was investigated in [46], [47] for single-agent applications. A key advantage of Bayesian models in RL is that ensembling techniques support the implementation of well-informed active exploration, or data-collection, schemes, which target regimes with high epistemic uncertainty [48]–[50].

C. Main Contributions

The main contributions of this paper are as follows.

- We introduce a Bayesian DT framework for the control, monitoring, and analysis of a communication system. In the proposed framework, the DT maintains a model of the PT dynamics via a distribution over model parameters, supporting ensembling-based control, prediction of observed and unobserved dynamics, and counterfactual analysis. The model at the DT can incorporate domain knowledge about the communication systems (see, e.g., [51]), including traffic and channel models, and is trained based on data collected from the PT. Data-collection policies can be optimized over successive rounds based on available data at the DT.
- We investigate and detail the ensemble-based DT functionalities of control via MARL, monitoring for anomaly detection, prediction with uncertainty quantification, and data-collection optimization via directed model-based exploration.
- We present an application of the proposed general framework to a multi-access PT system consisting of a RAN. For this system, we carry experiments that validate the advantages of the proposed Bayesian framework as compared to conventional frequentist model-based approaches for (*i*) optimal control, using performance metrics such as throughput and buffer overflow; (*ii*) anomaly detection, with performance evaluated via the receiving operating curve; (*iii*) prediction of buffer overflow events under a new control policy, assessed via accuracy and calibration metrics; and (*iv*) data-collection optimization, focusing on benefits in terms of data efficiency.

This work was partially submitted for conference publication as [52]. The conference version presents a partial description of the framework, including only a brief presentation of tabular model learning and of the DT functionality of anomaly detection. In contrast, this paper provides full details on the proposed framework, encompassing also neural Bayesian learning, data-collection optimization, prediction, and experimental results for data-collection optimization and prediction.

The rest of the paper is organized as follows. In Sec. II, we describe the PT system under study and its DT. Sec. III covers model learning at the DT, including both tabular and neural network-based approaches. Sec. IV details policy optimization for control, introducing also a solution to the problem of data-collection optimization. Sec. V addresses the monitoring functionalities of anomaly detection, prediction and counterfactual analysis. The application of the proposed framework to a multi-access system is provided in Sec. VI, and Sec. VII presents numerical results. Sec. VIII concludes the paper.

II. Physical Twin and Digital Twin Systems

In this paper, we study a Bayesian methodology for the DT-based optimization and monitoring of a telecommunications network, which constitutes the PT. In this section, we describe the system under study by first providing a general overview of the interactions between the DT and the PT; then detailing the general assumptions made on the ground-truth dynamic model followed by the PT; and finally explaining the parametric model of the PT assumed by the DT. The next section will then describe the model learning process at the DT.

A. Overview

The system under study encompasses a multi-agent PT, which describes a telecommunications network, and a single DT located in the cloud, for a large PT system, or at the edge, for a local PT system [53]. The network elements may be mobile devices and/or central units or distributed units of a 5G system [54]. Note that we focus on the case of a single DT, and leave the important problem of coordination among multiple DTs to future work [29], [55]. The DT collects data from the PT, either periodically or in an adaptive manner, and the data is used to optimize a model of the PT dynamics. The model learned at the DT is used to operate the PT, as well as to provide monitoring functionalities such as anomaly detection, prediction of the PT future possible states, and counterfactual analysis [4].

As detailed in Sec. II-B, the PT system under study consists of multiple network elements, such as mobile devices and infrastructure nodes, which are generically referred to as *agents*. Without loss of generality, the PT system at a given time can be described as being in a specific *state*. The state of the system may include, for instance, traffic load conditions at radio units and packet queue lengths at the devices. Furthermore, the PT state evolves over time according to a ground-truth *transition model* that depends on the agents' actions.

As detailed in Sec. II-C, the DT collects data from the PT over dedicated periods of time (phase ① in Fig. 1). The time interval between two data collection phases may vary, depending also on the result of diagnostic tests at the DT of current PT behavior, e.g., via anomaly detection (see Sec. V-A). Based on the data obtained in each data collection period, the DT constructs a model of the transition dynamics of the PT (phase ② in Fig. 1).

The model is used by the DT to recommend control policies to the PT (phase ③ in Fig. 1), as well as to carry out monitoring functionalities such as anomaly detection (phase ⑤ in Fig. 1), prediction (phase ⑥ in Fig. 1), and counterfactual analysis (phase ⑦ in Fig. 1). For example, the control policy may dictate channel access strategies or scheduling algorithms. We refer to Sec. VI for a specific instantiation of the framework for a multi-access system.

An essential aspect of the model learned at the DT is the quality of its *uncertainty quantification* [13]. In fact, it is critically important for the DT to know what it knows, i.e., to be aware of which operating regimes of the PT are well described by the DT model; as well as to know what it does not know, i.e., to be aware of the operating regimes in which the DT model may fail to correctly describe the operation of the PT. A *poorly calibrated* DT model, i.e., a model that cannot properly quantify its epistemic uncertainty, may yield unsafe control decisions for the PT; provide incorrect predictions; and fail to recognize abnormal PT behavior [13].

Data collection phases in successive periods may be carried out by the PT with the supervision of the DT, which may recommend specific data collection strategies (phase ④ in Fig. 1). Uncertainty awareness at the DT is also essential for the optimization of the data-collection policy. In fact, a well-calibrated model enables the DT to assess which operating regimes of the PT call for additional information to be collected to refine or correct the model.

B. Physical Twin

The PT system of interest consists of K agents, indexed by integer $k \in \mathcal{K} = \{1, \ldots, K\}$, that operate over a discrete time index $t = 1, 2, \ldots$ The time index runs over the relevant time units for the system of interest, which are typically time slots or frames. The agents make decisions at each time t that affect the evolution of the overall state of the system.

Formally, at each time t, each agent k takes an *action* a_t^k from a discrete set of possible actions. For instance, a mobile device may decide whether to transmit or not in a given time slot t. The action is selected by following a *policy* that leverages information collected by the agent regarding the current *state* s_t of the overall system.

The state s_t is a vector encompassing all the variables necessary to describe the evolution of the system from time t onwards. State variables may be specific to different local parts of the network, and may be functionally and semantically distinct. For example, a state variable may describe the current traffic conditions at a base station or the quality of the wireless channel on a particular link. The state s_t evolves according to some *ground-truth transition probability* $T(s_{t+1}|s_t, a_t)$. Specifically, the probability distribution of the next state $s_{t+1} \sim T(s_{t+1}|s_t, a_t)$ is modelled as a *Markov decision process* (MDP), and only depends on the current state s_t and joint action $a_t = (a_t^1, \ldots, a_t^K)$ of all agents.

At each time t, each agent k observes a function o_t^k of the overall state s_t. This captures the fact that an agent k typically has access only to *local* information about the state of the system, such as the buffer queue length for a device or the traffic load for a base station. We restrict our framework to the case of *jointly observable* states [35], in which the state s_t can be identified based on the collection of the observations o_t^k of all agents $k \in \mathcal{K}$ at time t. Mathematically, this means the state s_t is assumed to be a function of the collection $o_t = (o_t^1, \ldots, o_t^K)$ of all agents' observations.

It is assumed that agents cannot communicate with each other, and thus the overall information available at agent k at time t amounts to its *action-observation history* $h_t^k = (o_1^k, a_1^k, o_2^k, \ldots, a_{t-1}^k, o_t^k)$. Accordingly, the behavior of agent k is defined by a policy π^k that assigns a probability $\pi^k(a_t^k|h_t^k)$ to each possible action a_t^k based on the available information h_t^k. Note that the presented framework is general enough to subsume the case of a single, possibly composite, PT agent by setting $K = 1$. The more general multi-agent setting under study represents well many telecommunication settings of interest (see Sec. I-B3), and has been studied as a use case for DT platforms in, e.g., [56].

C. Digital Twin

The DT maintains a *model* of the PT ground-truth dynamics $T(s_{t+1}|s_t, a_t)$. To this end, the DT assumes a family of parametric models $T_\theta(s_{t+1}|s_t, a_t)$ that are determined by a parameter vector θ. In the model learning phase, the parameter vector θ is optimized based on data collected from the PT. As we will detail next, the model class $T_\theta(s_{t+1}|s_t, a_t)$ should account for any known structure of the PT. For instance, the DT may be aware that some of the actions in a_t only affect a subset of the state variables in s_t.

In order to account for information available at the DT about the structure of the PT, we partition the state s_t into M distinct subsets $\{s_t^i\}_{i=1}^M$ of state variables, such that each subset s_t^i of state variables is a geographically and/or semantically distinct unit. For instance, a subset s_t^i may correspond to the queue lengths of a subset of devices connected to the same base station; while a subset s_t^j, with $j \neq i$, may describe the channel conditions for all devices connected to a base station.

Given the state subset $\{s_t^i\}_{i=1}^M$ and actions $\{a_t^k\}_{k\in\mathcal{K}}$ of all agents, we introduce a graph with M current state-nodes, one for each subset s_t^i; K action-nodes, one for each action a_t^k; and M future state-nodes, one for each subset s_{t+1}^i. The graph

describes a factorization of the transition probability of the form

$$T_\theta\left(s_{t+1}|s_t,a_t\right)=\prod_{i=1}^{M}T^i_{\theta^i}\left(s^i_{t+1}\middle|s_t^{>i},a_t^{>i}\right), \tag{1}$$

where $s_t^{>i}$ and $a_t^{>i}$ represent the collections of state variables and actions that are considered to directly affect the evolution of state variables in subset s^i_{t+1}. We represent such dependencies by adding a directed edge from action-nodes $a_t^{>i}$ and state-nodes $s_t^{>i}$ to state-node s^i_{t+1}. We refer to Fig. 2 for an example. Note that subsets $s_t^{>i}$ and $a_t^{>i}$ may be empty. For instance, as depicted in the example in Fig. 2, state variables that define the channel qualities are generally not affected by the agents' actions (unless solutions such as intelligent reflective surfaces are used [57]).

We denote as $\theta^i \subseteq \theta$ the subset of model parameters that directly account for the modelled dependence between variables s^i_{t+1} and $(s_t^{>i},a_t^{>i})$. Accordingly, the DT defines M independent parametric models $T^i_{\theta^i}(s^i_{t+1}|s_t^{>i},a_t^{>i})$ that, following the factorization in (1), define the overall dynamic model $T_\theta(s_{t+1}|s_t,a_t)$ of the PT with model parameters $\theta=\{\theta^i\}_{i=1}^M$.

A table summarizing the notations can be found in the Appendix A.

III. Model learning at the DT

In this section, we will detail the model learning phase (Fig. 2 and phase ② in Fig. 1), during which the DT uses the data collected from the PT to train the model parameters θ of model (1). We first discuss the data-collection phase (phase ① in Fig. 1), and then present two Bayesian learning methods with different scalability properties.

A. Data Collection

At the beginning of each data collection phase, the DT may provide the PT with a *data-collection policy* $\pi_d=\{\pi_d^k(a_t^k|h_t^k)\}_{k\in\mathcal{K}}$, with each agent k receiving policy $\pi_d^k(a_t^k|h_t^k)$. These policies may be designed by the DT based on information about the PT prior to the data collection phase. Alternatively, the agents may follow fixed exploration policies, such as distributions $\pi_d^k(a_t^k|h_t^k)$ that assign equal probability to all possible actions a_t^k for each agent.

Starting from an initial state s_1 of the PT, all agents in the PT execute the policy π_d during T time steps. After time T, each agent k communicates its sequence of observations $\{o_t^k\}_{t=1}^T$ and actions $\{a_t^k\}_{t=1}^T$ to the DT. Based on this information, and given that the states are assumed to be jointly observable (see Sec. II-B), the DT can recover the dataset $\mathcal{D}_T^{\pi_d}=\{(s_t,a_t,s_{t+1})\}_{t=1}^T$ of the T experienced transitions.

Sec. IV-C will discuss how the DT can optimize the data-collection policy π_d, while the next subsection covers the model learning phase (phase ② in Fig. 1).

B. Bayesian Learning

Based on the dataset $\mathcal{D}_T^{\pi_d}$, the DT seeks to optimize the parametric models in (1) to approximate the ground-truth unknown transition distribution $T(s_{t+1}|s_t,a_t)$. To this end, we propose that the DT adopts *Bayesian learning* in order to obtain a well-calibrated model. Bayesian learning aims at evaluating the posterior distribution $P(\theta|\mathcal{D}_T^{\pi_d})$ of the unknown model parameters θ. We define a factorized prior distribution $P(\theta)=\prod_{i=1}^M P(\theta^i)$ on the model parameters. The prior distribution $P(\theta^i)$ can encode both domain knowledge and previous experience obtained from previous data-collection

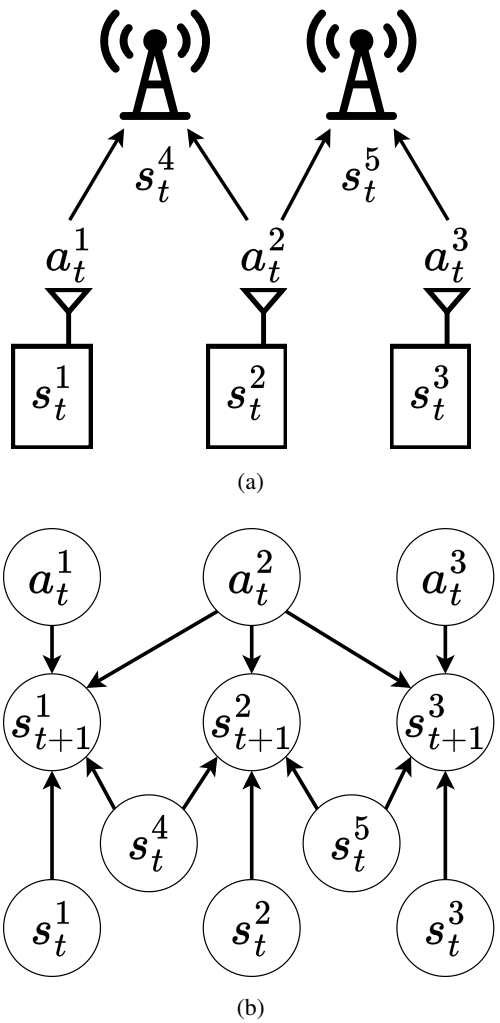


Fig. 2: (a) Example of a PT consisting of three devices and two base stations. The internal states s_t^1, s_t^2 and s_t^3 of the three devices may include local battery levels and queue lengths. State variables s_t^4 and s_t^5 describe the propagation conditions on the shared links from the devices to the base stations. Note that, in this example, only the second device is in the coverage range of both base stations. The actions a_t^1, a_t^2 and a_t^3 of each respective device may include channel access decisions.
(b) Graph representing a possible factorization (1) assumed at the DT for the state and action variables from time step t to time step $t+1$ for the system described in panel (a). Accordingly, the DT assumes that the state of a device at time $t+1$ is affected by the corresponding device's state at time t, as well as the actions of the devices connected to the same base station and the channel state for the given base station.

phases. In particular, in some settings, some of the parameters θ^i may be known to the DT. In this case, the prior is concentrated at the known value, and the posterior $P(\theta^i|\mathcal{D}_T^{\pi_d})$ trivially coincides with the prior.

Given the factorization in (1), the posterior distribution $P(\theta|\mathcal{D}_T^{\pi_d})$ also factorizes as $P(\theta|\mathcal{D}_T^{\pi_d}) = \prod_{i=1}^M P(\theta^i|\mathcal{D}_T^{\pi_d})$, where the posterior distribution $P(\theta^i|\mathcal{D}_T^{\pi_d})$ is given by

$$\begin{aligned} P\left(\theta^i\middle|\mathcal{D}_T^{\pi_d}\right) &\propto P(\theta^i) P\left(\mathcal{D}_T^{\pi_d}\middle|\theta^i\right) \\ &= P(\theta^i) \prod_{t=1}^{T} T_{\theta^i}^i \left(s_{t+1}^i \middle| s_t^{>i}, a_t^{>i}\right). \end{aligned} \tag{2}$$

As we will discuss in the rest of this section, depending on the size of the state and action spaces, computing the exact posterior in (2) may not be feasible, and one should resort to function approximations.

C. Tabular Bayesian Learning

In this subsection, we consider small-scale models, in which: (i) the state variable subsets s_t^i take values in a small discrete set $\mathcal{S}^i$; and (ii) each conditional distribution $T_{\theta^i}^i(s_{t+1}^i|s_t^{>i}, a_t^{>i})$ can be expressed as $T_{\theta^i}^i(s_{t+1}^i|x_t^i)$, where x_t^i is a function of variables $(s_t^{>i}, a_t^{>i})$ that can take a small number of values in a set $\mathcal{X}^i$. In this case, the parameters θ^i may be chosen to directly represent the transition probabilities, i.e., we can set $T_{\theta^i}^i(s_{t+1}^i|x_t^i) = \theta^i_{s_{t+1}^i|x_t^i}$ with $(s_{t+1}^i, x_t^i) \in \mathcal{S}^i \times \mathcal{X}^i$. Note that we have the conditions

$$\sum_{s^i \in \mathcal{S}^i} \theta^i_{s^i|x^i} = 1, \text{ and, } \theta^i_{s^i|x^i} \in [0,1], \tag{3}$$

for all $s^i \in \mathcal{S}^i$ and $x^i \in \mathcal{X}^i$.

Exact computation of the posterior distributions $\{P(\theta^i_{s^i|x^i}|\mathcal{D}_T^{\pi_d})\}_{s^i \in \mathcal{S}^i}$ for $i \in \{1,\dots,M\}$ and $x^i \in \mathcal{X}^i$ can be done using the Dirichlet-Categorical model (see, e.g., [10]). To this end, we define the prior Dirichlet distribution $P(\{\theta^i_{s^i|x^i}\}_{s^i \in \mathcal{S}^i}) \sim \mathrm{Dir}(\{\alpha^i_{s^i|x^i,0}\}_{s^i \in \mathcal{S}^i})$ with parameters $\alpha^i_{s^i|x^i,0} > 0$ for $s^i \in \mathcal{S}^i$, such that we have

$$P\left(\left\{\theta^i_{s^i|x^i}\right\}_{s^i \in \mathcal{S}^i}\right) = \frac{\prod_{s^i \in \mathcal{S}^i} \theta^{i\ \alpha^i_{s^i|x^i,0}-1}_{s^i|x^i}}{B\left(\{\alpha^i_{s^i|x^i,0}\}_{s^i \in \mathcal{S}^i}\right)}, \tag{4}$$

where the beta function $B(\{\alpha^i_{s^i|x^i,0}\}_{s^i \in \mathcal{S}^i})$ is taken as a normalizing constant and depends only on the Dirichlet parameters. Accordingly, the prior $P(\theta^i)$ factorizes as $P(\theta^i) = \prod_{x^i \in \mathcal{X}^i} P(\{\theta^i_{s^i|x^i}\}_{s^i \in \mathcal{S}^i})$, and represents prior knowledge or belief about the respective transition model $T_{\theta^i}^i$. Given the available experience $\mathcal{D}_T^{\pi_d}$, the posterior distribution $P(\{\theta^i_{s^i|x^i}\}_{s^i \in \mathcal{S}^i}|\mathcal{D}_T^{\pi_d})$ for $x^i \in \mathcal{X}^i$ is given by the Dirichlet distribution $\mathrm{Dir}(\{\alpha^i_{s^i|x^i,T}\}_{s^i \in \mathcal{S}^i})$ with the updated parameters

$$\alpha^i_{s^i|x^i,T} = \alpha^i_{s^i|x^i,0} + \sum_{t=1}^{T} \mathbb{1}_{\left\{s_{t+1}^i = s^i, x_t^i = x^i\right\}} \tag{5}$$

for all $s^i \in \mathcal{S}^i$; where the indicator function $\mathbb{1}_{\left\{s_{t+1}^i = s^i, x_t^i = x^i\right\}}$ is equal to 1 whenever we have $(x^i, s^i) = (x_t^i, s_{t+1}^i) \in \mathcal{D}_T^{\pi_d}$, and 0 otherwise. Therefore, we update the Dirichlet parameters by counting the number of experienced transitions (x_t^i, s_{t+1}^i) for all $t \in \{1,\dots,T\}$.

With tabular learning, the number of parameters to be optimized is the same for both frequentist and Bayesian frameworks, with the former relying on maximum likelihood (ML) or maximum a posteriori (MAP) estimates. This is generally the case in conjugate models, for which the posterior distribution can be evaluated exactly (see, e.g., [10], [58]).

D. Neural Bayesian Learning

For more complex problems, computing the exact posteriors $P(\theta^i|\mathcal{D}_T^{\pi_d})$ in (2) is typically intractable, and the DT must resort to using approximation methods. To illustrate this approach, we specifically introduce M neural networks (NNs), one per unknown factor $T_{\theta^i}^i$ in (1). For $i \in \{1,\dots,M\}$, the vector θ^i defines the parameters of the NN $T_{\theta^i}^i$ that takes as input the state and action variables contained in $s_t^{>i}, a_t^{>i}$ at some time t, and outputs a probability distribution $T_{\theta^i}^i(s_{t+1}^i|s_t^{>i}, a_t^{>i})$ over the set of possible states $s_{t+1}^i \in \mathcal{S}^i$ at time $t+1$.

In order to approximate the posterior distribution $P(\theta^i|\mathcal{D}_T^{\pi_d})$, we define here a conventional solution based on mean-field variational inference (VI) [59]. Other approximate inference algorithms, such as Markov chain Monte Carlo (MCMC), would also be applicable [10].

In the most common implementation of VI, for each factor $i \in \{1,...,M\}$, one assumes a Gaussian prior given by $P(\theta^i) = \mathcal{N}(\theta^i|0,\Sigma_p^i)$, where $\Sigma_p^i = \mathrm{Diag}(\sigma_{p,1}^{i\ 2},...,\sigma_{p,P^i}^{i\ 2})$ is a diagonal covariance matrix with $\sigma_{p,j}^i > 0$ for $j \in \{1,\dots,P^i\}$, and where P^i denotes the number of parameter in the NN, i.e., the size of vector θ^i. The posterior $P(\theta^i|\mathcal{D}_T^{\pi_d})$ is approximated through the parameterized distribution $q(\theta^i|\phi^i) = \mathcal{N}(\theta^i|\mu^i,\Sigma^i)$ with mean vector $\mu^i = (\mu_1^i,\dots,\mu_{P^i}^i)$ and diagonal covariance matrix $\Sigma^i = \mathrm{Diag}(\sigma_1^{i\ 2},...,\sigma_{P^i}^{i\ 2})$, with $\sigma_j^i > 0$ for all $j \in \{1,\dots,P^i\}$. Variational parameters $\phi^i = (\mu_1^i,\dots,\mu_{P^i}^i,\sigma_1^i,\dots,\sigma_{P^i}^i)$ are optimized by addressing the problem of minimizing the variational *free energy* [10], i.e.,

$$\begin{aligned} \arg\min_{\phi^i} \quad & \left\{\mathbb{E}_{\theta^i \sim q(\theta^i|\phi^i)}\left[-\log\left(P\left(\mathcal{D}_T^{\pi_d}\middle|\theta^i\right)\right)\right]\right. \\ & \left. +\mathrm{KL}\left(q(\theta^i|\phi^i)||P(\theta^i)\right)\right\}, \end{aligned} \tag{6}$$

where

$$\mathrm{KL}\left(P(X)||Q(X)\right) = \mathbb{E}_{X \sim P(X)}\left[\log\left(\frac{P(X)}{Q(X)}\right)\right] \tag{7}$$

is the Kullback-Leibler (KL) divergence between two distributions P and Q. The free energy criterion in (6) is also known as the negative *evidence lower bound* (ELBO) in the machine learning literature.

While frequentist NNs directly minimize the log-loss $-\log(P(\mathcal{D}_T^{\pi_d}|\theta^i))$ with respect to the P^i parameters in θ^i, optimization of Bayesian NNs via the presented, conventional, VI solution, minimizes the free energy with respect to the $2P^i$ parameters in vector ϕ^i. Using the *reparameterization trick* [59], problem (6) can be addressed iteratively through stochastic gradient descent on an optimization space that encompasses twice the number of parameters as for frequentist learning on the same NN architecture.

IV. Policy Optimization

In this section, we discuss the policy optimization phase (phase ③ in Fig. 1), in which the DT leverages the approximate posterior $P(\theta|\mathcal{D}_T^{\pi_d})$ obtained during the model learning phase (see Sec. III) to produce optimal control policies for the multi-agent PT system. We will also describe the proposed procedure to design efficient data collection policies for the data collection phase (phase ④ in Fig. 1).

A. Setting

During policy optimization (phase ③ in Fig. 1), the DT aims at optimizing the *decentralized policy* $\pi = \{\pi^k(a_t^k|h_t^k)\}_{k\in\mathcal{K}}$ of the K agents so as to maximize some user-specified performance criterion. This criterion is defined by a *reward* function $r(s_t, a_t, s_{t+1})$, which determines the *total discounted return*

$$G_t = \sum_{\tau=t}^{+\infty} \gamma^{\tau-t} r(s_\tau, a_\tau, s_{\tau+1}), \tag{8}$$

for some exponential discounting factor $\gamma \in [0, 1]$ when the PT applies the policy π. The optimal control problem consists of the maximization of the average long-term reward [8]

$$\max_{\pi} \mathbb{E}_\pi(G_1). \tag{9}$$

This amounts to a Decentralized MDP (Dec-MDP) [35].

We emphasize that the DT has only access to the ensemble of models $T_\theta(s_{t+1}|s_t, a_t)$ given by the posterior distribution $P(\theta|\mathcal{D}_T^{\pi_d})$, and not to the ground-truth distribution $T(s_{t+1}|s_t, a_t)$, when addressing problem (9). In particular, the DT cannot directly interact with the PT during the policy optimization phase, and must solely rely on the observed data $\mathcal{D}_T^{\pi_d}$.

Given that all policies are issued by the central DT platform, policy optimization can naturally rely on CTDE methods characterized by centralized training at the DT and decentralized execution at the PT. This class of approaches bypasses non-stationarity issues that affect decentralized learning schemes [36].

B. Control Policy Optimization

Among possible CTDE methods (see Sec. I-B3), we focus on the COunterfactual Multi-Agent (COMA) algorithm in [41], a state-of-the-art CCDA method. The key distinction between the approach adopted here and the conventional COMA implementation is the fact that the model $T_\theta(s_{t+1}|s_t, a_t)$ assumed here is stochastic in the sense that the model parameter vector θ is distributed according to the (approximate) posterior distribution $P(\theta|\mathcal{D}_T^{\pi_d})$.

The proposed approach addresses the problem (9) via model-generated virtual rollouts at the DT. In a manner similar to [50], we account for the epistemic uncertainty encoded by the posterior $P(\theta|\mathcal{D}_T^{\pi_d})$ by periodically sampling a parameter vector $\theta \sim P(\theta|\mathcal{D}_T^{\pi_d})$ during policy optimization so as to produce the next state $s_{t+1} \sim T_\theta(s_{t+1}|s_t, a_t)$ in the virtual rollouts.

As is typical in CCDA algorithms [40], in a manner similar to standard actor-critic algorithms [8], the DT maintains a centralized critic $Q_w(s_t, a_t)$, with parameter vector w, as well as the decentralized policies $\pi_v = \{\pi_v^k(a_t^k|h_t^k)\}_{k\in\mathcal{K}}$, with common parameter vector v. During *policy evaluation*, the critic $Q_w(s_t, a_t)$ aims at approximating the *Q-value* $Q^{\pi_v}(s, a) = \mathbb{E}_{\pi_v}[G_t|s_t = s, a_t = a]$, i.e., the average future return under policy π_v starting from a given global state s and joint action a. Then, during *policy improvement*, policies $\pi_v^k(a_t^k|h_t^k)$ for all agents $k \in \mathcal{K}$ are updated to maximize the expected return in (9). This is done by using the centralized critic $Q_w(s_t, a_t)$ to reward actions that enhance the performance at the system level. As we will detail next, during the policy optimization phase, we alternate between policy evaluation and policy improvement steps until convergence of the decentralized policy π_v. Upon convergence, only the learned policies $\pi_v^k(a_t^k|h_t^k)$ need to be transmitted by the DT to their respective agents.

During policy evaluation, the policy π_v is kept constant and the critic Q_w is optimized by leveraging virtual rollouts $(s_1, a_1, r_2, s_2, a_2, ...)$ obtained by following policy π_v within model $T_\theta(s_{t+1}|s_t, a_t)$. Since rollouts represent only a finite number of terms in (8), the return G_t under policy π_v is approximated using the *n-step truncated λ-return* estimator defined as [8]

$$G_{t:t+n}^\lambda = (1-\lambda)\sum_{l=1}^{n-1} \lambda^{l-1} G_{t:t+l} + \lambda^{n-1} G_{t:t+n}, \tag{10}$$

with $\lambda \in [0, 1]$, and

$$G_{t:t+l} = \sum_{l'=0}^{l-1} \gamma^{l'} r(s_{t+l'}, a_{t+l'}, s_{t+l'+1}) + \gamma^l Q_{\bar{w}}(s_{t+l}, a_{t+l}). \tag{11}$$

The *target critic* $Q_{\bar{w}}$ in (11) is used to stabilize the training procedure and shares the same architecture as Q_w, with parameters $\bar{w}$ periodically copied from w [60]. Accordingly, the critic loss function is defined as

$$\mathcal{L}_w = \mathbb{E}_{\pi_v}\left[\left(G_{t:t+n}^\lambda - Q_w(s_t, a_t)\right)^2\right], \tag{12}$$

and the parameters w are obtained iteratively through gradient descent, with target parameters $\bar{w}$ updated every $N_{\text{target}} \geq 1$ iterations.

After $N_{\text{critic}} \geq 1$ policy evaluation steps, a policy improvement step is carried through gradient ascent with respect to parameters v using the policy gradient theorem with a baseline [8]. Accordingly, for each agent $k \in \mathcal{K}$, the gradient is given by:

$$\nabla_v J = \mathbb{E}_{\pi_v}\left[\sum_{k\in\mathcal{K}} \nabla_v \log\left(\pi_v^k(a_t^k|h_t^k)\right) A^k(s_t, a_t)\right], \tag{13}$$

where $A^k(s_t, a_t)$ is the *counterfactual baseline* used by COMA, and defined as

$$\begin{aligned} A^k(s_t, a_t) =& Q_w(s_t, a_t) \\ &- \sum_{a^k\in\{0,1\}} \pi_v^k(a^k|h_t^k) Q_w\left(s_t, (a_t^{-k}, a^k)\right), \end{aligned} \tag{14}$$

where $a_t^{-k} = \{a_t^{k'}\}_{k'\neq k}$ denotes the actions of all agents except agent k at time step t. By marginalizing the contribution

of agent k in the baseline, $A^k(s_t, a_t)$ quantifies the effect the action a_t^k of agent k has on the return as compared to its default behavior $a_t^k \sim \pi_v^k(a_t^k|h_t^k)$. This in turn helps mitigate the credit assignment problem [41].

In order to encourage exploration of the (virtual) state-action space during the first policy optimization iterations, we draw inspiration from the SAC algorithm [61] and use the alternative reward

$$r_e(s_t, a_t, s_{t+1}) = r(s_t, a_t, s_{t+1}) - \alpha_e \log\left(\pi_v(a_t|s_t)\right), \quad (15)$$

with *temperature* hyperparameter $\alpha_e > 0$. The alternative reward r_e in (15) adds an exploration bonus based on the entropy of the policy π_v, rewarding policies that are stochastic and with high entropy, which in turn enables undirected exploration of the state-action space. After a given number of policy improvement steps, we return to the original reward definition r until convergence of the control policy π_v.

C. Data Collection Optimization

As discussed in Sec. III-A, the data-collection policy $\pi_d(a_t|s_t)$ can be optimized by the DT based on the available data and on the DT's assessment about operating regimes characterized by more significant model uncertainty. For this purpose, the DT uses the available posterior parameter distribution $P(\theta|\mathcal{D}_T^{\pi_d})$ to identify transitions (s_t, a_t, s_{t+1}) that yield high epistemic uncertainty, i.e., where models T_θ drawn from distribution $P(\theta|\mathcal{D}_T^{\pi_d})$ disagree more significantly [10], [62]. The resulting disagreement metric is used to engineer a data collection reward r_d. With such reward function, the data collection policy π_d is optimized at the DT by following the approach described in Sec. IV-A with r_d in lieu of r.

The reward function r_d should capture the extent to which the ensemble of models $T_\theta(s_{t+1}|s_t, a_t)$ with $\theta \sim P(\theta^i|\mathcal{D}_T^{\pi_d})$ disagree on the prediction of the next state s_{t+1} given the previous-step state-action pair (s_t, a_t) [49]. Among the methods presented in Sec. I-B4, one way to gauge this disagreement is to use the mutual information $\mathrm{I}(s_{t+1}; \theta|s_t, a_t)$ evaluated under the posterior $P(\theta|\mathcal{D}_T^{\pi_d})$ [49]. Accordingly, the data collection reward is defined as

$$\begin{aligned} r_d(s_t, a_t) =& \mathcal{H}\left(\mathbb{E}_{\theta\sim P(\theta|\mathcal{D}_T^{\pi_d})}\left[T_\theta(\cdot|s_t, a_t)\right]\right) \\ &- \mathbb{E}_{\theta\sim P(\theta|\mathcal{D}_T^{\pi_d})}\left[\mathcal{H}\left(T_\theta(\cdot|s_t, a_t)\right)\right], \end{aligned} \quad (16)$$

where $\mathcal{H}(P(\cdot)) = \mathbb{E}_{s\sim P(s)}\left[-\log\left(P(s)\right)\right]$ represents Shannon's entropy for the argument distribution. Note that the entropy terms in (16) are evaluated with respect to the distribution of the next state s_{t+1}. In (16), the first term measures the uncertainty on the next state s_{t+1} for the ensemble model, while the second term represents the average uncertainty associated with each member model $T_\theta(s'|s_t, a_t)$ of the ensemble (see also [10]).

V. Monitoring Functionalities

In this section, we discuss three typical functionalities that may be run at the DT in addition to control, namely anomaly detection (phase ⑤ in Fig. 1), prediction (phase ⑥ in Fig. 1), and counterfactual analysis (phase ⑦ in Fig. 1). These functionalities are selected as representatives of tasks that are facilitated by the use of uncertainty-aware Bayesian models.

A. Anomaly Detection

Anomaly detection aims at detecting significant changes in the dynamics of the PT. To formulate this problem, assume that, during the operation of the system following policy optimization (phase ③ in Fig. 1), the DT has access to the information $\mathcal{D}_{T^{\mathrm{M}}}^{\pi} = \{(s_t, a_t, s_{t+1})\}_{t=1}^{T^{\mathrm{M}}}$ about the state-action sequence experienced by the PT within some *monitoring time window* T^{M} under the optimized policy π. The DT tests if the collected data $\mathcal{D}_{T^{\mathrm{M}}}^{\pi}$ is consistent with the data reported by the PT during the most recent model learning phase (phase ① and ② in Fig. 1), or rather if it provides evidence of changed conditions or anomalous behavior.

While frequentist learning is known to perform poorly for detection of out-of-distribution, or abnormal, samples, Bayesian learning has the key advantage of being capable of quantifying epistemic uncertainty via *disagreement-based test metrics*, a property also used in Sec. IV-C (see, e.g., [63]). While in Sec. IV-C disagreement was evaluated on next-state predictions, here the disagreement is defined in terms of the log-likelihood of the observed data. Accordingly, we define as

$$LL\left(\mathcal{D}_{T^{\mathrm{M}}}^{\pi}|\theta\right) = \sum_{\tau=1}^{T^{\mathrm{M}}} \log\left(T_\theta(s_{t+1}|s_t, a_t)\pi(a_t|s_t)\right) \quad (17)$$

the log-likelihood of model θ for the reported experience $\mathcal{D}_{T^{\mathrm{M}}}^{\pi}$, where $\pi(a_t|s_t) = \prod_{k\in\mathcal{K}} \pi^k(a_t^k|h_t^k)$. We then consider the test metric given by the variance

$$\begin{aligned} \mathbb{E}_{\theta\sim P(\theta|\mathcal{D}_T^{\pi_d})}\Big[\Big(&LL\left(\mathcal{D}_{T^{\mathrm{M}}}^{\pi}|\theta\right) \\ &- \mathbb{E}_{\theta\sim P(\theta|\mathcal{D}_T^{\pi_d})}\left[LL\left(\mathcal{D}_{T^{\mathrm{M}}}^{\pi}|\theta\right)\right]\Big)^2\Big], \end{aligned} \quad (18)$$

estimated using samples from distribution $P(\theta|\mathcal{D}_T^{\pi_d})$. A larger variance provides evidence of a large epistemic uncertainty, which is taken to indicate an anomalous observation $\mathcal{D}_{T^{\mathrm{M}}}^{\pi}$ as compared to the model learning conditions.

B. Prediction

One of the key motivations behind the model-based approach adopted by the DT paradigm is the possibility of predicting future states of the PT system by simulating the operation of the system via the model. While frequentist models would generally provide unreliable measures of prediction uncertainty, Bayesian models can not only provide useful point predictions but also well-calibrated error bars.

To describe the problem, we define a *prediction time lag* T^{H}, corresponding to the number of time steps in the future we wish to predict, and a *target metric* y_p, which is a function of future trajectories $\mathcal{D}_{T^{\mathrm{H}}}^{\pi} = \{(s_t, a_t, s_{t+1})\}_{t=1}^{T^{\mathrm{H}}}$ within the prediction time window duration T^{H}, starting from a known state s_1. We also assume that the agents follow a known policy π. As an example, the metric of interest y_p may be the average number of packet losses for a subset of devices connected to the same base station over the next T^H time steps (see Sec. VII-F).

Under these conditions, the DT can roll out the model defined by transitions T_θ and policy π in order to estimate statistics of the target metric y_p. With a Bayesian model,

such statistics are further averaged over the posterior distribution $P(\theta|\mathcal{D}_T^{\pi_d})$, providing a reliable measure of prediction uncertainty. Accordingly, prediction using a Bayesian model requires a number of samples that is larger as compared to its frequentist counterpart by a factor given by the number of models sampled from the posterior.

C. Counterfactual Analysis

The predictive methodology described in the previous subsection is also a useful tool for counterfactual analysis of the PT behavior [7]. In such analysis, one wishes to assess the impact that changes in the system, as described by the ground-truth dynamics T, would have on some target metrics of interest. To this end, one could roll out different models T_θ or policies π implementing the given changes of interest, and then evaluate measures such as the average treatment effect [64].

VI. Application to a Multi-Access System

In order to illustrate the operation and the benefits of the proposed framework for the implementation of a DT platform, in the rest of the paper we focus on a multi-access IoT-like wireless network as the PT system to be controlled and monitored [15]–[17]. This system is implemented as a numerical simulator, which is available on-line [65].

A. Setting

As illustrated in Fig. 1, the PT system under study comprises K sensing devices that obtain data with correlated data arrivals both in time [66] and across devices [15], and communicate with a common base station (BS) over a channel with an unknown distribution. Time is slotted, and each device may transmit in a slot if its buffer is not empty.

With t denoting the time slot index, and following the notation in Sec. II-B, each device $k \in \mathcal{K}$ observes its *local state* $o_t^k = (q_t^k, g_t^k, d_t^k)$, where $q_t^k \in \{0, 1, \ldots, Q_{\max}^k\}$ with $Q_{\max}^k \geq 1$ is the number of packets in the device's buffer; $g_t^k \in \{0, 1\}$ is a binary variable indicating if a new packet is generated ($g_t^k = 1$) at time t or not ($g_t^k = 0$); and $d_t^k \in \{0, 1\}$ indicates whether a packet sent at the previous time step $t-1$ from device k was successfully delivered at the BS ($d_t^k = 1$) or not ($d_t^k = 0$). Satisfying the joint observability assumption (see Sec. II-B), the *overall state of the PT* is fully identified given the joint observations of all devices and is represented by $s_t = o_t = (o_t^1, \ldots, o_t^K)$.

1) Policies: The *access policy* of device k is given by the distribution $\pi^k(a_t^k|h_t^k)$, where we have $a_t^k = 1$ if the device attempts to transmit the first packet in its buffer, and $a_t^k = 0$ if it stays idle during slot t. Finally, we define the (binary) packet-generation vector as $g_t = (g_t^1, \ldots, g_t^K)$, the successful packet-delivery vector as $d_t = (d_t^1, \ldots, d_t^K)$, and the packet-transmission vector as $a_t = (a_t^1, \ldots, a_t^K)$.

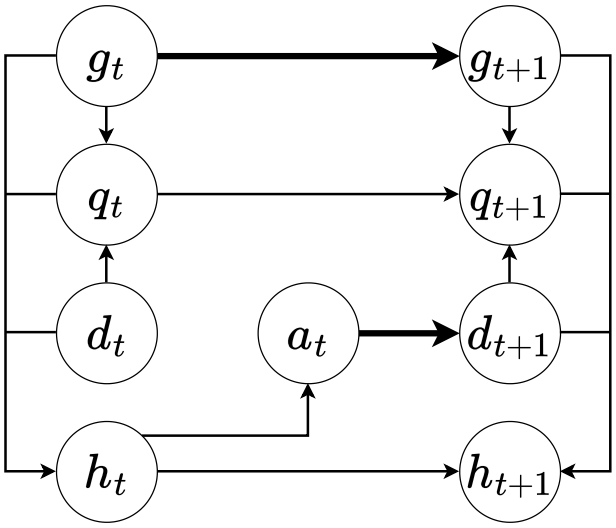


Fig. 3: Dependency graph of the multi-access system. Thin lines represent a 1 to 1 relationship per device (independent between devices) while thick lines represent a many to many relationship (correlated between devices)

2) Buffers: Each device k maintains a *first-in first-out buffer* of maximum capacity $Q_{\max}^k$, where the buffer state q_t^k evolves according to the deterministic update $P(q_{t+1}^k|q_t^k, d_{t+1}^k, g_{t+1}^k)$ given by

$$q_{t+1}^k = \min(Q_{\max}^k, q_t^k + g_{t+1}^k - d_{t+1}^k). \tag{19}$$

A device k can transmit a packet only if its buffer is not empty, and action a_t^k is automatically set to take value $a_t^k = 0$ otherwise, resulting in the condition $a_t^k \leq q_t^k$. If device k generates a new packet when the buffer is full and transmission fails, i.e., if we have the equalities $q_t^k = Q_{\max}^k$, $g_{t+1}^k = 1$, and $d_{t+1}^k = 0$, a *buffer overflow* event occurs at time step $t+1$. In this case, the oldest packet in the buffer is deleted without being sent, and the newly generated packet at time $t+1$ is added to the buffer as per the update rule in (19).

3) Packet generation: The *packet generation mechanism* is modelled as a Markov model $P(g_{t+1}|g_t)$. To account for spatial correlation, we partition the devices into clusters $\{\mathcal{C}^i\}_{i=1}^C$ with $\mathcal{C}^i \subseteq \mathcal{K}$, $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset$ if $i \neq j$ and $\bigcup_{i=1}^C \mathcal{C}^i = \mathcal{K}$, where each cluster $\mathcal{C}^i$ contains devices with correlated packet arrivals. Accordingly, the data-generation dynamics factorize without loss of generality as

$$P(g_{t+1}|g_t) = \prod_{i=1}^{C} P\left(g_{t+1}^{\mathcal{C}^i} \middle| g_t^{\mathcal{C}^i}\right), \tag{20}$$

where $g_t^{\mathcal{C}^i} = \{g_t^k\}_{k \in \mathcal{C}^i}$ for $i \in \{1, \ldots, C\}$.

4) Channel: The *shared channel* is described by the input-output distribution $P(d_{t+1}|a_t)$, where packet delivery from agent k can be successful ($d_{t+1}^k = 1$) only if a packet was transmitted ($a_t^k = 1$), i.e., we have $a_t^k \geq d_{t+1}^k$. For each successfully decoded packet, the BS sends back an acknowledgement (ACK) message to the sending device k over an error-free channel on the control plane.

As an example to be adopted in the next section, in a *multi-packet reception* (MPR) channel, the number of successfully delivered packets $n_{t+1}^{\mathrm{Rx}} = \sum_{k\in\mathcal{K}} d_{t+1}^k$ depends on the number of simultaneous transmissions $n_t^{\mathrm{Tx}} = \sum_{k\in\mathcal{K}} a_t^k$, and the delivered packets are taken uniformly across all the agents that transmit. Accordingly, the channel distribution is given by [67]

$$P(d_{t+1}|a_t) = P(n_{t+1}^{\mathrm{Rx}}|n_t^{\mathrm{Tx}}) \times \frac{\prod_{k\in\mathcal{K}} \mathbb{1}_{\{a_t^k \geq d_{t+1}^k\}}}{\binom{n_t^{\mathrm{Tx}}}{n_{t+1}^{\mathrm{Rx}}}}. \quad (21)$$

B. DT Model

Following the system description in the previous section, the DT model assumes the factorization (1) illustrated in Fig. 3, which is of the form

$$T_\theta(s_{t+1}|s_t,a_t) = P_{\theta^{\mathrm{G}}}(g_{t+1}|g_t) \times P_{\theta^{\mathrm{C}}}(d_{t+1}|a_t) \times \prod_{k\in\mathcal{K}} P(q_{t+1}^k|q_t^k, d_{t+1}^k, g_{t+1}^k), \quad (22)$$

where the deterministic queue dynamics $P(q_{t+1}^k|q_t^k, d_{t+1}^k, g_{t+1}^k)$ defined by (19) are assumed to be known to the DT, and the model parameters $\theta = \{\theta^{\mathrm{G}}, \theta^{\mathrm{C}}\}$ determine the packet generation and channel models, respectively. The DT is also assumed to be aware of the cluster partitions $\{\mathcal{C}^i\}_{i=1}^C$ in (20), e.g., based on the network topology, so that the data generation model $P_{\theta^{\mathrm{G}}}(g_{t+1}|g_t)$ consists of C independent models $T_{\theta^{\mathrm{G},i}}(g_{t+1}^{\mathcal{C}^i}|g_t^{\mathcal{C}^i})$ with parameters $\theta^{\mathrm{G}} = \{\theta^{\mathrm{G},i}\}_{i=1}^C$. As for the channel, the DT optimizes an MPR model $T_{\theta^{\mathrm{C}}}(n_{t+1}^{\mathrm{Rx}}|n_t^{\mathrm{Tx}})$ of the unknown ground-truth distribution $P(n_{t+1}^{\mathrm{Rx}}|n_t^{\mathrm{Tx}})$ of the number of received packets given the number of transmitted packets.

VII. Numerical Results

In this section, we present numerical results related to the multi-access system introduced in the previous section. The main goal is to analyze the advantages of the proposed Bayesian framework at the DT for control, anomaly detection, prediction, and data collection optimization.

A. Setup

Consider $K = 4$ sensing devices equipped with a buffer of capacity $Q_{\max}^k = 1$ packet, with all buffers being initially empty. This scenario is of interest for devices that transmit updates, discarding previous packets from the queue as outdated. Devices 1 and 2 form the cluster $\mathcal{C}^1$, while devices 3 and 4 form cluster $\mathcal{C}^2$. The data generation distribution within each cluster does not depend on previously generated data, and is such that both devices cannot simultaneously generate a packet, with a new packet being generated at either device with probability 0.4. This capture a situation in which devices monitor distinct parts of a process, e.g., the location of a target in distinct spatial regions. The channel allows for the successful transmission of a single packet with probability 1; while, for two simultaneous transmissions, one packet is received with probability 0.8 and both packets are received with probability 0.2. More than two simultaneous transmissions cause the loss of all packets.

B. Implementation

1) Data Collection: Unless stated otherwise, we adopt a random data collection policy that sets $\pi_d^k(a_t^k = 1|h_t^k) = q_t$ for all $k \in \{1,2,3,4\}$ with probability q_t uniformly and independently selected in the interval $[0,1]$ at each step t.

2) Model Learning: Model learning at the DT is carried using the Categorical-Dirichlet model as described in Sec. III-C with all prior Dirichlet parameters set to 0.01. The DT adopts a memoryless model $T_{\theta^{\mathrm{G},i}}(g_{t+1}^{\mathcal{C}^i})$ for the data generation process with model parameters $\theta^{\mathrm{G},i} = \{\theta^{\mathrm{G},i}_{g'^{\mathcal{C}^i}}\}_{g'^{\mathcal{C}^i}\in\{0,1\}^{|\mathcal{C}^i|}}$ for $i \in \{1,2\}$. Furthermore, the channel model is defined by the model parameters $\theta^{\mathrm{C}} = \{\theta^{\mathrm{C}}_{n^{\mathrm{Rx}}|n^{\mathrm{Tx}}}\}_{n^{\mathrm{Rx}}\leq n^{\mathrm{Tx}}\in\{0,\ldots,K\}}$.

3) Reward: In a similar manner to [17], we assume that the reward in (8) takes the form

$$r(s_t,a_t,s_{t+1}) = \sum_{k\in\mathcal{K}} \beta^k r^k(o_t^k, a_t^k, o_{t+1}^k), \quad (23)$$

with

$$r^k(o_t^k,a_t^k,o_{t+1}^k) = \begin{cases} +\xi & \text{if } d_{t+1}^k = 1 \\ -\xi & \text{if } q_t^k = Q_{\max}^k,\ g_{t+1}^k = 1 \text{ and } d_{t+1}^k = 0 \\ -1 & \text{otherwise,} \end{cases} \quad (24)$$

where the first condition corresponds to successful packet delivery and the second condition to buffer overflow. The constants $\{\beta^k\}_{k\in\mathcal{K}}$ and $\xi > 0$ are hyperparameters under the control of the network operator at the DT. In our experiments, we set $\beta^k = 1$ for all $k \in \{1,2,3,4\}$, $\xi = 50$, and the discount parameter in (8) is set to $\gamma = 0.95$.

4) Actor and Critic: The critic $Q_w(s_t,a_t)$ and the policies $\pi_v^k(a_t^k|h_t^k)$ for the COMA algorithm presented in Sec. IV-A are implemented as feedforward neural networks. Specifically, the policy $\pi_v^k(a_t^k|h_t^k)$ takes as input its current observation o_t^k, along with the positional input $p_t = (t \mod L)$, where $L = 4$ is a hyperparameter, resulting in a policy of the form $\pi^k(a_t^k|o_t^k,p_t)$. More precisely, each neural network $\pi_v^k(a_t^k|h_t^k)$ outputs L probabilities $\{\pi_v^k(a_t^k|o_t^k,p)\}_{p=0}^{L-1}$ such that $\pi_v^k(a_t^k|h_t^k) = \prod_{p=0}^{L-1} \mathbb{1}_{\{p=p_t\}} \pi_v^k(a_t^k|o_t^k,p)$. Partitioning time into frames of L slots, $\pi_v^k(a_t^k = 1|o_t^k,p)$ can be interpreted as the probability of sending a packet during slot p within the current frame. The adoption of more complex policies using recurrent neural networks (RNNs) [68] is left for future work.

C. Benchmarks

Throughout the experiments, we compare the performance of the proposed Bayesian framework to the two following benchmarks. The first is a *frequentist* model-based approach, which obtains a maximum a posteriori (MAP) estimate $\theta^{\mathrm{MAP}} = \arg\max_\theta P(\theta|\mathcal{D}_T^{\pi_d})$ of the model parameter vector θ during model learning with all Dirichlet prior parameters set to 1.01. This choice guarantees well-defined solutions for the MAP problem. The frequentist approach uses the single optimized model $T_{\theta^{\mathrm{MAP}}}(s_{t+1}|s_t,a_t)$ for policy optimization, anomaly detection, and prediction. For policy optimization, we also consider an *oracle-aided* model-free scheme, in which the policy optimizer is allowed to interact with the ground-truth distributions (20) and (21) until convergence.

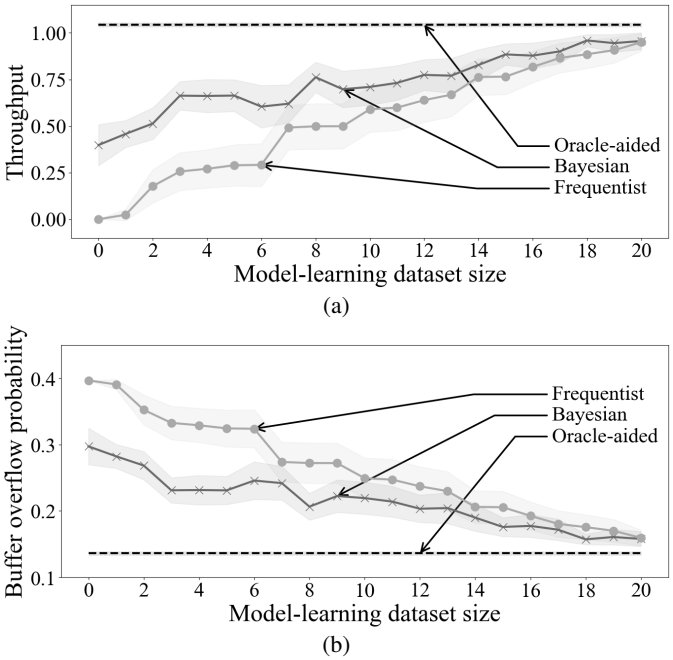


Fig. 4: Throughput (a) and buffer overflow probability (b) as a function of the size of the dataset available in the model learning phase for the proposed Bayesian model-based approach, as well as the oracle-aided model-free and frequentist model-based benchmarks. Metrics are averaged over time and over 50 independent model learning and policy optimization cycles.

D. Policy Evaluation

In this section, we evaluate the performance of policy optimization in the ground-truth environment by using the following metrics: *(i)* the *throughput*, i.e., the average number of packets successfully sent at each time step (Fig. 4a); and *(ii)* the average probability of *buffer overflow* across all devices (Fig. 4b). We focus on the impact of the size of the model learning dataset $\mathcal{D}_T^{\pi_d}$ by varying the number of random data collection steps T from 0 to 20 prior to the model learning phase. The results are averaged over 50 independent data collection, model learning and policy optimization cycles (phases ①, ② and ③ in Fig. 1).

From Fig. 4, we observe that, in regimes with high data availability during the model learning phase, i.e., with large T, both Bayesian and frequentist model-based methods yield policies with similar performance to the oracle-aided benchmark. In the low-data regime, however, Bayesian learning achieves superior performance as compared to its frequentist counterpart with, for instance, a 20% increase in throughput at $T = 10$. With frequentist learning, which disregards epistemic uncertainty, policy optimization is prone to *model exploitation*, whereby the optimized policy is misled by model errors into taking actions that are unlikely to be advantageous in the ground-truth dynamics. By using an ensemble of models with distinct transition dynamics in state-action space regions with high epistemic uncertainty, Bayesian learning reduces the sensitivity of the optimized policy to model errors.

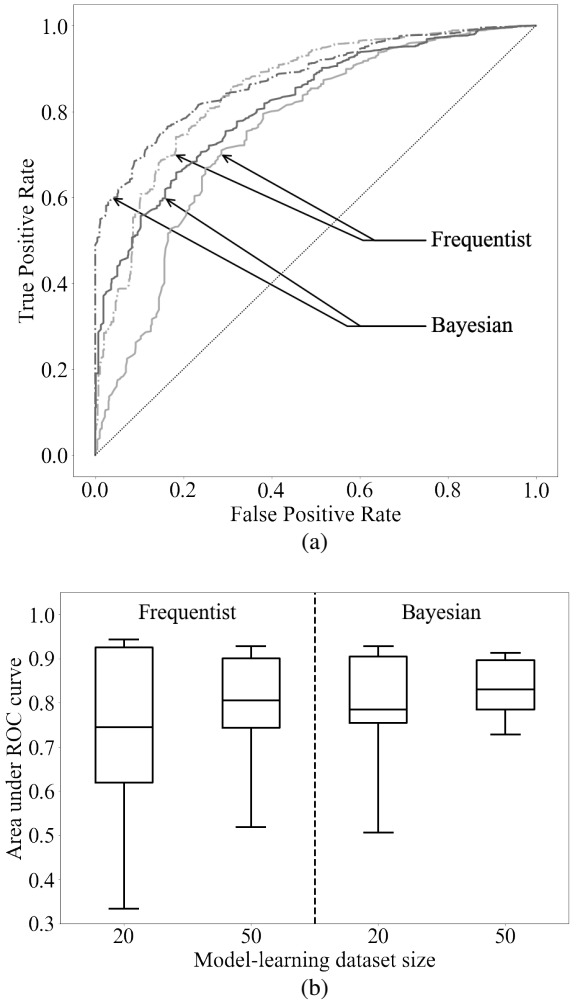


Fig. 5: Mean receiver operating characteristic (ROC) curves (a) and area under ROC curves (AUC) (b) for the Bayesian and frequentist anomaly detection tests. Solid lines in (a) represent model learning dataset sizes of $T = 20$ steps, while dashed lines correspond to dataset sizes of $T = 50$ steps. Mean AUCs in (b) are represented by an horizontal bar, while boxes denote the 25% and 75% quantiles and whiskers denote the 10% and 90% quantiles. Results are obtained from 50 independent data collection and model learning cycles.

E. Anomaly Detection

We now consider the performance of anomaly detection, as defined in Sec. V-A, by assuming that an anomalous event occurs when device 2 is disconnected, resulting in an anomalous packet-generation distribution $\tilde{P}(g_{t+1})$ for which a packet is generated at device 1 only with probability 0.4, and no packet is generated either at device 1 or 2 with probability 0.6. To focus on such anomalies at the packet generation level, we use the log-likelihood $LL(\mathcal{D}_{T^{\mathrm{M}}}^{\pi}|\theta) = \sum_{t=1}^{T^{\mathrm{M}}} \log(T_{\theta^{\mathrm{G}},1}(g_{t+1}^{\mathcal{C}^1}))$ in the disagreement metric (18). Furthermore, as mentioned in Sec. VII-C, we consider as benchmark a standard test based on the log-likelihoods $LL(\mathcal{D}_{T^{\mathrm{M}}}^{\pi}|\theta^{\mathrm{MAP}})$ obtained from MAP-based frequentist learning.

For each model learning dataset size $T = 20$ and $T = 50$,

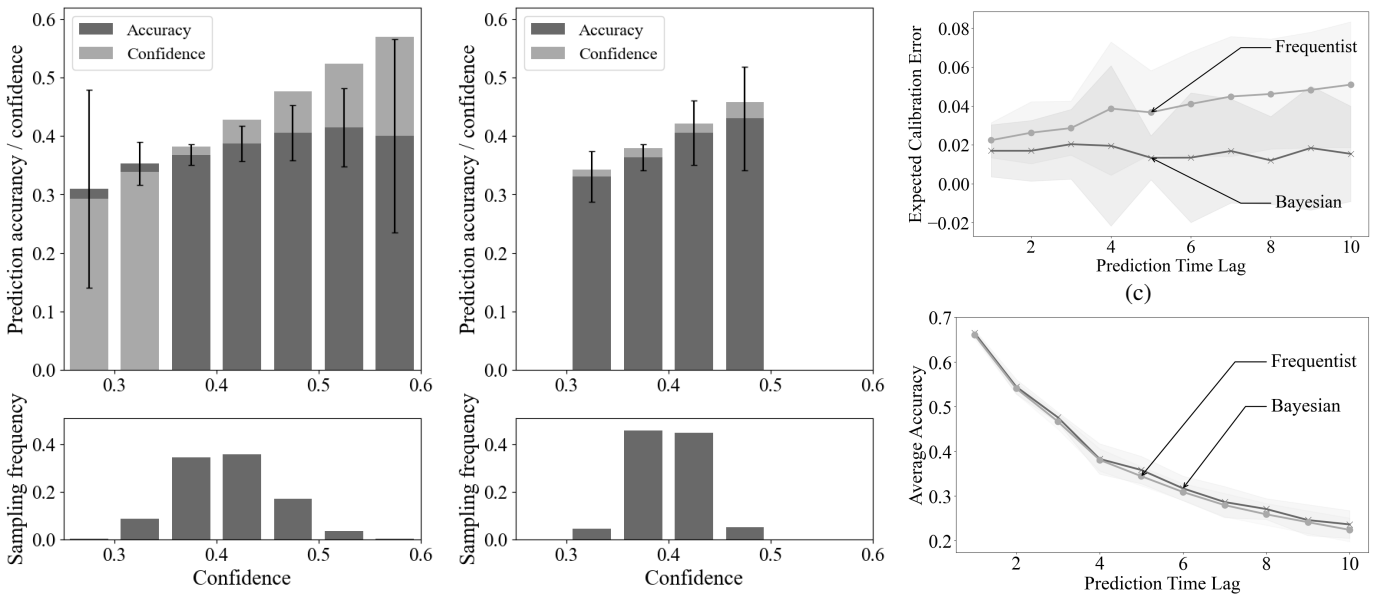


Fig. 6: Reliability plots for packet drop prediction with time lag $T^{\mathrm{H}} = 4$ for (a) the frequentist MAP model and (b) the Bayesian model. (c) Expected calibration error and (d) accuracy of the predictions for both Bayesian and frequentist models as a function on the prediction time lag T^{H}, ranging from $T^{\mathrm{H}} = 1$ to $T^{\mathrm{H}} = 10$. All the results are averaged over 20 independently learned models.

we compute the Bayesian disagreement metrics and frequentist log-likelihoods for 16000 independently sampled monitoring datasets $\mathcal{D}^{\pi}_{T^{\mathrm{M}}}$ with $T^{\mathrm{M}} = 1$, where half of the datasets are sampled from the ground-truth distribution under normal circumstances, while the other half is sampled with device 2 disconnected. We then report the false positive rates (FPR) and the true positive rates (TPR) of the anomaly detection tests in Fig. 5 by varying the detection threshold. The experiment is repeated 50 times over independent data collection and model learning phases (phases ① and ② in Fig. 1), while the optimized policy π used to report experiences $\mathcal{D}^{\pi}_{T^{\mathrm{M}}}$ remains the same.

For both model-learning dataset sizes of $T = 20$ and $T = 50$ steps in Fig. 5b, Bayesian anomaly detection achieves, on average, a higher area under the receiver operating characteristic (ROC) curve; with a 5% average area increase and a 22% larger area at the 25% quantile for $T = 20$ compared to its frequentist counterpart. From Fig. 5a, the proposed Bayesian framework is also observed to uniformly outperform the frequentist ROC curve for $T = 20$ steps, while providing higher performance at lower FPR for $T = 50$ steps. For instance, at a TPR of 0.75 in Fig. 5a, the Bayesian anomaly detector has a FPR of 0.30 for a model learning dataset size of $T = 20$ and a FPR of 0.15 for a dataset size of $T = 50$; whereas the frequentist benchmark has a FPR of 0.34 for $T = 20$ and 0.21 for $T = 50$. These results suggest that measuring epistemic uncertainty, instead of likelihood, can yield more effective and robust monitoring solutions.

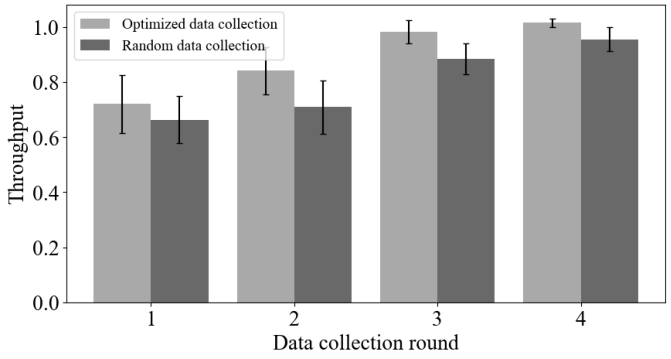
F. Prediction

In this section, we are interested in predicting the number of packet drops, i.e., buffer overflows, experienced across all devices starting from a uniformly sampled state s_1. We collect $T = 100$ data samples using a random data collection policy π_d, train a Bayesian model $P(\theta|\mathcal{D}^{\pi_d}_T)$, and use it to produce an optimized policy π as described in Sec. IV-A. Following Sec. V-B, we define our target metric over the time lag $T^{\mathrm{H}} \in \{1, \dots, 10\}$ as

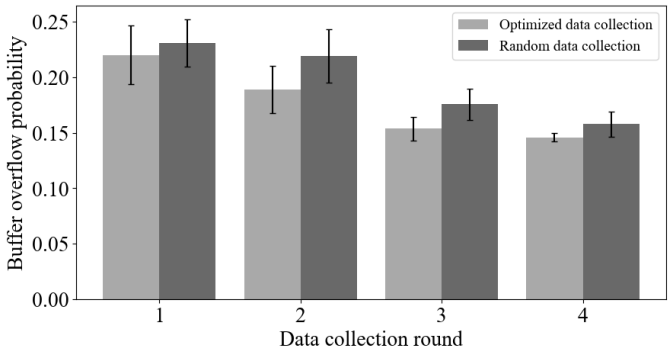
$$y_p = \sum_{t=1}^{T^{\mathrm{H}}} \sum_{k \in \mathcal{K}} \mathbb{1}_{\{q^k_t = Q^k_{\max}, g^k_{t+1} = 1, d^k_{t+1} = 0\}}, \tag{25}$$

where the state variables of future trajectories in $\mathcal{D}^{\pi}_{T^{\mathrm{H}}}$ are taken with respect to the optimized policy π. Note that, since the optimized policy π differs from the data collection policy π_d, the datasets $\mathcal{D}^{\pi_d}_T$ and $\mathcal{D}^{\pi}_{T^{\mathrm{H}}}$ are drawn from two distinct distributions. Therefore, the number of packet drops y_p cannot be predicted from the currently available data $\mathcal{D}^{\pi_d}_T$, and the accuracy of the prediction depends on how well the learned model T_θ at the DT can generalize to new, unseen, conditions.

In order to estimate the packet-drop rate y_p, we roll out 10-steps trajectories from s_1 using the learned model. Furthermore, for the Bayesian model, we average the confidence of each prediction over 20 sampled models T_θ with $\theta \sim P(\theta|\mathcal{D}^{\pi_d}_T)$, with 100 trajectories per model; while, for the frequentist MAP benchmark, we only average 100 trajectories over the single model $T_{\theta^{\mathrm{MAP}}}$ with $\theta^{\mathrm{MAP}} = \arg\max_\theta P(\theta|\mathcal{D}^{\pi_d}_T)$. The predicted outcome y_p in (25) is tested against 100 outcomes sampled from the ground-truth environment with policy π and starting state s_1. We average



(a)



(b)

Fig. 7: (a) Throughput and (b) buffer overflow probability as a function of the number of data collection rounds using a random (dark gray) and an optimized (light gray) data collection policy, as described in Sec. IV-C. All the results are averaged over 50 independent data collection, model learning and policy optimization cycles.

our results over 20 independent data collection and model learning cycles (phases ① and ② in Fig. 1) for 200 uniformly sampled starting states s_1.

We evaluate the performance both in terms of accuracy (Fig. 6d) and calibration (Fig. 6a-c). Calibration performance is evaluated using the standard *reliability plot* and *expected calibration error* (ECE) [69]. As seen in Fig. 6d, the prediction accuracy of the Bayesian and frequentist approaches are very similar for all values of $T^{\mathrm{H}} \in \{1, \ldots, 10\}$. However, as we increase the prediction time lag T^{H}, the frequentist approach tends to make incorrect decisions with a high level of confidence, while Bayesian learning correctly evaluates its confidence level.

To see this, we first observe the reliability plots in Fig. 6a-b, which are obtained for $T^{\mathrm{H}} = 4$. Reliability plots evaluate prediction accuracy as a function of the confidence level of the decision output by the model. Perfect calibration is obtained when the confidence (light gray) and accuracy (dark gray) bars are equal. As anticipated, the frequentist model is observed to be overconfident, while the Bayesian model provides a good match between confidence and accuracy at all confidence levels with a meaningful rate of occurrence (displayed at the bottom of the reliability plots). The ECE, which evaluates the average difference between confidence and accuracy [69] (Fig. 6c), confirms the advantages of Bayesian learning in terms of quality of uncertainty quantification.

G. Data Collection Optimization

An optimized data collection policy, as described in Sec. IV-C, can be useful to improve the estimate of the channel distribution $P(d_{t+1}|a_t)$ since the latter can be explored by controlling the number of transmitted packets. In this last experiment, we evaluate the advantages of data collection policy optimization across four data collection rounds.

During each round $i \in \{1, 2, 3, 4\}$, the DT collects information about $T^d = 5$ transitions in the ground-truth environment using the data collection policy $\pi_{d,i}$. The latter is optimized as discussed in Sec. IV-C using the available data $\mathcal{D}_{\leq i-1} = \bigcup_{j=1}^{i-1} \mathcal{D}_{T^d}^{\pi_{d,j}}$. Note that in this problem the data collection reward (16) can be evaluated in closed form using the digamma function [70].

We evaluate the advantage of the optimized data collection scheme by training a control policy π as detailed in Sec. IV-A using the model $P(\theta|\mathcal{D}_{\leq i})$ available at the end of each round, and evaluating its performance in the ground-truth environment in terms of throughput (Fig. 7a) and buffer overflow probability (Fig. 7b), as described in Sec. VII-D. The results presented in Fig. 7 are averaged over 50 independent data collection (with and without optimization), model learning and policy optimization cycles (phases ①, ②, ③ and ④ in Fig. 1).

Since the data collection policy $\pi_{d,1}$ is trained using the prior model $P(\theta)$ during the first round, the models T_θ with $\theta \sim P(\theta)$ tend to disagree under most transitions, and the performance of the optimized data collection scheme is close to its random counterpart. However, after the first round, the data collection reward (16) is able to target a smaller subset of transitions with higher epistemic uncertainty, yielding a 18.5% increase in throughput at the end of the second round compared to random exploration. As the number of rounds increases, the performance gap between the two collection strategies is reduced and we approach the optimal performance of the oracle-aided benchmark in Fig. 4.

VIII. Conclusions

This paper has proposed a Bayesian framework for the development of a DT platform aimed at the control, monitoring, and analysis of a communication system. By accounting for model uncertainty via ensembling, and compared to conventional single-model approaches, the proposed Bayesian DT framework was shown to obtain more reliable performance for multi-agent RL-based control, prediction, anomaly detection, and data collection in the regime of limited data available at the DT from the PT. For some quantitative examples, we demonstrated a 20% increase in throughput for multi-access transmission from IoT devices, with an additional 18.5% increase obtained by using an optimized data-collection policy; a 5% larger area under the ROC curve for anomaly detection; and a reduction by half of the calibration error for prediction. Future work may investigate the application of the Bayesian DT framework to other use cases in telecommunication [24],

[28]; the use of more complex policies accounting for partial observability at each agent [68]; as well as the presence of multiple interacting DTs and/or PTs [55], along with the optimal allocation of DTs across cloud and edge [29], [30].

APPENDIX A
TABLE OF NOTATIONS

Notation	Meaning
K	Number of agents in the PT system
o_t^k	Observation of agent k at time step t
s_t	Overall state of the PT system at time step t
s_t^i	i-th subset of state variables of the PT system at time step t
a_t^k	Action of agent k at time step t
a_t	Joint action of all agents at time step t
h_t^k	Action-observation history of agent k up to time step t
$\pi^k(a_t^k \vert h_t^k)$	Policy of agent k
$\pi(a_t \vert s_t)$	Decentralized policy of all agents
$\pi_d^k(a_t^k \vert h_t^k)$	Data-collection policy of agent k
$\pi_d(a_t \vert s_t)$	Data-collection policy of all agents
$T(s_{t+1} \vert s_t, a_t)$	PT ground-truth transition probability
$T_\theta(s_{t+1} \vert s_t, a_t)$	DT model of the PT transition probabilities with parameter θ
$T_{\theta^i}^i(s_{t+1}^i \vert s_t^{>i}, a_t^{>i})$	DT model of the transition probabilities of the i-th state subset with parameter θ^i
$\mathcal{D}_T^{\pi_d}$	Dataset containing T transitions collected from the PT system under policy π_d
$r(s_t, a_t, s_{t+1})$	Reward function for transition (s_t, a_t, s_{t+1})
$r_e(s_t, a_t, s_{t+1})$	Reward function with exploration bonus
α_e	Exploration bonus temperature parameter
$r_d(s_t, a_t)$	Data collection reward function
G_t	Total discounted return from time step t
γ	Discounting factor
T^{M}	Monitoring time window
$\mathcal{D}_{T^{\mathrm{M}}}^{\pi}$	Monitoring dataset containing T^{M} transitions collected from the PT under policy π
T^{H}	Prediction time lag
$\mathcal{D}_{T^{\mathrm{H}}}^{\pi}$	Predicted PT trajectory under policy π containing T^{H} transitions
y_p	Prediction target metric

TABLE I: Notations

REFERENCES

[1] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*. Springer, 2017, pp. 85–113.

[2] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.

[3] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in *in Proc. AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1818.

[4] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong *et al.*, "Network digital twin: Context, enabling technologies and opportunities," *arXiv preprint arXiv:2205.14206*, 2022.

[5] A. Akman, C. Li, L. Ong, L. Suciu, B. Sahin, T. Li, P. Stjernholm, J. Voigt, A. Buldorini, Q. Sun *et al.*, "ORAN use cases and deployment scenarios: Towards open and smart RAN," *O-RAN Alliance, White Paper, Feb*, 2020.

[6] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[7] L. Hui, M. Wang, L. Zhang, L. Lu, and Y. Cui, "Digital twin for networking: A data-driven performance modeling perspective," *arXiv preprint arXiv:2206.00310*, 2022.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[9] D. J. MacKay, D. J. Mac Kay *et al.*, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[10] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[11] T. Palmer, *The Primacy of Doubt: From climate change to quantum physics, how the science of uncertainty can help predict and understand our chaotic world*. Oxford University Press, 2022.

[12] E. Chen, L. Lin, and N. T. Dinh, "Advanced transient diagnostic with ensemble digital twin modeling," *arXiv preprint arXiv:2205.11469*, 2022.

[13] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, "A comprehensive review of digital twin–part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives," *arXiv preprint arXiv:2208.12904*, 2022.

[14] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5G networks using Open RAN architectures," *arXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2209.14171

[15] R. Kassab, A. Destounis, D. Tsilimantos, and M. Debbah, "Multi-agent deep stochastic policy gradient for event based dynamic spectrum access," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–6.

[16] A. Valcarce and J. Hoydis, "Toward joint learning of optimal MAC signaling and wireless channel access," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1233–1243, 2021.

[17] L. Miuccio, S. Riolo, S. Samarakoon, D. Panno, and M. Bennis, "Learning generalized wireless MAC communication protocols via abstraction," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 2322–2327.

[18] B. Han, M. A. Habibi, B. Richerzhagen, K. Schindhelm, F. Zeiger, F. Lamberti, F. G. Pratticò, K. Upadhya, C. Korovesis, I.-P. Belikaidis *et al.*, "Digital twins for industry 4.0 in the 6G era," *arXiv preprint arXiv:2210.08970*, 2022.

[19] L. Bariah and M. Debbah, "The interplay of AI and digital twin: Bridging the gap between data-driven and model-driven approaches," *arXiv preprint arXiv:2209.12423*, 2022.

[20] M. Tariq, F. Naeem, and H. V. Poor, "Toward experience-driven traffic management and orchestration in digital-twin-enabled 6G networks," *arXiv preprint arXiv:2201.04259*, 2022.

[21] T. H. Luan, R. Liu, L. Gao, R. Li, and H. Zhou, "The paradigm of digital twin communications," *arXiv preprint arXiv:2105.07182*, 2021.

[22] O. Hashash, C. Chaccour, and W. Saad, "Edge continual learning for dynamic digital twins over wireless networks," *arXiv preprint arXiv:2204.04795*, 2022.

[23] O. Hashash, C. Chaccour, W. Saad, K. Sakaguchi, and T. Yu, "Towards a decentralized metaverse: Synchronized orchestration of digital twins and sub-metaverses," *arXiv preprint arXiv:2211.14686*, 2022.

[24] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4968–4977, 2020.

[25] C. Zhou, J. Gao, M. Li, X. S. Shen, and W. Zhuang, "Digital twin-empowered network planning for multi-tier computing," *Journal of Communications and Information Networks*, vol. 7, no. 3, pp. 221–238, 2022.

[26] X. Wang, L. Ma, H. Li, Z. Yin, T. Luan, and N. Cheng, "Digital twin-assisted efficient reinforcement learning for edge task scheduling," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–5.

[27] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[28] B. Sheen, J. Yang, X. Feng, and M. M. U. Chowdhury, "A digital twin for reconfigurable intelligent surface assisted wireless communication," *arXiv preprint arXiv:2009.00454*, 2020.

[29] J. Jagannath, K. Ramezanpour, and A. Jagannath, "Digital twin virtualization with machine learning for IoT and beyond 5G networks: Research directions for security and optimal control," in *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, 2022, pp. 81–86.

[30] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 219–16 230, 2021.

[31] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, "A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies," *Structural and Multidisciplinary Optimization*, vol. 65, no. 12, pp. 1–55, 2022.

[32] M. Matulis and C. Harvey, "A robot arm digital twin utilising reinforcement learning," *Computers & Graphics*, vol. 95, pp. 106–114, 2021.

[33] K. T. Park, Y. H. Son, S. W. Ko, and S. D. Noh, "Digital twin and reinforcement learning-based resilient production control for micro smart factory," *Applied Sciences*, vol. 11, no. 7, p. 2977, 2021.

[34] C. Cronrath, A. R. Aderiani, and B. Lennartson, "Enhancing digital twins through reinforcement learning," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 293–298.

[35] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[36] G. J. Laurent, L. Matignon, L. Fort-Piat *et al.*, "The world of independent learners is not Markovian," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 15, no. 1, pp. 55–64, 2011.

[37] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," *Neurocomputing*, vol. 190, pp. 82–94, 2016.

[38] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[39] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[40] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang, "Off-policy multi-agent decomposed policy gradients," *arXiv preprint arXiv:2007.12322*, 2020.

[41] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[42] M. P. Mota, A. Valcarce, J.-M. Gorce, and J. Hoydis, "The emergence of wireless MAC protocols with multi-agent reinforcement learning," in *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021, pp. 1–6.

[43] N. Tao, J. Baxter, and L. Weaver, "A multi-agent, policy-gradient approach to network routing," in *In: Proc. of the 18th Int. Conf. on Machine Learning*. Citeseer, 2001.

[44] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, "Dynamic Bayesian network for aircraft wing health monitoring digital twin," *Aiaa Journal*, vol. 55, no. 3, pp. 930–941, 2017.

[45] J. Yu, Y. Song, D. Tang, and J. Dai, "A digital twin approach based on nonparametric Bayesian network for complex system health monitoring," *Journal of Manufacturing Systems*, vol. 58, pp. 293–304, 2021.

[46] M. O. Duff, *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

[47] P. Poupart, N. Vlassis, J. Hoey, and K. Regan, "An analytic solution to discrete Bayesian reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 697–704.

[48] I. Osband, D. Russo, and B. Van Roy, "(More) efficient reinforcement learning via posterior sampling," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[49] P. Shyam, W. Jaśkowski, and F. Gomez, "Model-based active exploration," in *International conference on machine learning*. PMLR, 2019, pp. 5779–5788.

[50] Q. Zhang, C. Lu, A. Garg, and J. Foerster, "Centralized model and exploration policy for multi-agent RL," *arXiv preprint arXiv:2107.06434*, 2021.

[51] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *arXiv preprint arXiv:2012.08405*, 2020.

[52] C. Ruah, O. Simeone, and B. Al-Hashimi, "Digital twin-based multiple access optimization and monitoring via model-driven Bayesian learning," *arXiv preprint arXiv:2210.05582*, 2022.

[53] O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro, "Placement of social digital twins at the edge for beyond 5G IoT networks," *IEEE Internet of Things Journal*, 2022.

[54] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[55] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[56] I. Vilà, O. Sallent, and J. Pérez-Romero, "On the design of a network digital twin for the radio access network in 5g and beyond," *Sensors*, vol. 23, no. 3, p. 1197, 2023.

[57] J. Chen, Y.-C. Liang, Y. Pei, and H. Guo, "Intelligent reflecting surface: A programmable wireless environment for physical layer security," *IEEE Access*, vol. 7, pp. 82 599–82 612, 2019.

[58] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[59] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[60] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[61] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[62] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[63] E. Daxberger and J. M. Hernández-Lobato, "Bayesian variational autoencoders for unsupervised out-of-distribution detection," *arXiv preprint arXiv:1912.05651*, 2019.

[64] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer, 2004, vol. 26.

[65] C. Ruah, "Bayesian Digital Twin Repository," 2022. [Online]. Available: https://github.com/kclip/bayesian-dt

[66] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," in *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*. VDE, 2013, pp. 1–5.

[67] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: From signal processing to optimal medium access control," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 108–112, 2001.

[68] P. Zhu, X. Li, P. Poupart, and G. Miao, "On improving deep reinforcement learning for POMDPs," *arXiv preprint arXiv:1704.07978*, 2017.

[69] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[70] M. Scutari, "Dirichlet Bayesian network scores and the maximum relative entropy principle," *Behaviormetrika*, vol. 45, no. 2, pp. 337–362, 2018.

Clement Ruah is currently pursuing his Ph.D. degree in Machine Learning at King's College London, United Kingdom. He received his postgraduate degree in Engineering from the French "grande ecole" CentraleSupelec, and a Master's degree in Biomedical Engineering (neuroscience stream) with distinction from Imperial College London in 2018. His research interest include model-based machine learning, Bayesian learning and multi-agent reinforcement learning.

Osvaldo Simeone is a Professor of Information Engineering with the Centre for Telecommunications Research at the Department of Engineering of King's College London, where he directs the King's Communications, Learning and Information Processing lab. He received an M.Sc. degree (with honors) and a Ph.D. degree in information engineering from Politecnico di Milano, Milan, Italy, in 2001 and 2005, respectively. From 2006 to 2017, he was a faculty member of the Electrical and Computer Engineering (ECE) Department at New Jersey Institute of Technology (NJIT), where he was affiliated with the Center for Wireless Information Processing (CWiP). His research interests include information theory, machine learning, wireless communications, neuromorphic computing, and quantum machine learning. Dr Simeone is a co-recipient of the 2022 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Vehicular Technology Society Jack Neubauer Memorial Award, the 2019 IEEE Communication Society Best Tutorial Paper Award, the 2018 IEEE Signal Processing Best Paper Award, the 2017 JCN Best Paper Award, the 2015 IEEE Communication Society Best Tutorial Paper Award and of the Best Paper Awards of IEEE SPAWC 2007 and IEEE WRECOM 2007. He was awarded an Open Fellowship by the EPSRC in 2022 and a Consolidator grant by the European Research Council (ERC) in 2016. His research has been also supported by the U.S. National Science Foundation, the European Commission, the European Research Council, the Vienna Science and Technology Fund, the European Space Agency, as well as by a number of industrial collaborations including with Intel Labs and InterDigital. He was the Chair of the Signal Processing for Communications and Networking Technical Committee of the IEEE Signal Processing Society in 2022, as well as of the UK & Ireland Chapter of the IEEE Information Theory Society from 2017 to 2022. He was a Distinguished Lecturer of the IEEE Communications Society in 2021 and 2022, and he was a Distinguished Lecturer of the IEEE Information Theory Society in 2017 and 2018. Prof. Simeone is the author of the textbook "Machine Learning for Engineers" published by Cambridge University Press, four monographs, two edited books, and more than 200 research journal and magazine papers. He is a Fellow of the IET, EPSRC, and IEEE.

Bashir M. Al-Hashimi (Fellow, IEEE) is an ARM Professor of computer engineering and Vice President (Research & Innovation) of King's College London. He worked in the electronics design industry for eight years prior to embarking on an academic career with the School of Electronics and Computer Science, in the University of Southampton, in 1999. In 2008, he founded the Arm-ECS industry-academia centre of research excellence in energy-efficient computing. As an interdisciplinary researcher, he has successfully led a number of large-scale interdisciplinary research programmes funded by the EPSRC and industry. He has supervised 40 Ph.D. students to successful completion, published 380 referred technical papers, and authored or co-authored seven books.