# Digital Twin Model Selection for Feature Accuracy in Wireless Edge Networks

Hong Chen*, Terence D. Todd*, Dongmei Zhao* and George Karakostas†
*Department of Electrical and Computer Engineering
†Department of Computing and Software
McMaster University
Hamilton, Ontario, CANADA
Email: {chenh151,todd,dzhao,karakos}@mcmaster.ca

***Abstract*—Digital twins (DTs) are virtual implementations of real physical systems (PSs) that interact with other objects on their behalf. Each PS periodically communicates with its digital twin so that the state of the DT is always sufficiently current. Using these updates, a DT can provide features that represent the real behavior of its PS using models that yield differing levels of system accuracy. In this paper, we study the DT model selection problem in wireless networks where the DTs of multiple PSs are hosted at an edge server (ES). The accuracy obtained from a given model is a function of its required amount of PS input data, the updating frequency, and the amount of computational capacity needed at the ES. The objective is to maximize the minimum achieved accuracy among the requested features by making appropriate model selections subject to wireless channel and ES resource availability. The problem is first formulated as an NP-complete integer program. The paper then uses relaxation and dependent rounding, and introduces a polynomial time approximation algorithm to obtain good solutions. A variety of simulation results are presented that demonstrate the excellent performance of the proposed solution.**

***Index Terms*—Digital twin, digital twin features and models, digital twin model selection for accuracy, edge server.**

## I. Introduction

A digital twin (DT) is a virtual representation of a real physical system (PS) that can represent both the static and dynamic behavior of the PS [1]. In order to maintain real-time synchronization between the PS and its DT, the PS periodically sends state-related data to a server that hosts the DT. At the server, the data is processed so that PS features can be provided by the DT to other objects on its behalf [2]. The DT concept has attracted a lot of attention from both industry and academia since it was proposed over a decade ago. By taking advantage of the resources available in the digital space, DTs have proven to be beneficial in various application areas, such as predictive maintenance and optimization in industrial manufacturing [3] and intelligent network resource management [2], [4].

Ideally, a DT should exactly reflect all features of its PS at all times [5]. This objective may be impractical and unnecessary, since it may require excessive network resources [1], [6] and achieve feature accuracies that are not needed by a requesting application. A DT implementation is considered *fully functional* if it can provide sufficient information of the PS required by an application in terms of age of information (AoI) [7] and/or level of accuracy [1], [5], [6], [8].

A real PS usually has a large number of features. The level of accuracy of each feature that should be supported by its DT depends on the requirements of the applications that interact with it. For example, the work in [9] proposes an approach enabling Accuracy-as-a-Service for resistance-based sensors in production systems. Depending on the accuracy level required by the production control, a standard model or more accurate individual models may be used. In general, higher accuracy models require larger volumes of data and higher data processing loads at the DT [9]–[13].

The accuracy level of the features supported by a digital twin is an important indicator of its performance [5]. The work in [10] proposes a deep lifelong learning method for DT-driven defect class recognition, where the learning models provide higher recognition accuracy for new defect classes compared to pre-trained DT models. Reference [11] analyzes the geometric accuracy of DTs for structural health monitoring purposes. The Level of Geometric Accuracy (LOGA) is introduced as a way of measuring the twinning quality of the resulting DTs. This work provides prospective twinning methods and deviation analysis for DT-driven structural health monitoring. The work indicates that the choice of the twinning methods and LOGA depends heavily on how the DT is used and what kind of data is required when providing information about geometric accuracy.

Communication between the PS and its DT often involves wireless transmission. In order to maintain tight synchronization between the two, the DT should often be placed at an edge server (ES) that is located within the same wireless network as the PS. In this case, the quality of a DT feature is affected by both the wireless transmission quality and the computational capacity of the ES that hosts the DT. For an edge network that supports DTs for a large number of PSs, an issue is how to provide the best DT feature accuracy under limited wireless channel and edge computing resources.

In this paper, we study the DT model selection accuracy problem in a wireless edge network, where multiple PSs communicate with their DTs at an edge server that is located close to the PS's serving base station (BS). The PS features can be implemented by different models with corresponding accuracies. Each model is characterized by its required amount of input data from the PS, the data updating frequency, and the amount of computational load needed at the ES. The goal is to maximize the minimum achieved accuracy among the PS features that are requested of the DTs, by choosing the appropriate set of models, subject to the wireless channel and ES resource availability. The problem is formulated as an integer program (IP) problem whose solution is an NP-complete problem. The paper relaxes the IP, and uses dependent rounding to address the problem. A polynomial time approximation algorithm is introduced that gives approximate solutions with asymptotic bounds on constraint violation. The provably small bounds imply that available resources need to be augmented only by small amounts in order to render our solutions feasible for the system. A variety of simulation results are presented that demonstrate the excellent performance of the proposed solution. The results include comparisons to the optimal solutions that can be obtained using exhaustive enumeration for small problem instances.

The remainder of the paper is organized as follows. The system model and proposed optimization problem are described in Section II. In Section III, the solution to the formulated optimization problem is presented. Simulation results are demonstrated in Section IV. Finally, the conclusions are drawn in Section V.

## II. System Model and Problem Formulation

We consider a wireless network that connects an ES with multiple PSs, when ES hosts a DT for each PS. The ES is located at a BS, as shown in Fig. 1. Each DT handles multiple PS features. As is typical for DTs, periodically the PS collects data to upload to the DT or receives data from the DT, so that the different features implemented at the DT are synchronized with the PS state.

Let $N$ be the total number of the PSs in the system and $K$ be the total number of the features. Define $\beta_{i,k} \in \{0,1\}$ as a binary variable representing the demand of PS $i$ for feature $k$: if $\beta_{i,k} = 1$, PS $i$ requires feature $k$, otherwise $\beta_{i,k} = 0$. Let $T_{i,k}$ be the data refreshing period of feature $k$ in PS $i$. Each feature can be realized by one of the different models in the DT with corresponding accuracy. There are $M_k$ models that can be used by feature $k$ to realize its function. Let $\mathcal{I}, \mathcal{K}, \mathcal{M}_k$ be the sets of PSs, features, and models for feature $k$, respectively.

Let $\Phi_{i,k,m}$ be the accuracy that feature $k$ of the DT can achieve when it is realized using model $m$ by DT $i$, and $s_{i,k,m}$ (in bits) be the input data needed by DT $i$ in time period $T_{i,k}$ realizing feature $k$ using model $m$. Let $x_{i,k,m} \in \{0,1\}$ be the decision variable indicating whether the DT of PS $i$ chooses to use model $m$ to realize feature $k$. Note that $x_{i,k,m} = 0$ if $\beta_{i,k} = 0$. We assume that each feature can choose one and only one model to realize its function, i.e.,

$$\sum_{m=1}^{M_k} x_{i,k,m} = 1, \quad \forall i, k : \beta_{i,k} = 1. \tag{1}$$

When feature $k$ is implemented by the $i$-th DT (i.e., $\beta_{i,k} = 1$), the achieved accuracy for this feature is

$$\Psi_{i,k} = \sum_{m=1}^{M_k} x_{i,k,m} \Phi_{i,k,m}. \tag{2}$$

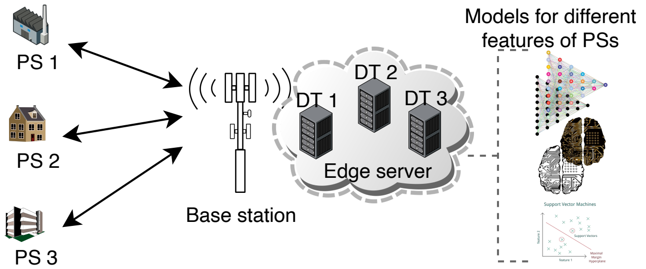

Fig. 1. System Model

The objective studied in this paper is the maximization of the minimum accuracy achieved by our choice of models for any feature implemented by any DT, i.e., $\max_x \min_{i,k:\beta_{i,k}=1} \Psi_{i,k}$, while respecting computational and data transmission constraints, as well as the periodicity of DT updates.

More specifically, when PS $i$ transmits data to its DT, the wireless transmission rate $R_i$ of PS $i$ should be at least the rate required for the timely transmission of data between the $i$-th pair of PS and DT within time period $T_{i,k}$ for each feature $k$ used, i.e.,

$$R_i \geq \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{s_{i,k,m}}{T_{i,k}} x_{i,k,m}, \tag{3}$$

where $R_i$ can be calculated by $R_i = w\log_2(1 + \frac{P_i^{\mathrm{T}} g_i}{\sigma^2})$, in which $w$ is the wireless channel bandwidth, $P_i^{\mathrm{T}}$ and $g_i$ are the wireless transmission power and the link gain from PS $i$ to BS, respectively, and $\sigma^2$ denotes the noise power at the BS receiver input.

Let $F$ be the available computation rate of ES in CPU cycles per second. If $f_{i,k,m}$ denotes the CPU cycles needed by the $i$-th DT in order to process feature $k$ using model $m$, the following capacity constraint must hold:

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{f_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq F. \tag{4}$$

As a result of the discussion above, we obtain the following IP formulation of the problem:

$$\max_{x} \min_{i,k:\beta_{i,k}=1} \Psi_{i,k} \text{ s.t.} \tag{IP}$$

$$\sum_{m=1}^{M_k} x_{i,k,m} = 1, \forall i, k : \beta_{i,k} = 1 \tag{5}$$

$$\sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{s_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq w\log_2(1 + \frac{P_i^{\mathrm{T}} g_i}{\sigma^2}), \forall i \tag{6}$$

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{f_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq F, \tag{7}$$

$$x_{i,k,m} \in \{0,1\}, \forall i, k, m \tag{8}$$

Equivalently, the problem can be formulated as follows:

$$\max_{x,\tau} \tau \text{ s.t.} \tag{IP'}$$

$$\Psi_{i,k} \geq \tau, \forall i, k : \beta_{i,k} = 1$$

$$\sum_{m=1}^{M_k} x_{i,k,m} = 1, \forall i,k : \beta_{i,k} = 1$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M_k} \frac{s_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq w\log_2(1+\frac{P_i^{\mathrm{T}} g_i}{\sigma^2}), \forall i$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M_k} \frac{f_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq F,$$

$$x_{i,k,m} \in \{0,1\}, \forall i,k,m$$

$$\tau \geq 0.$$

We observe that (2) implies that given $i, k$, the optimal $\Psi_{i,k}$ can take one of the $\Phi_{i,k,m}$ values, where $m = 1, 2, \ldots, M_k$. Therefore, the optimal $\tau$ can take one of at most $N\sum_k M_k$ values. Hence, in what follows we will assume that we have already 'guessed' (i.e., we try all possible values of) $\hat{\tau}$, which is the optimal $\tau$. Given $\hat{\tau}$, we set $x_{i,k,m} := 0$ for all $i, k, m$ that correspond to a $\Phi_{i,k,m} < \hat{\tau}$ in (2), and reduce problem (IP') to the following, simpler, feasibility problem:

$$\max_x \; 0 \quad \text{s.t.} \tag{FIP}$$

$$\sum_{m=1}^{M_k} x_{i,k,m} = 1, \forall i,k : \beta_{i,k} = 1 \tag{9}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M_k} \frac{s_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq w\log_2(1+\frac{P_i^{\mathrm{T}} g_i}{\sigma^2}), \forall i \tag{10}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M_k} \frac{f_{i,k,m}}{T_{i,k}} x_{i,k,m} \leq F, \tag{11}$$

$$x_{i,k,m} \in \{0,1\}, \forall i,k,m \tag{12}$$

Deciding whether problem (FIP) is feasible is NP-complete, which can be proved by a reduction of the well-known Knapsack problem.

To summarize, solving problem (IP) is reduced to the following steps:

1) Sort numbers $\{\Phi_{i,k,m}\}$ in decreasing order $\mathcal{O} = \{\Phi_1, \Phi_2, \Phi_3, \ldots\}$.
2) Starting with $\Phi_1$, set $\hat{\tau}$ equal to the current element of $\mathcal{O}$.
3) Set $x_{i,k,m} = 0$ if $\Phi_{i,k,m} < \hat{\tau}$ or $\beta_{i,k} = 0$.
4) Solve (FIP).
5) If (FIP) is feasible then return $\hat{\tau}$, else let $\hat{\tau}$ equal the next element in $\mathcal{O}$, or return **Infeasible** if the latter does not exist.

The general structure of our method is given in Algorithm 1.

## III. An $O(\frac{\ln N}{\ln\ln N})$-approximation algorithm

Since solving problem (IP), or, equivalently, (IP'), is NP-complete, we propose a polynomial-time approximation algorithm, that will guarantee a solution $\tau_s$ of (IP') with $\tau_s \geq \tau_{opt}$, where $\tau_{opt}$ is the optimal solution of (IP'), but by violating constraints (6), (7) by a factor of at most $O(\frac{\ln N}{\ln\ln N})$.

We note that by relaxing constraints (8) in (IP) and subsequent problems to $x_{i,k,m} \geq 0$, $\forall i,k,m$, problem (IP) (and all subsequent problems in Section II) becomes a linear programming (LP) problem. When lines 8-9 in Algorithm 1 are applied to the linear relaxation of (FIP) instead of (FIP), the algorithm returns solution $\tau_f \geq \tau_{opt}$, unless the relaxed problem is infeasible (and, therefore, the original problem is also infeasible). If $x_f$ is the optimal fractional solution achieving $\tau_f$, then $x_f$ can be computed in polynomial time. Our approximation algorithm will use the *dependent rounding* of [14] to round $x_f$ to values 0 or 1 with the required guarantees. Note that lines 3-7 of Algorithm 1 guarantee that all $\Phi_{i,k,m}$ with $x_{i,k,m} > 0$ are no worse than $\tau_f$, and, therefore, the rounded solution $\tau_s$ cannot be worse than $\tau_f$.

**Algorithm 1** General solution method

```
 1: Sort numbers $\{\Phi_{i,k,m}\}$ in decreasing order $\mathcal{O} = \{\Phi_1, \Phi_2, \Phi_3, \ldots\}$
 2: for $\hat{\tau} = \Phi_1, \Phi_2, \Phi_3, \ldots$ do
 3:     for All $i, k, m$ do
 4:         if $\Phi_{i,k,m} < \hat{\tau} \vee \beta_{i,k} = 0$ then
 5:             $x_{i,k,m} = 0$
 6:         end if
 7:     end for
 8:     Solve (FIP)
 9:     if (FIP) feasible then return solution $\hat{x}, \hat{\tau}$
10:     end if
11: end for
12: return Infeasible
```

First, we give a high-level description of the dependent rounding procedure of [14]. Assume we are given a bipartite graph $(V_1, V_2, E)$ with bipartition $(V_1, V_2)$ and a value $x_{i,j} \in [0,1]$ for each edge $(i,j) \in E$. Initialize $y_{i,j} = x_{i,j}$ for each $(i,j) \in E$. Values $y_{i,j}$ will be probabilistically modified in several (at most $|E|$) iterations such that $y_{i,j} \in \{0,1\}$ at the end, at which point we will set $X_{i,j} := y_{i,j}$ for all $(i,j) \in E$, where $X_{i,j}$ are the (random) rounded final values for edges $(i,j) \in E$.

The iterations that modify $y$ proceed as follows: We call an edge $(i,j)$ *floating* if its value $y_{i,j}$ is not integral (i.e., $y_{i,j} \in (0,1)$). Let $\tilde{E} \subseteq E$ be the current set of floating edges. If $\tilde{E} = \emptyset$, the process terminates by setting $X_{i,j} := y_{i,j}$ for all $(i,j) \in E$. Otherwise, find a maximal path $S$ in the subgraph $(V_1, V_2, \tilde{E})$ in $O(|V_1| + |V_2|)$ time running depth-first-search (DFS). Partition the edge-set of $S$ into two alternating matchings $A$ and $B$. We define

$$\alpha := \min\{\gamma > 0 : (\exists (i,j) \in A : y_{i,j} + \gamma = 1) \vee (\exists (i,j) \in B : y_{i,j} - \gamma = 0)\} \tag{13}$$

$$\beta := \min\{\gamma > 0 : (\exists (i,j) \in A : y_{i,j} - \gamma = 0) \vee (\exists (i,j) \in B : y_{i,j} + \gamma = 1)\}. \tag{14}$$

Then we execute the following randomized step:

- With probability $\beta/(\alpha+\beta)$ set $y_{i,j} := y_{i,j} + \alpha \; \forall (i,j) \in A$, $y_{i,j} := y_{i,j} - \alpha \; \forall (i,j) \in B$, and
- with probability $\alpha/(\alpha+\beta)$ set $y_{i,j} := y_{i,j} - \beta \; \forall (i,j) \in A$, $y_{i,j} := y_{i,j} + \beta \; \forall (i,j) \in B$.

In either case, at least one edge $(i,j) \in \tilde{E}$ will stop being floating, i.e., $y_{i,j} \in \{0,1\}$, and, therefore, after at most $|E|$

iterations or $O(|E|(|V_1|+|V_2|))$ time, all values $y$ will become integral and the rounding process terminates. Algorithm 2 codifies the dependent rounding procedure. Note that dependent rounding is a randomized algorithm, and the final rounded solution $X$ are random variables.

**Algorithm 2** Dependent rounding

```
Bipartite graph $(V_1, V_2, E)$, $x_{i,j} \in [0,1]$ for each edge $(i,j) \in E$
$y_{i,j} = x_{i,j},\ \forall (i,j) \in E$; $\tilde{E} = E$
for all $(i,j) \in \tilde{E}$ do
    if $y_{i,j} \in \{0,1\}$ then
        $\tilde{E} := \tilde{E} \setminus \{(i,j)\}$
    end if
end for
while $\tilde{E} \neq \emptyset$ do
    Maximal path $S = DFS(V_1, V_2, \tilde{E})$; $S = A \cup B$ for alternating matchings $A, B$
    Obtain $\alpha$ and $\beta$ from (13) and (14)
    With probability $\beta/(\alpha+\beta)$, $y_{i,j} := y_{i,j} + \alpha,\ \forall (i,j) \in A$, and $y_{i,j} := y_{i,j} - \alpha,\ \forall (i,j) \in B$
    With probability $\alpha/(\alpha+\beta)$, $y_{i,j} := y_{i,j} - \beta,\ \forall (i,j) \in A$, and $y_{i,j} := y_{i,j} + \beta,\ \forall (i,j) \in B$
    for all $(i,j) \in \tilde{E}$ do
        if $y_{i,j} \in \{0,1\}$ then
            $\tilde{E} := \tilde{E} \setminus \{(i,j)\}$
        end if
    end for
end while
$X_{i,j} = y_{i,j},\ \forall (i,j) \in E$
return $X_{i,j}$
```

We apply this framework to the fractional solution of LP relaxation of (FIP), that can be obtained in polynomial time. The fractional solution corresponds to a bipartite graph $G = (V_1, V_2, E)$, with $V_1 = \{(i,k) : \beta_{i,k} = 1\}$, $V_2 = \{m \in \mathcal{M}_k,\ \forall k\}$, and $E = \{((i,k),m) : x_{i,k,m} > 0, \forall\ i,k,m\}$, i.e., $G$ is a collection of stars with centres $(i,k)$. If $x$ is the fractional solution of the linear relaxation of (FIP), let

$$D^{\rm LP} := \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{f_{i,k,m}}{FT_{i,k}} x_{i,k,m},$$

$$R_i^{\rm LP} := \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{s_{i,k,m}}{R_i T_{i,k}} x_{i,k,m},\ \forall i.$$

The application of part (i) of Theorem 3.1 of [14] (proved by [15]) on the (randomly) rounded values $X_{i,k,m}$ implies the following Chernoff-type bounds:

**Theorem 3.1:**

$$\Pr\left[\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{f_{i,k,m}}{FT_{i,k}} X_{i,k,m} \geq (1+\varepsilon)\right] \leq \frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}, \quad (15)$$

$$\Pr\left[\exists i : \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{s_{i,k,m}}{R_i T_{i,k}} X_{i,k,m} \geq (1+\varepsilon)\right] \leq \frac{Ne^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}. \quad (16)$$

*Proof:* Note that $0 \leq \frac{f_{i,k,m}}{FT_{i,k}}, \frac{s_{i,k,m}}{R_i T_{i,k}} \leq 1,\ \forall i,k,m$, since, otherwise, the optimal LP solution sets $x_{i,k,m} = 0$. Also, assuming the solution of (FIP) does not result in infeasibility, i.e., $D^{\rm LP} \leq 1$ and $R_i^{\rm LP} \leq 1\ \forall i$ for some $\hat{\tau}$, the properties of dependent rounding imply that $E[\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{f_{i,k,m}}{FT_{i,k}} X_{i,k,m}] = D^{\rm LP} \leq 1$ and $E[\sum_{k=1}^{K} \sum_{m=1}^{M_k} \frac{s_{i,k,m}}{R_i T_{i,k}} X_{i,k,m}] = R_i^{\rm LP} \leq 1,\ \forall i$. Hence, the conditions of Theorem 3.1 of [14] are satisfied, and we have the inequalities of the theorem statement. ■

As in Theorem 3.2 of [14], setting $\varepsilon = O(\frac{\ln N}{\ln \ln N})$ guarantees a violation of at most a factor $O(\frac{\ln N}{\ln \ln N})$ of the constraints of (FIP) with probability larger than a constant. By repeating the experiment a constant number of times, the probability of success can be made bigger than any constant, e.g., 0.75. As a result, with high probability the available resources need to be augmented only by small amounts in order to render our solutions feasible for the system. Also, note that although the upper bound of $O(\frac{\ln N}{\ln \ln N})$ for resource augmentation is guaranteed with high probability, the randomized nature of our algorithm suggests that we can have better results if we run it a few times and keep the best solution. We explore this heuristic further in Section IV.

## IV. Simulation Results

In this section, we present simulation results to demonstrate the performance of the proposed solution. We consider a single-BS network with the BS located at the center and the PSs uniformly distributed in the circular coverage area of radius 150 m. Path loss is used for the link gains and the path loss exponent is 3. Default parameters used in the simulation are summarized in Table I, where $U[a,b]$ denotes the uniform distribution between $a$ and $b$. Similar parameter values were also used in [2], [4] and [16].

TABLE I
Default Parameters

| Parameter | Value |
|---|---|
| $T_{i,k}$ | $U[1,5]$ s |
| $s_{i,k,m}$ | $U[30,150]$ M bits |
| $f_{i,k,m}$ | $U[50,500]$ M CPU cycles |
| $\Phi_{i,k,m}$ | $U[70,100]$ % |
| $F$ | 2 GHz |
| $P_i^{\rm T}$ | 0.1 W |
| $w$ | 6 MHz |
| $\sigma^2$ | -90 dBm |

In the first set of simulation, there are 6 PSs, the number of features ($K$) for each PS is varied during the simulation, and each feature has 6 models whose accuracy levels are randomly generated based on distribution given in Table I. The simulations are performed on a server with Ubuntu 18.04.6 LTS, Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz and 196 GB memory. Table II shows the running time of the proposed approximation solution and the optimum solution. As $K$ increases, the size of the problem increases, and the running time of both solutions increases. For the optimum solution, the running time increases exponentially and quickly becomes prohibitively long, e.g., more than 24 days when $K = 10$. Although the running time of our proposed solution is longer

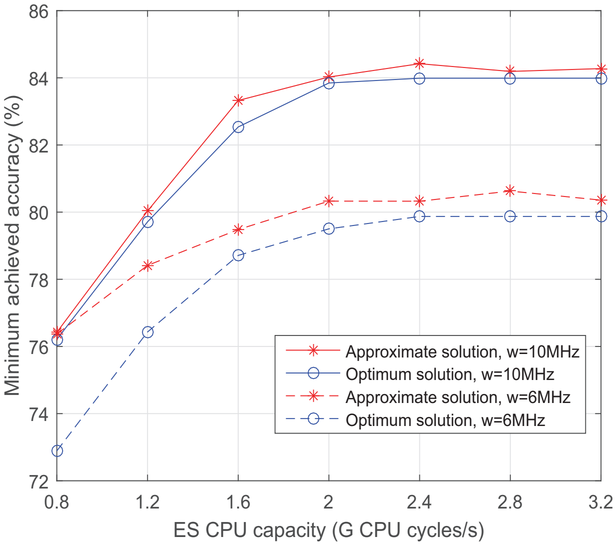

Fig. 2. Minimum achieved accuracy versus ES computation capacity

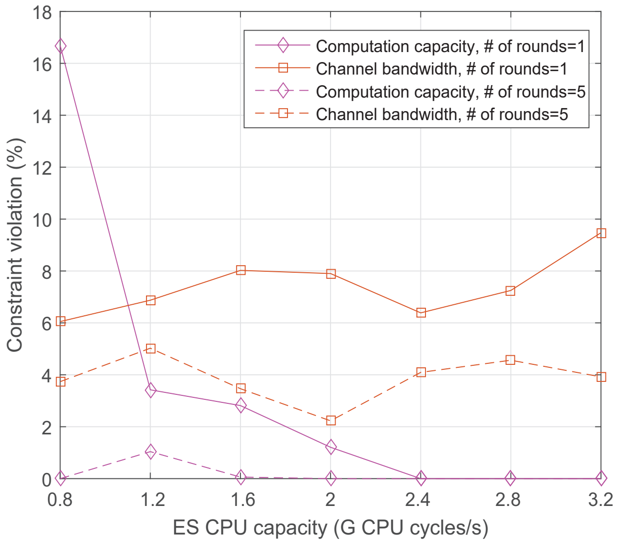

Fig. 3. Constraint violation versus ES computation capacity

than the optimum solution (and still short) when $K$ is small (e.g., 2 and 4), it increases much slower and is about 0.0032% of the running time of the optimum solution when $K = 10$. This demonstrates that our proposed solution is highly efficient when the system size is large.

TABLE II
COMPARISON OF RUNNING TIME

| Number of features ($K$) | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| Proposed solution (s) | 4.07 | 5.36 | 27.04 | 34.03 | 68.62 |
| Optimum solution (s) | 0.06 | 0.09 | 32.5 | 16236 | 2120443 |

In the second set of simulation, we compare the minimum achieved accuracy using the proposed solution and the optimum. Due to the long running time of the optimum solution, the comparison can only be performed for small size systems. In the simulation, there are 3 PSs, each PS requires 5 features, and each feature can be implemented by using one of the 4 models (with different accuracy). The simulation results are averaged over 100 independent experiments, each of which is for one set of randomly generated PS locations and feature and model parameters. Fig. 2 shows the minimum achieved accuracy of features versus the ES CPU capacity $F$. First of all, the minimum achieved feature accuracy increases with the ES CPU capacity in general. The increase is more significant when $F$ is relatively small and gradually becomes saturated as the wireless transmission rates eventually become the performance bottleneck. The figure shows that, when $F$ is sufficiently large, the minimum achieved accurate is much higher when the channel bandwidth $w$ is 10 MHz than $w$ is 6 MHz. It is also seen that the minimum achieved accuracy using the proposed approximate solution is higher than the optimum one, and the approximate solution achieves this at the price of violating the constraints, which is shown below.

Fig. 3 shows the constraint violation resulted from dependent rounding in the proposed approximation solution as the ES CPU capacity changes. Both the wireless bandwidth constraints (i.e., (10)) and the computation constraint (i.e., (11)) are considered. For the computing constraint, the violation is calculated as percentage changes to the right-hand value after running the algorithm. For the bandwidth constraints, the percentage change to the right-hand value is calculated for each of the $N$ constraints and then the maximum is taken. The violation is collected after running the dependent rounding for one round and 5 rounds. The figure shows that as $F$ increases, the computation constraint violation decreases in general. When $F$ is sufficiently large, there is no violation in the computation constraint. Note that the results of the dependent rounding are random, and therefore, the change of the constraint violation is not monotonic. The changes in bandwidth constraint violation are mainly due to the random effect of the dependent rounding. However, for both the bandwidth and computation constraints, running additional rounds of the dependent rounding helps reduce the constraint violation.

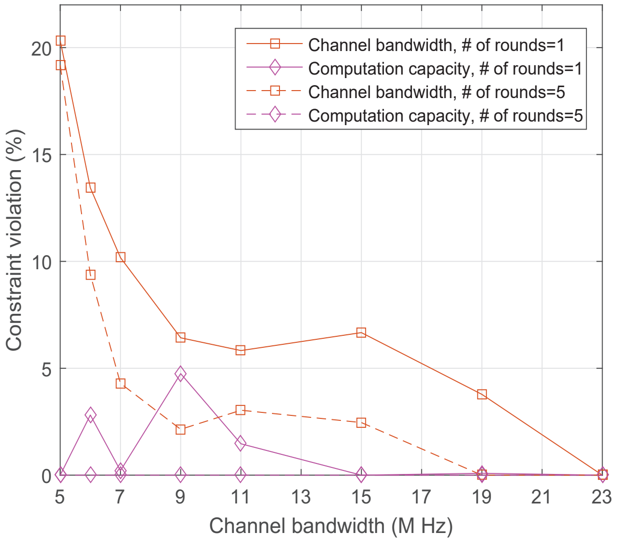

Fig. 4. Constraint violation versus wireless channel bandwidth

Fig. 4 shows the constraint violation as the wireless channel bandwidth changes. The observations are similar as in Fig. 3. As the channel bandwidth increases, the bandwidth constraint violation decreases in general with some fluctuations due to random nature of the dependent rounding. Increasing the channel bandwidth does not obviously affect the computation constraint violation. However, increasing the number of rounds of running the dependent rounding helps reduce the violation in both types of constraints.

Finally, by comparing the simulation results and the theorem 3.1, we found that the proposed solution achieves close-to-optimum feature accuracy, its violation of constraints is significantly below the theoretical bound, and it is highly efficient in solving large size problems.

## V. Conclusions

DTs provide features that represent the real behavior of their PS using models that yield differing levels of system accuracy. In this paper, we have considered the DT model selection problem in wireless networks where the DTs of multiple PSs are hosted at an ES. The objective is to maximize the minimum achieved accuracy among the requested features by making appropriate model selections subject to wireless channel and ES resource availability. The problem was formulated as an NP-complete integer program and the paper then used relaxation and dependent rounding to address the problem. A polynomial time approximation algorithm was introduced that finds approximate problem solutions that provide an asymptotic bound on constraint violations. A variety of simulation results were presented that demonstrate the excellent performance of the proposed solution.

## References

[1] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, G. Karakostas, H. Wu, and X. Shen, “Digital twins from a networking perspective,” *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 525–23 544, 2022.

[2] 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.

[3] B. A. Talkhestani and M. Weyrich, “Digital twin of manufacturing systems: a case study on increasing the efficiency of reconfiguration,” *at-Automatisierungstechnik*, vol. 68, no. 6, pp. 435–444, 2020.

[4] Y. Dai and Y. Zhang, “Adaptive digital twin for vehicular edge computing and networks,” *Journal of Communications and Information Networks*, vol. 7, no. 1, pp. 48–59, 2022.

[5] B. R. Barricelli, E. Casiraghi, and D. Fogli, “A survey on digital twin: Definitions, characteristics, applications, and design implications,” *IEEE access*, vol. 7, pp. 167 653–167 671, 2019.

[6] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, “Holistic network virtualization and pervasive network intelligence for 6g,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–30, 2021.

[7] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, and G. Karakostas, “Digital twin placement for minimum application request delay with data age targets,” *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[8] F. Tang, X. Chen, T. K. Rodrigues, M. Zhao, and N. Kato, “Survey on digital twin edge networks (diten) toward 6g,” *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1360–1381, 2022.

[9] V. Stegmaier, G. Ghasemi, N. Jazdi, and M. Weyrich, “An approach enabling accuracy-as-a-service for resistance-based sensors using intelligent digital twins,” *Procedia CIRP*, vol. 107, pp. 833–838, 2022, leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022.

[10] G. Yiping, L. Xinyu, and L. Gao, “A deep lifelong learning method for digital twin-driven defect recognition with novel classes,” *Journal of Computing and Information Science in Engineering*, vol. 21, no. 3, 02 2021.

[11] R. Lu, C. Rausch, M. Bolpagni, I. Brilakis, and C. T. Haas, “Geometric accuracy of digital twins for structural health monitoring,” in *Structural Integrity and Failure*. IntechOpen, 2020.

[12] G. M. Paldino, F. De Caro, J. De Stefani, A. Vaccaro, D. Villacci, and G. Bontempi, “A digital twin approach for improving estimation accuracy in dynamic thermal rating of transmission lines,” *Energies*, vol. 15, no. 6, 2022.

[13] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, “Machine intelligence at the edge with learning centric power allocation,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7293–7308, 2020.

[14] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, “Dependent rounding and its applications to approximation algorithms,” *Journal of the ACM (JACM)*, vol. 53, no. 3, pp. 324–360, 2006.

[15] A. Panconesi and A. Srinivasan, “Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds,” *SIAM J. Comput.*, vol. 26, no. 2, pp. 350–368, 1997.

[16] Z. Zhou, Z. Jia, H. Liao, W. Lu, S. Mumtaz, M. Guizani, and M. Tariq, “Secure and latency-aware digital twin assisted resource scheduling for 5g edge computing-empowered distribution grids,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4933–4943, 2022.