Differentially Private Federated Multi-Task Learning Framework for Enhancing Human-to-Virtual Connectivity in Human Digital Twin

Samuel D. Okegbile, *Member, IEEE,* Jun Cai, *Senior Member, IEEE,* Hao Zheng, Jiayuan Chen, and Changyan Yi, *Member, IEEE*

***Abstract*—Ensuring reliable update and evolution of a virtual twin in human digital twin (HDT) systems depends on any connectivity scheme implemented between such a virtual twin and its physical counterpart. The adopted connectivity scheme must consider HDT-specific requirements including privacy, security, accuracy and the overall connectivity cost. This paper presents a new, secure, privacy-preserving and efficient human-to-virtual twin connectivity scheme for HDT by integrating three key techniques: differential privacy, federated multi-task learning and blockchain. Specifically, we adopt federated multi-task learning, a personalized learning method capable of providing higher accuracy, to capture the impact of heterogeneous environments. Next, we propose a new validation process based on the quality of trained models during the federated multi-task learning process to guarantee accurate and authorized model evolution in the virtual environment. The proposed framework accelerates the learning process without sacrificing accuracy, privacy and communication costs which, we believe, are non-negotiable requirements of HDT networks. Finally, we compare the proposed connectivity scheme with related solutions and show that the proposed scheme can enhance security, privacy and accuracy while reducing the overall connectivity cost.**

***Index Terms*—Blockchain, digital twin, federated multi-task learning, privacy, virtual twin.**

I. Introduction

DIGITAL twin (DT) continues to attract wide attention, especially in communications networks [1], healthcare [2], [3], and manufacturing [4], [5] because of its ability to improve the current systems by leveraging newly emerged algorithms including machine learning, optimization and artificial intelligence as well as communication technologies, such as edge intelligence, security and privacy-preservation [6]–[8]. When adopted, DT is capable of allowing a digital representation of real-world equipment, process, objects or environment by creating corresponding virtual twins (VTs) in the virtual space [9]. Specifically, in human-centric systems such as medical cyber-physical systems, the human digital twin (HDT) facilitates the co-evolution of both humans and VTs [2], and thus can transform the current healthcare systems, environmental monitoring systems, and other applications by integrating human behaviour and activities.

Generally, an HDT encompasses any human being, otherwise called a physical twin (PT), located in the physical environment, its digital replica, i.e., its corresponding VT located in the virtual environment, and a mapping between these two environments through reliable data links [2]. This mapping is expected to ensure continuous and reliable interactions between human-virtual twin pairs considering the dynamic nature of the physical environment. While HDT can significantly improve the quality of services and experiences in the physical environment, the diverse requirements [1]–[3], [9] in terms of latency, privacy, security, reliability, data rate, and other user-defined performance metrics make it very complicated and challenging to achieve a reliable mapping or connectivity between physical and virtual environments [10]. Furthermore, there are currently insufficient possibilities for the physical-virtual environment synchronizations to establish closed loops, a lack of high-fidelity and quantification models as well as difficulties in obtaining accurate predictions of complex physical systems [4]. All these make HDT suffer in many aspects in terms of accuracy, security, privacy, synchronization and connectivity.

To address security and privacy issues, blockchain and federated learning (FL) techniques have been widely adopted in recent DT solutions [3], [11], [12] owing to their ability to support the training of machine learning models in a decentralized manner. However, the adoption of blockchain often relies on high latency and energy-intensive consensus algorithms [13], [14], which cannot meet the specific requirements of HDT. In addition, although FL continues to receive wide considerations when providing solutions to address privacy concerns, privacy leakage has been reported to remain a potential issue. For instance, when clients synchronize their learned model parameters with the global server, an attacker may infer some data properties or recover the raw data based on this shared information [15], [16]. Moreover, conventional FL suffers from many other challenges such as statistical heterogeneity, high computation and communication costs, and limited fault tolerance. Thus, an effective solution for HDT integrating blockchain and FL has to be proposed, which can accelerate the learning process without sacrificing accuracy, privacy and communication costs.

S. D. Okegbile and J. Cai are with the Network Intelligence and Innovation Lab (NI^2L), Department of Electrical and Computer Engineering, Concordia University, Montreal QC H3G 1M8, Canada (Emails: samuel.okegbile@concordia.ca; jun.cai@concordia.ca).

H. Zheng, J. Chen and C. Yi are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 211106, China (Emails: zhenghao@nuaa.edu.cn; jiayuan.chen@nuaa.edu.cn; changyan.yi@nuaa.edu.cn).

This research is supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, Concordia University PERFORM Research Chair program, and National Natural Science Foundation of China (NSFC) under Grant 62002164.

A. Contributions

Although connectivity problems in HDT [3], DT-empowered 6G networks [11] and DT edge networks [12] have earlier been studied, they may suffer from high latency that characterizes traditional blockchain-enabled systems, as well as data leakages and statistical heterogeneity issues that characterize the conventional FL schemes. In addition, synchronization accuracy is an important metric in HDT for sufficient performance evaluations. However, such a metric is difficult to obtain and has been neglected in previous works. Moreover, the effects of scheduled offloading rates with queueing constraints for a status update and privacy budget on both the synchronization cost and the overall HDT system performance have never been studied.

To address all these issues, in this paper, we proposed a new HDT framework, which integrates differential privacy, federated multi-task learning (FML) [15] and quality training-based validation process (a newly proposed lightweight blockchain-enabled consensus method) in the presence of heterogeneous environments. The proposed framework finds the balance between synchronization accuracy and connectivity cost without compromising security and privacy which, we believe, are non-negotiable requirements of HDT networks. To the best of our knowledge, such a framework that captures statistical heterogeneity, synchronization accuracy, synchronization cost and other related performance has never been presented. The contributions of this paper are thus summarized as follows:

- We propose a secure differentially private federated multi-task learning (DPFML) framework for HDT by integrating the DPFML and a computational-efficient blockchain-enabled validation process to provide a secure, privacy-preserving and more accurate human-to-virtual twin connectivity solution.
- We analytically study the connectivity cost of the proposed DPFML-enabled connectivity scheme to investigate the influence of some important system parameters, including privacy budget, on synchronization cost as well as the long-term average connectivity cost, overall time cost and energy cost. To capture the validation cost inherent in blockchain, we propose a new consensus mechanism based on the quality of trained models during FL, called proof of model quality (PoMQ).
- Following this, we formulate a connectivity problem in the proposed DPFML-based framework as a Markov decision process (MDP) to minimize the connectivity cost. To solve the MDP, we propose a deep reinforcement learning (DRL) algorithm using the deep deterministic-policy gradient (DDPG) approach.
- Finally, we compare the proposed framework with existing frameworks through simulation and demonstrate the ability of the proposed solution to offer synchronization accuracy and reduced connectivity cost without compromising security and privacy.

B. Organization

The remainder of this paper is structured as follows. Section II reviews related works, while Section III introduces the details of the proposed system model. In Section IV, we present the analysis of the connectivity cost – time, privacy and energy. The formulated MDP-based optimization problems and DRL-based solutions are presented in Section V. Section VI discusses the simulation results and the performance of the proposed scheme, while Section VII concludes this paper.

II. Related Work

In this section, we discuss some of the earlier presented related frameworks and solutions. For clarity, we categorized these existing works into three: the human digital twin, FL in DT applications and differentially private solutions for FL.

A. Human digital twin

HDT is an emerging technology that is recently attracting more consideration in many domains, including medical, sports and manufacturing [17]. It relies on the concept of DT to create a virtual replica of human, body organs or habits in the virtual environment [2]. Note that DT provides enhanced system performance by combining both system models and analyses with real-time measurements for any individual system. It facilitates model evolutions over the lifecycle of any physical system while supporting the derivation of solutions with the ability to aid real-time optimizations of such a physical system.

Similar to DT, HDT possesses the potential to revolutionize the practice of human system integration by adopting real-time sensing and feedback to tightly couple measurements of human performance, behaviour, as well as the influence of environment throughout a product's life cycle, on human modelling to improve system design and performance [17]. Unlike the VT in the conventional DT, however, human VTs often possess distinct underlying variability among each other as well as dependence between humans and products in the physical environment. Since each human VT (referred to as VT henceforth for simplicity) evolves with data from its counterpart PT, located in the physical environment, its design and implementation are known to be very difficult.

A DT solution was presented in [18] for elderly healthcare services, while [19] discussed a deep neural-based model for capturing bi-directional context relationships when predicting lung cancer. A software-based HDT was similarly presented in [20] for tracking fitness-related measurements describing an athlete's behaviour on consecutive days, while the work in [21] presented a cardio twin architecture for the detection of ischemic heart disease. In [22], a DT ecosystem for health and well-being was presented. It is worth mentioning that, although the majority of [18]–[22] discussed the importance of connectivity in HDT, none of these works delved into investigating and modelling this connectivity scheme. In [3], an edge-assisted connectivity framework for HDT was presented. While the presented framework is interesting, statistical heterogeneity, synchronization accuracy and other related performance issues such as data leakages were not considered. This paper addresses these limitations by proposing a connectivity scheme that considered all important HDT-specific requirements including statistical heterogeneity, synchronization accuracy and data leakages.

B. FL in DT applications

One of the underlying limitations of DT and HDT applications is privacy. Since every physical object must continuously share its data with its corresponding VT, privacy becomes an important concern. To address this, the work in [3] adopted FL to preserve data privacy in HDT networks. Similarly, FL was adopted in [10], [11] to learn a behavioural model from user data towards achieving low latency in DT-empowered 6G networks and in [12] to achieve efficient communication in DT edge networks since offloading all running data to the VT can incur a large amount of communication resource, cost, and time while leading to privacy issues. In a similar work, the authors in [23] carried out an optimization of FL using the DRL method to construct the DT-empowered industrial IoT model. The work proposed an asynchronous FL scheme that is capable of addressing the discrete effects caused by heterogeneous industrial IoT devices. A cooperative FL was developed in [24] to facilitate DT construction in resource-limited smart devices. An iterative double auction-based joint cooperative FL with an update verification scheme was designed.

In [25], an FL-based anomaly detection model was proposed using DT by utilizing edge cloudlets when running anomaly detection models locally, while the work in [26] presented a blockchain-enabled adaptive asynchronous FL paradigm for privacy-preserving and decentralized DT networks. However, these works do not consider possible data leakages in conventional FL algorithms which is important in HDT. Similarly, connectivity problems were only studied in HDT [3], and DT-enabled wireless networks [10]–[12], where the influence of statistical heterogeneity and synchronization accuracy on the connectivity cost was not considered. Sharing gradients as in federated averaging can lead to data leakage. As a result, cryptographic-based approaches have also been explored in some FL-based research [27] although such approaches are computationally inefficient in large-scale machine-learning models. Recently, differential privacy (DP) is being explored in FL to reduce the possibility of information leakages by hiding the contribution of each client during training thereby ensuring privacy guarantees.

C. DP solutions for FL

DP solutions are efficient techniques that can provide privacy guarantees in machine learning [28]. It is therefore unsurprising that such approaches have recently been attracting a lot of interest in FL-based research. By adding artificial noise to the learned model parameters or datasets, DP can protect nodes' privacy with limited computation. It, however, tends to reduce the overall accuracy as the privacy protection level increases. Thus, a trade-off exists between accuracy and privacy. In [16], a differentially private FL framework was adopted to prevent privacy leakages during data sharing to model the contribution, computation, communication, and privacy costs of each participant. Also, security and privacy concerns in the standard FL continue to hinder its wide adoption in urban applications, hence a differentially private asynchronous FL scheme was proposed in [29] for resource sharing in vehicular networks by integrating DP and FL techniques.

Table I
COMMON NOTATIONS USED

Notation	Definition
$\mathcal{L}_i$	Learning task of local aggregator or client i
M	Total number of related local aggregators
V	Total number of validators
D_i; $D_i(t)$	Data size of client i; Data size of client i at time t
o_i	Status update offloading scheduling rate of client i
ϱ_i	Service rate of client i
σ	Gaussian noise variance
N_R	Number of communication rounds
G_S	Local gradient sensitivity
g^t	Gradient
η	Learning rate
N_E	Number of local epochs
$\epsilon; \delta$	Privacy budget; Additive term
θ^μ	Weights parameter of the actor network
$\theta^\mathcal{O}$	Weights parameter of the critic network
c_r	CPUs required to execute a sample of training data
c_i	CPU frequency of any local aggregator i
κ_i	Coefficient dependency on the chip architecture
γ	Discount factor
θ_{off}	Offloading threshold
θ_{pvy}	Privacy cost threshold
θ_{cmp}	Computation cost threshold

While a differentially private FL framework can ensure privacy guarantees when adopted, the inherent issues of statistical heterogeneity are the main concern. Hence, any differentially private FL-based framework can suffer from accuracy degradation when used in a network with non-independent and identically distributed (non-iid) data. To address these issues associated with differentially private FL, especially in HDT, where data from PTs are arbitrarily heterogeneous with fundamental statistical heterogeneity issues, while also noting the presence of many external conditions that can influence the behaviour of each PT such as environment and genetic information, we propose DPFML framework following the privacy-aware multi-task learning approach, first discussed in [15]. This ensures a federated optimization of heterogeneous tasks while protecting the local model gradient information using DP.

A DPFML-enhanced framework can enable federated optimization of heterogeneous client tasks while protecting the local model gradient information through DP. Such a technique can prevent privacy leakage, and ensure accuracy, privacy and reduced communication costs when properly adopted. Since in HDT networks, many external factors may influence the performance of the entire system, while some unique structures may exist among different people [30], it is desirable to simultaneously achieve learning models for multiple related tasks through an efficient multi-task learning framework. The proposed DPFML-enabled HDT connectivity solution can learn customized context-aware policies from multiple users and environments in a privacy-preserving manner. The definitions of some common notations used throughout this paper are summarized in Table I.

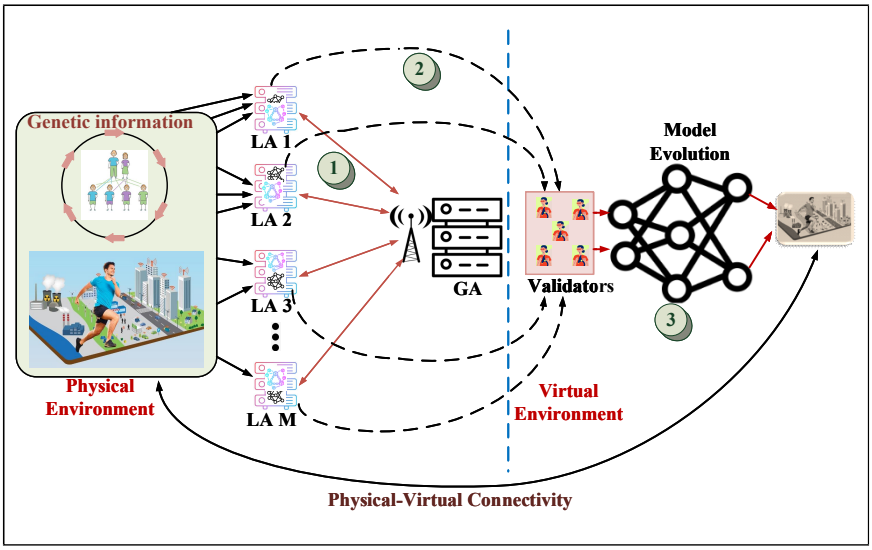

Figure 1. Overview of the system model.

III. System Model

We consider a DPFML-enabled HDT system, where the physical and the virtual environments are connected through a blockchain and FML-enabled connectivity scheme as shown in Fig. 1. Data are generated by each PT (based on its update scheduling rate), located in the physical environment, to maintain a reasonable synchronization with its counterpart VT, located in the virtual environment. The physical environment is equipped with multiple local aggregators (LAs) corresponding to different entities of the physical environment, such as the typical PT, genetic information, environmental factors, etc. Each LA is connected with various sensing devices for data capturing and produces both shared and task-specific parameters, from its locally trained model and based on its local data. The shared parameter is then offloaded to the global aggregator (GA) for aggregation following the standard FL during the first phase. At the beginning of the second phase, each LA will forward its locally trained task-specific model to the validators, located in the virtual environment, to evaluate its quality, before the VT model updating and evolution in the final phase. The main components of the system model are summarized as follows.

- **Environment:** The physical environment depicts the real-world space, where every PT interacts with other entities within its environment and maintains a dependent relationship with such related entities. A virtual replica of each PT is maintained in the virtual environment.
- **Local Aggregator:** Also called client, each LA represents a distinct entity in the physical environment. Each of these clients consists of several sensing devices to regularly collect data from the physical environment. The collected data by each LA are aggregated, subject to its update scheduling rate o_i, and are used to support learning during the FML. In the proposed HDT framework, each LA is responsible for updating the corresponding aspect of its associated VT, using the task-specific models, after the training quality requirements have been satisfied.
- **Global aggregator:** The GA is a central server that facilitates aggregation of shared parameters during FML, and also provides the training requirement thresholds to validators during validation. At every communication round, the global model (which contains the aggregated shared parameters) is used to locally train each task-specific model at each LA.
- **Validators:** Validators are essential components of the blockchain system that ensures the reliability of every update from each client before triggering the model evolution process in the virtual environment. Without this, the system cannot guarantee accurate and authorized model evolution of any corresponding VT. The blockchain also keeps the records of previous model evolution activities to ensure traceability. Since each LA is responsible for updating its associated VT, it becomes imperative to have an independent validation process.
- **Model evolution:** Model evolution is an essential process in HDT. It involves the process of updating any typical VT based on the current state of its counterpart PT and relies on a timely, reliable, secure and privacy-preserving PT-VT connectivity. At any time, the VT in the proposed DPFML-enabled HDT system is updated in the virtual environment using the task-specific parameters received from its counterpart physical pair. Since this model evolution process ensures that each VT is a true replica of its paired PT, it is an important part of any HDT framework and its construction is beyond the scope of this current paper.

Assume that there are M LAs, each with a learning task $\mathcal{L}_i, \forall i \in \{1, 2, ..., M\}$ as in Fig. 1. Each LA contains a training dataset $\mathcal{D}_i = \cup_{j=1}^{D_i}\{(x_{i,j}, y_{i,j})\}$, generated from different sensing devices as in [3], where D_i is the data size, $x_{i,j}$ is the data of size j collected by client i and $y_{i,j}$ is the label of $x_{i,j}$. In practice, the general aim of the HDT framework is to maintain the VT of each physical entity (e.g., body organ, habit, eating pattern, etc.) in the virtual environment while capturing the dependence of such a physical entity on other

related components observed in the physical environment. As a result, LAs in the HDT framework are related to each other (e.g., a dependence exists between any PT, and its environment and genetic information) such that locally trained models from different LAs share some common underlying representation. To capture this in the system modelling, the hard parameter sharing technique [31] can be incorporated into the standard FL to obtain FML. This hard parameter-sharing technique has been earlier adopted in neural networks. With this, the shared feature representation can be learned through joint optimization of different tasks via the parameter sharing in the proposed DPFML framework.

While model sharing among various LAs can reduce the effect of insufficient data and improve the overall system accuracy, it also comes with a risk of privacy leakages and statistical heterogeneity. The adoption of the DP technique ensures that privacy leakages are prevented through applications of Gaussian noises at each LAs. This, however, comes at the expense of accuracy. Thus, we incorporate a double-layer multi-task learning technique [15], where shared parameters are used to improve the training performance at each LA, while task-specific parameters are used to achieve personalization.

A. Federated multi-task learning model

To properly capture statistical heterogeneity, each LA learns a domain classifier to capture transferable feature representations (i.e., shared parameters) across tasks through the hard parameter-sharing technique. The transferable feature representations are offloaded to the GA every communication round for aggregation. After aggregation, the GA forwards the global model to all related LAs. Each LA uses this global model to improve the training of its task-specific models. The aim is to reinforce each task by taking advantage of the interconnections among related tasks while considering both the inter-task relevance and the inter-task difference. Each LA through its domain classifier classifies every feature as either a sharable or task-specific feature by minimizing the distribution difference [32] between its shared and global parameters. This distribution difference can be obtained following the maximum mean discrepancy [33] as

$$M_{MD}(i,GA) = \Big|\Big|\frac{1}{n_i}\sum_{k=1}^{n_i}\Phi(x_i^k) - \frac{1}{n_{GA}}\sum_{l=1}^{n_{GA}}\Phi(x_{M+1}^l)\Big|\Big|_{\mathcal{H}}^2, \tag{1}$$

where n_i and n_{GA} are the number of samples drawn from any LA i and GA, respectively and $\Phi(.)$ is the nonlinear mapping into reproducing kernel Hilbert space $\mathcal{H}$. Also, $x_i^k, \forall k = \{1, ..., n_i\}$ and $x_{M+1}^l, \forall l = \{1, ..., n_{GA}\}$ are the feature vectors of LA i and GA, respectively, while $||.||$ represents the norm. For each feature x_i^k in D_i, the chances that x_i^k is a sharable feature can be obtained through the instance weight

$$\mathcal{Q}(x_i^k) = \frac{P(x_i^k \in \mathcal{D}_{\text{shared}})}{1 - P(x_i^k \in \mathcal{D}_{\text{shared}})}, \tag{2}$$

where $\mathcal{D}_{\text{shared}}$ is a vector containing the sharable parameters. If we assume a logistic regression-based domain classifier, then the mapping $\Phi(i, \text{GA})$ maps the global feature vector to the local feature vector of LA i, while $\Phi(\text{GA}, i)$ maps the local feature vector of LA i to the global feature vector, such that

$$\min_{\Phi(i,\text{GA})}\Bigg\{\Big|\Big|\text{diag}(\mathcal{Q}_i)\Big(x_i^k - x_{M+1}^l\Phi(i,\text{GA})\Big)\Big|\Big|_F^2 + \lambda\sum_{l=1}^{n_{GA}}\Big|\Big|\Phi_l(i,\text{GA})\Big|\Big|_F^2\Bigg\}, \tag{3}$$

$$\min_{\Phi(\text{GA},i)}\Bigg\{\Big|\Big|x_i^k - x_{M+1}^l\Phi(\text{GA},i)\Big|\Big|_F^2 + \lambda\sum_{k=1}^{n_i}\Big|\Big|\Phi_k(\text{GA},i)\Big|\Big|_F^2\Bigg\}, \tag{4}$$

where diag(.) transforms an input vector to a diagonal matrix, $||.||_F$ is the Frobenius norm and $\lambda > 0$ is the regularization parameter. At any round, features with less difference to the global parameter are classified as shared parameters and features with more difference are classified as task-specific parameters. Let the top layers of the double-layer multi-task learning framework, as presented in Fig. 2, capture the task-specific features $\mathcal{T}_f$, while the lower ones capture the shared features $\mathcal{S}_f$. Then, we can define $\mathcal{T}_f$ and $\mathcal{S}_f$ as

$$\begin{aligned}\mathcal{T}_f &= \{\omega_1, \omega_2, ..., \omega_M\},\\ \mathcal{S}_f &= \{\omega_{M+1,1}, \omega_{M+1,2}, ..., \omega_{M+1,M}\},\end{aligned} \tag{5}$$

where ω_i and $\omega_{M+1,i}$ represent the task-specific parameters and the shared parameters of any client i, respectively. As shown in Fig. 2, the local optimization is carried out by each client subject to its local objective function f_i, given as

$$f_i(\omega_i, \omega_{M+1}) = \frac{1}{D_i}\sum_{j=1}^{D_i}\ell(x_{i,j}, y_{i,j}, \omega_i, \omega_{M+1}), \tag{6}$$

where ℓ is the loss function and ω_{M+1} is the global model. During the optimization process, ω_{M+1} is shared among all clients, while the GA carries out the aggregation of local models at each communication round and then distributes it back to all clients after updating the global model. With this, each task aims to learn a function f_i, while the global objective is obtained as

$$f(\omega_{M+1}) = \sum_{i=1}^{M}\Omega_i f_i(\omega_{M+1,i}, \omega_{M+1}), \tag{7}$$

where $\Omega_i = \Big[\frac{D_i}{\sum_{i=1}^{M} D_i}\Big]$ captures the weight of the local model for each LA $i \in \{1, 2, ..., M\}$.

Let all participants (i.e., LAs and the GA) be honest but curious. As a result, it is possible that any participant may maliciously attempt to infer some vital information when interacting with other participants. To prevent such a potential privacy violation scenario, randomized noises such as Gaussian noise is introduced at the gradient level following the DP approach. With that, the gradient contribution of each LAs during communication and aggregation can be protected.

The proposed DPFML framework is summarized in Algorithm 1, where each of the M LAs begins the FL process when at least any LA $i \in \{1, 2, ..., M\}$ has a status to update following its update scheduling rate o_i, subject to the approval of the GA. As soon as the GA approves the commencement of any status update, each related LA receives the global model

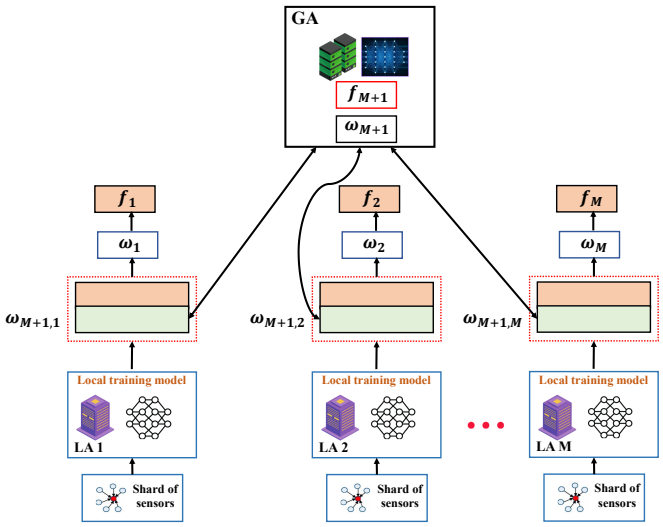

Figure 2. The method of federated multi-task learning.

ω_{M+1}^t from the GA and updates its locally trained model by computing the gradient for both shared and task-specific layers. Note that the Gaussian noise $n_{M+1,i} \sim \mathcal{N}(0, \sigma^2 G_S^2)$ is only added to the shared models since only the lower layers are shared to capture transferable feature representation. Thus, we reduce the effect of noise on the overall accuracy of the model. The convergence of Algorithm 1 is demonstrated through the simulation presented in Section VI.

DPFML framework ensures privacy guarantees without sacrificing much of the trained model accuracy. Such a framework aims to prevent any attacker or eavesdropper [34] from extracting sensitive information during model exchanges among LAs and the GA. It relies on a common standard method when measuring privacy risk, called $(\epsilon, \delta) - \text{DP}$, where $\epsilon > 0$ is the privacy budget and $\delta \in (0, 1)$ is the additive term. Hence, the possibility that $\epsilon -$ differential privacy is violated is captured by probability δ. A lower ϵ suggests that the clients have a lower risk of privacy leakage. With DPFML, each client adds artificial Gaussian noise during local training at every round such that $(\epsilon, \delta) -$DP of its local datasets is always guaranteed. To ensure accuracy and convergence, the GA determines the privacy budget $\epsilon_{\min} \leq \epsilon \leq \epsilon_{\max}$. When $\epsilon < \epsilon_{\min}$, the added noise is too large thus the training cannot converge. Similarly, when $\epsilon > \epsilon_{\max}$, the added noise is too small and privacy cannot be protected. The GA may also specify the training data size N_i, and the corresponding reward R_i to facilitate fairness during the validation process.

Note that any global model at each round includes aggregated uploaded noisy local models. At every communication round $0 \leq t \leq N_R - 1$, the global model is trained using the local models received from all LAs to find a global model parameter ω_{M+1} that minimizes global loss while ensuring privacy guarantee. With this

$$\begin{aligned} &\underset{\omega_{M+1} \in \{\omega_{M+1}^t, \forall t < N_R\}}{\arg\min} \quad f(\omega_{M+1}) \\ &\text{s.t.} \quad Pr(\omega_{M+1,i} \in \mathbb{R}_d) \leq \exp(\epsilon) Pr(\omega'_{M+1,i} \in \mathbb{R}_d) + \delta, \end{aligned} \tag{8}$$

where $Pr(\omega_{M+1,i} \in \mathbb{R}_d) \leq Pr(\omega'_{M+1,i} \in \mathbb{R}_d)$ captures the $(\epsilon, \delta) -$ DP guarantees for $\omega_{M+1,i}$. The parameters $\omega_{M+1,i}, \omega'_{M+1,i} \in \mathbb{R}_d$ are generally called the neighbouring model parameters [28], such that the sensitivity function can be defined as

$$\Delta_f = \max_{\omega_{M+1,i}, \omega'_{M+1,i}} \left|\left| f(\omega_{M+1,i}) - f(\omega'_{M+1,i}) \right|\right|. \tag{9}$$

This Δ_f depicts the maximum value by which any local model function f changes if noise is added to $\omega_{M+1,i}$ and it captures the similarity between any neighbouring model parameters $\omega_{M+1,i}$ and $\omega'_{M+1,i}$.

Algorithm 1 DPFML scheme.

INPUT: M clients, Gaussian noise variance σ, number of rounds N_R, local gradient sensitivity G_S, number of local epochs N_E, learning rate η.
OUTPUT: Global model ω_{M+1}^{t+1}, task-specific model $\omega_i, \forall i$.
INITIALIZE: ω_i, for all $i \in \{1, 2, ..., M\}$
For each round $t = 0$ to $N_R - 1$ **do**
  The GA broadcasts $\omega_{M+1}^t$to all LAs
  For LA $i \in \{1, 2, ..., M\}$ **do**
    Synchronize local parameters $\omega_{M+1,i}^t \leftarrow \omega_{M+1}^t$
    For local epoch $L_E = 0$ *to* N_E **do**
      Compute gradients for shared layers as $g_{M+1,i}^t = \partial_{\omega_{M+1}} f_i(\omega^t)$
      Perturb gradients for shared layers as $\tilde{g}_{M+1,i}^t = g_{M+1,i}^t + n_{M+1,i}, \forall n_{M+1,i} \sim \mathcal{N}(0, \sigma^2 G_S^2)$
      Update parameters for shared layers as $\omega_{M+1,i}^{t+1} = \omega_{M+1,i}^t - \eta(\tilde{g}_{M+1,i}^t)$
      Compute gradients for task-specific layers as $g_i^t = \partial_{\omega_i} f_i(\omega^t)$
      Update parameters for task-specific layers as $\omega_i^{t+1} = \omega_i^t - \eta g_i^t$
    End For
  End For
  Each LA offloads model weight $\omega_{M+1,i}^{t+1}$ to the GA
  The GA aggregates the received weights as $\omega_{M+1}^{t+1} = \frac{1}{M} \sum_{i=1}^{M} \omega_{M+1,i}^{t+1}$
End For
RETURN ω_{M+1}^{t+1}, ω_i, $\forall i$

B. Blockchain-enabled validation model

HDT relies on accurate modelling of the VT to guarantee performance. However, some LAs may attempt to manipulate the system either by providing an untrained model or misleading data for model updating in the virtual environment. Similarly, a multidimensional information asymmetry [16] may also exist among participants, where selfish participants manipulate their costs to receive more rewards from the system. To ensure that the final model from each LA is accurate and has not been modified through malicious activities, we propose a PoMQ consensus mechanism which offers the validation process in terms of the quality of the trained model during FML rather than solving computational-inefficient hashing puzzles, as in the proof of work. With the PoMQ, the validation process can be carried out before

model updating and evolution in the virtual environment. The PoMQ protocol is made up of V validators located in the virtual environment. Multiple validators are necessary to eliminate the possibility of malicious validation. These validators are responsible for validating the training quality of each model. After validation, each validator broadcasts its validation decision to other validators to reach a consensus. A virtual model is updated using any learned model only if the majority of V consent.

The proposed PoMQ evaluates each learned model based on computation cost, communication cost and privacy cost. With this, validators can investigate whether the expenditure by each LA corresponds to the expected costs. The VT is, therefore, only updated when the consensus decision signifies conformity to the requirements. More details on the analysis of the validation process are provided in the next section.

IV. Performance Analysis

In this section, we first analyze the proposed framework by investigating the physical-virtual environment connectivity cost from time, privacy and energy perspectives. We then obtain analysis for synchronization accuracy before presenting the resulting optimization problem in Section V.

A. DPFML model

To ensure accurate VT model updating and evolution, data captured from its environment and other individuals with similar behaviour are also used through multi-task learning to improve the performance while boosting the effective sample size for each LA. At every round, each LA carries out local training following (6) and subsequently performs shared and task-specific features classification. The shared model is then offloaded to the GA for aggregation. It is worth noting that the cost of achieving a secure and privacy-preserving connectivity scheme may undermine its benefits. Thus, we aim to minimize the synchronization cost while ensuring accuracy and reliability. Compared to the local training time, the features extractions time at each LA is negligible. Thus, we focus on the local training time to estimate the required time, privacy and energy cost when updating any typical VT.

Let c_r and c_i represent the number of CPUs required to train one sample of training data and the CPU cycle frequency of any LA i, respectively. The time cost for local model training over N_R rounds can be derived as

$$T_{\text{LA}}^{\text{cmp}} = \max_{i\in[M]} \left(\sum_{t=0}^{N_R-1} \frac{c_r D_i(t)}{c_i} \right), \tag{10}$$

and the corresponding energy cost can be derived as

$$E_{\text{LA}}^{\text{cmp}} = \sum_{i=1}^{M} \sum_{t=0}^{N_R-1} \kappa_i c_0 D_i(t) c_i^2, \tag{11}$$

where κ_i is the capacitance coefficient depending on the chip architecture and c_0 captures the number of floating operations required to train or compute each sample for N_E. Similarly, for global aggregation, the total time and energy costs can be respectively approximated as

$$\begin{aligned} T_{\text{agg}}^{\text{cmp}} &= \sum_{t=0}^{N_R-1} \Big(\frac{c_{agg} \sum_{i=1}^{M} |\omega_{M+1,i}(t)|}{c_{GA}} \Big), \\ E_{\text{agg}}^{\text{cmp}} &= \sum_{t=0}^{N_R-1} \kappa_{GA} c_0 \Big(\sum_{i=1}^{M} |\omega_{M+1,i}(t)| \Big) c_{GA}^2, \end{aligned} \tag{12}$$

where c_{agg} is the number of CPUs required to aggregate one unit of data and c_{GA} is the CPU cycle frequency of the GA.

During each round, each LA perturbs its shared model parameters through the DP technique. If we assumed Renyi DP [16], then the injected noise σ_i to achieve $(\epsilon, \delta)-$DP guarantee for each LA after N_R-round training can be computed as

$$\sigma_i = \Big(\frac{14\alpha\eta^2 N_E N_R}{|L_B| D_i \{\epsilon - \log\left(\frac{1}{\delta}\right)/(\alpha-1)\}} \Big)^{\frac{1}{2}}, \tag{13}$$

where $|L_B|$ is the size of the local mini-batch and $\alpha = ([2\log(1/\delta)]/\epsilon) + 1$ given that

$$\alpha - 1 \leq \Big[\frac{2\alpha_i^2}{3} \Big] \log \Big(\frac{1}{\alpha |L_B| / D_i (1+\delta^2)} \Big). \tag{14}$$

From (13), it is clear that σ_i depends on ϵ, while ϵ is inversely proportional to privacy protection. The strictest privacy is achieved when $\epsilon = 0$. Under this, it is impossible to differentiate any two locally trained models. Let the privacy cost be defined as the economical loss due to the potential privacy leakage which can be formulated as

$$\psi = \frac{1}{\epsilon_{\max}} \sum_{i=1}^{M} \sum_{t=0}^{N_R-1} \epsilon_i v_i |\omega_{M+1,i}(t)|, \tag{15}$$

where v_i represents this economical loss per unit shared model from privacy leakage.

B. Communication and validation model

During FML, each LA offloads its shared model to the GA at the end of every round for aggregation. The total offloading time cost then can be calculated as

$$T_{\text{LA}}^{\text{off}} = \sum_{t=0}^{N_R-1} \max_{i\in[M]} \Big(\frac{|\omega_{M+1,i}(t)|}{r_i(t)} \Big), \tag{16}$$

where r_i is the data rate between any LA i and the GA, and can be obtained during round t as

$$r_i(t) = B_0 \log_2 \big(1 + \frac{h_{i,\text{GA}}(t) P_i(t)}{N}\big). \tag{17}$$

The parameter N is the thermal noise signal power, B_0 is the bandwidth, P_i depicts the offloading power of any LA i and $h_{i,\text{GA}}$ is the channel gain between any LA i and the GA. From the GA to all LAs, the transmission time is assumed to be negligible due to ample resources available in the GA [12]. Hence, the focus is only on offloading. Similarly, the total energy cost during offloading of shared model parameters is given as

$$E_{\text{LA}}^{\text{off}} = \sum_{i=1}^{M} \sum_{t=0}^{N_R-1} t_i(t) \frac{N}{h_{i,\text{GA}}(t) P_i(t)} \Big[\exp\Big(\frac{r_i(t)}{B_0} - 1 \Big) \Big], \tag{18}$$

where t_i is the time allocated to each LA.

After N_R-round training, any learned model $f_i(\omega_i)$ will be further validated by the group of validators following the PoMQ consensus protocol to determine whether the received model satisfies the pre-defined requirements. Given that the total communication cost, total training cost and total privacy cost incurred during the training of such a learned model, as received from the GA, are given as $C_{\text{tot}}^{\text{off}}$, $C_{\text{tot}}^{\text{cmp}}$ and $C_{\text{tot}}^{\text{pvy}}$ respectively. Then each validator $m_j \in [V]$ rates such a learned model $f_i(\omega_i)$ as

$$R_{m_j}(i) = \begin{cases} 1 & \text{if } C_{\text{tot}}^{\text{off}}(i) \geq \theta_{\text{off}}, C_{\text{tot}}^{\text{cmp}}(i) \geq \theta_{\text{cmp}}, \text{and} \\ & C_{\text{tot}}^{\text{pvy}}(i) \geq \theta_{\text{pvy}} \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where the thresholds of offloading, computation and privacy costs are respectively given as θ_{off}, θ_{cmp} and θ_{pvy}. A learned model will be ultimately applied to the corresponding VT if

$$2 \sum_{m_j=1}^{V} R_{m_j}(i) > V. \tag{20}$$

To estimate the validation cost, we consider the transmission of the final learned model to the virtual environment for the validation, the computation process at each validator and the decision exchange among validators after validation. The total validation time cost of any model $f_i(\omega_i)$ is given as

$$T_{\text{val}}(i) = \frac{|\omega_{f,i}|}{r_i} + \max_{m_j \in [V]} \Big\{ \frac{c_v |R_{m_j}(i)|}{c_{m_i}} + \frac{|R_{m_j}(i)|}{r_v} \Big\}, \tag{21}$$

where $|\omega_{f,i}|$ is the size of the final model $f_i(\omega_i)$ after N_R communication rounds, c_v is the number of CPUs required to validate one sample of the final learned model, c_{m_i} is the validation capacity, $|R_{m_j}(i)|$ is the size of the decision message and r_v is the data rate among validators, assumed to be constant owing to the pre-defined communication subchannels among validators. Similarly, the energy cost can be obtained as

$$\begin{aligned} E_{\text{val}} = \frac{N}{P_i h_{i,\text{val}}} \Big[\exp\Big(\frac{r_i}{B_0} - 1 \Big) \Big] + \\ \sum_{m_j=1}^{V} \Big\{ \frac{N}{P_j h_{j,k}} \Big[\exp\Big(\frac{r_v}{B_0} - 1 \Big) \Big] + \kappa_v c_0 |R_{m_j}(i)| c_{m_i}^2 \Big\}. \end{aligned} \tag{22}$$

C. Connectivity cost

Connectivity cost captures the cost of updating the VT model every time an update is scheduled. It captures the cost of maintaining secure and reliable synchronizations between any PT and its counterpart VT. Since M LAs participate in a model update, the connectivity cost includes the cost of FML at each participating node, the validation cost, the privacy cost as well as communication cost. Given any typical physical-virtual twin pair, the overall time cost to complete a single status update can be obtained as

$$\begin{aligned} C_{\text{time}} = \sum_{t=0}^{N_R-1} \Big\{ \Big(\frac{c_{agg} \sum_{i=1}^{M} |\omega_{M+1,i}(t)|}{c_{GA}} \Big) + \\ \max_{i \in [M]} \Big(\frac{|\omega_{M+1,i}|}{r_i} + \frac{c_r D_i(t)}{c_i} \Big) \Big\} + \Big(\frac{|\omega_{f,i}|}{r_i} + \\ \max_{m_j \in [V]} \Big\{ \frac{c_v |R_{m_j}(i)|}{c_{m_i}} + \frac{|R_{m_j}(i)|}{r_v} \Big\} \Big). \end{aligned} \tag{23}$$

Similarly, the overall energy cost to ensure synchronization of any single model update is obtained as

$$\begin{aligned} C_{\text{ene}} = \sum_{t=0}^{N_R-1} \Big\{ \kappa_{GA} c_0 \Big(\sum_{i=1}^{M} |\omega_{M+1,i}(t)| c_{GA}^2 \Big) + \\ \sum_{i=1}^{M} \Big(\kappa_i c_0 D_i(t) c_i^2 + t_i(t) \frac{N}{h_{i,\text{GA}}(t) P_i(t)} \Big[\exp\Big(\frac{r_i(t)}{B_0} - 1 \Big) \Big] \Big) \Big\} \\ + \frac{N}{P_i h_{i,\text{val}}} \Big[\exp\Big(\frac{r_i}{B_0} - 1 \Big) \Big] + \\ \sum_{m_j=1}^{V} \Big\{ \frac{N}{P_j h_{j,k}} \Big[\exp\Big(\frac{r_v}{B_0} - 1 \Big) \Big] + \kappa_v c_0 |R_{m_j}(i)| c_{m_i}^2 \Big\}. \end{aligned} \tag{24}$$

Generally, the connectivity cost depends on the scheduling rate o_i. Given that $\mathcal{U}$ is the number of status updates that have been scheduled over a known time interval, its probability mass function can be expressed as

$$P(\mathcal{U}|o_i) = \frac{\exp(-o_i) o_i^{\mathcal{U}}}{\mathcal{U}!}, \forall \mathcal{U} \in \mathbb{N}. \tag{25}$$

The long-term average connectivity cost per any arbitrary VT model update can therefore be obtained as

$$\begin{aligned} C_{\text{conn}} = \lim_{\mathcal{U} \to \infty} \frac{1}{\mathcal{U}} \sum_{u=1}^{\mathcal{U}} \Big(\frac{1}{\epsilon_{\max}} \sum_{i=1}^{M} \sum_{t=0}^{N_R-1} \epsilon_i v_i |\omega_{M+1,i}^u(t)| \\ + C_{\text{time}}^u + C_{\text{ene}}^u \Big). \end{aligned} \tag{26}$$

Note that to update any arbitrary VT, the privacy, time and energy costs should be considered. Hence, (26) is obtained by averaging (15), (23) and (24) over $\mathcal{U}$.

D. Synchronization accuracy

It is essential to investigate the synchronization accuracy. Unfortunately, such a metric is very difficult to define since it captures the degree of similarity between any PT and its counterpart VT at any time. If we consider that any final model after the validation process could capture the true corresponding update in the physical environment, then we may evaluate synchronization accuracy as a function of the synchronization time [35] and the FML loss. That is, lower synchronization time and FML loss depict higher synchronization accuracy. For this purpose, we define a new term called synchronization gap, which is the time since the last status update was generated in the physical environment. With this, the synchronization gap, at any known FML loss, is inversely proportional to accuracy.

To obtain the synchronization gap, we first consider each generated status update u to intuitively pass through the FML learning and validation process following the first-come-first-serve (FCFS) approach. We further assume that this status update generation is the same as the status update arrival while the arrival and the service time (i.e., the time from arrival till model evolution or updating) follow random processes. The service time of any status depicts the time at which the VT is successfully updated using such a status update. Define that the

inter-arrival time of status updates $X_{(u)}$ from any tagged LA is an independent and identically distributed (i.i.d.) exponential random variable with $E[X] = \frac{1}{o_i}$ [36], [37]. Let ϖ_k represents the times at which status is received at the tagged VT for its update, then at any time t, the index and the timestamp of the most recently received status are respectively given as

$$\begin{aligned} \mathcal{J}(t) &= \max\{k|\varpi_k \leq t\}, \\ u(t) &= \varpi_{\mathcal{J}(t)}. \end{aligned} \tag{27}$$

The synchronization gap at time t can then be expressed as

$$S_{\text{gap}}(t) = t - u(t), \forall t \geq 0. \tag{28}$$

In the absence of newly received model updates, the synchronization gap increases linearly with time and is reduced to a smaller value when a new model update is received. For any update u, the processing time of any generated status update can be obtained as

$$P_{\text{time}}^{(u)} = W_{\text{time}}^{(u)} + C_{\text{time}}^{(u)}, \tag{29}$$

where $W_{\text{time}}^{(u)}$ is the waiting time for any status update u and $C_{\text{time}}^{(u)}$ is the service time of u following (23). Obviously, $W_{\text{time}}^{(u)} = 0$ if any status update u is generated when the VT is already updated with previously generated status $u-1$. However, if u is generated when $u-1$ is still in the system (i.e., $u-1$ has not triggered an update of the VT), $W_{\text{time}}^{(u)} = (P_{\text{time}}^{(u-1)} - X_{(u)})^+$ captures the average waiting time. For explanation purpose, we assumed that o_i follows the Poisson point process, while the service times of the rate ϱ_i are similarly i.i.d. exponentials with average $C_{\text{time}}^{(u)}$, then the average synchronization gap is provided in Proposition 1.

Proposition 1: The average synchronization gap when the status arrival rate is Poisson while the service times are i.i.d. exponentials can be calculated as

$$\begin{aligned} S_{\text{gap}}^{\text{fc}} = C_{\text{time}}^{(u)} \Big\{ & \frac{o_i C_{\text{time}}^{(u)}(1 - o_i C_{\text{time}}^{(u)}) + (1 - o_i C_{\text{time}}^{(u)})}{o_i C_{\text{time}}^{(u)}(1 - o_i C_{\text{time}}^{(u)})} \\ & \frac{(o_i C_{\text{time}}^{(u)})^3}{o_i C_{\text{time}}^{(u)}(1 - o_i C_{\text{time}}^{(u)})} \Big\}. \end{aligned} \tag{30}$$

Proof: The proof follows from the probability density function of the system with Poisson process arrival rate and exponential service times given as

$$\mathcal{D}_{\text{fc}}(t) = \varrho_i \Big(1 - \frac{o_i}{\varrho_i}\Big) \exp\Big(- \varrho_i \Big[1 - \frac{o_i}{\varrho_i}\Big] t\Big). \quad \blacksquare \tag{31}$$

Note that (30) follows from a single-server FCFS queue with infinite buffer size. Such a scheme, though simple, may not be suitable in the HDT system, where any VT is expected to reflect the latest status of its counterpart PT. Instead of processing an old status, we can discard it and simply process the latest status. Following this and to obtain a low synchronization gap, we introduced a non-preemptive single-server last-come-first-serve (LCFS) queue with a buffer of size 2 and queue displacement policy, where the system at any time can only consist of a maximum of two status updates – one currently under processing and the other on the queue. On the arrival of another status update, the newly arrived status displaces the status waiting in the queue as shown in Fig. 3.

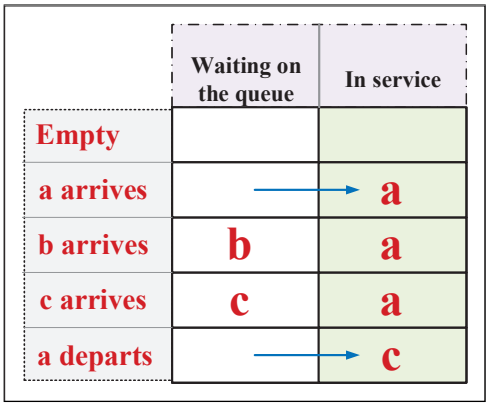

Figure 3. A 2 buffer size non-preemptive single-server LCFS queue with queue displacement policy.

Let the arrival and service follow Poisson and general distributions, respectively. We applied the classical embedding technique under the assumption that the queue system is stationary and sampled at certain epochs, such that service completion (i.e., the successful updates of a VT) has a Markovian property. With that, the synchronization gap can have the same distribution for all t while the distribution of any $C_{\text{time}}^{(u)}$ of rate ϱ_i is given as g with

$$g(x) = \mathbb{P}(C_{\text{time}}^{(u)} \leq x). \tag{32}$$

Proposition 2: If C_{time} is i.i.d. exponential in steady-state, the density of S_{gap} at any time t can be obtained under a non-preemptive single-server LCFS queue system with buffer size 2 and queue displacement as

$$\begin{aligned} \mathcal{D}_{S_{\text{gap}}}^{\text{lcfs}}(t) = & \frac{o_i[(\rho+2)(\rho-1)]t - \rho^2 + \rho + 3}{\rho^3 - 1} \exp(-\varrho_i t) \\ & + \frac{o_i(\rho+1)t + \rho(\rho+3) + 3}{\rho(\rho+1) + 1} \exp(-\varrho_i[\rho+1]t) \\ & - \frac{\rho}{\rho - 1} \exp(-o_i t), \end{aligned} \tag{33}$$

where $\rho = \frac{o_i}{\varrho_i}$ depicts the traffic intensity.

Proof: The proof can be obtained by inverting the Laplace transform of $S_{\text{gap}}(t)$ at any time $t = 0$, given as [38], [39]

$$\begin{aligned} \mathbb{E}[e^{-sS_{\text{gap}}(0)}] = & \frac{\varrho_i}{\varrho_i + s} \Big(\frac{2\varrho_i o_i + \varrho_i^2 + o_i^2 + \varrho_i s}{(\varrho_i + o_i)(o_i + \varrho_i + s)} \Big) \\ & \Big\{ \Big(\frac{o_i}{s + o_i} \Big) \Big(\frac{\varrho_i^2}{o_i^2 + o_i\varrho_i + \varrho_i^2} \Big) \Big(\frac{o_i + \varrho_i}{\varrho_i + o_i + s} \Big) + \Big(\frac{\varrho_i}{\varrho_i + s} \Big) \\ & \frac{o_i^2 + \varrho_i o_i}{o_i^2 + \varrho_i o_i + \varrho_i^2} \Big\}, \end{aligned} \tag{34}$$

where for $o_i = \varrho_i = 1$,

$$\mathcal{D}_{S_{\text{gap}}}^{\text{lcfs}}(t) = \frac{1}{3}[(2t+7)\exp(-2t) + (6t-7)\exp(-t)], \tag{35}$$

and $\lim_{o_i \to \infty} \mathcal{D}_{S_{\text{gap}}}^{\text{lcfs}}(t) = t\exp(-t)$. The synchronization gap in such a case can be expressed as

$$S_{\text{gap}}^{\text{lcfs}} = \frac{o_i^4(2o_i + 7\varrho_i) + o_i^2\varrho_i^2(8o_i + 7\varrho_i) + \varrho_i^4(4o_i + \varrho_i)}{o_i\varrho_i(o_i + \varrho_i)^2(o_i^2 + \varrho_i o_i + \varrho_i^2)}. \tag{36}$$

V. Problem formulation and optimization

Asides from the synchronization gap, it is also important to reduce the occurrence of loss during the FML. This is expected to improve the synchronization accuracy at any time by simultaneously minimizing the two functions following

$$\min_{o_i,V,M} \Big(w_1 S_{\text{gap}} + \frac{w_2}{D_i}\sum_{j=1}^{D_i} \ell(x_{i,j}, y_{i,j}, \omega_i, \omega_{M+1})\Big), \tag{37}$$

$$\text{s.t.} \quad o_i \in [0,1], \tag{37a}$$

$$w_1 + w_2 = 1, \forall w_2 \gg w_1, \tag{37b}$$

$$\sum_{i\in[M]} i \leq M, \tag{37c}$$

$$\sum_{j}^{V} m_j \leq V. \tag{37d}$$

The parameters ω_1 and ω_2 are weight factors that ensure an effective combination of the synchronization time and the FML loss. The constraint in (37a) ensures the values for o_i while (37b) ensures that the two weights sum up to 1. In addition, (37c) and (37d) ensure that the total number of participating LAs and validators do not exceed M and V, respectively. A simultaneous minimization of the synchronization gap and loss in (37) is not straightforward since the synchronization gap depends on not only the overall time cost, but also the FML. We can thus leverage the overall time cost to further reduce the synchronization gap in (36).

From (37), we know that an increase in the number of rounds N_R can improve the training accuracy at the expense of connectivity cost, while an increase in the privacy protection level (i.e., a lower ϵ) improves privacy, but decreases the accuracy. Moreover, if we increase the computation overhead (by increasing the added noise), we can have higher privacy but lower connectivity cost. That means a trade-off exists between accuracy and connectivity cost, accuracy and privacy, and connectivity cost and privacy. In this section, we attempt to minimize the connectivity cost of the proposed DPFML-enabled HDT framework without compromising accuracy and privacy.

A. Problem formulation

We aim to find the balance between synchronization accuracy, privacy cost and connectivity cost. The objective function can be formulated as

$$O = \Theta_1(C_{\text{time}} + C_{\text{ene}} + f_i(\omega_i, \omega_{M+1})) - (1-\Theta_1)\Theta_2\psi_{\text{cost}}, \tag{38}$$

where Θ_1 $(0 < \Theta_1 < 1)$ represents the weight factor necessary to combine two objective functions and Θ_2 captures the mapping factor to ensure that the two objectives functions are at the same scale. Note that to maximize the synchronization accuracy, we simply minimize the FML loss in (38), and take (36) as a baseline synchronization gap which can be further reduced through the time cost C_{time}. That is, given (36), the gap can be further reduced by minimizing C_{time}. Thus, the optimization problem is obtained as

$$\begin{aligned}\min_{o_i,V,M,\epsilon,N_R} [&\Theta_1(C_{\text{time}} + C_{\text{ene}} + f_i(\omega_i, \omega_{M+1})) - \\ &(1-\Theta_1)\Theta_2\psi_{\text{cost}}],\end{aligned} \tag{39}$$

$$\text{s.t.} \quad o_i \in [0,1], \tag{39a}$$

$$\sum_{i\in[M]} i \leq M, \tag{39b}$$

$$\sum_{j}^{V} m_j \leq V, \tag{39c}$$

$$\epsilon_{\min} \leq \epsilon \leq \epsilon_{\max}, \tag{39d}$$

$$c_i^{\min} \leq c_i \leq c_i^{\max}, \forall i \in [M], \tag{39e}$$

$$c_{m_j}^{\min} \leq c_{m_j} \leq c_{m_j}^{\max}, \forall m_j \in [V], \tag{39f}$$

where $c_i^{\min}$ and $c_i^{\max}$ represent the minimum and maximum computation capacity of each LA respectively, while $c_{m_j}^{\min}$ and $c_{m_j}^{\max}$ are the minimum and maximum validation capacities of each validator respectively. The constraint (39d) ensures that the privacy budget is within the acceptable range while (39e) ensures that the CPU frequency of any LA i is within the acceptable range. Similarly, (39f) ensures that the validation capacity of any validator m_j is also within the acceptable range. Obviously, problem (39) is a nonconvex optimization problem and thus its solution is difficult to obtain in close form. In what follows, we transform the original problem into an MDP problem and provide solutions using the DRL algorithm.

B. MDP problem and solution

We defined the tuple $(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}, \mathcal{R}^{(t)})$, where $\mathcal{S}^{(t)}$, $\mathcal{A}^{(t)}$ and $\mathcal{R}^{(t)}$ are the state space, action space and reward, respectively, at each round t. In the proposed framework, the agent is the typical LA that aims to update its counterpart VT by collaborating with related LAs during the FML. Thus, the state space includes the achievable data rate r, the computation capacity c, the validation capacity c_m, the learned parameter $\omega(t)$ and the global loss value function $f(\omega_{M+1})$. The state space is given as

$$\mathcal{S}^{(t)} = \{r(t), c(t), c_m(t), f(\omega_{M+1}(t))\}. \tag{40}$$

Similarly, the action space includes the scheduling rate o, the number of validators V, the number of LAs M, the privacy budget ϵ and the number of rounds N_R. The action space is given as

$$\mathcal{A}^{(t)} = \{o(t), V(t), M(t), \epsilon(t), N_R(t)\}, \tag{41}$$

while the reward function is obtained from (38) as

$$\mathcal{R}^{(t)} = -O(t), \tag{42}$$

if all constraints in (39) are satisfied and zero if otherwise. Given that $\gamma \in [0,1]$ is the discount factor, each agent aims to maximize the cumulative reward

$$\mathbb{E}\Big[\sum_{t=0}^{N_R-1} \gamma \mathcal{R}(\mathcal{S}^{(t)}, \mathcal{A}^{(t)})\Big]. \tag{43}$$

C. DRL solution using DDPG

To solve the MDP problem, we adopted the DDPG algorithm [40] owing to its ability to achieve improved performance, for continuous action space, compared to other

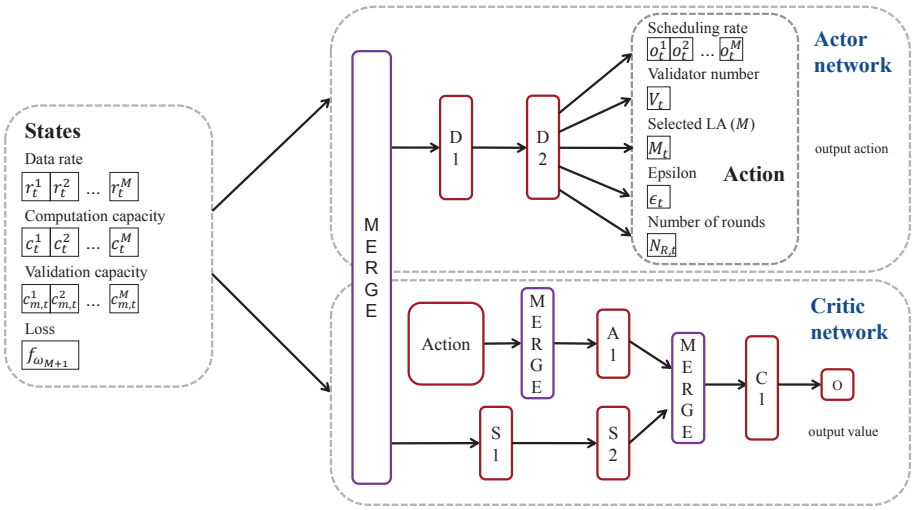

Figure 4. Structure of the DDPG.

algorithms. While Q-learning algorithms and their variants are suitable for low-dimensional, discrete state and action spaces, most of these algorithms do not easily converge to optimal behaviour. In addition, such models are susceptible to the curse of dimensionality when too many decision variables are involved [41]. The Deep Q network algorithm can handle problems in high-dimensional continuous state space, albeit under a discrete action space. DDPG algorithm generally relies on deep neural networks to create two approximation functions of the actor-critic algorithm. The actor network is described as a policy function $\mu(\mathcal{S}|\theta^{\mu})$ with parameter θ^{μ} while the critic network is described as an action-value function $\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})$ with parameter $\theta^{\mathcal{O}}$.

Define the Bellman equation as

$$\mathcal{O}^{\mu}(\mathcal{S}^{(t)},\mathcal{A}^{(t)}) = \mathbb{E}[\mathcal{R}^{(t)} + \gamma\mathcal{O}^{\mu}(\mathcal{S}^{(t+1)},\mu(\mathcal{S}^{(t+1)}))], \quad (44)$$

where the loss of $\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})$ can be obtained, if μ' represents the target actor network, as

$$L(\theta^{\mathcal{O}}) = \mathbb{E}_{\mu'}[(\mathcal{O}^{\mu}(\mathcal{S}^{(t)},\mathcal{A}^{(t)}|\theta^{\mathcal{O}}) - \{\mathcal{R}^{(t)} + \gamma\mathcal{O}(\mathcal{S}^{(t+1)},\mu(\mathcal{S}^{(t+1)})|\theta^{\mathcal{O}})\})^2]. \quad (45)$$

To update the policy function $\mu(\mathcal{S}|\theta^{\mu})$, the chain rule is applied to the Bellman equation from the start distribution J [40] based on the actor parameters

$$\bigtriangledown_{\theta^{\mu}} J \approx \mathbb{E}_{\mu'}[\bigtriangledown_{\theta^{\mu}}\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})|_{\mathcal{S}=\mathcal{S}^{(t)},\ \mathcal{A}=\mu(\mathcal{S}^{(t)}|\theta^{\mu})}]$$
$$= \mathbb{E}_{\mu'}[\bigtriangledown_{\mathcal{A}}\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})|_{\mathcal{S}=\mathcal{S}^{(t)},\ \mathcal{A}=\mu(\mathcal{S}^{(t)})}\ \bigtriangledown_{\theta^{\mu}}\mu(\mathcal{S}|\theta^{\mu})|_{\mathcal{S}=\mathcal{S}^{(t)}}]. \quad (46)$$

To minimize loss, the parameter $\bigtriangledown_{\theta^{\mathcal{O}}} L$ is estimated using the algorithmic differentiation technique [42], such that the action-value function $\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})$ parameters are updated using the gradient descent as

$$\theta^{\mathcal{O}} \leftarrow \theta^{\mathcal{O}} - \eta^{\text{critic}}_{\text{rate}} \bigtriangledown_{\theta^{\mathcal{O}}} L, \quad (47)$$

given that $\eta^{\text{critic}}_{\text{rate}}$ is the learning rate of the critic network. In addition, the algorithmic differentiation technique is also adopted to obtain gradients $\bigtriangledown_{\mathcal{A}}\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})|_{\mathcal{S}=\mathcal{S}^{(t)},\ \mathcal{A}=\mu(\mathcal{S}^{(t)})}$ and $\bigtriangledown_{\theta^{\mu}}\mu(\mathcal{S}|\theta^{\mu})|_{\mathcal{S}=\mathcal{S}^{(t)}}$, such that

$$\bigtriangledown_{\theta^{\mu}} J \approx \frac{1}{N_{\text{batch}}}\sum_{j}[\bigtriangledown_{\mathcal{A}}\mathcal{O}(\mathcal{S},\mathcal{A}|\theta^{\mathcal{O}})|_{\mathcal{S}=\mathcal{S}^{(t)},\ \mathcal{A}=\mu(\mathcal{S}^{(t)})} \bigtriangledown_{\theta^{\mu}}\mu(\mathcal{S}|\theta^{\mu})|_{\mathcal{S}=\mathcal{S}^{(t)}}], \quad (48)$$

where N_{batch} is the mini-batch size selected by the agent during the learning process. Similarly,

$$\theta^{\mu} \leftarrow \theta^{\mu} - \eta^{\text{actor}}_{\text{rate}} \bigtriangledown_{\theta^{\mathcal{O}}} J, \quad (49)$$

where $\eta^{\text{actor}}_{\text{rate}}$ is the learning rate of the actor network. With these, the target critic and actor networks can be respectively obtained using the update rate $\tau^{\text{rate}} \ll 1$ as

$$\theta^{\mathcal{O}'} \leftarrow \tau^{\text{rate}}\theta^{\mathcal{O}} + (1-\tau^{\text{rate}})\theta^{\mathcal{O}'},$$
$$\theta^{\mu'} \leftarrow \tau^{\text{rate}}\theta^{\mu} + (1-\tau^{\text{rate}})\theta^{\mu'}. \quad (50)$$

This work used multiple DDPG agents to simulate the proposed framework. Its general structure is shown in Fig. 4, where $D1$, $D2$, $A1$, $S1$, $S2$ and $C1$ capture the hidden layers of such a framework. The details of the simulation and results are provided in the Section VI.

VI. Numerical Results

We first implemented the proposed DPFML framework by adopting TensorFlow and LEAF library [43] – an open-source library that provides a modular benchmarking framework for federated settings with applications including federated learning, multi-task learning, meta-learning, and on-device learning. Since HDT data are expected to be non-iid, we used the CelebA datasets available in the LEAF library. To minimize the required connectivity cost in the proposed framework, we incorporated the DDPG algorithm into the DPFML framework. We compared the proposed framework with three other baseline frameworks: the conventional federated averaging (FeDAvg), the FeDAvg with DP (DPFedAvg) and the DPFML with standard validation method (vDPFML). The vDPFML is simply an implementation of the proposed framework with

Table II
PARAMETERS USED FOR SIMULATION

Parameter	Value
M, V	$[1, 22]$
D_i	$[1, 5]$ MB
c_r, c_v	1 GHz
c_i	$[2, 3.5]$ GHz
N_R	$[10, 50]$
κ_i, κ_{GA}; κ_v	10^{-27}
c_{agg}	1 GHz
c_{GA}	20 GHz
v_i	$[0.01, 0.012, 0.014, 0.016]$
$\epsilon_{\max}$	50
B_0	1 MHz
$\lvert R_{m_j} \rvert$	10 MB
c_{m_i}	$[5, 12.5]$ GHz
r_v	2
$\eta_{\text{rate}}^{\text{critic}}$	0.001
$\eta_{\text{rate}}^{\text{actor}}$	0.0005
N_{batch}	10
γ	0.99
τ^{rate}	0.05

a standard blockchain consensus algorithm instead of the proposed PoMQ consensus mechanism.

We carried out several experiments and simulations to demonstrate the performance of the proposed framework. In the simulations, we used a computer system with 10 CPU cores. The CPU is Intel(R) Core(TM) $i9 - 10900X$ with 3.70 GHz. Generally, to complete any arbitrary VT model update, we carried out the local training, local parameters synchronization and communication rounds in the regions $[0, N_E]$, $[1, M]$ and $[1, N_R]$, respectively. Therefore, the complexity of Algorithm 1 is around $\mathcal{O}(N_E M N_R)$. Except otherwise stated, the parameters used for simulations are presented in Table II. These parameters were selected based on similar works [3], [11], [12]. We set the sizes of the hidden layers in Fig. 4 as follows: $D1 = 128$, $D2 = 32$, $S1 = 64$, $S2 = 32$, $A1 = 32$, and $C1 = 16$.

In Fig. 5, we demonstrate the ability of the proposed DPFML scheme to reach convergence faster than other schemes. The highest loss is observed in the DPFedAvg scheme due to the incorporation of the privacy budget (as in the proposed DPFML and vDPFML schemes), through the addition of the Gaussian noise. The DPFML scheme achieved the least loss as N_R increases which justifies the suitability of the proposed scheme to perform efficiently in the presence of non-iid data. Although the FeDAvg has been demonstrated to perform well when iid datasets are used, we claim that data is HDT systems are expected to be non-iid. Thus, such a framework may not be suitable.

A similar result is observed in Fig. 6, where the loss is obtained as ϵ increases. At a lower ϵ, the loss is higher since a large amount of noise is added. As ϵ increases, the loss is observed to reduce although remains constant for DPFeDAVg as ϵ increases beyond 35. In addition, the standard FeDAvg remains fixed since privacy is not considered, thus the privacy budget was set to the maximum under such a case, i.e., $\epsilon = \epsilon_{\max}$, while the DPFML scheme showed improved performance compared to the other schemes. The proposed

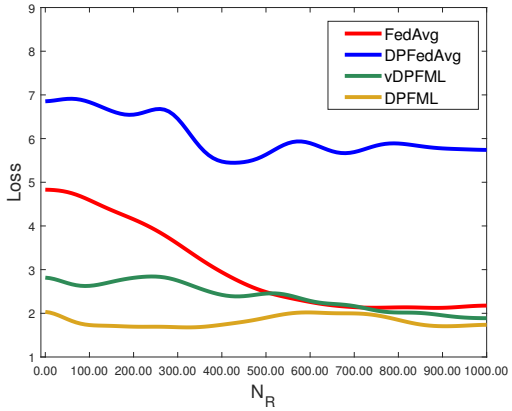

Figure 5. Performance in terms of the learning loss with respect to N_R.

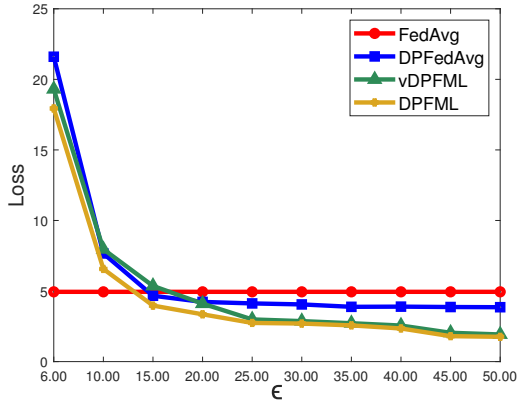

Figure 6. Performance in terms of the learning loss with respect to ϵ.

scheme ensures better performance without compromising privacy.

Next, we investigate the performance of the DPFML framework using accuracy as a metric. While the accuracy in FeDAvg increases with N_R as shown in Fig. 7, the DPFedAvg continues to produce an accuracy closer to zero even as N_R increases. Conversely, the DPFML achieved the best accuracy within the first 100 rounds and remain almost constant afterwards. This confirms the ability of the DPFML framework to reach convergence with a limited number of communication rounds. Likewise, the proposed DPFML scheme produces an improved performance compared to the FeDAvg as ϵ increases. This is depicted in Fig. 8. Although the standard FeDAvg produced a better accuracy when ϵ is closer to zero, this is unsurprising since such an approach provides no privacy. With DPFeDAvg, the accuracy was observed to be very low. This further confirms that FeDAvg is unsuitable for HDT frameworks where privacy is an important constraint since such an approach underestimates privacy.

To compare the performance in terms of the average connectivity cost, we investigate the long-term average connectivity cost ($\mathcal{U} = 100$) as a function of c_i. As presented in Fig. 9, the long-term average connectivity cost decreases as c_i increases in all cases since an improved computation capacity can reduce latency significantly thereby reducing the time cost.

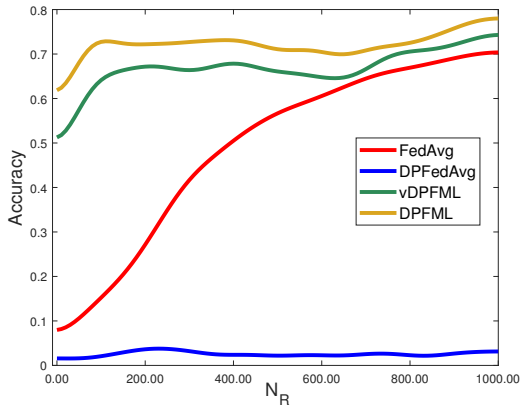

Figure 7. Performance in terms of the learning accuracy with respect to N_R.

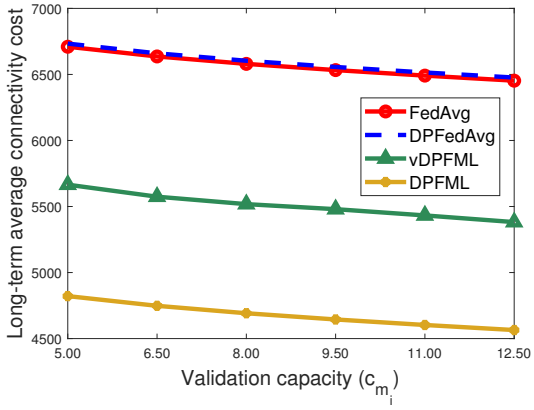

Figure 10. Impact of validation capacity on the average connectivity cost.

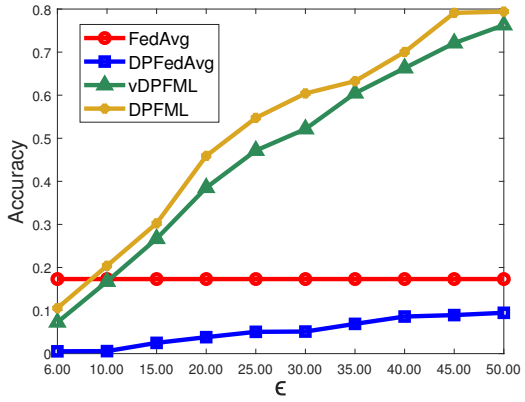

Figure 8. Performance in terms of the learning accuracy with respect to ϵ.

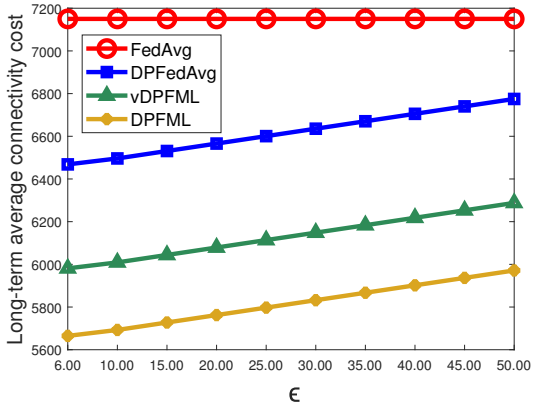

Figure 11. Impact of privacy budget.

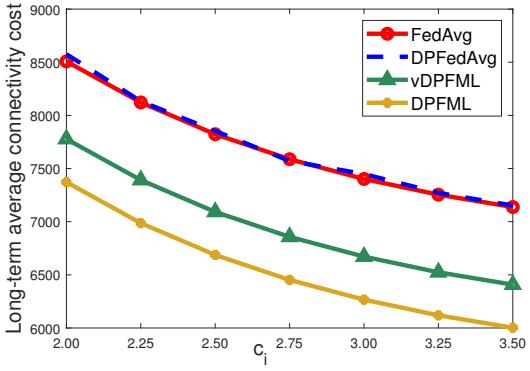

Figure 9. Performance in terms of the average connectivity cost.

However, the DPFML scheme requires the lowest cost while the FeDAvG and DPFeDAvg require more cost to ensure the timely synchronization of any PT-VT pair. A similar result is obtained when long-term average connectivity cost was investigated as c_{m_i} increases in Fig. 10, confirming that the standard validation process in blockchain may not be suitable in HDT systems because they require more time and energy to validate every transaction. In Fig. 11, we also confirmed that the connectivity cost indeed increases with ϵ. Interestingly, the cost is highest when no privacy or less privacy constraint is implemented reflecting the level of potential threats.

We note that the time cost in the DPFeDAvg is slightly higher than the conventional FeDAvg since the introduction of noise means more communication round is required to reach convergence than in its standard version. As shown in Fig. 12, the time cost increases with the V since more cost is incurred to reach consensus as V increases. With a minimum time cost for the DPFML, the synchronization gap is further reduced between any PT-VT pair. Interestingly, the energy cost increases in Fig. 13 as c_i increases since the agents have learnt the optimal parameters to ensure reduced cost and an increase in capacity can only increase the energy cost but cannot improve significantly the overall performance.

VII. Conclusion

HDT is a new technology that can transform many aspects of our current environment. To realize any HDT system, there must be reliable connectivity between any PT-VT pair to ensure timely synchronization between them. In this paper, we investigate the connectivity problem in the HDT framework and proposed the DPFML technique to achieve connectivity between any PT-VT pair. This is necessary since connectivity costs must be reduced to ensure timely synchronization without compromising privacy. To further reduce cost, we proposed a new consensus protocol, called the PoMQ, and formulated the connectivity problem as an MDP problem to allow optimization through the DDPG algorithm. We compared the

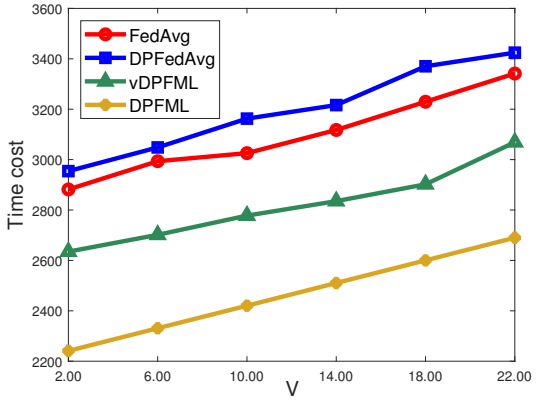

Figure 12. Performance in terms of the time cost.

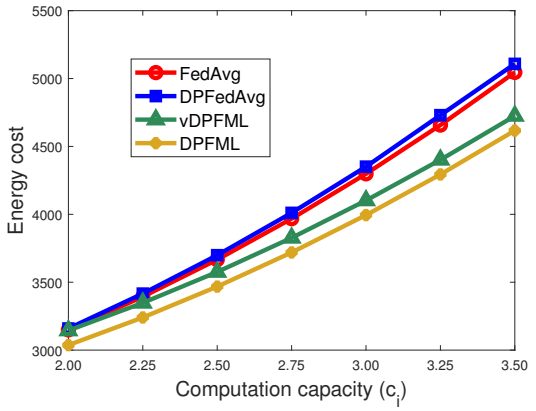

Figure 13. Performance in terms of the energy cost.

proposed solution with the existing ones and conclude that the proposed scheme is suitable for HDT systems where datasets are expected to be non-iid while privacy and security are also essential parameters.

REFERENCES

[1] 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, Feb. 2022.

[2] S. D. Okegbile, J. Cai, C. Yi, and D. Niyato, "Human Digital Twin for Personalized Healthcare: Vision, Architecture and Future Directions," *IEEE Network*, July 2022., DOI: 10.1109/MNET.118.2200071.

[3] S. D. Okegbile, and J. Cai, "Edge-assisted human-to-virtual twin connectivity scheme for human digital twin frameworks," *in IEEE VTC Conference*, Helsinki, Jun. 2022, pp. 1–6.

[4] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP annals*, vol. 66, no. 1, pp. 141–144, Jan. 2017.

[5] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics,* vol. 18, no. 2, pp. 1377–1386, Feb. 2021.

[6] J. Chen, C. Yi, S. D. Okegbile, J. Cai and X. S. Shen, "Networking Architecture and Key Supporting Technologies for Human Digital Twin in Personalized Healthcare: A Comprehensive Survey," *arXiv: 2301.03930*, Jan. 2023.

[7] H. Xiang, K. Wu, J. Chen, C. Yi, J. Cai, D. Niyato and X. S. Shen, "Edge Computing Empowered Tactile Internet for Human Digital Twin: Visions and Case Study," *arXiv:2304.07454*, Apr. 2023.

[8] J. Chen, C. Yi, H. Du, D. Niyato, J. Kang, J. Cai and X. S. Shen, "A Revolution of Personalized Healthcare: Enabling Human Digital Twin with Mobile AIGC," *arXiv:2307.12115*, Jul. 2023.

[9] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, May 2021.

[10] 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. 16219–16230, Jul. 2021.

[11] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[12] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things Journal*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[13] S. D. Okegbile, J. Cai, and A. S. Alfa, "Performance analysis of blockchain-enabled data sharing scheme in cloud-edge computing-based IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21520–21536, Nov. 2022.

[14] S. D. Okegbile, J. Cai, and A. S. Alfa, "Practical Byzantine fault tolerance-enhanced blockchain-enabled data sharing system: Latency and age of data package analysis," *IEEE Transactions on Mobile Computing*, Nov. 2022, DOI: 10.1109/TMC.2022.3223306.

[15] H. Wu, C. Chen, and L. Wang, "A theoretical perspective on differentially private federated multi-task learning," Nov. 2020, arXiv:2011.07179.

[16] M. Wu, D. Ye, J. Ding, Y. Guo, R. Yu, and M. Pan, "Incentivizing differentially private federated learning: A multidimensional contract approach," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10639–10651, Jan. 2021.

[17] M. E. Miller, and E. Spatz, "A unified view of a human digital twin," *Human-Intelligent Systems Integration*, vol. 4, pp. 23–33, Mar. 2022, DOI: https://doi.org/10.1007/s42454-022-00041-x.

[18] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Lui, Z. Pang, and M. Deen, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49088–49101, Apr. 2019.

[19] J. Zhang, L. Li, G. Lin, D. Fang, Y. Tai, and J. Huang, "Cyber resilience in healthcare digital twin on lung cancer," *IEEE Access*, vol. 8, pp. 20190011201913, Oct. 2020.

[20] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, and S. Valtolina, "Human digital twin for fitness management," *IEEE Access*, vol. 8, pp. 26637–26664, Feb. 2020.

[21] R. Martinez-Velazquez, R. Gamez, and A. El Saddik, "Cardio Twin: A Digital Twin of the human heart running on the edge," *in IEEE International Symposium on Medical Measurements and Applications*, Istanbul, Jun. 2019, pp. 1–6.

[22] A. El Saddik, H. Badawi, R. Velazquez, F. Laamarti, R. Diaz, N. Bagaria, and J. Arteaga-Falconi, "Dtwins: a digital twins ecosystem for health and well-being," *IEEE COMSOC MMTC Commun. Front*, vol. 14, pp. 39–43, May 2019.

[23] W. Yang, W. Xiang, Y. Yang, and P. Cheng, "Optimizing Federated Learning with Deep Reinforcement Learning for Digital Twin Empowered Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1884–1893, Feb. 2023.

[24] L. Jiang, H. Zheng, H. Tian, S. Xie, and Y. Zhang, "Cooperative federated learning and model update verification in blockchain empowered digital twin edge networks," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11154–11167, Jul. 2022.

[25] D. Gupta, O. Kayode, S. Bhatt, M. Gupta, and A. S. Tosun, "Hierarchical federated learning based anomaly detection using digital twins for smart healthcare," *in IEEE International Conference on Collaboration and Internet Computing*, Atlanta, Dec. 2021, pp. 16–25.

[26] Y. Qu, L. Gao, Y. Xiang, S. Shen, and S. Yu, "FedTwin: Blockchain-Enabled Adaptive Asynchronous Federated Learning for Digital Twin Networks," *IEEE Network*, vol. 36, no. 6, pp. 183–190, Jul. 2022.

[27] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp.50–60, May 2020.

[28] A. El Ouadrhiri, and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22359–22380, Feb. 2022.

[29] Y. Lu, X Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, Sept. 2019.

[30] V. Smith, C. K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *in Advances in neural information processing systems*, Long Beach, CA, USA, vol. 30, 2017, pp. 1–11.

[31] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1 pp. 41–75, Jul. 1997.
[32] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *in Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.
[33] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, Jul. 2006.
[34] S. D. Okegbile, and O. I. Ogunranti, "Users emulation attack management in the massive internet of things enabled environment," *ICT Express*, vol. 6, no. 4, pp. 353-356, Dec. 2020.
[35] J. C. Eidson, and K. B. Stanton, "Timing in cyber-physical systems: The last inch problem," *in IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication*, Beijing, Oct. 2015, pp. 19–24.
[36] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," *in IEEE Proceedings IEEE INFOCOM*, Mar. 2012, pp. 2731–2735.
[37] S.D. Okegbile, and B. T. Maharaj, "Age of information and success probability analysis in hybrid spectrum access-based massive cognitive radio networks," *Applied Sciences*, vol. 11, no. 4, pp. 1940, Feb. 2021.
[38] G. Kesidis, T. Konstantopoulos, and M. A. Zazanis, "Age of information using Markov-renewal methods," *Queueing Systems*, pp. 1-36, Aug. 2022.
[39] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "A general formula for the stationary distribution of the age of information and its application to single-server queues," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8305–8324, Aug. 2019.
[40] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," Sept. 2015, arXiv:1509.02971.
[41] Y. Liang, C. Guo, Z. Ding, and H. Hua, "Agent-based modeling in electricity market using deep deterministic policy gradient algorithm," IEEE transactions on power systems, vol. 35, no. 6, pp. 4180–4192, Jun. 2020.
[42] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Marchine Learning Research*, vol. 18, no. 153, pp. 1–43, 2018.
[43] S. Caldas et al., "LEAF: A benchmark for federated settings," 2018, arXiv:1812.01097.

Samuel D. Okegbile received the Ph.D. degree in computer engineering from the University of Pretoria, Pretoria, South Africa, in 2021. He is currently a Postdoctoral Fellow in the Network Intelligence and Innovation Laboratory, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada. His research interests are in the area of pervasive and mobile computing which includes various interesting topics in the human digital twin, internet of things, data sharing, artificial intelligence, wireless communication networks, and blockchain. He has received several awards, including the Horizon postdoctoral scholarship, the SENTECH scholarship and the University of Pretoria Doctoral Scholarship. He is also a regular reviewer for some IEEE journals and conferences and served as the Publication Chair for the 2023 Biennial Symposium on Communications.

Jun Cai (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Waterloo, ON, Canada, in 2004. From 2004 to 2006, he was a Postdoctoral Fellow with the Natural Sciences and Engineering Research Council of Canada (NSERC), McMaster University, Canada. From 2006 to 2018, he was with the Department of Electrical and Computer Engineering, University of Manitoba, Canada, where he was a Full Professor and the NSERC Industrial Research Chair. In 2019, he joined the Department of Electrical and Computer Engineering, Concordia University, Canada, as a Full Professor and the PERFORM Centre Research Chair. His current research interests include edge/fog computing, eHealth, radio resource management in wireless communications networks, and performance analysis. He served as the Registration Chair for QShine 2005, the Track/Symposium Technical Program Committee (TPC) Co-Chair for the IWCMC 2008, the IEEE Globecom 2010, the IEEE VTC 2012, the IEEE CCECE 2017, and the IEEE VTC 2019, and the Publicity Co-Chair for the IWCMC 2010, 2011, 2013, 2014, 2015, 2017, and 2020, the TPC Co-Chair for the IEEE GreenCom 2018 and the General chair for the 2023 Biennial Symposium on Communications. He also served on the Editorial Board of the IEEE Internet of Things Journal, the IET Communications, and Wireless Communications and Mobile Computing. He received the Best Paper Award from Chinacom in 2013, the Rh Award for outstanding contributions to research in applied sciences in 2012 from the University of Manitoba, and the Outstanding Service Award from the IEEE Globecom 2010.

Hao Zheng received the B. S. degree in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, where he is currently pursuing the M. S. degree in Computer Science and Technology. His research interests include reinforcement learning, human digital twin (HDT), edge computing and control optimization for servoing systems.

Jiayuan Chen received the M.S. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, where he is currently pursuing the Ph.D. degree in Computer Science and Technology. His research interests include reinforcement learning, mechanism design and distributionally robust optimization with applications in resource management and decision making for edge computing, human digital twin (HDT), and mobile artificial intelligence-generated content (AIGC).

Changyan Yi (S'16-M'18) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Manitoba, MB, Canada, in 2018. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His research interests include stochastic optimization, mechanism design, game theory, queueing scheduling and machine learning with applications in resource management and decision making for edge computing and edge intelligence, mobile and human digital twin, ubiquitous intelligent network and industrial cyber-physical system.