Towards Efficient Decentralized Federated Learning: A Survey

Saqr Khalil Saeed Thabet[1], Behnaz Soltani[1], Yipeng Zhou[1], Quan Z. Sheng[1], and Shiting Wen[2(✉)]

[1] Macquarie University, Sydney, NSW 2109, Australia
{saqr.thabet,behnaz.soltani,yipeng.zhou,michael.sheng}@mq.edu.au
[2] Ningbo Tech University, Ningbo, Zhejiang 315211, People's Republic of China
wensht@nbt.edu.cn

Abstract. Federated Learning (FL) is a distributed machine learning technique that has been increasingly adopted across many applications due to its capability to disseminate clients' local knowledge while preserving their privacy. Systems that rely on Centralized Federated Learning (CFL) require a central entity to aggregate a global model. However, this centralized structure can result in higher latency and increased vulnerability to attacks or failures. Decentralized Federated Learning (DFL), which relies on direct communication between clients to train models collaboratively, has emerged as an efficient alternative to CFL by avoiding dependence on a central server. In this study, we conduct a comprehensive survey of various approaches proposed to optimize the performance and efficiency of DFL in terms of memory, communication, and computation, and address divergent datasets among clients. First, we introduce the DFL framework and highlight the pertinent challenges. Then, we explore the existing methods and categorize them based on their mechanisms to address system heterogeneity and data heterogeneity in DFL. Finally, we highlight some application scenarios of DFL.

Keywords: Decentralized Federated Learning · Communication Efficiency · Computation Efficiency · Memory Efficiency · Data Heterogeneity

1 Introduction

Recently, mobile devices such as smartphones, smart IoT devices, and vehicles can generate vast amounts of data that need to be processed. Machine learning is increasingly utilized to handle this data to achieve smart mobile applications. However, transmitting clients' private data to a central entity (e.g., edge server) for training can result in privacy risk, cause long communication latency, and

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020021, and the Ningbo Science and Technology Special Projects under Grant Nos. 2022Z235 and 2022Z095.

Q. Z. Sheng et al. (Eds.): ADMA 2024, LNAI 15388, pp. 208–222, 2025.
https://doi.org/10.1007/978-981-96-0814-0_14

exhaust available resources [1]. Federated Learning (FL), which is an effective framework for training machine learning models in distributed systems, was suggested by Google, allowing clients to cooperatively train local models without sharing their data [2]. The privacy-preserving attribute is assured as each client trains its local model on-device using its own dataset. Then, these local model parameters are transferred to the FL server for aggregation. The FL server aggregates the local models to generate a global model, which is sent back to the clients for further training on their local data. These steps are repeated for several iterations, leading to model convergence. In traditional Centralized FL (CFL), all clients need to communicate with a central server during the model training, leading to several concerns, such as a single point of failure and communication congestion, which limit the system's scalability. Hence, Decentralized FL (DFL) is proposed to mitigate these disadvantages by distributing communication and computation burdens more evenly across clients, leading to more resilient and efficient training [3]. In DFL, clients do not rely on a central server, as local models are exchanged directly between clients in a peer-to-peer fashion [4]. Under the DFL settings, each client develops a different aggregated model, influenced primarily by the diversity of the neighboring clients.

In this work, we aim to discuss the factors that influence the efficiency of DFL and review various proposed mechanisms. The main contributions of this study are as follows:

- We discuss the major challenges influencing the DFL architecture.
- We review the state-of-the-art memory, communication, and computation-efficient methods used in DFL by categorizing them according to their different approaches. Then, we identify the different methods used to overcome data heterogeneity in DFL.
- We explore different applications of DFL in the field of recommender systems, connected and automated vehicles, and healthcare.

2 Decentralized Federated Learning

2.1 DFL Systems

In essence, the process flow in DFL can be described as follows:

1. *Initialization* – Each client determines the FL task and defines the network topology (communication links between clients).
2. *Local Training* – Before each communication round, clients perform multiple local iterations to update their local model using their local data samples.
3. *Participant Selection* – Every local client selects n number of neighboring clients to participate in the training process.
4. *Local Model Exchange* – At each communication round, clients exchange their local models with their selected participants.
5. *Consensus (Aggregation)* – Every client aggregates the received model updates, integrating other clients' knowledge into its local model in order to achieve a consensus on the global update.

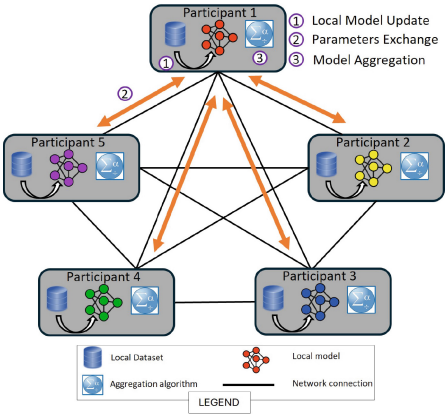

Fig. 1. An overview of decentralized federated learning.

Steps (2)–(5) repeat until the system converges, as shown in Fig. 1. FedAvg [2] is a widely employed aggregation algorithm, which is commonly optimized to fit decentralized scenarios. Decentralized Stochastic Gradient Descent (D-SGD) [5] applies SGD to the DFL network, allowing decentralized FL communication to be asynchronous, local, and time-varying. Each client first generates and updates their models locally using SGD. Then, these models are exchanged with neighboring nodes and averaged. However, various data distributions across clients, different network structures, and uneven computation, memory, and communication capabilities can degrade the performance of the DFL process.

2.2 Related Work

This section summarizes key survey works on DFL. The study in [6] details the DFL frameworks most commonly used. The suggested solutions in the study focus on blockchain-related frameworks and participant reward methods to tackle privacy and security issues in DFL. In [7], the authors conduct a comprehensive survey investigating optimized DFL models and algorithms based on network topology solutions. The work reviews various network topologies with different participants and optimization solutions to tackle challenges such as system and data heterogeneity, privacy, and communication efficiency. In [8], different approaches for federated evaluation are examined, given the absence of a centralized entity. The DFL framework is discussed in a survey [9], which compares many aspects of CFL and DFL. The work categorizes DFL based on iteration order, communication protocol, network topology, paradigm proposal, and temporal variability. However, the DFL challenges were not the main focus of the work.

2.3 Challenges in Decentralized Federated Learning

There are several challenges in DFL that limit the performance of the training process:

- *System Heterogeneity*: Clients have different computation, memory, and communication capabilities, which can influence the performance of the training process. Communication among clients may occur over various networks, from LANs to WANs, including core networks and cellular networks, resulting in limited available bandwidth. Many factors can impact the transmission bandwidth between clients. For example, when clients are wirelessly connected to the network, link instability and client mobility cause the wireless link conditions to fluctuate over time [10], leading to varying transmission rates for communication routes between clients [11]. Participating clients in DFL are typically devices with limited memory. They are expected to store not only local datasets but also a large neural network deployed across the distributed system. The computational capacity of these devices must be able to handle large processes while maintaining standard accuracy. For instance, ResNet-50, which has 50 convolutional layers, requires more than 100 MB of memory and more than 3.8 billion floating-point multiplications when processing a single image.
- *Data Heterogeneity*: Since datasets at each client are generated independently based on the client's behavior, function, and location, data distributions are usually different. Datasets across clients are non-Independent and Identically Distributed (non-IID). Thus, local datasets may not accurately represent the population distribution, affecting the convergence rate of model training [12].

Since DFL was first introduced, there have been many attempts to optimize the system, especially to handle clients' data heterogeneity, reduce communication overhead, and preserve available resources. Optimization of the DFL system is addressed from different angles, focusing on leveraging different aspects of DFL resources, such as memory, communication, and computation, and considering the heterogeneous nature of data distribution across clients. Figure 2 shows a relation map between key mechanisms, methodologies, and solution approaches for various proposed frameworks. In the figure, we briefly describe different types of underlying mechanisms and methodologies that contribute to improving DFL performance. Additionally, we overview the associated solution approaches. It is worth mentioning that several mechanisms can be categorized in multiple ways.

3 System Heterogeneity

3.1 Memory Efficient Techniques

Compression Methods. The nature of training highly over-parameterized models in distributed FL necessarily requires strong computation and communication overhead. It is evident that over the past years, improvements in computation have overcome those of network bandwidth [13]. Hence, distributed

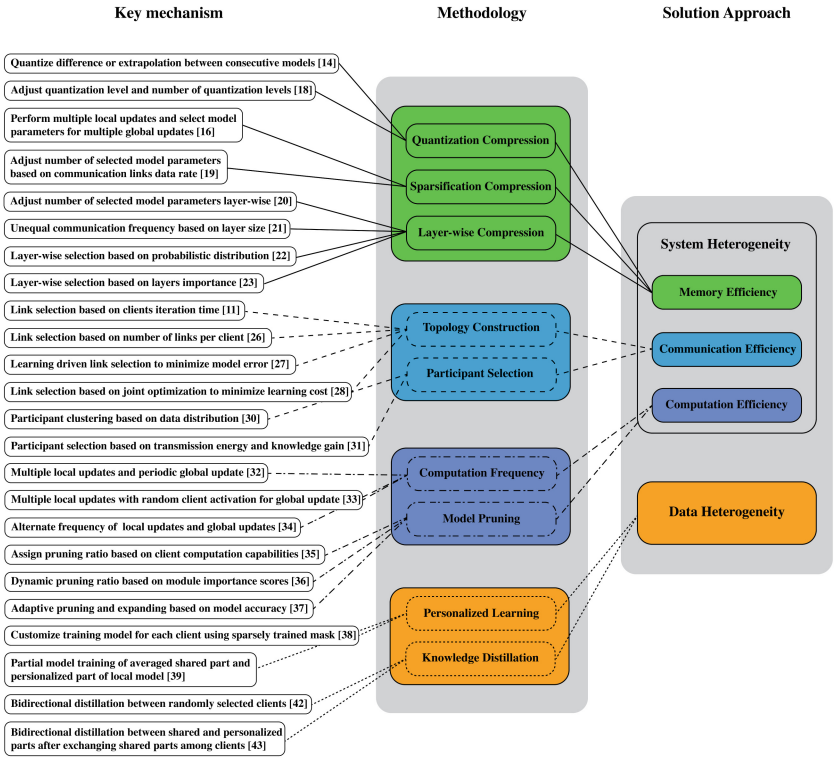

Fig. 2. Overview of the existing works for leveraging DFL framework efficiency.

learning systems often lack the adequate communication capability to handle distributed training. Compression plays a crucial role in reducing communication traffic. The most common compression techniques are Quantization [14], which lowers the precision of data presentation, and Sparsification [15], which first evaluates data importance before dropping unimportant data from the traffic. Several researchers have proposed studies pertinent to utilizing compression to reduce overhead and speed up the convergence rate in DFL.

Model/Gradient Compression. It is usually a straightforward approach to reduce communication traffic of the model training in DFL by compressing the model or the gradient exchanged between clients. In [14], two quantized algorithms are designed for DFL. Under this framework, nodes exchange quantized gradients, while unquantized models are employed for local updates. Namely, extrapolation and difference compression: the former quantizes extrapolation between two consecutive local models, and the latter quantizes the difference between two successive local models. Performing multiple local updates in a round makes the descent direction locally optimal but not globally due to local drift or inferior model consensus. Thus, DFL with Compressed Communication

(C-DFL) [16] aims to improve model consensus by considering both multiple local updates and multiple inter-node communication. The resulting communication overhead is addressed by implementing sparsification and randomized gossip compression schemes introduced in [17]. In [18], the authors design a doubly adaptive DFL framework that considers matching the time-varying convergence rate by jointly adjusting quantization levels and a number of quantization levels. The former allows the quantization to accommodate the dynamic distribution of the exchanged model parameters, and the latter minimizes communicated data during the training. The framework adopts Lloyd-Max quantizer (LM-DFL) to mitigate the quantization distortion. If a fixed compression ratio is assumed while exchanging models among clients with different transmission links, it can potentially result in transmission time lags. In AdapCom-DFL [19], the communication latency constraint adaptively adjusts the sparsification compression ratio for every device based on the data rate of the communication links. Subsequently, the authors formulate a regularized Fastest Mixing Markov-Chain (FMMC) problem aiming to prune communication links with low data rates. In the optimized network topology, more power is allocated to selected links to improve their compression ratio.

Layer-Wise Compression. Instead of applying compression techniques to the entire model, an alternative approach involves applying compression in a layer-wise manner. A Layer-Wise Adaptive Gradient Sparsification (LAGS-SGD) has been proposed in [20], where each client independently performs sparsification on each layer with an adaptive ratio to balance the communication size of each layer with the convergence speed. Different layers exert different impacts during the model training process, and disregarding their importance when implementing compression may result in extended convergence time. Layers with a large number of parameters generally encode the information in a redundant manner and, therefore, can be highly compressed or shared less frequently. For example, the authors in Layer-Wise Federated Group ADMM (L-FGADMM) [21] consider applying an unequal communication frequency on each layer, where the largest layers are communicated less frequently than the other layers. In Layer-based Random SGD (LR-SGD) [22], a layer-based random sparsification method is proposed, as layers are not compressed but rather randomly selected to be exchanged with other clients. The selection is determined based on two different probabilistic distributions; in the first, all layers have an equal probability of being selected, while in the second, the first and last layers are more likely to be chosen compared to the middle layers. A layer selection optimizer is used in [23] to select the most informative layers, those expected to contribute significantly to the global model quality. Layers are sorted based on the squared norm of their gradients with respect to the trainable parameters characterizing each layer. Given that this metric remains static for several rounds, some layers' parameters might be rarely exchanged, which may affect the convergence process. Thus, a randomized layer selection policy is integrated to ensure that clients exchange a fair share of neighboring model layers during training rounds.

3.2 Communication Efficient Techniques

Communication schemes determine how different clients in the overlay network behave when transmitting, receiving, and aggregating parameters from their neighbors. DFL can accommodate various communication modes, including synchronous, asynchronous, or semi-synchronous modes. Each mode is effective given suitable conditions and considerations under specific communication scenarios, such as computational and communication cost, convergence speed, and accuracy. In principle, the DFL communication mechanism leverages peer-to-peer (P2P) communication to exchange local model parameters between clients [24]. In that perspective, communication topologies such as ring, star, and mesh are fundamental to P2P DFL, as they dictate the network's communication protocol and knowledge exchange process. Gossip communication [25] is a P2P transmission method that relies on P2P sampling, which provides every client with a set of neighbors (called a view) to gossip with. This set of neighbors periodically changes to give clients a chance to interact with new neighbors within the network. Communication schemes implemented in DFL networks need to manage a dynamic and heterogeneous topology, with participants frequently altering their location or role within the network. Thus, adapting to topology changes and effectively selecting participants is essential.

Topology Construction. Under the decentralized learning architecture, aggregation occurs along the topology links, affecting the training efficiency. Communication between clients based on a fixed topology conflicts with dynamic network conditions and bandwidth reallocation. In [11], high-speed links and high-bandwidth clients are selected to form the network topology. The decision is made based on the iteration time of each client, which is collected periodically and reflects the link speed between clients. In [26], the authors propose minimizing the total power consumption by optimizing the link selection under the given network topology. Based on the user-defined cardinality (number of links) ratio, the algorithm selects and augments link sets to satisfy the link cardinality ratio. Then, a minimum-cost spanning tree of the network is formed to ensure connectivity and calculate power consumption; eventually, redundant links are deleted. Deep reinforcement learning (DRL) is exploited in [27] to develop a learning-driven link selection method. The DRL process depends on minimizing the model error among mobile devices under the energy consumption and bandwidth constraints of each iteration. The authors in [28] suggest a joint optimization for computing power, wireless resource allocation, link selection, and aggregation weight. The joint optimization minimizes the total learning cost, which is defined as the weighted sum of energy consumption and learning latency. Within a specific network topology, the computing power and wireless resource allocation are optimized through alternating optimization, and the aggregation weight is determined using semidefinite optimization. Based on the network scale, two approaches are proposed for solving link selection: (i) a global search algorithm is designed for small-scale networks, and (ii) a tabu search-based meta-heuristic algorithm is employed for large-scale networks.

Participant Selection. In FL, participant selection can heavily impact the training time and final model accuracy. Clients' contributions to training performance vary due to differences in computational and communication resources, along with heterogeneous datasets. In DFL, participant selection aims to improve the generalization performance of each client's local model. Hence, every client must exhaustively share its local models with all connected neighbors despite the induced latency of such communication burden [29]. In [30], a participant clustering approach is adopted. First, the sparsity and density of the data distribution of each client are represented with the help of the Mergeable Counting Bloom Filter (MergeCBF), employing locally sensitive hash functions. Subsequently, the resulting bit arrays are transmitted to nearby clients, where each client performs bitwise OR operations to measure data complementarity. Lastly, clients with complementary data are clustered together, leading to groups of clients, each with uniform data distribution. Based on the working nature of mobile devices, some clients may have lower-performing models. This can adversely impact federated averaging in neighboring clients with high-performing models. On the other hand, the high-performing model can incredibly improve lower-performing models' efficiency. In that regard, Opportunistic Communication Efficient DFL (OCD-FL) [31] introduces a knowledge gain measure to recognize peers with low and high-performing models. The designed peer selection problem aims to efficiently select neighbors for collaboration. It considers both the amount of energy required for transmission and the knowledge gained as a result of the collaboration.

3.3 Computation Efficient Techniques

In CFL, the central server is responsible for executing most of the exhaustive computations necessary for the system to operate effectively. These include coordinating all participants and aggregating the global model. In DFL, all the different computations are transferred back to the clients. Thus, it is essential to ensure computation efficiency for the implementation of DFL.

Computation Frequency. Improving the updating scheme can improve the computation efficiency in DFL. In [32], the Cooperative SGD considers periodically exchanging model parameters after a certain number of local updates. In MATCHA [33], the authors extend the previous work to improve convergence speed. Given that the network topology of the proposed framework is first decomposed into different sets of clients, the work proposes to randomly select different sets of neighbors to exchange models at each communication round. In Local Decentralized SGD (LD-SGD) [34], the authors investigate two arbitrary update schemes to alternate the frequency of local and global updates. The first scheme adds multiple global updates, leading to better convergence while increasing communication time. The second scheme decays the interval of local updates gradually, reducing computation but affecting convergence speed. Similar to [33], this work also adopts randomly activating a small group of devices for the global updates step at each round.

Model Pruning. Discarding redundant weights (e.g., filters, neurons, and connecting feature maps) from the local training model contributes to reducing inference computation. Model Pruning and Topology Construction (MOTOR) [35] proposes applying different pruning ratios according to the client's computation capabilities. This approach allows clients to train models of different sizes and structures. Both pruning ratio decision and topology construction are jointly optimized using the consensus speed metric to update the network topology and adapt the pruning ratio accordingly. In the Dynamic Aggregation Decentralized Personalized FL (DA-DPFL) [36], a dynamic pruning policy is implemented to update the pruning ratio at each communication round. The pruning-based (masked) PFL framework removes or rescales model masks depending on the importance scores, which are computed from the magnitude of model weights and gradients. Adaptive model Pruning-Expanding DFL (FedPE) [37] suggests performing one-shot pruning on local data to address clients' limited computation capacity. The initially pruned models are further pruned or expanded according to the accuracy variations of the local models to ensure model convergence.

4 Data Heterogeneity

When different clients have different data distributions, the local model parameters at different clients converge to different optimum points, diverging from the global optimum. Therefore, mitigating the challenges posed by data heterogeneity is crucial to ensure effective convergence. However, in DFL, there is no central server coordinating clients' efforts to develop global information, and thus, alleviating this challenge for DFL requires new approaches.

Personalized Federated Learning (PFL). Creating personalized models for each client makes the consensus model more robust to address the data distribution variance across clients. When applying PFL on DFL, full model aggregation with neighboring clients causes inferior representation ability. This results from the loss of unique information within each client. In Dis-PFL [38], each client has its own customized model. This model is pruned by a sparsely trained mask in order to improve client adaptability to their local data and reduce communication costs. The suggested decentralized sparse training indicates that during the entire local training and P2P communication process, each local model only retains a fixed set of active parameters. In DFedAlt [39], partial personalization is explored under the DFL setting, where each client's local model is decomposed into two components: a shared part and a personalized part. While the personalized part for each client is not shared, the shared part is averaged with those of neighboring clients. The adopted partial model training alternatively updates the parameters of the two parts in a decentralized manner, resulting in the development of partially personalized models.

Knowledge Distillation. Correcting the deviation of various local models from the global model in terms of parameters can efficiently mitigate the model drift problem caused by data heterogeneity. Knowledge distillation transfers the feature presentation learned by a more complex model, referred to as the teacher model, to a simpler model, known as the student model [40]. This knowledge transfer process ensures that the features or predicted outputs of the student model are aligned with those of the teacher model. Traditionally, knowledge distillation involves unidirectional learning from the teacher model to the student model. However, in [41], a bidirectional or Mutual Knowledge Transfer (MKT) between different sets of student models is proposed. This method improves learning performance for each student model compared to conventional separate learning. In [42], the authors propose DFL via MKT (Def-KT), where in each round, two steps are implemented consecutively: model updating and model fusion. First, each client updates its model by local model training, and then this fine-tuned model is sent to another randomly selected client. Next, these clients fuse the received models and their local models using MKT instead of averaging them. This enables the two models, which inherit different knowledge, to learn from each other. In FedDLM [43], each client creates two models: a private (teacher) model and a public (student) model. As private models maintain their parameters locally, public models from different clients exchange parameters and share knowledge. Then, the private and public models of each client simultaneously transfer each other's knowledge via bidirectional distillation, leading to a private model that is well adapted to the user's local data.

5 Applications of Decentralized Federated Learning

5.1 Recommender Systems

In today's digitized world, the abundance of information often makes it challenging to distinguish relevant data from irrelevant ones. Thus, recommender systems have become indispensable in the era of information overload, offering users assistance in making decisions by providing personalized suggestions. However, their accuracy relies on access to extensive user data (e.g., social information, location logs, ratings, and clicks). This reliance raises significant privacy and security concerns, further compounded by the inherent heterogeneity of the data distribution among users. Therefore, DFL addresses these issues by enabling models to be trained locally and remain private. In [44], a DFL framework is applied to perform recommendation system tasks, utilizing gossip communication to exchange subsampled (compressed) module updates between neighboring clients. During model aggregation, the model's age factor is considered, giving more weight to models that have processed more data. In PEPPER [45], the recommender system framework relies on asynchronous gossip communication, allowing clients to train and exchange models asynchronously. When the received models start to improve the local training, these good neighbors are remembered to form personalized peer-sampling, which periodically updates the node's views.

Similarly, the received model quality is used in performance-based aggregation, where clients prioritize models that perform well on their data.

5.2 Connected and Automated Vehicles

Applying machine learning for key tasks in Connected and Automated Vehicles (CAV) results in generating several Terabits of raw data per day, per vehicle. The data sources can range from vehicle identifiers and component conditions to perception sensor measurements and vehicle-to-everything (V2X) communications. As vehicle automation increases, exchanging large portions of data becomes impractical. This is due to the stringent requirements for full self-driving scenarios, such as low latency and high accuracy. DFL holds the potential to meet these requirements by sharing only model parameters to achieve consensus among vehicles. Furthermore, DFL provides scalability by supporting numerous vehicles without the need for a central entity. As part of the continual road mapping, the readings of Lidar sensors are fed into a Deep Learning (DL) model (PointNet) to identify the actual types of road actors. However, the volume of exchanged information is substantial, resulting in inevitable communication latency. A modular DFL is proposed in [46] to implement PointNet-compliant architecture on Lidar sensor input. This modular approach aims to reduce communication overhead by enabling collaborative learning between neighboring clients over a specific range of model layers. Naturalistic Driving Action Recognition (NDAR) helps detect driving behavior, thus reducing accident risks. NDAR models are constructed based on in-cabin camera feeds, requiring the use of various clients' feeds to train accurate models, which raises privacy concerns. FedPC [47] introduces a P2P DFL framework that utilizes gossip communication to exchange models between clients to deliver precise and personalized NDAR models. This approach also mitigates the security and privacy risks associated with traditional model aggregation. Additionally, continual learning is incorporated to retrain the previous client model, treating it as the initial model for the current client.

5.3 Healthcare

With technological advancement, the healthcare industry has evolved to integrate a growing number of Internet-connected devices (e.g., IoT) and to generate more records. Simultaneously, cooperation between hospitals and research centers is crucial in modern healthcare systems to tackle the challenges posed by limited healthcare data samples, especially for machine learning applications. However, preserving patient privacy and data integrity is an essential standard that can impede transferring patient records between different parties or devices. Therefore, it is critical to adopt technologies that support decentralized learning. DFL, for instance, ensures knowledge sharing between different parties or devices while protecting users' privacy. In [48], the authors design an IoT-based decentralized architecture to train on skin images for detecting skin diseases without compromising users' data. The framework utilizes transfer learning to address the lack of extensive labeled data. Additionally, an automated data acquisition

process is proposed to generate more training samples, relying on model prediction ratings for user-captured images. Robust and Privacy-preserving Decentralized Deep Federated Learning (RPDFL) [49] is a DFL framework with a cross-silo architecture designed to address data diversity among different healthcare organizations. The authors suggest organizing healthcare organizations in a ring network structure. The communication necessary to update the global model is performed using Ring-Allreduced for efficient bandwidth usage. To protect the exchanged models, a secret sharing protocol is used, and an edge-dropping technique is adopted to overcome struggling nodes.

6 Conclusion

Decentralized federated learning (DFL) enables direct communication between clients to collaboratively train models, eliminating the need for a central server. However, several challenges can adversely impact training performance in the DFL environment. In order to overcome the challenges related to system heterogeneity and data heterogeneity, various optimization methods have been proposed. In this work, we review the existing studies and discuss different approaches to address the aforementioned challenges. Finally, we identify several applications of DFL, including recommender systems, connected and automated vehicles, and healthcare.

References

1. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., Rellermeyer, J.S.: A survey on distributed machine learning. ACM Comput. Surv. (CSUR) **53**(2) (2020)
2. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR (2016)
3. Liu, Q., et al.: Asynchronous decentralized federated learning for collaborative fault diagnosis of PV stations. IEEE Trans. Netw. Sci. Eng. **9**(3), 1680–1696 (2022)
4. Beltrán, E.T.M., et al.: Decentralized federated learning: fundamentals, state of the art, frameworks, trends, and challenges. IEEE Commun. Surv. Tutorials **25**, 2983–3013 (2022)
5. Nedic, A., Ozdaglar, A.: Distributed subgradient methods for multi-agent optimization. IEEE Trans. Autom. Control **54**(1), 48–61 (2009)
6. Witt, L., Heyer, M., Toyoda, K., Samek, W., Li, D.: Decentral and incentivized federated learning frameworks: a systematic literature review. IEEE Internet Things J. **10**(4), 3642–3663 (2023)
7. Jiajun, W., Dong, F., Leung, H., Zhu, Z., Zhou, J., Drew, S.: Topology-aware federated learning in edge computing: a comprehensive survey. ACM Comput. Surv. **56**(10), 1–41 (2024)
8. Soltani, B., Zhou, Y., Haghighi, V., Lui, J.C.S.: A survey of federated evaluation in federated learning. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI 2023, pp. 6769–6777 (2023)

9. Yuan, L., Wang, Z., Sun, L., Philip, S.Y., Brinton, C.G.: Decentralized federated learning: a survey and perspective. IEEE Internet Things J. (2024)
10. Zhou, Y., et al.: The role of communication time in the convergence of federated edge learning. IEEE Trans. Veh. Technol. **71**(3), 3241–3254 (2022)
11. Zhou, P., Lin, Q., Loghin, D., Ooi, B.C., Wu, Y., Yu, H.: Communication-efficient decentralized machine learning over heterogeneous networks. In: Proceedings of the 37th International Conference on Data Engineering (ICDE), pp. 384–395. IEEE (2021)
12. Soltani, B., Haghighi, V., Mahmood, A., Sheng, Q.Z., Yao, L.: A survey on participant selection for federated learning in mobile networks. In: Proceedings of the 17th ACM Workshop on Mobility in the Evolving Internet Architecture, pp. 19–24. ACM (2022)
13. Luo, L., Nelson, J., Ceze, L., Phanishayee, A., Krishnamurthy, A.: Parameter hub: a rack-scale parameter server for distributed deep neural network training. In: Proceedings of the ACM Symposium on Cloud Computing, SoCC'18, pp. 41–54. ACM (2018)
14. Tang, H., Gan, S., Zhang, C., Zhang, T., Liu, J.: Communication compression for decentralized training. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, pp. 7663–7673. Curran Associates Inc. (2018)
15. Stich, S.U., Cordonnier, J.-B., Jaggi, M.: Sparsified SGD with memory. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, pp. 4452–4463. Curran Associates Inc. (2018)
16. Liu, W., Chen, L., Zhang, W.: Decentralized federated learning: balancing communication and computing costs. IEEE Trans. Signal Inf. Process. Over Netw. **8**, 131–143 (2022)
17. Koloskova, A., Stich, S., Jaggi, M.: Decentralized stochastic optimization and gossip algorithms with compressed communication. In: Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 3478–3487. PMLR (2019)
18. Chen, L., Liu, W., Chen, Y., Wang, W.: Communication-efficient design for quantized decentralized federated learning. IEEE Trans. Signal Process. **72**, 1175–1188 (2024)
19. Mengxuan, D., Zheng, H., Gao, M., Feng, X.: Adaptive decentralized federated learning in resource-constrained IoT networks. IEEE Internet Things J. **11**, 10739–10753 (2024)
20. Shi, S., Tang, Z., Wang, Q., Zhao, K., Chu, X.: Layer-wise adaptive gradient sparsification for distributed deep learning with convergence guarantees. arXiv:1911.08727 (2019)
21. Elgabli, A., Park, J., Ahmed, S., Bennis, M.: L-FGADMM: layer-wise federated group ADMM for communication efficient decentralized deep learning. In: Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE (2020)
22. Zhang, Z., Hu, Y., Ye, Q.: LR-SGD: layer-based random SGD for distributed deep learning. In: Proceedings of the 8th International Conference on Computing and Artificial Intelligence, pp. 6–11. ACM (2022)
23. Barbieri, L., Savazzi, S., Nicoli, M.: A layer selection optimizer for communication-efficient decentralized federated deep learning. IEEE Access **11**, 22155–22173 (2023)
24. Chen, Z., Weixian Liao, P., Tian, Q.W., Wei, Yu.: A fairness-aware peer-to-peer decentralized learning framework with heterogeneous devices. Future Internet **14**(5), 138 (2022)

25. Hu, C., Jiang, J., Wang, Z.: Decentralized federated learning: a segmented gossip approach. t arXiv:1908.07782 (2019)
26. Kuo, J.-J., Ching, C.-W., Huang, H.-S., Liu, Y.-C.: Energy-efficient topology construction via power allocation for decentralized learning via smart devices with edge computing. IEEE Trans. Green Commun. Netw. **5**(4), 1806–1819 (2021)
27. Hongli, X., Chen, M., Meng, Z., Yang, X., Wang, L., Qiao, C.: Decentralized machine learning through experience-driven method in edge networks. IEEE J. Sel. Areas Commun. **40**(2), 515–531 (2022)
28. Liu, S., Guanding, Yu., Wen, D., Chen, X., Bennis, M., Chen, H.: Communication and energy efficient decentralized learning over D2D networks. IEEE Trans. Wirel. Commun. **22**(12), 9549–9563 (2023)
29. Su, D., Zhou, Y., Cui, L.: Boost decentralized federated learning in vehicular networks by diversifying data sources. In: Proceedings of the 30th International Conference on Network Protocols (ICNP), pp. 1–11. IEEE (2022)
30. Liu, S., Liu, Z., Zhiwei, X., Liu, W., Tian, J.: Hierarchical decentralized federated learning framework with adaptive clustering: bloom-filter-based companions choice for learning non-IID data in IoV. Electronics **12**(18), 3811 (2023)
31. Masmoudi, N., Jaafar, W.: OCD-FL: a novel communication-efficient peer selection-based decentralized federated learning. arXiv:2403.04037 (2024)
32. Wang, J., Joshi, G.: Cooperative SGD: a unified framework for the design and analysis of communication-efficient SGD algorithms. arXiv:1808.07576 (2018)
33. Wang, J. Sahu, A.K., Yang, Z., Joshi, G., Kar, S.:. MATCHA: speeding up decentralized SGD via matching decomposition sampling. In: Proceedings of the 6th Indian Control Conference (ICC), pp. 299–300. IEEE (2019)
34. Li, X., Yang, W., Wang, S., Zhang, Z.: Communication-efficient local decentralized SGD methods. arXiv:1910.09126 (2019)
35. Jiang, Z., Yang, X., Hongli, X., Wang, L., Qiao, C., Huang, L.: Joint model pruning and topology construction for accelerating decentralized machine learning. IEEE Trans. Parallel Distrib. Syst. **34**(10), 2827–2842 (2023)
36. Long, Q., Wang, Q., Anagnostopoulos, C., Bi, D.: Decentralized personalized federated learning based on a conditional sparse-to-sparser scheme. arXiv:2404.15943 (2024)
37. Yi, L., Shi, X., Wang, N., Zhang, J., Wang, G., Liu, X.: FedPE: adaptive model pruning-expanding for federated learning on mobile devices. IEEE Trans. Mobile Comput. 1–18 (2024)
38. Dai, R., Shen, L., He, F., Tian, X., Tao, D.: DisPFL: towards communication-efficient personalized federated learning via decentralized sparse training. In: Proceedings of the International Conference on Machine Learning, pp. 4587–4604. PMLR (2022)
39. Shi, Y., et al.: Towards more suitable personalization in federated learning via decentralized partial model training. arXiv:2305.15157 (2023)
40. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
41. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4320–4328. IEEE (2018)
42. Li, C., Li, G., Varshney, P.K.: Decentralized federated learning via mutual knowledge transfer. IEEE Internet Things J. **9**(2), 1136–1147 (2021)
43. Huang, Y., Kong, L., Li, Q., Zhang, B.: Decentralized federated learning via mutual knowledge distillation. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), pp. 342–347. IEEE (2023)

44. Hegedűs, I., Danner, G., Jelasity, M.: Decentralized recommendation based on matrix factorization: a comparison of gossip and federated learning. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 317–332. Springer (2019)
45. Belal, Y., Bellet, A., Mokhtar, S.B., Nitu, V.: Pepper: empowering user-centric recommender systems over gossip learning. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **6**(3), 1–27 (2022)
46. Barbieri, L., Savazzi, S., Nicoli, M.: Decentralized federated learning for road user classification in enhanced V2X networks. In: Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6. IEEE (2021)
47. Yuan, L., Ma, Y., Su, L., Wang, Z.: Peer-to-peer federated continual learning for naturalistic driving action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5249–5258. IEEE (2023)
48. Elayan, H., Aloqaily, M., Guizani, M.: Deep federated learning for IoT-based decentralized healthcare systems. In: Proceedings of the International Wireless Communications and Mobile Computing (IWCMC), pp. 105–109. IEEE (2021)
49. Tian, Y., Wang, S., Xiong, J., Bi, R., Zhou, Z., Bhuiyan, M.Z.A.: Robust and privacy-preserving decentralized deep federated learning training: focusing on digital healthcare applications. IEEE/ACM Trans. Comput. Biol. Bioinf. **21**(4), 890–901 (2023)