Polymorphic Learning of Heterogeneous Resources in Digital Twin Networks

Chengzhuo Han[*], Tingting Yang[†‡], Xianbin Li[†] Kai Wang[§] and Hailong Feng[§]

[*] School of cyber Science and Engineering, Southeast University, China
[†] Navigation College, Dalian Maritime University, China
[‡] Peng Cheng Laboratory, China
[§] National Institute of Defense Technology Innovation, Academy of Military Science China, China
{hcz_dmu, yangtingting820523, cxywangkai, lixianbincn}@163.com

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838407

***Abstract*—The emergence of digital twin technology is expected to reduce the significant cost of traditional physical debugging. Moreover, it can also fully combine IoT real-time data, large data analysis, and simulation, to make the best decisions. In the digital twin network, the data distribution and computing power of different device nodes shows great heterogeneity, but the internal node data are independent and identically distributed. Therefore, our learning algorithm should be able to learn not only the network commonality, but also the node specificity. In order to provide secure and personalised services for different device nodes, a polymorphic learning (PL) framework is proposed in this paper. PL divides the classical neural network model into a homomorphic model and a polymorphic model to realise flexible network control. Finally, the advantages of the PL algorithm in the collaborative optimisation of different nodes are proven through simulation experiments, and the algorithm running process is simulated through a classic case, with the final results proving the superiority of the algorithm.**

***Index Terms*—Polymorphic learning algorithm, Digital twin, Heterogeneous resources**

I. Introduction

Digital twin is a real-time mirror of physical entities in the digital world. It has the characteristics of virtual and real fusion, real-time intersection, loss, iterative operation, and data driving of all elements, processes, and services [1] [2] [4]. In digital twin, the efficient of the privacy security protection mechanism is especially important. How to ensure high accuracy and reliability, while reducing additional communication delay is an especially important issue [3]. At the same time, in digital twin, the computing capabilities of different device nodes show great heterogeneity, and the data inside the nodes are independent and equally distributed [5]. Each digital twin node has different data characteristics, and the data sets of these nodes may be non-independent and identically distributed [7]. Therefore, our learning algorithm should be able to learn both the universality and the particularity of the nodes in digital twin.

Currently, it is widely used in industrial smart cities, health care, environmental protection, and other fields. In [8], an adaptive artificial intelligence framework based on efficient feature selection is proposed to match the resource allocation of cybertwins. The framework customises available computing resources to suit the artificial intelligence model complexity. In [9], the authors proposed a mathematical total service delay model of a multi-access edge computing system based on cybertwins, which takes advantage of the feature that cybertwins have a control plane and the cooperation between cybertwins can be realised. The features are then transmitted to the local server. In this study, physical network data and information are placed onto twin platforms, which increases user data security risks, such as leakage or attack. The PL algorithm proposed in this paper divides the learning network into a homomorphic network and a polymorphic network. Only homomorphic features are transmitted in the process of parameter aggregation, to protect node privacy. The PL algorithm can reduce the time delay of model convergence by only aggregating homomorphic models while ensuring the inference accuracy.

Digital twin is applied in the digital transformation of traditional networks by new-generation digital technologies such as cloud computing, big data, mobile internet, artificial intelligence, and blockchain [6]. In order to realise the intelligent routing management mechanism of the distributed active digital twin, and make use of the powerful network learning ability to serve different users and different businesses on demand, a polymorphic learning (PL) framework is proposed in this paper. The framework divides a neural network model into a homomorphic model and a polymorphic model to achieve flexible control and management of the network. This provides real-time, flexible, efficient, and intelligent active network topology, along with dynamic routing according to the dynamic service requirements of different users. In addition, the PL framework can fulfil the efficient utilisation of global network resources. For the homomorphic model parameters, global data features are represented by a weighted average method, while for the polymorphic model, customer data features are represented by different parameter values. The framework optimises the proportion of dynamic parameters from the perspective of training delay according to equipment communication and computing resources. During the training, the global model partitioning ratio was dynamically adjusted according to the model accuracy, and the final model was optimised according to client reasoning accuracy.

Based on the simulated data, we replicate the PL algorithm

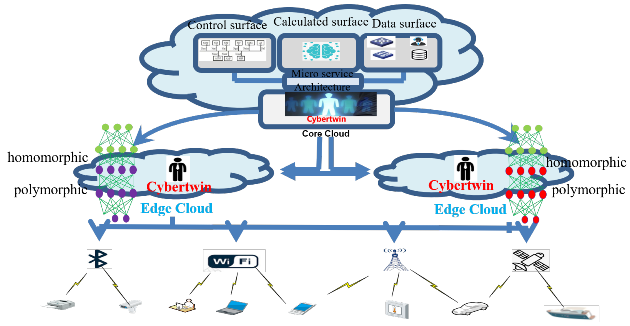

Fig. 1. Cybertwin heterogeneous neural network architecture based on SDN

segmentation model process in this paper. Experiments show that our algorithm is more accurate than the classical federated learning algorithm when the nodes have heterogeneity and specific features.

1) A PL framework for the flexible control and management of digital twin is proposed. The data distribution and node composition to which the PL algorithm can be applied are described in section II.
2) The framework divides the neural network model into homomorphic and polymorphic models to achieve the flexible control and management of the network. The homomorphic model parameters represent the global data characteristics by a weighted average method, while the polymorphic model represents the client data characteristics by different parameter values.
3) We briefly introduce the workflow of the polymorphic learning algorithm. Experimental results show the superiority of the PL algorithm in heterogeneous node resource processing.

II. System Model

A. Edge heterogeneous model

Although the cloud may be powerful enough, the huge volume of data still puts a heavy burden on communication networks. In the PL framework, neural network models are divided into homomorphic frame models and polymorphic models, as shown in Fig. 1. Since the polymorphic model communication resources can be saved during the process of parameter aggregation, PL can effectively reduce the communication time delay when dealing with heterogeneous node data communication problems.

A neural network with N-layer neurons has $N-1$ splitting points. At the splitting point i ($0 \le i \le N-1$) and ($i \in Z$), the model is divided into two parts: Homomorphic model and polymorphic model. The proportion of homomorphic model parameters is a_i, and the proportion of polymorphic model parameters is $1-a_i$. Therefore, the number of homomorphic model parameters of splitting point i is

$$V_s = V * a_i. \tag{1}$$

where, V is the total number of local model parameters. The number of polymorphic parameters of splitting point i as

$$V_n = V * (1-a_i). \tag{2}$$

B. Edge computing

As mentioned above, distributed nodes and digital twin servers exchange data with each other during each communication. Specifically, each time period is further divided into multiple time periods. Each node must transmit information within its own time period. According to Ref. [17], the standard maximum length of each timeframe for information transmission is 10 ms. In fact, the parameters transmission delay is measured in seconds, which is much longer than the duration of a time period [18]. Therefore, we can evaluate the node data transmission rate i [19] by replacing the parallel channel capacity with the average channel capacity of the node information uplink channel, which can be expressed as

$$R_i^U = WE_h\left\{log_2\left(1+\frac{p_i^U|h_{\mathrm{i}}^{\mathrm{U}}|^2}{N_0}\right)\right\}, \tag{3}$$

where $h_{\mathrm{i}}^{\mathrm{U}}$ uplink channel power gain of node i, p_i^k for the corresponding transmission power, E_h as the node information uplink channel power gain expectation, N_0 for the channel noise power, W as the total bandwidth of the entire digital twin system.

After the homomorphic parameters are aggregated by the global digital twin system, the virtual base station (BS) mapping broadcasts the global homomorphic parameters to all digital nodes, which in turn update the physical nodes. In this way, all physical nodes can synchronously receive the global parameters. Suppose $h_{\mathrm{i}}^{\mathrm{D}}$ represents the down channel power gain of user i and p^D represents the transmitted power of all users. Therefore, the downlink data rate is

$$R^D = W\min_i\left\{E_i\left\{log_2\left(1+\frac{p^D|h_{\mathrm{i}}^{\mathrm{D}}|^2}{N_0}\right)\right\}\right\}. \tag{4}$$

III. Problem Formulation

In this section, we first propose an indicator to quantify the learning efficiency of digital twin models. On this basis, the overall learning delay of each communication cycle was analysed, and the optimisation problem of system learning efficiency maximisation is given.

A. Model learning delay analysis

As mentioned above, our main objective is to improve the PL system inference accuracy and model convergence time delay. Therefore, it is necessary to optimise the transmission time delay between nodes and digital twin networks. Latency in communication during learning is analysed in detail below.

1) Compute the local model. The local training computation time delay of node i is represented by T_i^l.

2) Upload shared parameters. Therefore, the information transmission delay of physical node I can be expressed as T

$$T_i^L = \frac{V_s}{\tau_i R_i^U}, \forall i \in I. \tag{5}$$

Where τ_I is the ratio of the node i to the whole time slot during a certain period of time in a channel, and V is the number of learning model parameters, which is constant for all nodes in the network.

3) For all nodes, the delay in downloading the homomorphic parameter as

$$T^D = \frac{V_s}{R_D} \tag{6}$$

.

4) Update the local learning model. T_i^U is the model update delay for node i.

As the shared rate a_i value is exceedingly small, its data transmission delay is negligible. In addition, edge servers generally have strong integrated capacity, therefore, the aggregation delay is also negligible.

B. Problem Formulation

In this work, we improved the overall learning efficiency of the system through the division of local learning models and the reasonable allocation of network communication resources. According to Ref. [20], we used the following criteria to judge whether the sensory system training effect achieved the expected outcome.

Definition: The overall learning efficiency of digital twin networks can be defined as:

$$E = \frac{\sum_{i=0}^{I} e_i}{T} \tag{7}$$

, where e_i represents the convergence rate of the best splitting point of user i.

The delay of node and digital whorl network in each communication round of PL communication system is given

$$T = \max_{i \in I} \left\{ T_i^U + T_i^L \right\} + \max_{i \in I} T_i^T + T^D. \tag{8}$$

IV. Problem Resolution

In this section, we propose a joint optimisation scheme of user split point and bandwidth allocation to ensure that the delay of all users is approximately equal.

A. PL Workflow

The PL workflow includes initialisation, split model, training, and aggregation, as shown in Figure.2.

Initialisation: During the initialisation phase, the PL server first waits for other devices to join the PL digital twin network learning task. The nodes participate in training by sending the initial segmentation information of their local learning model to the digital twin network. After determining the nodes participating in the learning training, the digital twin initialises the optimiser and weights of the neural network model and collects the polymorphic model segmentation information of the nodes participating in the learning training.

Split Model: Based on the information received from the client, we determined the client model as homomorphic, considering the computing power, upload and download efficiency, channel state, and other factors. The proportion of homomorphic divisions of a model divided by the PL Server requires a trade-off between communication overhead and model precision. We discuss this tradeoff in section III-B.

Training: At the beginning of each communication wheel, the central server sends the proportion of homomorphic portions to the client. Each client uses local stochastic gradient descent(SGD) to train the local model (made up of homomorphic and polymorphic parts) and returns the updated model to the response server. Local equipment only affects the time each customer spends on training and does not add additional communication overheads. The client loops through this process several times. All clients then send updates of the homomorphic parameters of the model to the PL Server.

Aggregation: Firstly, the PL server aggregates all the updates using the FedAvg algorithm. The PL server then sends the updated model to the client and starts the next round of synchronisation. The obvious difference between the PL algorithm and standard FL algorithm in the model integration stage is that PL can achieve the separation of isomorphic features and heterogeneous features. Therefore, PL learning architecture is more suitable than traditional architecture to resolve the heterogeneous node data resource allocation problem.

A flow chart of the polymorphic segmentation point adaptive algorithm is shown in the figure.3

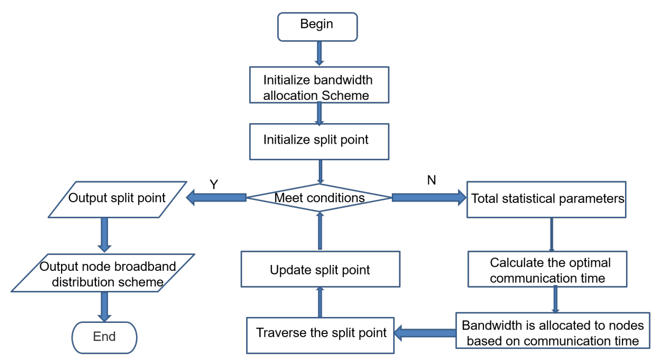

Fig. 3. The adaptive bandwidth allocation algorithm flowchart

V. Algorithm simulation and result discussion

This section first verifies the performance of the algorithm in solving the heterogeneous nodes resource allocation problem. Then, the simulation of the selection process of segmentation points verifies the superiority of the algorithm in dealing with heterogeneous data.

A. PL algorithm for heterogeneous data processing performance experiment

This section preliminarily verifies the algorithm performance when dealing with heterogeneous data, through a comparison test of the PL algorithm and classical ML algorithm via linear regression. In order to increase the prediction difficulty, we introduced the noise parameter $NOSIE \in (0, 0.05)$.

The neural network used in this experiment is shown in Fig.4. As shown in the Fig.4, *inputs* layer and hidden layer

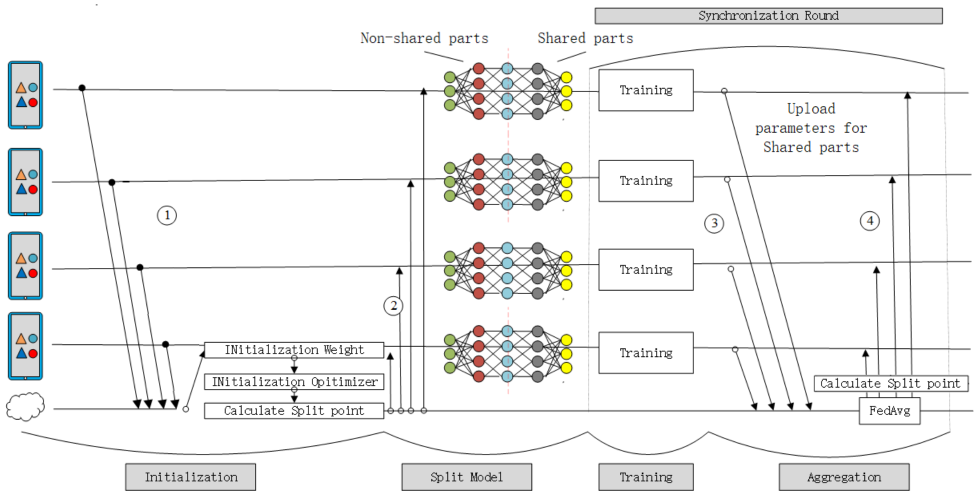

Fig. 2. The Workflow of SFL

Algorithm 1 : *Find the best cut point and bandwidth allocation*

input: V, $\mathrm{T}_i^{\mathrm{U}}$, $\mathrm{T}_i^{\mathrm{L}}$, τ_i, R^{D}
A(Set of splitting points)
output: $\mathrm{R}_i^{\mathrm{U}}$,
a(Optimal splitting point)
$result \leftarrow 0$, $a^* \leftarrow 0$
while Meet conditions **do**
  while a in A **do**
    $\mathrm{T}_i^{\mathrm{T}} \leftarrow \frac{\mathrm{aV}}{\tau_i \mathrm{R}_i^{\mathrm{U}}}$
    $\mathrm{T}^{\mathrm{D}} \leftarrow \frac{\mathrm{aV}}{\mathrm{R}^{\mathrm{D}}}$
    $T \leftarrow \max_{\mathrm{i} \in \mathrm{I}}\{\mathrm{T}_i^{\mathrm{U}} + \mathrm{T}_i^{\mathrm{L}}\}$
    $T \leftarrow \mathrm{T} + \mathrm{T}_i^{\mathrm{T}} + \mathrm{T}^{\mathrm{D}}$
    $E = \frac{f(a)}{T}$
    if $result < E$ **then**
      $relust \leftarrow E$,
      $a \leftarrow a^*$
    end if
  end while
  while a in A **do**
    $\mathrm{T}_i^{\mathrm{T}} \leftarrow \frac{\mathrm{aV}}{\tau_i \mathrm{R}_i^{\mathrm{U}}}$
    $\mathrm{T}^{\mathrm{D}} \leftarrow \frac{\mathrm{aV}}{\mathrm{R}^{\mathrm{D}}}$
    Calculate the optimal latency T_{VB}^* by solving 5 with bisection method.
    Determine the bandwidth R_i^U as 3.
  end while
end while

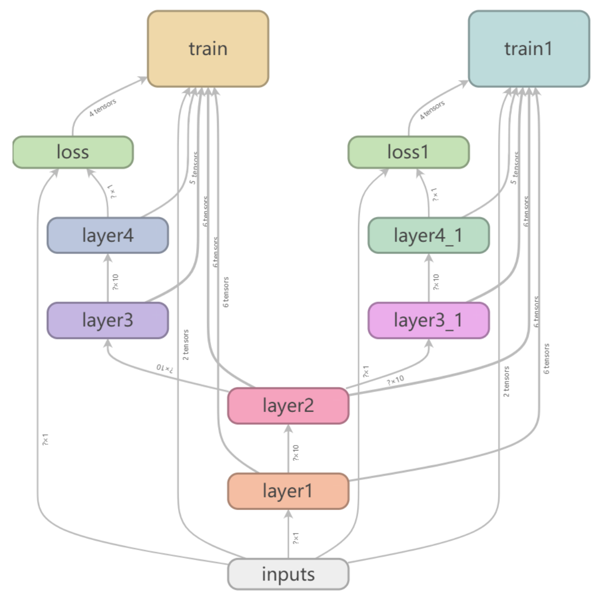

Fig. 4. Model of cutting edge

layer1 and *layer2* are the polymorphic parameters of the model, that do not need uploading to the edge server. The remaining model parameters are homomorphic parameters that need to be uploaded to the edge server. The communication time delay in the learning process mainly stems from the information transmission process when the isomorphic model parameters are aggregated through the central cloud.

The experimental results are shown in Fig.5. Wherein, the abscissa denotes the learning times of the local model, and the ordinate is the local model loss function. As can be seen from Fig.5, the PL model can achieve high precision when dealing with heterogeneous node resource allocation. This is because although the node data is heterogeneous, the two nodes are similar with similar properties, and the polymorphic parameter model has better performance.

B. PL algorithm in communication feasibility analysis experiment

In this experiment, K nodes are stochastically distributed in a digital twin network. Digital twin networks have a range of 500m. All nodes are connected to digital twin through wireless channels. We assume that the ratio of calculation delay to transmission delay is 1:1. The proposed PL algorithm is implemented and evaluated by modifying the TensorFlow machine learning framework. In the evaluation, the dataset of the simulation experiment is numbered in Table I.

According to formula 7, we can calculate the value of E as

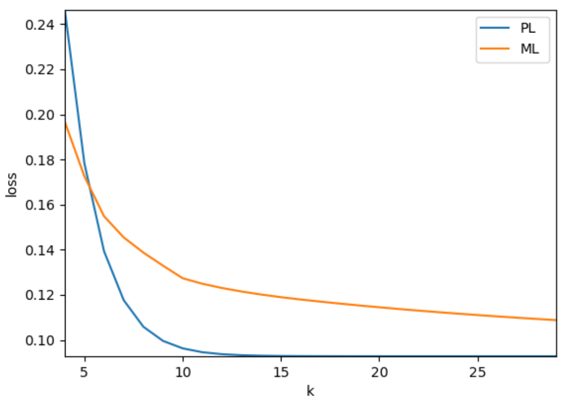

Fig. 5. The experimental results of the control group learning data as IID

TABLE I

	X	Y:Data_A	Y:Data_B	Size
1	[- 1,1]	X^3+ NOISE	X^3 + NOISE + 1	300
2	[- 1,1]	X^3 +2X^2 NOISE	X^3 +2X^2 + NOISE + 2	300
3	[- 1,1]	2X^3 + X^3+NOISE	2X^3 + X^3+NOISE + 5	300

shown in Table II. So, the model splitting point for group 1 is '3', group 2 is '1', and group 3 is '2'.

TABLE II

	1	2	3	4	5
1	0.50	0.50	0.83	0.85	0.96
2	0.50	0.58	0.62	0.66	0.52
3	0.71	0.83	0.50	0.66	0.52

VI. Conclusions

In this paper, the algorithm optimisation and broadband allocation of a digital twin system based on the PL algorithm are studied. A PL algorithm is proposed to effectively deal with digital twin heterogeneous data. Based on accurate quadratic function data, the simulation results verify the performance of our method. In future work, we aim to apply a learning algorithm to realise intelligent algorithm optimisation. In addition, continuous dynamic adjustment algorithms applicable to training will also be considered.

Acknowledgement

Key Projects of the National Defense Foundation Strengthening Plan under Grant 2020-JCJQ-ZD-020-05, National Defense Technology Innovation Research Institute under Grant 2020KCYWXGK4098, Talent Program of Liaoning Province under Grant XLYC1807149, National Key R&D Program of China under Grant 2020YFB1806800.

References

[1] Q. Yu, J. Ren, Y. Fu, Y. Li and W. Zhang, "Cybertwin: An Origin of Next Generation Network Architecture," IEEE Wireless Communications, vol. 26, no. 6, pp. 111-117, Dec. 2019.

[2] F. Wang, L. Yang, X. Cheng, S. Han and J. Yang, "Network softwarization and parallel networks: beyond software-defined networks," in IEEE Network, vol. 30, no. 4, pp. 60-65, July-August 2016.

[3] H. Liang and W. Zhang, "A Barter and Combinatorial Auction Based Hierarchical Resource Trade Mechanism for Cybertwin Network," 2020 3rd International Conference on Hot Information-Centric Networking, 2020, pp. 84-89.

[4] Y. Guan, R. Lu, Y. Zheng, S. Zhang, J. Shao and G. Wei, "Toward Privacy-Preserving Cybertwin-Based Spatio-Temporal Keyword Query for ITS in 6G Era," in IEEE Internet of Things Journal, 2021.

[5] J. Li, W. Shi, Q. Ye, S. Zhang, W. Zhuang and X. Shen, "Joint Virtual Network Topology Design and Embedding for Cybertwin-Enabled 6G Core Networks," in IEEE Internet of Things Journal, 2021.

[6] H. Xu, J. Wu, J. Li and X. Lin, "Deep Reinforcement Learning-based Cybertwin Architecture for 6G IIoT: An Integrated Design of Control, Communication, and Computing," in IEEE Internet of Things Journal, 2021.

[7] S. Yan, Q. Ye and W. Zhuang, "Learning-Based Transmission Protocol Customization for VoD Streaming in Cybertwin-Enabled Next Generation Core Networks," in IEEE Internet of Things Journal ,2021.

[8] S. Shen, C. Yu, K. Zhang and S. Ci, "Adaptive Artificial Intelligence for Resource-constrained Connected Vehicles in Cybertwin-driven 6G Network," in IEEE Internet of Things Journal, 2021.

[9] T. K. Rodrigues, J. Liu and N. Kato, "Application of Cybertwin for Offloading in Mobile Multi-access Edge Computing for 6G Networks," in IEEE Internet of Things Journal, 2021.

[10] W. Sun, J. Liu and Y. Yue, "AI-Enhanced Offloading in Edge Computing: When Machine Learning Meets Industrial IoT," in IEEE Network, vol. 33, no. 5, pp. 68-74, Sept.-Oct. 2019.

[11] Wang, J. , and G. Joshi . "Cooperative SGD: A unified Framework for the Design and Analysis of Communication-Efficient SGD Algorithms." (2018).

[12] B. Zhang, A. Davoodi and Y. H. Hu, "Exploring Energy and Accuracy Tradeoff in Structure Simplification of Trained Deep Neural Networks," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 4, pp. 836-848, Dec. 2018.

[13] B. Yang, X. Cao, X. Li, Q. Zhang and L. Qian, "Mobile-Edge-Computing-Based Hierarchical Machine Learning Tasks Distribution for IIoT," in IEEE Internet of Things Journal, vol. 7, no. 3, pp. 2169-2180, March 2020.

[14] Z. Tang, S. Shi and X. Chu, "Communication-Efficient Decentralized Learning with Sparsification and Adaptive Peer Selection," in Proc. IEEE International Conference on Distributed Computing Systems, 2020, pp. 1207-1208.

[15] F. Sattler, S. Wiedemann, K. -R. Mller and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3400-3413, Sept. 2020.

[16] Dong Y , Giannakis G B , Chen T , et al. "Communication-Efficient Robust Federated Learning Over Heterogeneous Datasets,". arXiv e-prints, 2020.

[17] Std-T, A. "Evolved Universal Terrestrial Radio Access; Physical channels and modulation." 3GPP, TS 36.211, v8.8.0 (2009).

[18] Lin, Y. , et al. "Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training." (2017).

[19] J. Ren, G. Yu, Y. Cai and Y. He, "Latency Optimization for Resource Allocation in Mobile-Edge Computation Offloading," in IEEE Transactions on Wireless Communications, vol. 17, no. 8, pp. 5506-5519, Aug. 2018.

[20] J. Ren, G. Yu and G. Ding, "Accelerating DNN Training in Wireless Federated Edge Learning Systems," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 1, pp. 219-232, Jan. 2021.