K-DUMBs IoRT: Knowledge Driven Unified Model Block Sharing in the Internet of Robotic Things

Muhammad Waqas Nawaz, Olaoluwa Popoola, Muhammad Ali Imran
Qammer H. Abbasi
Email: m.nawaz.1@research.gla.ac.uk,
{olaoluwa.popular, qammer.abbasi, muhammad.imran }@glasgow.ac.uk
James Watt School of Engineering, University of Glasgow, Glasgow, U.K

***Abstract*—6G is expected to revolutionize the Internet of things (IoT) applications toward a future of completely intelligent and autonomous systems. Conventional machine-learning approaches involve centralizing training data in a data center, where the algorithms can be used for data analysis and inference. To promote green computing in IoT applications, Machine-2-Machine (M2M) technologies are largely focused on lowering energy consumption and creating effective IT infrastructure. In this paper, we introduce an AI-enabled One-Shot Interference(O-SI) Knowledge-Driven unified model block sharing (K-Dumbs) framework in which actionable knowledge is aggregated from the training perception robots to facilitate others at the Edge in the vicinity. To demonstrate the practicality of the proposed concept, we explore a K-Dumb Fed-Average (FedAvg) algorithm to meet the massively distributed and unbalanced pattern and privacy requirement of the Internet of Robotic Things(IoRT). Simulation results show that, when compared to traditional Federated Learning (FL) systems, the proposed K-Dumb FedAvg architecture delivers higher information-sharing and learning quality. In addition, we validate our method using MNIST handwritten digits for training image processing with an accuracy that is close to the centralized solution for up to 80% reduction in the amount of exchange data with the O-SI method. Furthermore, the suggested solution reduces IoRT energy consumption by up to 10 times and protects privacy.**

***Index Terms*—Machine-2-Machine (M2M), Internet of Robotic Things (IoRT), One-shot Inference (O-SI), Federated Learning (FL)**

I. INTRODUCTION

Machine learning (ML) has received a lot of attention and is expected to be key to the creation of sixth-generation (6G) mobile networks [1]. It is essential to anticipate the potential applications, methodologies, use cases, and problems of 6G technology at this time [2]. The most recent applications, including Internet of things (IoT), the Internet of robotic things (IoRT), Artificial intelligence (AI), Cognitive Networks (CN), IP multimedia subsystem (IMS), eHealth, Internet of Vehicles (IoV), not limited to the above application will demonstrate the 6G cellular network [3]. The 6G next generation of cellular technology as illustrated in Fig 1. The 2030 agenda includes investigations and development on 6G wireless networks in order to meet the demands of the intelligent information society [4]. 6G will be a key enabler providing low latency and supporting high throughput and ultra-high-quality videos [5]. The intelligent information society is expected to have highly digitized, intelligent, autonomous, global data-driven, and unconstrained wireless connectivity [6].

One of the main challenges in 6G knowledge aggregation is to handle the vast amounts of data generated by the numerous devices and sensors in the network [7]. To overcome this challenge, various ML and data analysis techniques can be used, such as Federated learning (FL), edge computing, and data fusion. Meanwhile, with data from millions/billions of devices, and limited connectivity resources, uploading data from all edge devices to a parameter server for centralized ML is quite infeasible [8]. For these reasons, it is preferable to implement distributed learning algorithms that allow devices to collaborate to develop a unified learning model using local training [9]. FL is one of the most promising distributed ML frameworks that allow users to benefit from the shared model built using this rich data without having to store it centrally [10]. The merging of IoT, AI, and Robotics accelerates the development of IoRT application, which increases pragmatically aware decision-making assistance for complicated machine intelligence operations [11]

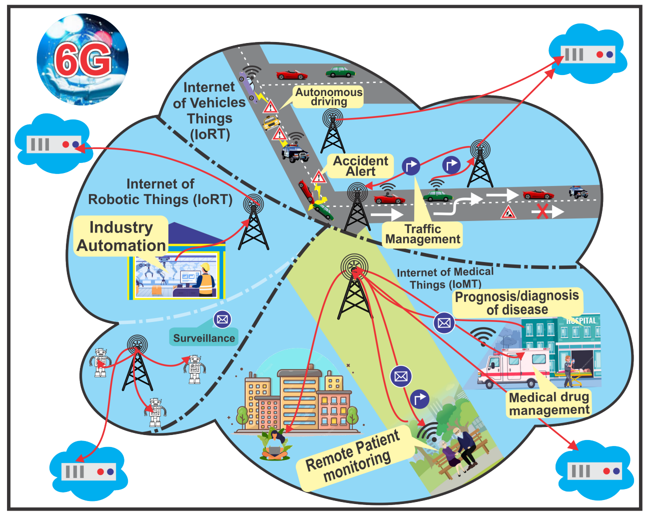

Fig. 1: AI enabled 6G networks

In this paper, we introduce a framework on *AI-enabled knowledge sharing in Dumb robots using FedAvg* to achieve collaborative intelligence sharing. We also use distributive model selection aggregation at the centralized levels to share the average weight with unidentified domains to transfer knowledge. To show knowledge aggregation we simulate a robot vehicle perception scenario where a new vehicle gains knowledge and intelligence, without training, but by leveraging on trained information of previous vehicles. This is done by training models locally in the form of supervised classifiers to identify handwritten digit recognition. After acquiring knowledge from the models locally, numerous nodes will upload their weights to the central server, where the selection and aggregation of the global model will take place. Robust perception algorithm intelligently shares the global aggregated weights to new robot vehicles which is immediately able to complete the perception task without any local training. This enables vehicles learn quickly, reduces latency in task completion, and support reduced energy AI by the acquisition of intelligence with reduced training requirement.

In Section II, describes the related work on FL and ML offloading and edge computing to support the important development based on knowledge sharing. In Section III, we discuss the architecture of the K-Dumbs decision framework as well as the problem formulation. Section IV describes the K-Dumb FedAvg mechanism in detail. Section V discusses simulation results. Section VI concludes the paper and discusses future work.

II. Related Work

Recent studies centered on edge caching using classic probability-based approaches. The authors in [12] proposed Heterogeneous distributed machine learning FL technology that allows the machine to train a shared prediction model jointly. Vermesan, O et al. [13] presented a comprehensive examination of the IoRT concept, technologies, architectures, and applications, as well as a prediction of future challenges, developments, and applications. The ie of distributed mobile edge optimization through FL [14] and intelligent network-based robots [15] has drawn a lot of interest in the convex environment to control communication [16], and certain algorithms do have a special emphasis on communication effectiveness, averaging in distributed primal-dual optimization [17], communication efficient using an approximate newton-type [18].In [19] discussed trading computation for communication and [20] efficient communication optimization of empirical loss. Computational harvesting [21] using on-board computing and self-orchestration on edge clouds relates to moving (computational or storage) workloads away from the centralized cloud and toward the endpoints (often the sources of data) offloading [22].

ML technologies based on artificial intelligence (AI) have become quite popular on the Internet of Vehicles (IoVs) with the development of intelligent transportation systems [23], [24]. IoVs tend to communicate data across cars and infrastructures because they are built with AI-enabled onboard sensors. Data sharing includes, but is not limited to, the sharing of processing, telecommunication, and spectral resources; there is also knowledge traded during the ML process [25]. Through the sharing of knowledge, vehicles can learn through by sharing their experience. It can enhance decision-making abilities while also accelerating the learning process [26]. For instance, the traffic flow models that have been learned can represent the common understanding of collaborative vehicle sensing. Based on the data it has gathered, a single IoVs train a unique algorithm model of current traffic pattern. By combining the learning models from all cars, a complete model may be created. In this situation, information sharing reflects swarm intelligence, which is significant for future Intelligent Transportation System (ITS) applications like traffic management and autonomous driving [27], [28].

Although there are many different types of applications for networked intelligent robots, most of them still aren't user-friendly. Artificial Intelligence will play an important role in promoting knowledge aggregation in the networked intelligent robot system to be investigated in the future 6G and beyond green IoT. We explore One-shot Inference sharing algorithm family , where AI-enable unified global model are shared with new domain to reap the benefits of data aggregation. The contributions of this paper are:

- Introducing novel K-Dumbs in robotics architecture to support green computing in IoRT and Vehicular Robots, reducing power consumption and Improving efficiency.
- K-Dumbs FedAvg involves one-shot communication between the server and client
- K-Dumbs uses a simple and useful algorithm that can be used in this situation to reduce the computational cost, preserving data privacy and security.
- K-Dumbs identified the problem to reduce the communication overhead and enable knowledge sharing.

III. System Model

While the goal of K-transfer to move knowledge from one domain to another, the aim of K-sharing to reduce the distribution divergence between several domains is the main concept. Many advancements can be understood as adapting the model's structure (and consequently the loss function) to be more amenable to optimization by straightforward gradient-based methods. The recent multitude of successful Deep Learning (DL) applications has almost exclusively relied on variants of stochastic gradient descent (SGD). The system model shown in Fig 2.

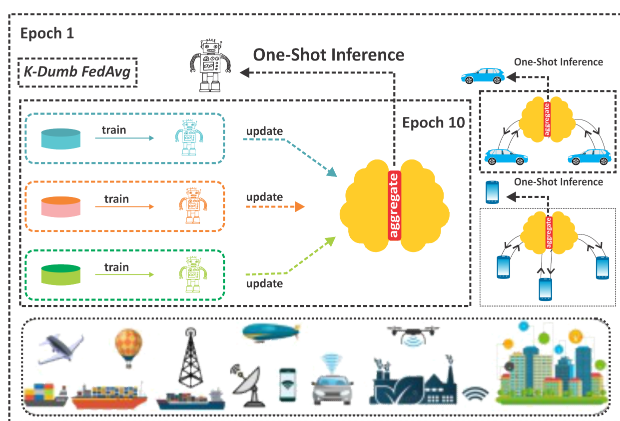

Fig. 2: Basic K-Dumb FedAvg Architecture

SGD can be naively applied to the federated optimization problem, where one batch gradient calculation is performed each round of communication (let's say on a client chosen at random). This method is computationally efficient, but it requires a large number of training rounds to develop good models. In our MNIST dataset, we take this baseline in our experiments and select a M fraction of clients. In a typical FedSGD implementation with $M = 1$ as well as a set learning rate n, each client k computes $a_k = \nabla F_k(z_k)$

average gradient of the current model on the it local data z_t, and central server combines these gradients and update the $z_{t+1} \leftarrow z_t - \eta \sum_{k=1}^{K} \frac{n_i}{n} a_k$, since $\sum_{k=1}^{K} \frac{n_i}{n} a_k = \nabla f(z_t)$ The equivalent update is provided by $\forall k \; z_{t+1} \leftarrow z_t - \eta a_k$ and then $z_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \; z_{t+1}^k$. In other words, each client performs one step of gradient descent on the current model using its local data, and the server then computes a weighted average of the resulting models. After we've built the algorithm in this manner, we may add further work to each client by iterating the local update $z^t \leftarrow z^t - \eta \nabla F_k(z^k)$ multiples time before the averaging steps. These steps are stopped after the desired epoch to have final global model that is the combination of all local weighted average. The final weighted average of resulting models are shared with new domain $\forall k$ $z_{t+1} \leftarrow z_t - \eta \nabla \sum f a_{k(global)}$ to minimize the communication overheads. we call this approach *K-Dumb FedAvg*. Three important parameters determine how much computation is done: *M*, the percentage of clients that compute each round; *E*, the number of training passes each client does over its local dataset each round; and *B*, the local mini-batch size that is used for client updates. Averaging models in parameter space could generate an arbitrarily terrible model for general non-convex objectives. In model transfer, the convolution layers try to extract low-level characteristics concerning activity recognition. We do not alter these layers' parameters during back-propagation since we keep them and the max-pooling layers frozen. The fully connected layers are at a higher level, they focus on learning unique qualities for the task and user.Hence, throughout training, we update their parameters. The weighted average shared with new domain without leaking any information from the user's, protect privacy and security. In the FedAvg technique, it has been demonstrated that averaging the models with shared initialization can achieve good performance in loss reduction [29].

IV. K-Dumb FedAverage

A. Federated Optimization

For optimization, the recent slew of successful deep learning applications has almost entirely depended on versions of stochastic gradient descent (SGD). By starting with SGD, the method is designed exclusively for FL. Each client iteratively cycles through the present system with its own data, and the server computes an arithmetic mean of the total. Examples of ideal problems for FL include personalizing for recommendation systems, speech recognition and natural language processing, and computer vision applications [30]. FL is derived from a common distributed optimization issue in which:

- **non-IID** The amount of training data that each user shares is determined by their usage, the local data set of any individual user will not be typical of the population distribution.
- **Unbalanced** The amount of local training data will vary depending on which customers utilize the service or app more frequently than others.
- **Widely Disseminated** We predict that the number of clients taking part in optimization will be significantly greater than the average number of times per client.
- **Limited communication** Edge devices generally have poor, expensive, or no internet connectivity.

These challenges lie outside the purview of the current effort; instead, we employ a controlled setting conducive to experimentation. However, it still handles the critical challenges of client availability as well as unbalanced and non-IID data.

B. Edge Intelligence

The phenomenal increase in the number of connected devices distributed widely inside the IoT has caused the volume of data collected and exchanged to increase at breakneck speed [31]. The inefficiencies of centralizing all this data on the cloud have resulted in the emergence of new computer and networking paradigms in recent years [32]. There are benefits to processing near the data sources in terms of uploading latency and bandwidth.

The inherent benefits of data privacy another important advantage because raw data does not travel very far. Furthermore, the data is used to feed increasingly advanced AI models, particularly Deep Learning (DL), is becoming more and more popular across a broad range of industries and application areas. The dangers and disadvantages of sharing personal information online have been more evident in recent years. FL is the answer to edge computing that utilizes DL technology while safeguarding data privacy [33].

C. Problem Conceptulization

In this section, we focus on the optimization of non-IID (non-Independent and Identically distributed) and IID properties, as well as the importance of communication restrictions. An implemented distributed optimization system must deal with various real-world challenges, including client data sets that vary as new data is added and removed and client availability, which is directly related to the local data distribution. We implement Federated Averaging with Local Stochastic Gradient Descent (FedAvg-SGD), each participating percepton robot trains a local model using SGD on its own data, and periodically updates the global model by sending its model parameters to the central server. The central server aggregates the parameters from all devices using the FedAvg method, which takes the average of all the parameters to obtain a new global model. This procedure is carried out repeatedly until it reaches a predetermined criterion or convergence. The new global model is then broadcasted to the new domain of robots which was not part of the training initially. The new perception robot reaps the benefits of knowledge sharing, which uses the inference of the globally trained model to initialize digit recognition.

We make the assumption of a synchronous updating mechanism with communication rounds. There are ***M*** clients in all, each with its own set of local data. A random proportion of **C** of clients is chosen at the start of each round, and the server broadcasts the current global model state to each of these clients [29]. For efficiency, we only select a subset of clients as adding more clients after certain point results in diminishing returns. Then, each selected client does the local calculation and sends an update to the clouds based on the global degree and its local data set. The changes are subsequently applied to the server's global state, and the cycle begins again.

While we concentrate on the non-convex cost function of neural networks, the method we use also applies to any finite-sum objective of this type [29].

$$\min_{z\in\mathbb{R}^d} f(z) \quad \text{where} \quad f(z) \overset{def}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(z) \tag{1}$$

Technically, the learning objective for user i is denoted as

$$\min_{\Theta^i} \mathcal{L}_i = \sum_{k=1}^{n^k} \ell(y_i^k, f_i(\mathbf{x}_k^i))$$

where l indicates the loss of the network,$(\mathbf{x}_i, y_i)_{i=1}^k$ are the subset of the data size k, θ denotes the parameter to be learned, i.e., the weight (z) and bias. Deep Neural networks are trained through a mathematical operation sequence that transforms input data into an output prediction. Backpropagation is a process that uses optimization prediction and the true label to compute gradients of the neural network parameters with respect to a loss function. These gradients are then used to update the parameters in the direction of minimizing the loss function, which improves the network's prediction accuracy over time. Thus, As a result, we may rewrite objective (1) as

$$f_i(z) = \sum_{k=1}^{K} \frac{n_k}{n} F_k \quad \text{where} \quad f(z) = \frac{1}{n}\sum_{i=1}^{n} f_i(z)$$

Where n denotes the number of devices involved in training and$z \in \mathbb{R}^d$ denotes the d parameters of a global model (e.g., weights of a neural network) The partition can also be formed by distributing the data uniformly across the device, assuming a set of data i device assigned to a fixed client M, usually the IID assumption. The number of local updates per round for a client with $u_k = E\frac{n_k}{B}$; Pseudo-code is given in Algorithm 1.

Algorithm 1 *K-Dumb FedAvg* [29]: The *M* clients are indexed by *k*; *B* is the local minibatch size, *E* local epochs, learning rate *n*.

Require: Server executes z_0
Ensure: for each round *t* = 1,2,.... do
  $m \leftarrow max(C.K, 1)$
  $S_t \leftarrow$ random set of m clients.
  $k \in S_t$ **in parallel do**
  for each client k.
  $z_{t+1}^k \leftarrow$ Client Update(k, z_t)
  Client Update(k, z_t) : ||*Run on client k*
  $B \leftarrow (split\ P_k\ into\ batches\ of\ size\ B)$
  for each local epoch *i* from 1 to *E do*
    for batch $\in B$ **do**
      $z \leftarrow z - n\nabla l(z; b)$
Ensure: return w to server
  Obtained $z\nabla_{\text{optimized}}$
  Share with New node k_n

The fundamental concept is to use the data aggregation mechanism to extract the knowledge from the global model to share with other devices to reduce energy consumption, improve reliability and to satisfy the low latency requirements.

V. Simulation Results

Our motivation comes from tasks like object detection and image classification, where sophisticated models can considerably improve robotics usability. For each of these tasks, we first selected a small-enough surrogate dataset that allowed us to carefully investigate the *K-Dumb FedAvg* hyperparameters. Finally, we analyze K-Dumb on substantial modeling to demonstrate its performance on a real-world problem with a natural data split between clients. Our preliminary research comprises two model families and one datasets. MNIST data sets are used for handwritten digit recognition problems with our two of family algorithms: 1) A simple Multilayer Perceptron Network (MLP) with two hidden layers of 200 units each, activated by ReLu. 2) A Convolutional Neural Network (CNN) with a (5×5) convolutional layer, followed by 2 pooling layers and a final softmax output layer to obtain the global weight of the training and test data set. The data partition was done in two ways **IID** and **non-IID**.

We are primarily concerned with standard performance, but K-Dumb FedAvg is also effective at optimizing test loss, even when test-set accuracy has plateaued. The below observation for all two models class displays the MNIST CNN and MLP (multi-layer perceptron) communication rounds [29] vs one-shot Inference.

Model	Epoch	IID	Non-IID (equal)
MLP	10	91.25%	75.94%
CNN	10	97.17%	85.97%
MLP(One-shot Inf)	**1**	**89.67%**	**74.89%**
CNN(One-shot Inf)	**1**	**95.47%**	**83.62%**

TABLE I: Comparison between rounds of communication efficiency

According to Table 1 results comparison, if the level of knowledge sharing needs to be improved, the level of access to training in the actual model must first be increased for convergence. Table 1 shows the model's average accuracy in comparison with rounds of communication. Figure 3 shows the average accuracy of One-shot Inference accuracy against communication rounds for the MNIST CNN. In this scenario, a fraction of users were selected to train the MNIST models and it appears to achieve a target accuracy with a set of communication rounds. Figure 4 shows MLP compare (IID & non-IID) with a fraction of users *C*=0.1 and optimizer *n*, presents the One-shot Inference average accuracy vs. communication rounds loss on MNIST dataset.

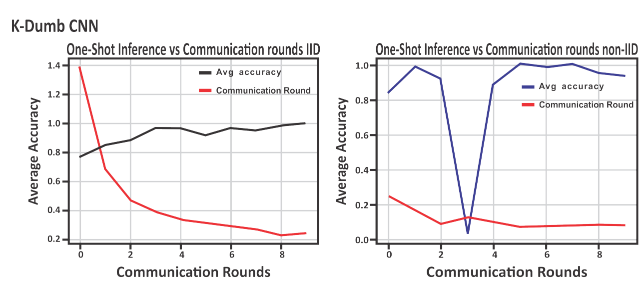

Fig. 3: Avg-acc in CNN against Communication rounds

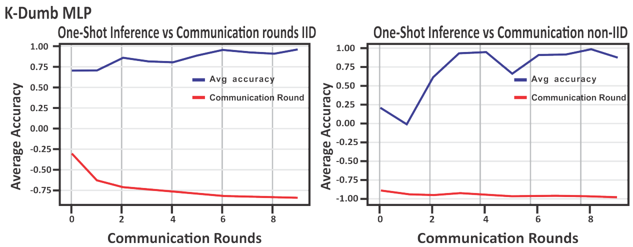

Fig. 4: Avg-acc in MLP against Communication rounds

K-Dumbs FedAvg predicts high-quality results by sharing weights only, preserving privacy and data security with only a single shot of communication rounds, as shown by the results in Fig 5 showing a range of model topologies, our tests illustrate that K-Dumbs FedAvg may be made practical for efficient knowledge sharing in the Internet of Robotic Things (IoT).

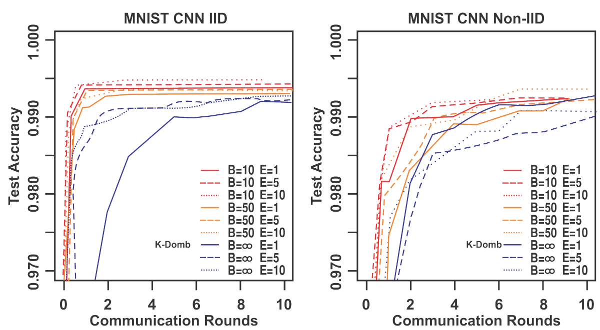

Fig. 5: Test-Accuracy in MNIST CNN of IID and non-IID

While partitioning the MNIST IID data, extra processing per client improves accuracy while decreasing the number of communication rounds required to reach the desired results. The non-IID are less efficient but still significant. For all two model classes, the *K-Dumbs FedAvg* converges to higher accuracy with the AI-enabled O-SI method.

VI. Conclusion

In this paper, we identify the unified weights-sharing model optimization technique in distributed devices through data aggregation on undefined nodes using a FedAvg privacy-preserving methodology. We presented a novel framework of data aggregation that integrates FL and K-Dumbs Average weights without sharing clients' data, or labels, as evidenced by findings on a number of model architectures: a multi-layer perceptron (mlp), convolutional neural network (CNN). K-Dumbs FedAvg provides many practical privacy benefits, giving robust guarantees through weight sharing, reducing computational load, and energy efficiency. Our proposed K-Dumbs FedAvg algorithm improves accuracy over traditional FL methods with a one-shot Inference technique. During the sharing process, the proposed AI-enabled ML algorithms efficiently reduce the communication rounds and achieve desirable knowledge aggregation performance without sacrificing data privacy. The key responsibility of the k-Dumb in robotics is to minimize the communication rounds to offload the computational complexity and share efficient knowledge. Future work to optimize the current model to automate the knowledge sharing with multiple training node selections to further enhance communication efficiency.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC).

References

[1] Rekkas, V.P. Sotiroudis, S. Sarigiannidis, P. Wan, S. Karagiannidis, G.K. Goudos, S.K. Machine Learning in Beyond 5G/6G Networks—State-of-the-Art and Future Trends. Electronics 2021, 10, 2786. https:weightrg/10.3390/electronics10222786

[2] M. Banafa a et al., "6G Mobile Communication Technology: Requirements, Targets, Applications, Challenges, Advantagea s, and Opportunities," Alexandria Engineering Journal, vol. 64, pp. 245AI-enabled3, doi: https://doi.org/10.1016/j.aej.2022.08.017.

[3] K. Davachieve. Berndt, "6G Vision and Requirements: Is There Any Need for Beyond 5G?," in IEEE Vehicular Technology Magazine, vol. 13, no. 3, pp. 72-80, Sept. 2018, doi: 10.1109/MVT.2018.2848498.

[4] R. Li, "Network 2030: Market drivers and prospects" In Proc. 1st International Telecommunication Union Workshop on Network 2030, Oct. 2018

[5] J. Liu, et al. "Online multi-workflow scheduling under uncertain task execution time in IaaS clouds," IEEE Trans. Cloud Comput., to be published,doi: 10.1109/TCC.2019.2906300

[6] M. Latva-aho, "Radio access networking challenges towards 2030", 1st International Telecommunication Union Workshop on Network 2030, Oct. 2018

[7] Padhi, P.K.; Charrua-Santos, F. 6G Enabled Industrial Internet of Everything: Towards a Theoretical Framework. Appl. Syst. Innov. 2021, 4, 11. https://doi.org/10.3390/asi4010011

[8] Gu, Renjie, Chaoyue Niu, Fan Wu, Guihai Chen, Chun Hu, Chengfei Lyu, and Zhihua Wu. 2021. "From Server-Based to Client-Based Machine Learning." ACM Computing Surveys 54 (1): 1–36. https://doi.org/10.1145/3424660.

[9] Kai Hu, Yaogen Li, Min Xia, Jiasheng Wu, Meixia Lu, Shuai Zhang, Liguo Weng, "Federated Learning: A Distributed Shared Machine Learning Method", Complexity, vol. 2021, Article ID 8261663, 20 pages, 2021. https://doi.org/10.1155/2021/8261663

[10] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor and S. Cui, "A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks," in IEEE Transactions on Wireless Communications, vol. 20, no. 1, pp. 269-283, Jan. 2021, doi: 10.1109/TWC.2020.3024629.

[11] Vermesan, O. et al. (2020) Internet of robotic things intelligent connectivity and platforms, Frontiers. Frontiers. Available at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00104/full (Accessed: March 8, 2023).

[12] Federated Learning: Collaborative Machine Learning Without Centralized Training Data. Accessed: Apr. 2017. [Online]. Available: http://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[13] Vermesan, O., Bacquet, J. (Eds). (2017). Cognitive Hyperconnected Digital Transformation: Internet of Things Intelligence Evolution (1st ed). River Publishers. https://doi.org/10.1201/9781003337584.

[14] W. L. et al., Federated learning in mobile edge networks: A comprehensive survey

[15] R. C. Luo, K. L. Su, S. H. Shen and K. H. Tsai, "Networked intelligent robots through the Internet: issues and opportunities," in Proceedings of the IEEE, vol. 91, no. 3, pp. 371-382, March 2003, doi: 10.1109/JPROC.2003.809198.

[16] Ohad Shamir and Nathan Srebro. Distributed stochastic optimization and learning. In Communication, Control, and Computing (Allerton), 2014.

[17] Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I Jordan, Peter Richtarik, and Martin Tak ´ a´c. Adding vs. ˘ averaging in distributed primal-dual optimization. In ICML, 2015.

[18] Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication efficient distributed optimization using an approximate newton-type method. arXiv preprint arXiv:1312.7853, 2013.

[19] Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In Advances in Neural Information Processing Systems,2013.

[20] Yuchen Zhang and Lin Xiao. Communication-efficient distributed optimization of self-concordant empirical loss.arXiv preprint arXiv:1501.00263, 2015.

[21] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. *In Advances in Neural Information Processing Systems, 2013.*

[22] Ahmed, Manzoor, et al. "A survey on vehicular task offloading: Classification, issues, and challenges." J. King Saud Univ. Comput. Inf. Sci 34 (2022): 4135-4162.

[23] Zhang, L. Dai, X. Li, Y. Liu, and L. Hanzo, "On low-resolution ADCs in practical 5G millimeter-wave massive MIMO systems," IEEE Commun. Mag., vol. 56, no. 7, pp. 205–211, Jul. 2018.

[24] K. Xiong, S. Leng, J. Hu, X. Chen, and K. Yang, "Smart network slicing for vehicular fog-RANs," IEEE Trans. Veh. Technol., vol. 68, no. 4, pp. 3075–3085, Apr. 2019.

[25] J. Zhang, L. Dai, S. Sun, and Z. Wang, "On the spectral efficiency of massive MIMO systems with low-resolution ADCs," IEEE Commun. Lett., vol. 20, no. 5, pp. 842–845, May 2016.

[26] X. Zhang, K.-K.-R. Choo, and N. L. Beebe, "How do i share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," IEEE Internet Things J., vol. 6, no. 4, pp. 6850–6861, Aug. 2019.

[27] B. Fan, S. Leng, and K. Yang, "A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks," IEEE Netw., vol. 30, no. 1, pp. 6–10, Jan./Feb. 2016.

[28] B. Ai et al., "On indoor millimeter wave massive MIMO channels: Measurement and simulation," IEEE J. Sel. Areas Commun., vol. 35, no. 7, pp. 1678–1690, Jul. 2017.

[29] McMahan, H. B. et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data." International Conference on Artificial Intelligence and Statistics (2016).

[30] Zhang, Q., Lu, J. Jin, Y. Artificial intelligence in recommender systems. Complex Intell. Syst. 7, 439–457 (2021). https://doi.org/10.1007/s40747-020-00212-

[31] Kumar, S., Tiwari, P. Zymbler, M. Internet of Things is a revolutionary approach for future technology enhancement: a review. J Big Data 6, 111 (2019). https://doi.org/10.1186/s40537-019-0268-2

[32] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[33] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. ACM Trans. Intell. Syst. Technol. 10, 2, Article 12 (March 2019), 19 pages. https://doi.org/10.1145/3298981