Available online at www.sciencedirect.com
ScienceDirect
Journal of the Franklin Institute xxx (xxxx) xxx
www.elsevier.com/locate/jfranklin

Towards efficient communications in federated learning: A contemporary survey

Zihao Zhao[a], Yuzhu Mao[a], Yang Liu[b], Linqi Song[c,d], Ye Ouyang[e], Xinlei Chen[a,f], Wenbo Ding[a,f,*]

[a] *Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, Guangdong, China*
[b] *Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China*
[c] *City University of Hong Kong, Hong Kong, China*
[d] *City University of Hong Kong Shenzhen Research Institute, Shenzhen, Guangdong, China*
[e] *AsiaInfo Technologies, Beijing, China*
[f] *RISC-V International Open Source Laboratory, Shenzhen, Guangdong, China*

Received 29 July 2022; received in revised form 14 November 2022; accepted 30 December 2022
Available online xxx

Abstract

In the traditional distributed machine learning scenario, the user's private data is transmitted between clients and a central server, which results in significant potential privacy risks. In order to balance the issues of data privacy and joint training of models, federated learning (FL) is proposed as a particular distributed machine learning procedure with privacy protection mechanisms, which can achieve multi-party collaborative computing without revealing the original data. However, in practice, FL faces a variety of challenging communication problems. This review seeks to elucidate the relationship between these communication issues by methodically assessing the development of FL communication research from three perspectives: *communication efficiency*, *communication environment*, and *communication resource allocation*. Firstly, we sort out the current challenges existing in the communications of FL. Second, we have collated FL communications-related papers and described the overall development trend of the field based on their logical relationship. Ultimately, we discuss the future directions of research for communications in FL.

* Corresponding author at: Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, Guangdong, China.

E-mail addresses: zhao-zh21@mails.tsinghua.edu.cn (Z. Zhao), myz20@mails.tsinghua.edu.cn (Y. Mao), liuy03@air.tsinghua.edu.cn (Y. Liu), linqi.song@cityu.edu.hk (L. Song), ye.ouyang@asiainfo.com (Y. Ouyang), chen.xinlei@sz.tsinghua.edu.cn (X. Chen), ding.wenbo@sz.tsinghua.edu.cn (W. Ding).

https://doi.org/10.1016/j.jfranklin.2022.12.053
0016-0032/© 2023 The Franklin Institute. Published by Elsevier Inc. All rights reserved.

Please cite this article as: Z. Zhao, Y. Mao, Y. Liu et al., Towards efficient communications in federated learning: A contemporary survey, Journal of the Franklin Institute, https://doi.org/10.1016/j.jfranklin.2022.12.053

1. Introduction

With the advances in deep learning (DL) models, recent years have witnessed a dawn of a new era of artificial intelligence. DL is now utilized in a variety of industries, including autonomous driving [1–3] and intelligent healthcare [4–6]. However, as the size of datasets and the complexity of the newly proposed neural networks increase, training DL models becomes significantly difficult. Consequently, several methods are offered so as to accelerate the training process of DL. For one thing, maximizing the use of hardware processing resources under appropriate software control is an excellent way to shorten training time, such as data parallelism [7,8]. For another, distributed machine learning is devised to address this issue by separating the large-scale learning process on one workstation into several small learning processes on a number of distributed workstations, which has been the most frequently adopted method in recent years. Notwithstanding, training data is usually fragmented and shared with different clients in most distributed machine learning procedures, whereas some data cannot be aggregated into a single central server since they are privacy-sensitive in nature. For instance, user behavior data in some online shopping websites may directly contain sensitive information, such as personal age, race, address; or it may indirectly carry implicit sensitive information, such as personal web browsing records and user political inclinations implied by content preferences. With the promulgation of privacy and data protection laws and regulations such as the General Data Protection Regulation (GDPR) and the improvement of people's awareness of privacy protection, more and more attention has been paid to the privacy and security of user data.

Thus, **federated learning (FL)** [9] is developed as a data privacy-aware distributed machine learning framework. Specifically, the client utilizes its own private data to train a local model and transmit it to the server side. Subsequently, the server aggregates these parameters to compute the global parameter and sends it back to all clients. Through the multiple rounds of learning and communication described above, FL eliminates the need to collect all private local data on a single central server, overcoming privacy and communication challenges in machine learning tasks, as the data are retained locally throughout the training process. Since the private property of FL, it is widely used in our daily life, such as mobile keyboard prediction [10], financial fraud detection [11], and precision medicine [12].

Despite the benefits that FL brings to us, it also faces several challenges. First of all, since the model information is high-frequently exchanged between clients and the server, the process is highly restricted by the communication conditions in FL. Therefore, the communication overhead is the main bottleneck of FL. Second, client drift is also a huge issue in FL. As an example, the clients suffer from: 1) *statistical (data) heterogeneity:* the data of each client may be not independent and identically distributed (non-iid); 2) *model heterogeneity:* the model structure of each client may be various. 3) *resource heterogeneity:* the computation, storage, and communication resources of clients may vary from one to another. These heterogeneities make the entire Fl system challenging to train. For instance, the statistical heterogeneity may limit the model convergence rate; the model heterogeneity may prevent the low bandwidth clients from receiving the cumbersome global model, resulting in the straggler issue; and the resource heterogeneity may also cause the straggler and dropout problems. Moreover, one easily overlooked but equally important challenge is the privacy issue in FL. Most people think that conveying the model parameters or model gradients instead of privacy data will not cause privacy leakage. Nonetheless, it has been demonstrated that this view is incorrect [13], and not transmitting privacy-sensitive local data still leaves security gaps. Therefore, a consummate trusted-FL system needs privacy-preserving techniques to prevent the leakage of local data. However, most proposed security strategies are very time-consuming [14], which also demonstrates the necessity of efficient protection approaches.

In this survey, we mainly discuss the first challenge, i.e., communications in FL. Some previous reviews have identified the main problems in FL communication from different viewpoints and classified existing research correspondingly. Shahid et al. [15] considered communication costs and provided an overview of related current methods such as client selection, local updating, and compression schemes. Moreover, Xu et al. [16] focused on compressed communication and introduced four compressors (quantization, sparsification, hybrid, and low-rank) in detail. Considering the training workflow of synchronized federated learning, Jiang et al. [17] discussed methods that aim to improve the training efficiency during different phases (client selection, configuration, and reporting) respectively. In addition to tackling the efficiency issue, Yang et al. [18] explored FL's applications in wireless communication and proposed to address some key open problems in wireless communication by FL methods, including communication delay, energy, reliability, and massive connectivity.

This work aims to provide a comprehensive description of communication problems in FL systems, and summarized the state-of-the-art research in all aspects involved so far. Specifically, we will cover the following aspects of the communication process: a) communication efficiency, b) communication environment, and c) communication resource allocation. The main contributions of this paper are trifold:

- We present a taxonomy of recent FL communication approaches and summarize the FL communication system framework with listed specific techniques in each field.
- We provide a comprehensive summary of recent communication algorithms in a table and sort them out in terms of **method**, **communication**, and **evaluation** objects.
- We propose some potential future research directions in the field of FL communications.

2. Problem statement and challenges

2.1. Federated learning

Assume an FL system contains N clients and all the training data and labels constitute an input space $\{\mathcal{X}_1, \dots, \mathcal{X}_N\}$ and a target space $\{\mathcal{Y}_1, \dots, \mathcal{Y}_N\}$. The i^{th} device in this FL system has its own local input space $\mathcal{X}_i \in \{\mathcal{X}_1, \dots, \mathcal{X}_N\}$ and target space $\mathcal{Y}_i \in \{\mathcal{Y}_1, \dots, \mathcal{Y}_N\}$, and will sample m_i instances with n_i features to build a local training dataset $\mathcal{D}_i = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \dots, (\boldsymbol{x}^{(m_i)}, y^{(m_i)})\}$ sampled from the local distribution $\mathbb{P}_i(\mathcal{X}_i, \mathcal{Y}_i)$, where $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n_i}$ and $y^{(i)} \in \mathbb{R}$. In traditional distributed machine learning, these training datasets are collected in a central server, while all private data are stored on the client's own devices in FL.

In FL, a group of clients train their local model W_i on their local private dataset $\mathcal{D}_i$ and then transmit the training results (e.g., model parameters or gradients) to the central server. Subsequently, the server will aggregate the received results to update the global parameter or gradient and send it back to the corresponding clients, in order to facilitate their local model updates. The whole procedure of FL is illustrated in Fig. 1 and the optimization problem of this FL system could be formulated as follows:

$$\min_{\boldsymbol{W}} f(\boldsymbol{W}) = \sum_{i=1}^{N} p_i f_i(\boldsymbol{W}),\, f_i(\boldsymbol{W}) = \mathbb{E}_{(\boldsymbol{x}^{(\zeta_i)}, \boldsymbol{y}^{(\zeta_i)}) \sim \mathcal{D}_i}\big[\mathcal{L}\big(\mathcal{F}_i\big(\boldsymbol{x}^{(\zeta_i)}; \boldsymbol{W}\big), \boldsymbol{y}^{(\zeta_i)}\big)\big], \tag{1}$$

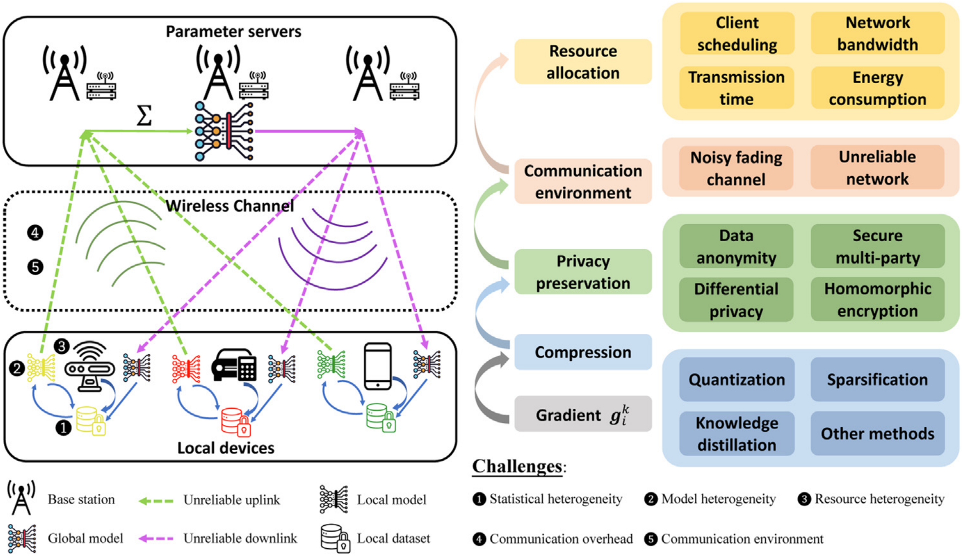

Fig. 1. Typical FL communication framework. The left part illustrates the training procedure of FL and proposes five dominating challenges of FL communication. The right shows the related works to tackle the aforementioned issues. After each client calculates its gradient g_i^k at kth global epoch, a compression method could be selected to efficiently train a global model. Finishing compression, clients could apply some privacy preservation algorithms to protect their gradient information further. Since the communication environment may be imperfect and the resource of each client may be imbalanced, the server could choose dynamical allocation strategies to mitigate their severe influence to FL system convergence.

where f denotes the global empirical risk, p_i denotes the aggregated weight of client i, f_i denotes the local empirical risk, $\boldsymbol{W}$ denotes the model parameter which optimizes the above objective function, p_i is the aggregation weight of each client (usually $1/N$), $\mathcal{L}$ and $\mathcal{F}_i$ are the loss function and the neural network function of ith client, respectively, and $(\boldsymbol{x}^{(\xi_i)}, \boldsymbol{y}^{(\xi_i)})$ denotes the mini-batch samples of local dataset D_i. In the most common FedAvg algorithm [9], the optimization objective $\boldsymbol{W}$ represents the global parameter $\boldsymbol{W_g}$ aggregated by the model parameter $\{\boldsymbol{W}_i\}_{i=1}^N$ of all clients. The whole FL training process (transmitting model parameter) is shown in Algorithm 1. Note that the optimal solution of the global empirical risk f and the local empirical risk f_i could be different. We define $f^* = \min_{\boldsymbol{W_g}} f(\boldsymbol{W_g}) = f(\boldsymbol{W_g^*})$ and $f_i^* = \min_{\boldsymbol{W}_i} f_i(\boldsymbol{W}_i) = f_i(\boldsymbol{W_i^*}) = \mathbb{E}_{(\boldsymbol{x}^{(\xi_i)}, \boldsymbol{y}^{(\xi_i)}) \sim \mathcal{D}_i}\big[[\mathcal{L}\big(\mathcal{F}_i\big(\boldsymbol{x}^{(\xi_i)}; \boldsymbol{W}_i^*\big), \boldsymbol{y}^{(\xi_i)}\big)\big]$.

2.2. Recent communication challenges in FL

- **Statistical (data) heterogeneity.** Most traditional deep learning techniques, such as face recognition [19,20] and object detection [21], assume that the training data are independent and identically distributed (iid). However, in practice, most of the training data are non-iid, and they will significantly influence the convergence rate of the entire FL process, potentially exacerbating communication overhead. In the FL setting, non-iidness means that the training data distributions of clients are different:

$$\mathbb{P}_i(\mathcal{X}_i, \mathcal{Y}_i) \neq \mathbb{P}_j(\mathcal{X}_j, \mathcal{Y}_j), \tag{2}$$

Algorithm 1 An example of FL training procedure (sending model parameters).

```
Input: The entire $N$ clients are indexed by $i \in \{1, 2, \ldots, N\}$; $D_i = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n_i}$ is the local dataset of client $i$; $T_g$ and $T_{loc}$ is the number of global epochs and local epochs, respectively, and $\alpha$ is the learning rate.
Server executes:
  Initialize $\boldsymbol{W}_g^0$
  for each round $t = 1, 2, \ldots, T_g$ do
    for each client $i$ in parallel do
      $\boldsymbol{W}_i^{t+1} \leftarrow ClientUpdate(i, \boldsymbol{W}_g^t)$
    end for
    $\boldsymbol{W}_g^{t+1} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{W}_i^{t+1}$
  end for
ClientUpdate$(i, \boldsymbol{W}_g^t)$:
for each local epoch from 1 to $T_{loc}$ do
  $\boldsymbol{W}_i^{t+1} \leftarrow \boldsymbol{W}_g^t - \alpha \nabla f_i(\boldsymbol{W}_g^t, D_i)$
end for
Return $\boldsymbol{W}_i^{t+1}$ to the server
```

for different ith and jth client. Moreover, some research [22] also considers that the data are non-iid if the expectation of local gradients and global gradients are different:

$$\mathbb{E}\left[\left\|\boldsymbol{g}_i^t - \bar{\boldsymbol{g}}^t\right\|\right] \neq 0, \tag{3}$$

where g_i^t denotes the uploaded gradient of the ith client and $\bar{g}^t$ denotes the average global gradient.

- **Model heterogeneity.** Since the resources of participants in FL vary widely, the size of the model they are able to train can also be different. Therefore, in each epoch, the uploaded model structure may be different:

$$\text{shape}(\boldsymbol{W}_i) \neq \text{shape}(\boldsymbol{W}_j), \tag{4}$$

where the $textshape(\cdot)$ operator outputs the shape of each input model $\boldsymbol{W}_i$.

- **Resource heterogeneity.** Due to the variety of different clients and communication environments, FL will be challenged in different ways. For example, the transmission channel may be noisy and fading, and the bandwidth B of the channel may be limited. Furthermore, the energy consumption E and time latency T of each participant may be constrained. Specifically, the energy consumption contains the energy of transmitting data E^U, receiving data E^R, and local computation and training E^C. Moreover, the time latency may also include uploading time T^U, receiving time T^R, and computing and training time T^C. Thus, the optimization problem can be formulated as follows:

$$\begin{aligned} \min \quad & f(\boldsymbol{W_g}) \\ \text{s.t.} \quad & B \leq \tilde{B} \\ & E^U + E^R + E^C \leq \tilde{E} \\ & T^U + T^R + T^C \leq \tilde{T}, \end{aligned} \tag{5}$$

where the $\tilde{B}, \tilde{E}, \tilde{T}$ denote the bandwidth, energy and time budget, respectively.

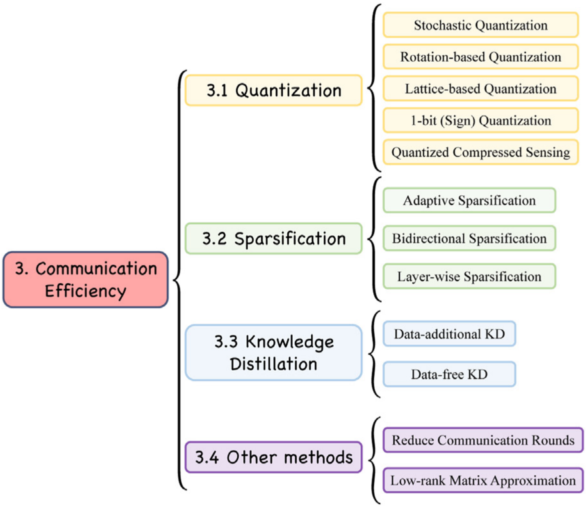
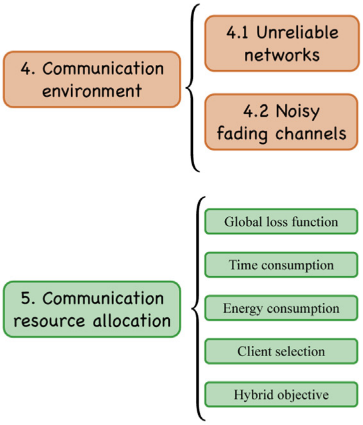
3. Communication Efficiency
3.1 Quantization
Stochastic Quantization
Rotation-based Quantization
Lattice-based Quantization
1-bit (Sign) Quantization
Quantized Compressed Sensing
3.2 Sparsification
Adaptive Sparsification
Bidirectional Sparsification
Layer-wise Sparsification
3.3 Knowledge Distillation
Data-additional KD
Data-free KD
3.4 Other methods
Reduce Communication Rounds
Low-rank Matrix Approximation
4. Communication environment
4.1 Unreliable networks
4.2 Noisy fading channels
5. Communication resource allocation
Global loss function
Time consumption
Energy consumption
Client selection
Hybrid objective

Fig. 2. The primary structure of this review's body part based on the categorization of communications in FL.

- **Communication overhead.** As the size of recently proposed neural networks increases, the communication process in FL becomes slower and slower. Thus, many communication-efficient FL algorithms are proposed to tackle this issue, and their pattern could be summarized as follows. Suppose client i has a dense model parameter $\boldsymbol{W}_i$ to transmit, and Q denotes the compression operator to compress $\boldsymbol{W}_i$ to a sparse one $Q(\boldsymbol{W}_i)$ and thus reduce communication cost. The optimization problem for a compression strategy Q and model parameters $\boldsymbol{W}_i$ could be formulated as:

$$\min_{Q,\boldsymbol{W}_i} f_i^Q(\boldsymbol{W}_i) + \lambda \mathrm{Bit}(Q(\boldsymbol{W})) + \mu \|\boldsymbol{W}_i - Q(\boldsymbol{W}_i)\|_2^2, \tag{6}$$

  where f_i^Q denotes the ith client's loss function of the compressed network, $\mathrm{Bit}(\cdot)$ denotes the summation of transmitted bits, and $\lambda, \mu > 0$ represent tuning hyper-parameters. The above problem seeks to optimize the model performance while subject to the constraint of a compression error regularizer. To solve this multi-goal optimization problem, a common method is the alternating direction method of multipliers (ADMM) [23] instead of stochastic gradient descent (SGD) [24].
- **Communication environment.** In a realistic communication environment, the communication channel is not perfect, and it may be noisy and fading, which slows down the convergence of model aggregation and reduces the performance of the global model. Assuming the transmitted global model is $\boldsymbol{W_g}$ and the received global model of clients is $\boldsymbol{W_g}'$, then the channel condition could be denoted as:

$$\boldsymbol{W_g}' = h\boldsymbol{W_g} + z, \tag{7}$$

  where h denotes the coefficients of the fading channel and z represents the additive channel noise. The remainder of this review is illustrated in Fig. 2.

3. Communication efficiency

In this section, we summarize three types of most commonly used communication-efficient FL methods into three parts, which are quantization-based, sparsification-based, and distillation-based strategies. Specifically, for quantization-based methods, we focus on the diverse designs of quantization operators, which give various ways of transforming a floating-point of 64/32 bits to a lower precision and thus determine the intrinsic property as well as the theoretical foundation for different FL frameworks featuring quantization. For sparsification-based methods, we classify existing works from the viewpoint of framework design, which provides different solutions to determine nonzero components. Moreover, given the special challenge of privacy protection in FL systems, the distillation-based strategies are classified into data-additional and data-free categories based on distinct levels of potential privacy leakage. Note that most of these communication methods are conducted to the transmitted gradients rather than model parameters. Additionally, there is also an extra section for other minor methods that are not frequently used. Finally, we elucidate the comparison of the cited methodologies in Table 1.

3.1. Quantization

Quantization [25] is a technique that decreases the model size by representing the bit width from a floating-point of 32 bits to a lower precision meanwhile retaining the model performance. Particularly, in the FL scenario, most quantization methods are proposed to compress the continuous model gradient value of each client into a discrete set after the local training process so as to reduce the representing bit.

Stochastic quantization. The stochastic quantization (SQ) is introduced in Alistarh et al. [26], which uses a gradient quantization method called QSGD to improve the communication transmission problem in parallel SGD computing, and focuses on solving the trade-off between the transmission channel bandwidth and convergence time. Specifically, the quantizer Q_{SQ} in Alistarh et al. [26] is defined as:

$$\mathrm{Q}_{\mathrm{SD}}(g_i) = \|\boldsymbol{g}\|_2 \cdot \mathrm{sign}(g_i) \cdot \begin{cases} \ell/s, & \text{with probability } 1 - \frac{g_i}{\|\boldsymbol{g}\|_2}s + \ell \\ (\ell+1)/s, & o.w. \end{cases}, \tag{8}$$

where $s \geq 1$, $0 \leq \ell < s$ are two tuning hyperparameters and $\|\cdot\|_2$ is the l^2-norm. By doing so, it preserves the statistical properties of the primary vector and introduces minimal variance. Then some QSGD-based variation algorithms are proposed. Reisizadeh et al. [27] introduce a framework called FedPAQ, which quantizes model updates by QSGD under a restricted partial client participation circumstance and reduces communication rounds by setting the synchronization of each client with the parameter server periodically. As an extension of FedPAQ, Haddadpour et al. [28] utilize the historical information of global models and theoretically analyze the proposed FedCOM under both homogeneous and heterogeneous local data. Furthermore, Das et al. [29] consider both heterogeneous local data and various noises of local stochastic gradients and propose FedGLOMO to reduce the variance of local updates by global aggregation with momentum. Dai et al. [30] present hyper-sphere quantization (HSQ) to build a global cookbook and quantize local updates based on this cookbook and SQ quantizer to reduce the communication cost further.

However, in the paper mentioned above, the quantization level could not dynamically change during the entire FL training process. To this end, Jhunjhunwala et al. [31] propose

Table 1
Classification and comparison of surveyed FL work on communication efficiency. *Amount/Round* means the strategy reduces the amount of communication or the rounds. *Partial node* indicates whether the mentioned method supports the situation that some clients participate while some clients drop the line. *Down/Up* implies the effectiveness of the cited method is on the downstream or the upstream.

		Method		Communication			Evaluation			
	Compression level	Ref.	Theoretical guarantee	non-iid	Amount/ round	Partial node	Down/Up	Datasets	# of devices	FL baselines
Stochastic Quantzation		[27]	√	✗	Amount	√	Both	MNIST, CIFAR-10	50	QSGD
		[121]	√	√	Amount	✗	U	MNIST	25	SignSGD, QSGD, D-DSGD
		[29]	√	√	Amount	✗	U	CIFAR10, FMNIST	50	FedAvg,FedPAQ
	mid	[30]	√	✗	Both	√	U	ILSVRC-12, CIFAR-10, CIFAR-100	1000, 10%	QSGD, TernGrad, SignSGD, SGD
		[31]	√	√	Both	✗	U	CIFAR-10, FMNIST	4, 8	NULL
	high	[28]	√	√	Amount	√	U	MNIST, CIFAR-10, FMNIST, EMNIST	100	FedAvg, FedPAQ, SCAFFOLD
Rotation-based Quantization	**low**	[35]	✗	✗	Both	✗	U	Geolife, MDC, Privamov	42, 14, 448	Geoi, TRL, PROM
	mid	[36]	√	✗	Both	✗	Both	CIFAR-10, CIFAR-100, Shakespeare	50, 50, 10	Hadamard, QSGD, EDEN
	high	[34]	√	✗	Amount	✗	U	MNIST, EMNIST, CIFAR-10, Shakespeare, Stack Overflow	10	FedAvg, TernGrad, 1-bit SQ
	low	[38]	✗	✗	Amount	✗	U	Gaussian iid matrix	from 3 to 15	Uniform quantizer
		[39]	√	✗	Amount	✗	U	MNIST, CIFAR-10	100/15, 10	FedAvg, QSGD, Uniform quantizer
	mid	[40]	√	✗	Both	✗	U	MNIST, Finger Movement	15	QSGD, Uniform quantizer
		[42]	√	√	Amount	✗	Both	MNIST, CIFAR-10	31	SignSGD

(*continued on next page*)

Table 1 (*continued*)

		Method		Communication			Evaluation			
	Compression level	Ref.	Theoretical guarantee	non-iid	Amount/ round	Partial node	Down/Up	Datasets	# of devices	FL baselines
Lattice-based Quantization	**high**	[43]	√	✗	Amount	✗	Both	MNIST, CIFAR-10	100	BAA
Quantized Compressed Sensing	**low**	[46]	✗	√	Both	√	Both	MNIST, FMNIST	10	FedAvg, SignSGD
	mid	[45]	√	✗	Amount	√	U	CIFAR-10, ImageNet	from 2 to 16	Random *k*, Top *k*, Threshold TernGrad, Adaptive Threshold
		[47]	√	✗	Amount	✗	Both	MNIST	10	NULL
	high	[48]	✗	√	Amount	✗	Both	MNIST	30	QCS-QIHT, QCS-Dither, SignSGD
Adaptive Sparsification	**low**	[63]	√	√	Both	√	Both	MNIST, CIFAR-10	32, 16	ADACOMM, ATOMO
	mid	[61]	√	√	Both	✗	Both	FEMNIST, CIFAR-10	156, 100	Periodic-*k*, Top-*k*, FedAvg
	high	[62]	√	✗	Amount	✗	U	MNIST, CIFAR-10, CIFAR-100, ImageNet, PTB, Wikitext-2	from 4 to 400	FedAVG
		[65]	√	✗	Amount	✗	U	MNIST, CIFAR-10, ImageNet	2, 4, 8	TernGrad
Bidirectional Sparsification		[67]	√	√	Amount	√	Both	CIFAR-10, MNIST, FEMNIST, KWS	100, 10%	FedAvg, SignSGD
		[68]	✗	✗	Amount	✗	U	CIFAR-10, ImageNet, Penn Treebank corpus	from 4 to 128	S-SGD, Top-*k*
	mid	[69]	√	✗	Amount	✗	Both	CIFAR-10, ImageNet, Stack Overflow	56	FedAvg, Top-*k*
	high	[70]	✗	✗	Amount	✗	Both	CIFAR-10	10	Top-*k*
	low	[78]	√	✗	Both	✗	U	CIFAR-10, ImageNet, PTB	16	MGS-SGD, Topk-SGD, P-SGD, LAGS-SGD

(*continued on next page*)

Table 1 (*continued*)

		Method		Communication			Evaluation			
	Compression level	Ref.	Theoretical guarantee	non-iid	Amount/ round	Partial node	Down/Up	Datasets	# of devices	FL baselines
Layer-wise sparsification	**mid**	[74]	√	✗	Amount	✗	U	CIFAR-10, Inception-v4, ImageNet, PTB	16	SLGS-SGD, Dense-SGD.
Data-additional FD	**low**	[80]	✗	√	Amount	✗	D	MNIST, FMNIST CIFAR-10, CIFAR-100	NULL	NULL
		[86]	✗	√	Amount	✗	U	MNIST	30	FedAvg, AFA, FedMGDA, FedDF
		[81]	✗	√	Amount	√	U	CIFAR-10, CIFAR-100, ImageNet, AG News, SST2	50, 40%	FedMD, FedAvg, FedProx
	mid	[87]	✗	√	Both	✗	Both	STL-10, CIFAR-10	20	FedAvg, FedDF
		[83]	✗	√	Amount	✗	Both	MNIST, FMNIST	0	FedAvg, FD
		[85]	✗	√	Both	√	U	MNIST, FMNIST, CIFAR-10	10, 80%	FedMD, DS-FL, MHAT
		[82]	✗	√	Both	✗	Both	MNIST, EMNIST, CIFAR-10, STL-10	20	FedAvg, FD
	high	[84]	✗	√	Both	√	U	STL-10, CIFAR-100, SVHN, ImageNet	100	FedDF, FedAvg
Data-free FD	**mid**	[88]	✗	√	Amount	✗	Both	MNIST	NULL	FAug
		[89]	✗	√	Amount	✗	U	MNIST, CIFAR-10	NULL	FedAvg
	high	[91]	✗	√	Both	√	Both	MNIST, EMNIST, CELEBA	20, 50%	FedAvg, FedProx, FedEnsemble, FedDfusion, FedDistill
		[92]	√	√	Both	√	Both	CIFAR-10, CIFAR-100, ImageNet, AG News, SST-5	NULL	FedAvg, FedProx, MOON, FedDistill, FedGen
		[93]	✗	√	Both	√	Both	CIFAR-10, CIFAR-100	100, 10%	FedAvg, FedProx, SCAFFOLD, FedDyn, MOON, FedGen, FedDF

(*continued on next page*)

Table 1 (*continued*)

		Method		Communication			Evaluation			
	Compression level	Ref.	Theoretical guarantee	non-iid	Amount/ round	Partial node	Down/Up	Datasets	# of devices	FL baselines
Reduce rounds	**low**	[9]	✗	√	Rounds	√	U	MNIST, Shakespeare	100,10%	NULL
		[112]	✗	√	Rounds	√	U	Adult, FMNIST	from 23 to 1101	AFL
	mid	[113]	✗	√	Rounds	√	U	CIFAR-10, CIFAR-100, CINIC-10, Shakespeare	100,16%	FedAvg, FedProx, SCAFFOLD
	high	[114]	√	✗	Both	✗	Both	MNIST, ijcnn1, covtype	10	GD,QGD,LAG
Low-rank Approximation	**mid**	[115]	✗	✗	Amount	√	U	CIFAR-10, Reddit dataset	100	NULL
	high	[90]	✗	√	Amount	✗	Both	MIND, ADR, CADEC, ADE, SMM4H	1, 2, 3, 4	FetchSGD, FedDropout, SCAFFOLD, FedPAQ

an adaptive quantization algorithm called AdaQuantFL, considering the trade-off between error and communication bits and allowing clients to automatically adjust the quantization level s of QSGD during the entire FL training process. Moreover, Amiri et al. [32] propose a lossy FL (LFL) approach to quantize both the global model parameters and the client model parameters to reduce the communication cost further, while most previous work assumes that the broadcast of the global model is perfect.

Rotation-based quantization. Although SQ is an efficient and convenient quantization strategy to assign each vector coordinate to a finite set of possibilities, recent research shows that SQ is sensitive to the vector distribution and the gap between the largest and smallest entries in the vector. Therefore, lots of studies concentrate on solving the Distributed Mean Estimation (DEM) problem. Specifically, Suresh et al. [33] devise a biased and deterministic quantization framework by applying a structured random rotation before quantization. Following this method, Vargaftik et al. [34] introduce a biased and unbiased compression technique, DRIVE, which could quantize the original vector into a 1-bit quantization level by random rotation in DEM. Although the previous work mostly defines the quantization through a set of discrete quantization points, Vargaftik et al. [35] build the theory on an interval-wise quantization and proposes EDEN, in which each coordinate is quantized to its interval's center of mass rather than the nearest quantization point. Therefrom, it decreases the entropy of the quantized vector and obtains a better estimation given a communication budget. Nonetheless, previous work requires that each client has an independent rotation matrix from other clients, which asymptotically increases the decoding time as the server must invert the rotation for each independent vector in every iteration. Basat et al. [36] propose a strategy named QUIC-FL to speed up the aggregation, which enables all clients to utilize the same rotation matrix, resulting in only a single inverse rotation on the server side.

Lattice-based quantization. Zamir and Feder [37] first introduce the lattice quantization with dithers to optimize every point to its closest lattice, which is much simpler and more efficient than optimal vector quantizers. Shlezinger et al. [38] firstly bring the lattice quantization into the FL framework. Let $\mathcal{L}$ be the lattice set, then the lattice quantizer $\mathrm{Q}_{\mathcal{L}}$ is defined by:

$$\mathrm{Q}_{\mathcal{L}}(\boldsymbol{g}) = \boldsymbol{l}_{\boldsymbol{x}} \text{ if } \left\|\boldsymbol{g} - \boldsymbol{l}_{\boldsymbol{g}}\right\| \leq \|\boldsymbol{g} - \boldsymbol{l}\| \text{ for every } \boldsymbol{l} \in \mathcal{L}. \tag{9}$$

Subsequently, Shlezinger et al. [39] take a more realistic and comprehensive situation into consideration and offer a universal vector quantization in FL called UVeQFed. Specifically, they consider the rate-constrained channels and implement the subtractive dithered lattice quantization to tackle the throughput-limited uplink problem and finally induce only a minimum distortion. Furthermore, except for channels with restricted rates, Chen et al. [40] further incorporate lattice quantization by selecting a subset of clients according to their probability of connection to the server and allowing the server to allocate bandwidth to minimize transmission delay and optimize the usage of wireless channels.

1-bit(sign) quantization. Previous quantization work usually compresses model updates or model gradients into a low-precise bit width, such as 8 bits or 4 bits. Despite this, Bernstein et al. [41] introduce a 1-bit quantization method named signSGD that represents the original gradient by its sign meanwhile maintaining the testing accuracy and convergence rate of the model, and its quantizer Q_{sign} is defined as:

$$Q_{\text{sign}}(\boldsymbol{g}) = \frac{\boldsymbol{g}}{\sqrt{\boldsymbol{g}^2}}. \tag{10}$$

Jin et al. [42] propose sto-signSGD that the quantization result of the gradient is a random variable corresponding to its sign rather than always fix, and then combine it with the majority vote rule to ensure the robustness of sto-signSGD under the client data heterogeneity situation in FL. Zhu et al. [43] employ the signSGD for Federated edge learning (FEEL) and design a sophisticated FEEL framework termed OBDA based on over-the-air majority-vote, which incorporate 1-bit gradient quantization and QAM modulation to archive communication efficiency.

Quantized compressed sensing (QCS). By taking advantage of the sparsity of the signal, compressed sensing (CS) [44] combines the sampling and compression stages of traditional signal processing methods into one step, thereby greatly reducing the number of samples required for accurate signal recovery. Specifically, CS first utilizes the matrix transformation to obtain a sparse representation of the primary dense signal and then reconstructs it from the observed data information by solving an optimization problem. Abdi and Fekri [45] first combine CS with quantization in a distributed deep learning scenario to obtain arbitrarily large compression gains. In particular, this paper introduced the QCS framework to solve the increasing gradient variance and the decrease in the convergence rate induced by quantization. Li et al. [46] and Fan et al. [47] develop a 1-bit QCS by utilizing binary iterative hard thresholding (BIHT) to reconstruct model updates/gradients rather than simply minimizing the mean squared error (MSE) [45]. However, since these two algorithms only use 1-bit quantization, they have a large quantization error due to this limitation. Therefore, Oh et al. [48] turn to study the multi-level scalar quantization method, which develops a framework called Q-EM-GAMP based on the expectation-maximization (EM) algorithm to serve as the reconstruction strategy to reduce the reconstruction error significantly.

Others. With the exception of the quantization algorithms mentioned above, there is still a variety of quantization For instance, without directly using the sign of the model gradient, He et al. [49] establish a non-uniform cosine-based quantization called CosSGD, which quantizes the angle vector with respect to the model gradient by incorporating cosine functions and the range of its minimal and maximal value. In addition, Malekijoo et al. [50] design FedZip which uses the quantization with k-means clustering as well as combining Top-z sparsification and Huffman encoding to maximize the compression rate. What's more, in order to solve the non-iid problem in FL, Philippenko and Dieuleveut [51] present Artemis which first compresses the model gradient by its memory term and then utilizes s-quantization to compress the difference. They find that using the memory mechanism could improve the convergence performance under the non-iid training data scenarios.

Chen et al. [52] consider the heterogeneous FL scenario that local clients hold various quantization levels (precision) and propose FedHQ to assign different aggregation weights to clients by optimizing the convergence upper bound. Cui et al. [53] design the MUCSC algorithm to compress the uploads by soft clustering and provide some theoretical properties of MUCSC. In addition, they also introduce a boosted MUCSC to tackle the situation with rather scarce network resource.

3.2. Sparsification

Sparsification techniques transform a full gradient to a sparse one with a subset of important elements and set other insignificant coordinates to zero. In a federated learning regime, top-k sparsification and rand-k sparsification are two commonly adopted methods [54]. In top-k sparsification, the subset of remained elements consists of k percent of the sparsification target with the greatest absolute values, while the remained values are selected randomly in rand-k sparsification. Although rand-k sparsification is an unbiased compression operator [55,56], it leads to larger compression errors and therefore has worse practical results compared to top-k sparsification in the high compression regime [54,57]. Generally speaking, sparsification techniques reduce communication overload in a more aggressive manner compared to quantization. For example, the top-k sparsification with error feedback can maintain the desired convergence rate and accuracy even with 99%-99.9% gradient elements zeroed out [58,59].

In contrast to sparsification in centralized learning, sparsification in federated learning has to consider not only the computation and storage efficiency but also the communication efficiency among distributed clients throughout the learning process. Therefore, most of the FL research on sparsification focus on improving existing sparsification techniques like top-k sparsification to meet the needs for frequent communications better.

Adaptive sparsification. In Sahu et al. [60]'s work, top-k is proven to be the communication-optimal sparsifier with a given k element budget per iteration from the optimization perspective, but a different hard-threshold sparsifier is further developed to consider optimality throughout the training instead of per iteration optimality. The proposed hard-threshold sparsifier adaptively determines the degree of sparsity k by a constant hard threshold and has been proven to be the optimal sparsifier that theoretically minimizes the compression error under a given budget throughout the FL training process [60]. Similarly, the adaptation of the sparsity degree k to minimize the overall training time has been proposed [61]. It is achieved by adjusting the degree of sparsity to come close to achieving the ideal balance between computation and communication in an online learning manner.

Some other work on adaptive sparsification focuses not only on adjusting the sparsity level d but also on co-adjustment with some other factors in federated learning, such as local update iterations and partial participation. For example, Sattler et al. [62] provides an information-theoretic way to analyze communication delay and error-accumulating sparsification techniques, both of which achieve information compression by delaying some updates and gathering gradient information before actual transmission. On the basis of these theoretical perspectives, the proposed Sparse Binary Compression (SBC) method adaptively trades off the temporal sparsity and gradient sparsity. Similarly, Nori et al. [63] considers both local update and sparsity budgets to characterize learning error and adaptively adjusts these two components to yield Fast FL (FFL). As another line of work, Abdelmoniem and Canini [64] design an adaptive compression control mechanism that better trades-off between training speed and accuracy by coupling network delays and compression control. The superiority of this method is demonstrated with top-k and rand-k.

To overcome the staleness of updates resulting from sparsified communication Li et al. [65] propose a General Gradient Sparsification (GGS) framework which performs gradient correction on the accumulated insignificant gradients for adaptive optimizers and updates the batch normalization layer with clients' local gradients. This method alleviates the impact of delayed gradient elements and thus further improves the performance of FL with sparsity. As another line of work, the linear FetchSGD sketch proposed by Rothchild et al. [66] allows error accumulation to be moved from local clients to the central server, eliminating the need for local client states. FetchSGD is an effective strategy to address the challenges of partial participation in FL systems.

Bidirectional sparsification. The aforementioned sparsification methods primarily consider compressing the upstream communication from clients to the central server. Due to the heterogeneity of local data, the sparsity patterns of updates from different clients can be very different. Sattler et al. [67] have claimed that if the amount of clients is larger than a threshold, the downstream update will be dense. To solve this problem, the sparse ternary compression (STC) framework is specifically designed to extend the top-k sparsification for enabling downstream compression. Shi et al. [68] introduce a global Top-k (gTop-k) sparsification method, which utilizes a tree structure to determine the global k largest absolute values of all clients. This method overcomes the challenge of the irregular non-zero gradient indices from different clients during aggregation and, therefore, successfully compresses the downstream communication as well.

Another way to achieve bidirectional communication compression is to reduce the extra budget caused by specifying non-zero location indices. Xu et al. [69] develop a synergistic combination of various compressors for both gradient values and indices. The proposed Bloom-filter-based index compressor can reduce 50% of transmission compared against raw $\langle key, value\rangle$ sparse representation. Besides, it is also effective to put limitations on transmitting new non-zero positions [70]. In this work, the proposed time-correlation sparsification (TCS) scheme utilizes the correlation between consecutive sparse representations in FL training to reduce the transmission of newly-computed non-zero indices. This method has been shown to achieve a higher compression level in downstream communication compared to upstream communication.

Layer-wise sparsification with pipeline. With the aim of further improving the communication efficiency of FL in terms of accelerating the entire training process, some works on FL with sparsity have considered making use of the models' layered structure to parallelize the communications with computations, which is referred to the pipeline [71–73]. To combine gradient sparsification with the idea of the pipeline, Shi et al. [74] develop a layer-wise adaptive gradient sparsification (LAGS) scheme, where the subset of remaining values for each layer is selected independently according to a given ratio, and the transmission of any sparsified gradient of any layer $l+1$' can be parallelized with the gradient calculation and sparsification of layer l, instead of waiting for the completion of the entire back-propagation before transmitting a single sparsified gradient.

However, an existing drawback of LAGS is that the sparsified gradient of each layer will invoke one independent communication, and thus the layer-wise communications require high communication start-up costs [74,75]. One intensively-studied way to alleviate the startup overload is merging gradients from multiple layers for one communication, but it will also cause more computation and waiting time [74–77]. To further trade-off gradient computation and layer-wise gradient sparsification, an optimal merging scheme named Optimal Merged Gradient Sparsification (OMGS) has been developed [78]. It formulates the trade-off as an optimization problem and minimizes the iteration time by maximizing the overlap between sparsified gradient computation and communication during training.

3.3. Distillation

Knowledge distillation (KD) [79] is further employed to FL to alleviate the communication bottleneck. Precisely, in order to reduce the model size, KD aims to transfer the information of a large model (teacher/mentor) to a small model (student/mentee) without adversely impacting the model's precision and convergence. Therefore, the teacher network is typically a network with a large number of parameters and a complicated structure, with excellent performance and generalization ability, whereas the student network has a modest number of parameters and a simple structure. Since the shape sizes of different student models' outputs are identical and irrelevant to the structure of student models, KD could be utilized to tackle the model heterogeneity issue as aforementioned by uploading the output of the local model without softmax (i.e., the logit vector) other than the model itself, and such method is called federated distillation (FD). In this section, we give a taxonomy of FD based on whether the client will upload a small subset of their private data to build a public dataset on the server side and divide FD into the data-additional FD (need to construct an addition dataset jointly) methods and the data-free FD methods.

Data-additional FD. At first, Li and Wang [80] propose FedMD that each client first trains the local models on a labeled public dataset and uploads their class scores (i.e., logit vector) rather than model parameters to the server, and subsequently, the server integrates them to obtain the knowledge from all clients. However, the training process of FedMD is simultaneously on both labeled public data and private data, which necessitates a considerable amount of local computation and is not suitable for resource-limited devices. Furthermore, the creation of public datasets necessitates careful deliberation and thus lacks generalization. Consequently, Lin et al. [81] propose FedDF to train the global model through an unlabeled dataset in an ensemble distillation manner and prove that FedDF is robust to the selection and combination of the FD dataset. Especially, the selection of an auxiliary dataset is set on the server side and effectively mitigates the computational pressure of local clients. Since the public dataset is unlabeled, the privacy leakage of FedDF is also much less than the FedMD's. Nonetheless, although the selection of the public dataset is moved to the server side, there are still no criteria for determining the size of the dataset. In Sattler et al. [82], they establish the Compressed Federated Distillation (CFD) developing "entropy", "certainty", and "margin" standards to determine the size of the public distillation dataset. Afterward, they also utilize a uniform quantization algorithm and delta coding to reduce the communication cost further.

The previous work mainly concentrates on reducing the communication cost of FD by either changing the size of the public dataset or the size of logit vectors, but none of them considers modifying the **aggregation strategy**. By contrast, Itahara et al. [83] propose DS-FL, which attempts to aggregate the local models via an entropy reduction strategy. Since the existence of data heterogeneity in FL, the entropy of global logits is quite high, which indicates the difficulty of training a model with high performance in non-iid scenarios. DS-FL reduces the entropy of global logits sharpening logits by adding the temperature factor T to the softmax to accelerate and stabilize the DS-FL. In addition, another aggregation algorithm named FedAUX is created by Sattler et al. [84], which calculates the certainty of each client's logits by contrastive logistic scoring and uses this certainty to determine the aggregation weights to yield a strong empirical performance on data with high distinction. Moreover, Li et al. [85] construct an adaptive aggregation strategy named pFedSD to dynamically modify the weight of each participant and improve the quality and performance of the global model. Specifically, it exploits the Jensen-Shannon divergence between the transmitted model outputs in two adjacent rounds as the weight of clients to measure the similarity of local models. Similarly, Sturluson et al. [86] also develop an adaptive aggregator named FedRAD based on median scores to reinforce the global model.

Combining both public dataset selection and non-trivial aggregation methods, Liu et al. [87] present FAS to actively sample client data to ameliorate the convergence of the training process under client-drift situations and provide its theoretical analysis. Specifically, they solve the non-iid problem by weighted logistic scores based on the discrepancy of clients' data and then add the differential privacy mechanism to the aggregation algorithm to preserve the safety of the local private data coupled with restricting the privacy loss.

Data-free FD. Although in the data-additional FD category, transferring logits can greatly reduce the communication overhead, it requires each client to sacrifice a part of its own private data to construct a public dataset, which might be unacceptable in some extremely private cases, such as a user's private information data on their personal devices. Therefore, data-free FD algorithms are proposed without sharing any private data to further preserve the security of all clients. The first data-free FD algorithm is developed by Jeong et al. [88] to handle the non-iid problem, in which clients periodically send their average local logits to the server and afterward receive the globally aggregated logit as the teacher's output to help train the local student model. However, the issue of model heterogeneity is not addressed in this research. To this end, Jiang et al. [89] present FedDistill and solve this problem by forcing the client to share a global model of the same structure while training another customized local model. Concretely, each client holds two models: a personalized large model that serves as the teacher model and is trained on local data, and a small global model received from the server. In each iteration, student models with the same structure are transmitted to the server and aggregated, which could still obtain all clients' information with arbitrary personalized large model structures. Following FedDistill's lead, Wu et al. [90] introduce FedKD including three different loss functions to encourage the training of the global model and matrix factorization to further reduce the communication overhead. Specifically, they utilize the Kullback–Leibler (KL) divergence loss to adaptively distill the knowledge of both teacher and student models.

Nevertheless, directly aggregating the global model might still be heavily affected by the heterogeneity issue, and its performance might be unsatisfactory. Thus, Zhu et al. [91] propose FedGEN to train a **generator** on the server side based on the transmitted classifiers received from clients. Specifically, in each FL iteration, clients convey their local label count information to the server and regularize the local training after obtaining the global lightweight generator.

After the release of FedGEN for training a generator to facilitate the FD procedure, some other recent work focuses on improving the **performance** of the generator model. For example, Yao et al. [92] propose FedGKD to utilize the averaged parameters of historical global models for the ensemble, which mitigate the client-drift problem caused by the non-iid local dataset. In addition, Zhang et al. [93] propose FedFTG that first generates pseudo-data to train the generator and then utilizes the hard samples to train the global model simultaneously. What's more, facing the data distribution shift issue, they also devise customized label sampling and label-wise ensemble algorithms to boost the convergence of the FD process.

Applications. Thanks to its special teacher-student fashion, other than employed to communication-efficient scenarios, KD is also frequently used in addressing the heterogeneity issues of clients, which is also called *personalized FL*. Cho et al. [94] propose a weighted consensus KD framework dubbed Fed-ET, which utilizes consensus distillation with diversity regularization to better extract the model knowledge based on heterogeneous scenarios. Additionally, Zhu et al. [95] devise a personalized framework FedResCuE, which trains a number of sub-models by pruning the full global model corresponding to a pruning sequence. In this way, each client just selects the pruning weight closing to its own local model and only receives a tiny model rather than the cumbersome global model. Besides, Zhang et al. [96] introduce a knowledge agnostic KD framework (even without any prior knowledge) called FedZKT transmitting a "personalized" generator based on the local model structure of each client and reducing the communication cost further. Furthermore, Ozkara et al. [97] develop a compressed personalized KD algorithm called QuPeD, which applies a soft quantization method by solving an optimization problem and utilizes KD to tackle the resource heterogeneity issue.

On account of the impressive feasibility and compression ability, KD has been wildly employed in massive FL scenarios. For instance, due to the high communication and computation overhead in FL protection methods such as local differential privacy (LDP) [98] and secure multi-party computation (SMPC) [99], it is complicated to incorporate these secure compute methods into real-world FL applications. Therefore, a variety of current research [100–106] considers combining KD with secure computation to boost the whole training process and privacy-preserving medical prediction [107,108]. Moreover, KD is also used in IoT and edge learning [109–111] to reduce communication costs.

3.4. Minors

Reduce communication rounds / fast convergence. In this article that proposes federated learning [9], they propose FedAvg to perform global aggregation after multiple iterations of local updates rather than directly collecting the model gradient in each iteration, which hugely reduces the communication frequency. Moreover, Li et al. [112] introduce q-FedAvg to dynamically select a subset of all clients and achieve faster convergence in terms of communication rounds. Subsequently, Hyeon-Woo et al. [113] boost the convergence rate via low-rank Hadamard product parameterization. Furthermore, [114] propose a criteria framework named LAQ to only update the compressed model gradient when the variation of the local model is large enough, and thus reduce the number of communication rounds.

Low-rank decomposition. Konečnỳ et al. [115] consider uploading structured models in FL and propose modifying the updated matrix to a sparse low-rank matrix to save communication cost, allowing the low-rank approximation in FL communication. Furthermore, Wu et al. [90] discover that the updated gradients have low-rank properties and thus utilize the singular value decomposition (SVD) [116] to decompose the model gradient matrix into smaller matrices, which significantly mitigates communication overhead. Additionally, Azam et al. [117] propose to recycle the gradients between communication rounds by utilizing the low-rank property, which reduces the transmission of model parameters to single scalars.

Topology design. Unlike the client-server architecture we mainly concentrate on, the topological network design for cross-silo FL [118] is also an area of interest. Marfoq et al. [119] claim that the high-speed access links are more efficient in exchanging information in cross-silo scenarios and introduce a novel topology design by optimizing the minimal cycle time problem and obtaining the largest throughput. Furthermore, Guo et al. [120] propose HL-SGD utilizing the device-to-device(D2D) communication capabilities to divide clients into a set of separate clusters with high D2D communication bandwidth and speeding global model convergence, without losing the model performance.

3.5. Comparison and discussion

In these aforementioned subsections, we mainly introduce three kinds of communication-efficient strategies. However, each of them has its own merits and demerits. For quantization methods, since most of the time quantizers are defined explicitly, the convergence analysis can be performed directly on them. Moreover, sparsification is more empirical and sometimes can mitigate more communication costs than quantization methods. Nevertheless, we cannot provide some theoretical analysis of this method. For knowledge distillation, it has an excellent performance in heterogeneous FL scenarios. But similar to sparsification, no theoretical analysis we can derive.

4. Communication environment

In the previous section, most of the papers assume that the communication channel is perfect and do not consider the resources of clients, such as the devices' energy and bandwidth. Furthermore, although the previous work proposes to compress the updated model or reduce the communication rounds to reduce the overall communication time, none of them study the problem under a time-limited scenario. In this section, we discuss the impact of different communication environments on model performance and convergence rate in FL and summarize the comparison of mentioned methods in Table 2.

4.1. Unreliable networks

In FL, mobile devices may frequently *drop offline* due to unreliable network factors such as unstable network links, insufficient device power supply, and slow device training, resulting in a decrease in the performance and convergence rate of the entire FL system. As such, Yu et al. [125] consider the unreliable network situation that all communication links have a certain packet loss rate to fail, and gives a theoretical analysis of this problem. However, it might be unreasonable to assume that each communication link has the same loss rate since it varies from the duration and packet size of different packets in reality. Consequently, Zhang et al. [126] design a new algorithm called ACFL to adaptively compress the information from the shared model based on the physical conditions of the current network, taking into account the drop rate and the total number of transmitted packets. Moreover, Salehi and Hossain [123] propose a different method to calculate the success probability of each link. Specifically, they utilize stochastic geometry tools to compute the loss rate and apply different weights depending on the scheduling policy and its transmission success probability to each client during global aggregation for better performance. Furthermore, Wu et al. [122] propose that SAFA facilitates the influence of straggling clients and outdated models in heterogeneous scenarios. In particular, trendy and deprecated clients take the most recent global model as the base model for the subsequent round of training, but tolerated clients can continue processing their old local findings.

Since the network's physical condition is hard to measure, Wu et al. [124] propose a reliability-agnostic framework called HybridFL, which explores the situation that the reliability of clients is agnostic under strong privacy-preserving conditions and tackles it by adding regional slack factors and adjusting client selection regionally.

The previous work reckons the unreliable network issue at the **physical layer** level, whereas some work discusses it from the point of view of the **transport layer** level. As an example, Ye et al. [127], Mao et al. [141] consider the unreliable communication protocol, such as the user datagram protocol (UDP), rather than the reliable transportation layer protocol, such as the transmission control protocol (TCP) using a link reliability matrix to optimize mixed weights. Furthermore, the gossip-based averaging protocol is the most commonly used fault-tolerant large-scale framework [142]. Nevertheless, the complexity of gossip would grow linearly with the number of clients, decelerating the convergence of FL and increasing the communication overhead. Thus, Ryabinin et al. [143] propose Moshpit All-Reduce which permits clients to

Table 2
Classification and comparison of surveyed FL work on the communication environment. *Client selection* indicates whether this method performs the client selection process. *Limited bandwidth* implies whether this strategy takes the limited bandwidth constraints into account.

	Down/ Up	Ref.	Methods		Communication			Evaluation		
			Theoretical guarantee	non-iid	Client selection	Limited bandwidth	Partial node	Datasets	# of devices	FL baselines
Unreliable Network	**Down**	[122]	✗	√	√	√	√	Boston Housing, MNIST, KDD Cup'99	5, 100, 500	FedAvg, FedCS
		[123]	√	√	√	✗	√	MNIST	100	FedAvg
	Both	[124]	√	✗	√	√	√	Aerofoil, MNIST	15, 500	FedAvg, HierFAVG
		[125]	√	✗	✗	✗	√	CIFAR-10, ATIS	16	NULL
		[126]	√	√	√	√	√	FEMNIST, Shakespeare	3500, 2288	FedAvg, C-FedAvg
		[127]	√	✗	✗	✗	√	CIFAR-10	8/16/24	DSGD
Noisy fading channel	**Down**	[128]	√	√	√	✗	✗	MNIST	200	BAA
		[121]	✗	√	✗	√	✗	MNIST	25	SignSGD, QSGD
		[129]	√	√	✗	✗	✗	MNIST	10	FedAvg, COTAF
		[130]	✗	√	✗	✗	✗	MNIST	10	NULL
		[131]	√	√	✗	✗	✗	Random Gaussian matrix	5	One-bit. Open-loop
		[132]	√	✗	✗	✗	✗	Million Song	100	LAQ-WK, FDM-GD, ECESA-DSGD
		[133]	√	✗	✗	✗	✗	MNIST	20	NULL
		[134]	✗	√	✗	✗	✗	FMNIST	from 2 to 40	FedAvg, Error-free channel

(*continued on next page*)

Table 2 (*continued*)

Down/ Up	Ref.	Methods		Communication			Evaluation		
		Theoretical guarantee	non-iid	Client selection	Limited bandwidth	Partial node	Datasets	# of devices	FL baselines
Up	[135]	√	✗	√	✗	√	Random Gaussian matrix	100	GBMA
	[136]	√	√	✗	√	✗	MNIST	40	NULL
Both	[137]	✗	√	✗	✗	✗	Random Gaussian matrix	5	MAP bound, Model-based detection, CL, NL
	[138]	√	√	✗	✗	√	MNIST, CIFAR-10, Shakespeare	2000 10%, 100 10%, 300, 10%	Noise-free channel, Equal power allocation
	[139]	✗	✗	√	✗	√	MNIST	100	LFL
	[140]	√	√	✗	√	✗	MNIST	30	NULL

dynamically select the group to which they belong, and each client only influences its current group after dropout.

4.2. Noisy fading channels

Recently, more and more research has focused on FL in practice, particularly over-the-air FL (OTA-FL), in which all devices transmit their data signals simultaneously through the MAC and perform computations through the wireless channel. OTA-FL could simultaneously utilize complete spectral and temporal resources and thus reduce the communication overhead in FL. However, it suffers from the noisy fading channel problem since it may cause serious problems, such as dropped packets or incorrect packet information.

Most related research aims to optimize over the **uplink** noisy fading multiple access channel (MAC). Zhu et al. [128] first assume the noisy fading channel following the iid Rayleigh fading and then design a fundamental aggregation algorithm called broadband analog aggregation (BAA) to utilize the wave superposition property of MAC for efficient update aggregation and resistance of fading channel. As an extension of BAA, Amiri and Gündüz [121] propose two more powerful and robust algorithms to tackle the fading channel issue. They create a digitally distributed SGD (D-DSGD) to select a single device for transmission in each iteration, whereas the D-DSGD lacks robustness. Thus, they develop compressed analog DSGD (CA-DSGD) to accelerate computation through sparsity and allocate the power alignment of each client at the server side for better efficiency and robustness. Nevertheless, the CA-DSGD's power alignment procedure could not be applied to the heterogeneous scenarios and controlled by each client. Subsequently, Yang et al. [129] consider the convergence impact of noise on OTA-FL in non-iid and heterogeneous conditions and proposes a more flexible framework ACPC-OTA-FL to permit each client to adaptively calculate its transmit power level and a number of local update rounds in order to maximize the utilization of computation and communication resources. In addition, Zhang and Tao [130] adjust the transmit power of a device on the fly depending on aggregated gradient estimates that have been collected in the past for high-performance and reliable over-the-air computation (AirComp) over fading channels. Another gradient aggregation algorithm called analog gradient aggregation (AGA) [131] adaptively change the receiver's parameter based on data and channel state information (CSI) under fading channel conditions. More directly, Sery and Cohen [132] claim that their proposed Gradient-Based Multiple Access (GBMA) framework does not need any power control or beamforming to get rid of fading effects, including Rayleigh, Rician, and Nakagami fading models. Instead, GBMA takes effect directly with noise distortion gradients.

Except for optimizing the problem of noisy fading uplink channel via the aggregation method and allocating transmit power, Hellström et al. [133] use retransmissions after the failure of communication to reduce such estimation errors and increase the convergence rate. Furthermore, Lin et al. [134] investigate the problem of stragglers in fading channel situations and solves it by assigning relays to permit clients to upload their models to the relay server and mitigate the problem of stragglers.

Although most work considers the uplink as the key to communication bottleneck and supposes the downlink is error-free, imperfect **downlink** transmission could still tremendously affect the convergence and performance of FL. Xia et al. [135] develop FedSplit to solve the ill-conditioned problem over noisy fading channels and recover the aggregation of local updates calculated by the end devices through AirComp. Moreover, Mashhadi et al. [137] establish a data-driven universal symbol detection procedure under the downlink fading channel and a new neural network based on the maximum a-posteriori probability (MAP) detector. Furthermore, Amiri et al. [136] tackle downlink fading broadcast downlink by organizing a digital approach to quantize the global model update and provides a theoretical convergence analysis for analog downlink transmission.

Comprehensively, some studies also simultaneously investigate both the noisy fading **downlink and uplink** channels. For instance, Wei and Shen [138] provide a rigorous convergence analysis toward standard FedAvg under non-iid client dataset, partial client participation, and noisy fading downlink and uplink channels conditions. Amiri et al. [139] select a subset of clients with the highest channel gain over both the uplink and downlink fading channel conditions, which results in better global performance and more accurate information exchange. However, the previous work does not consider the stochastic delay existing in time-varying fading channels. In order to solve the problem, Li et al. [140] examine the delay distribution for wireless FL systems with uplink and downlink transmission using both synchronous and asynchronous downlink transmission strategies by exploiting the combination of saddle point approximation, extreme value theory (EVT), and large deviation theory (LDT).

5. Communication resource allocation

Due to the difficulty posed by the heterogeneity of each device's resources, how to efficiently communicate model information in such scenarios has become a hot topic in recent years. In this section, we classify each related article according to the target of their objective functions because most of the research in this field utilizes the optimization method to solve the problem, and we list the comparison of these strategies in Table 3.

Global loss function. Shi et al. [144] establish an associated client scheduling and bandwidth allocation strategy based on optimizing the trade-off between latency and the number of training rounds. Additionally, Yu et al. [145] consider the hybrid vertically and horizontally partitioned client dataset and optimize both the communication and computational energy consumption to develop a scheduling optimization problem for a participant jointly and then minimize the global loss function. Furthermore, Mahmoudi et al. [146] propose an iteration-termination criterion FedCau by optimizing the computation and communication latency. Furthermore, they also combine FedCau with Top-q and LAQ compression methods to further reduce communication overhead.

However, the above papers may lack some generality since they all specify the type of resources, such as bandwidth and computational energy. To this end, Wang et al. [147] propose a more general method to dynamically minimize the loss function under a resource budget constraint without specifying a resource type and analyze the convergence bound for federated learning with a non-iid dataset.

Time consumption. Chen et al. [148] simultaneously minimize the communication time and convergence time to construct a user selection and power allocation framework. However, it only considers the transmission time constraint. Subsequently, Yang et al. [149] take the computation time into account and also develop a bisection search algorithm to find the optimal solution. Furthermore, Lu et al. [150] also consider the cost of the block validation process and introduce the digital twin wireless networks (DTWN) to alleviate the resource-constrained FL task in real-world environments.

Client selection. A greedy scheduling strategy is proposed by Nishio and Yonetani [151], which schedules as many devices as possible within a given time frame in each round. However, the deadlines are chosen experimentally, and it is tough to adapt to dynamic channels

Table 3
Classification and comparison of surveyed FL strategies on communication resource allocation. *Constraints* implies the constraints of the objective function optimized in each paper.

	Ref.	Method		Communication			Evaluation		
		Theoretical guarantee	non-iid	Client selection	Constraints	Partial node	Datasets	# of devices	FL Baselines
Global loss function	[144]	√	√	√	Time budget, Client subset	√	MNIST, CIFAR-10	20	Random Proportional fair
	[145]	✗	✗	✗	Energy budget	✗	MNIST	50	Loss, time
	[146]	√	√	✗	Latency budget	✗	MNIST	50, 100	LAQ, Top-*k*
	[147]	√	✗	✗	Different types of resources	✗	MNIST	from 5 to 500	SVM, *K*-means
Time consumption	[148]	√	✗	√	Different resources, Client subset	√	MNIST	from 3 to 20	FedAvg
	[149]	√	✗	✗	Bandwidth, Time budget, Uploaded data size, Transmit power	✗	Real open blog feedback dataset	50	EB-FDMA, FE-FDMA, TDMA
	[150]	✗	✗	✗	Digital twins, Subchannel, Training batch size	✗	CIFAR-10	100	FedAvg
Client selection	[151]	✗	√	√	Time budget, Communication rounds	√	CIFAR-10, FMNIST	1000, 10%	FedLim
	[152]	√	√	√	Power budget, Bandwidth, Time budget	√	MNIST	10	FedAvg

(*continued on next page*)

Table 3 (*continued*)

	Ref.	Method		Communication			Evaluation		
		Theoretical guarantee	non-iid	Client selection	Constraints	Partial node	Datasets	# of devices	FL Baselines
	[153]	✗	✗	✗	Bandwidth, Time budget	✗	MNIST, FMNIST	30	BBA
Energy consumption	[154]	✗	✗	√	Bandwidth, Time budget	√	MNIST	50	Optimal/ uniform bandwidth allocation
	[155]	✗	√	√	Bandwidth, Time budget	√	Real open blog feedback dataset	50	TDMA
Hybrid objective	[156]	✗	✗	√	Resource availability, Trust, Client subset	√	MNIST	12	NULL
	[157]	✗	✗	✗	Time budget, Energy budget	✗	MNIST	50	Loss, energy

and device computing power. To this end, Chen et al. [152] propose CEFL to reuse outdated local model parameters for client scheduling and theoretically analyze its convergence and communication characteristics. Nonetheless, the previous articles do not consider channel state information (CSI). Thus, Zhao et al. [153] propose two bandwidth allocation methods based on CSI and particle swarm optimization (PSO) and transform the optimization problem from minimizing global loss into maximizing the number of active clients.

Energy consumption. Zeng et al. [154] propose an energy-efficient joint bandwidth allocation and scheduling strategy by minimizing the total energy consumption. However, they assume that all clients have the same model structure and thus ignore the computational energy. Nevertheless, Yang et al. [155] further consider the computational energy and derive closed-form solutions for computation and transmission resources of a low-complexity iteration algorithm.

Hybrid objective. In the meanwhile, some research focuses on optimizing multiple objects in the objective function. For example, Imteaj and Amini [156] try to optimize the client resources such as memory, bandwidth, and battery life under the malicious client and straggler issue, but no theoretical guarantees are provided. Comprehensively, Zaw et al. [157] aim to optimize the global loss function and the entire communication and computation time and then reformulate the problem as a Generalized Nash Equilibrium Problem (GNEP) to establish a comprehensive theoretical convergence analysis.

6. Conclusions and future directions

Due to the complex wireless network environment, the heterogeneity of device data, the difference of device capabilities, and other factors, how to efficiently perform FL training in the edge network is a key issue currently facing. In this review, we discuss the solutions to this issue in three aspects: communication efficiency, communication environment, and communication resource allocation.

First, we present the communication efficiency strategies in FL. Specifically, we dive into the three main communication-efficient FL approaches: quantization, sparsification, and knowledge distillation. For each strategy, we examine its benefits, drawbacks, and prospective logical development trend in order to make the entire paragraph logically coherent.

Second, we discuss the influence of the harsh FL communication environment. We divide the harsh environment into two parts: 1) unreliable network: the connections between clients and servers are not guaranteed to succeed and thus may cause clients to go offline; 2) noisy fading channels: the communication channels have noise and fade along with the transmission distance.

Finally, we present the communication resource allocation algorithms, which mainly target solving an optimization problem based on some realistic constraints. Therefore, we classify such algorithms according to the optimization objective of the optimization problem. In particular, we find that the global loss function, time consumption, energy consumption, client selection, and hybrid objectives are the most frequently used objectives.

Even if the existing research is continually being refined, there are still some future directions we can investigate in this subject.

Computation consumption. As mentioned earlier, many communication compression algorithms run on the client side. Although there are some studies discussing computing time and computing power consumption in Section 5, few studies work on the specific computing resource consumption in the compression algorithm, which is exactly significant for resource-limited mobile devices. One work Mishchenko et al. [158] is related to this field, in which they propose a quantized parallel SGD where the gradient coordinates are rounded after scaling. Further work can be performed to extend it to popular optimization methods such as Adam.

A comprehensive communication system. Most of the previous studies considered only one compression method, and some studies considered the combination of two compression methods. However, it is possible that this combination is not the most efficient. In future work, the integration of multiple model compression methods to formulate an overall FL compression system and finding the balance of each compression method theoretically may become the next outlet for communication-efficient FL.

Exploring privacy while compressing. Privacy is also a widely studied field in FL. As mentioned in the introduction section, FL without privacy-preserving procedures may nevertheless leak local private information through the transmission of model parameters or model gradients. In order to solve this kind of data privacy leakage problem, some existing research uses cryptography to encrypt the model parameters sent by each participant to defend against model inference attacks, such as LDP and SMPC introduced in Section 3.3. In addition, other protective strategies, such as VerifyNet [159] and adversarial training [160], are effective during the training stage and preserve clients' privacy. Nonetheless, a substantial quantity of local calculation overhead will make the entire FL process extremely sluggish. Due to the heterogeneity of local client computing resources, some clients may take a long time to perform encryption algorithms while other clients can only wait, thereby further causing communication congestion. Consequently, the demand for efficient privacy-preserving strategies is increasing. Combining compression techniques and privacy-preserving techniques is one way to make the entire FL framework both efficient and secure. As an illustration, federated distillation only needs to transmit a few fully-connected layers or even logits between clients and the server, which simultaneously compresses the updated information and protects the data privacy.

Asynchronous aggregation. In this review paper, we mostly concentrate on synchronous FL aggregation. However, due to the fact that the available computation, communication, and data volumes on many work nodes are sometimes distinct, the time that work nodes transmit model parameters of local training at each round may vary from one to another. This will cause the parameter server to prolong the training time due to waiting for the slow node to upload the parameters (i.e., the straggler problem). In addition, since the local data on multiple work nodes usually may not follow the same probability distribution and their local models may also be different, it will cause the local models of different work nodes to converge in different directions, reducing the overall training speed. Therefore, the asynchronous aggregate update algorithm should also be considered when considering the communication overhead to make the model converge faster and obtain better model performance.

Integration with 5G and beyond. The rapid advances in wireless communication technologies also enhance the development of FL technologies. The bandwidth and processing speed constraints of FL are expected to be lifted by the expansion of the 5G network across the globe [161]. 5G network is estimated to provide a $10-100x$ increase in bandwidth and a $5-10x$ decrease in latency, enabling more devices on the Internet of Things (IoT) and the Internet of Vehicles (IoV) devices to learn from each other with federated learning. For example, China Mobile proposed the integration of FL with the distributed intelligent network architecture [162]. Internationally, industrial standards on FL in 5G networks have also been proposed in 3GP [163]. 6G communications can achieve ultra-low delay, superior reliability, and high energy efficiency [164]. Since 5G and 6G networks will enable a massive number of heterogeneous devices of higher intelligence learned from explosive data, the optimization of communication efficiency over the entire FL network will undoubtedly be more challenging. Heterogeneity in device capabilities, network connectivity, and energy level will also lead to complex network behaviors and cause more vulnerabilities and failures [165]. Therefore, it is important to perform real-life simulations and efficient training to accelerate convergence and identify bottlenecks in key applications and environments, such as AR/VR and intelligent transportation [166], to address these issues.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported in part by Tsinghua Shenzhen International Graduate School (QD2021013C). This work is also supported by Tsinghua University (AIR) - Asiainfo Technologies (China) Inc. Joint Research Center under grant no. 20203910074.

References

[1] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving, J. Field Robot. 37 (3) (2020) 362–386.
[2] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, H. Radha, Deep learning algorithm for autonomous driving using GoogLeNet, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 89–96.
[3] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, V.H.C. de Albuquerque, Deep learning for safe autonomous driving: current challenges and future directions, IEEE Trans. Intell. Transp. Syst. 22 (7) (2020) 4316–4336.
[4] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinform. 19 (6) (2018) 1236–1246.
[5] A.K. Sahoo, C. Pradhan, R.K. Barik, H. Dubey, DeepReco: deep learning based health recommender system using collaborative filtering, Computation 7 (2) (2019) 25.
[6] J.M. Philip, S. Durga, D. Esther, Deep learning application in IoT health care: a survey, in: Intelligence in Big Data Technologies–Beyond the Hype, Springer, 2021, pp. 199–208.
[7] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer, cuDNN: efficient primitives for deep learning, arXiv preprint arXiv:1410.0759 (2014).
[8] V. Sze, Y.-H. Chen, T.-J. Yang, J.S. Emer, Efficient processing of deep neural networks: atutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329.
[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.
[10] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, arXiv preprint arXiv:1811.03604 (2018).
[11] W. Yang, Y. Zhang, K. Ye, L. Li, C.-Z. Xu, FFD: a federated learning based method for credit card fraud detection, in: International Conference on Big Data, Springer, 2019, pp. 18–32.
[12] N. Rieke, J. Hancox, W. Li, F. Milletari, H.R. Roth, S. Albarqouni, S. Bakas, M.N. Galtier, B.A. Landman, K. Maier-Hein, et al., The future of digital health with federated learning, NPJ Digit. Med. 3 (1) (2020) 1–7.
[13] Z. Zhao, M. Luo, W. Ding, Deep leakage from model in federated learning, 2022.
[14] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. (2021) 1.
[15] O. Shahid, S. Pouriyeh, R.M. Parizi, Q.Z. Sheng, G. Srivastava, L. Zhao, Communication efficiency in federated learning: achievements and challenges, arXiv preprint arXiv:2107.10996 (2021).

[16] H. Xu, C.-Y. Ho, A.M. Abdelmoniem, A. Dutta, E.H. Bergou, K. Karatsenidis, M. Canini, P. Kalnis, Compressed Communication for Distributed Deep Learning: Survey and Quantitative Evaluation, Technical report, 2020.
[17] Z. Jiang, W. Wang, B. Li, Q. Yang, Towards efficient synchronous federated training: asurvey on system optimization strategies, IEEE Trans. Big Data (2022) 1.
[18] Z. Yang, M. Chen, K.-K. Wong, H.V. Poor, S. Cui, Federated learning for 6G: applications, challenges, and opportunities, Engineering 8 (2021) 33–41.
[19] X. Zhang, M. Hong, S. Dhople, W. Yin, Y. Liu, FedPD: a federated learning framework with optimal rates and adaptivity to non-iid data, arXiv preprint arXiv:2005.11418 (2020).
[20] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, T. Darrell, Face recognition with image sets using manifold density divergence, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE, 2005, pp. 581–588.
[21] P. Yu, Y. Liu, Federated object detection: optimizing object detection model with federated learning, in: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, 2019, pp. 1–6.
[22] M. Chen, B. Mao, T. Ma, FEDSA: a staleness-aware asynchronous federated learning algorithm with non-iid data, Future Gener. Comput. Syst. 120 (2021) 1–12.
[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. Learn. 3 (1) (2011) 1–122.
[24] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.
[25] R.M. Gray, D.L. Neuhoff, Quantization, IEEE Trans. Inf. Theory 44 (6) (1998) 2325–2383.
[26] D. Alistarh, D. Grubic, J. Li, R. Tomioka, M. Vojnovic, QSGD: communication-efficient SGD via gradient quantization and encoding, Adv. Neural Inf. Process. Syst. 30 (2017) 1707–1718.
[27] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, R. Pedarsani, FedPAQ: a communication-efficient federated learning method with periodic averaging and quantization, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2021–2031.
[28] F. Haddadpour, M.M. Kamani, A. Mokhtari, M. Mahdavi, Federated learning with compression: unified analysis and sharp guarantees, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 2350–2358.
[29] R. Das, A. Acharya, A. Hashemi, S. Sanghavi, I.S. Dhillon, U. Topcu, Faster non-convex federated learning via global and local momentum, arXiv preprint arXiv:2012.04061 (2020).
[30] X. Dai, X. Yan, K. Zhou, H. Yang, K.K. Ng, J. Cheng, Y. Fan, Hyper-sphere quantization: communication-efficient SGD for federated learning, arXiv preprint arXiv:1911.04655 (2019).
[31] D. Jhunjhunwala, A. Gadhikar, G. Joshi, Y.C. Eldar, Adaptive quantization of model updates for communication-efficient federated learning, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 3110–3114.
[32] M.M. Amiri, D. Gunduz, S.R. Kulkarni, H.V. Poor, Federated learning with quantized global model updates, arXiv preprint arXiv:2006.10672 (2020).
[33] A.T. Suresh, X.Y. Felix, S. Kumar, H.B. McMahan, Distributed mean estimation with limited communication, in: International Conference on Machine Learning, PMLR, 2017, pp. 3329–3337.
[34] S. Vargaftik, R.B. Basat, A. Portnoy, G. Mendelson, Y. Ben-Itzhak, M. Mitzenmacher, Drive: one-bit distributed mean estimation, Adv. Neural Inf. Process. Syst. 34 (2021) 362–377.
[35] S. Vargaftik, R.B. Basat, A. Portnoy, G. Mendelson, Y.B. Itzhak, M. Mitzenmacher, Eden: communication-efficient and robust distributed mean estimation for federated learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 21984–22014.
[36] R.B. Basat, S. Vargaftik, A. Portnoy, G. Einziger, Y. Ben-Itzhak, M. Mitzenmacher, Quick-fl: quick unbiased compression for federated learning, arXiv preprint arXiv:2205.13341 (2022).
[37] R. Zamir, M. Feder, On universal quantization by randomized uniform/lattice quantizers, IEEE Trans. Inf. Theory 38 (2) (1992) 428–436.
[38] N. Shlezinger, M. Chen, Y.C. Eldar, H.V. Poor, S. Cui, Federated learning with quantization constraints, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020a, pp. 8851–8855.
[39] N. Shlezinger, M. Chen, Y.C. Eldar, H.V. Poor, S. Cui, UVeQFed: universal vector quantization for federated learning, IEEE Trans. Signal Process. 69 (2020b) 500–514.
[40] M. Chen, N. Shlezinger, H.V. Poor, Y.C. Eldar, S. Cui, Communication-efficient federated learning, Proc. Natl. Acad. Sci. 118 (17) (2021). e2024789118

[41] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, A. Anandkumar, signSGD: compressed optimisation for non–convex problems, in: International Conference on Machine Learning, PMLR, 2018, pp. 560–569.

[42] R. Jin, Y. Huang, X. He, H. Dai, T. Wu, Stochastic-sign SGD for federated learning with theoretical guarantees, arXiv preprint arXiv:2002.10940 (2020).

[43] G. Zhu, Y. Du, D. Gündüz, K. Huang, One-bit over-the-air aggregation for communication-efficient federated edge learning: design and convergence analysis, IEEE Trans. Wirel. Commun. 20 (3) (2020) 2120–2135.

[44] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.

[45] A. Abdi, F. Fekri, Quantized compressive sampling of stochastic gradients for efficient communication in distributed deep learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 3105–3112.

[46] C. Li, G. Li, P.K. Varshney, Communication-efficient federated learning based on compressed sensing, IEEE Internet Things J. 8 (20) (2021) 15531–15541.

[47] X. Fan, Y. Wang, Y. Huo, Z. Tian, Communication-efficient federated learning through 1-bit compressive sensing and analog aggregation, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2021, pp. 1–6.

[48] Y. Oh, N. Lee, Y.-S. Jeon, Quantized compressed sensing for communication-efficient federated learning, in: 2021 IEEE Globecom Workshops (GC Wkshps), IEEE, 2021, pp. 1–6.

[49] Y. He, M. Zenk, M. Fritz, CosSGD: nonlinear quantization for communication-efficient federated learning, CoRR abs/2012.08241 (2020). arXiv:2012.08241.

[50] A. Malekijoo, M.J. Fadaeieslam, H. Malekijou, M. Homayounfar, F. Alizadeh-Shabdiz, R. Rawassizadeh, FEDZIP: a compression framework for communication-efficient federated learning, arXiv preprint arXiv:2102.01593 (2021).

[51] C. Philippenko, A. Dieuleveut, Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees, arXiv preprint arXiv:2006.14591 (2020).

[52] S. Chen, C. Shen, L. Zhang, Y. Tang, Dynamic aggregation for heterogeneous quantization in federated learning, IEEE Trans. Wirel. Commun. 20 (10) (2021) 6804–6819.

[53] L. Cui, X. Su, Y. Zhou, Y. Pan, Slashing communication traffic in federated learning by transmitting clustered model updates, IEEE J. Sel. Areas Commun. 39 (8) (2021) 2572–2589.

[54] N.F. Eghlidi, M. Jaggi, Sparse communication for training deep networks, arXiv preprint arXiv:2009.09271 (2020).

[55] S.U. Stich, J.-B. Cordonnier, M. Jaggi, Sparsified SGD with memory, Adv. Neural Inf. Process. Syst. 31 (2018) 4452–4463.

[56] J. Wangni, J. Wang, J. Liu, T. Zhang, Gradient sparsification for communication-efficient distributed optimization, Adv. Neural Inf. Process. Syst. 31 (2018) 1306–1316.

[57] S. Shi, X. Chu, K.C. Cheung, S. See, Understanding top-*k* sparsification in distributed deep learning, arXiv preprint arXiv:1911.08772 (2019).

[58] A.F. Aji, K. Heafield, Sparse communication for distributed gradient descent, arXiv preprint arXiv:1704.05021 (2017).

[59] Y. Lin, S. Han, H. Mao, Y. Wang, W.J. Dally, Deep gradient compression: reducing the communication bandwidth for distributed training, arXiv preprint arXiv:1712.01887 (2017).

[60] A. Sahu, A. Dutta, A. M Abdelmoniem, T. Banerjee, M. Canini, P. Kalnis, Rethinking gradient sparsification as total error minimization, Adv. Neural Inf. Process. Syst. 34 (2021) 8133–8146.

[61] P. Han, S. Wang, K.K. Leung, Adaptive gradient sparsification for efficient federated learning: an online learning approach, in: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), 2020, pp. 300–310.

[62] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Sparse binary compression: towards distributed deep learning with minimal communication, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.

[63] M.K. Nori, S. Yun, I.-M. Kim, Fast federated learning by balancing communication trade-offs, Trans. Commun. 69 (8) (2021) 5168–5182.

[64] A.M. Abdelmoniem, M. Canini, Dc2: delay-aware compression control for distributed machine learning, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[65] S. Li, Q. Qi, J. Wang, H. Sun, Y. Li, F.R. Yu, GGS: general gradient sparsification for federated learning in edge computing, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–7.

[66] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, R. Arora, FetchSGD: communication-efficient federated learning with sketching, in: H. Daumé, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 8253–8265.

[67] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-i.i.d. data, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2020) 3400–3413.

[68] S. Shi, Q. Wang, K. Zhao, Z. Tang, Y. Wang, X. Huang, X. Chu, A distributed synchronous SGD algorithm with global top-k sparsification for low bandwidth networks, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 2238–2247.

[69] H. Xu, K. Kostopoulou, A. Dutta, X. Li, A. Ntoulas, P. Kalnis, Deepreduce: a sparse-tensor communication framework for federated deep learning, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 21150–21163.

[70] E. Ozfatura, K. Ozfatura, D. Gündüz, Time-correlated sparsification for communication-efficient federated learning, in: 2021 IEEE International Symposium on Information Theory (ISIT), IEEE, 2021, pp. 461–466.

[71] Y. Li, M. Yu, S. Li, S. Avestimehr, N.S. Kim, A. Schwing, Pipe-SGD: a decentralized pipelined SGD framework for distributed deep net training, Adv. Neural Inf. Process. Syst. 31 (2018) 8056–8067.

[72] A. Harlap, D. Narayanan, A. Phanishayee, V. Seshadri, N. Devanur, G. Ganger, P. Gibbons, Pipedream: fast and efficient pipeline parallel DNN training, arXiv preprint arXiv:1806.03377 (2018).

[73] S. Shi, X. Chu, B. Li, MG-WFBP: efficient data communication for distributed synchronous SGD algorithms, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019a, pp. 172–180.

[74] S. Shi, Z. Tang, Q. Wang, K. Zhao, X. Chu, Layer-wise adaptive gradient sparsification for distributed deep learning with convergence guarantees, arXiv preprint arXiv:1911.08727 (2019b).

[75] Y. You, A. Buluç, J. Demmel, Scaling deep learning on GPU and knights landing clusters, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, pp. 1–12.

[76] A. Sergeev, M. Del Balso, Horovod: fast and easy distributed deep learning in tensorflow, arXiv preprint arXiv:1802.05799 (2018).

[77] X. Jia, S. Song, W. He, Y. Wang, H. Rong, F. Zhou, L. Xie, Z. Guo, Y. Yang, L. Yu, et al., Highly scalable deep learning training system with mixed-precision: training imagenet in four minutes, arXiv preprint arXiv:1807.11205 (2018).

[78] S. Shi, Q. Wang, X. Chu, B. Li, Y. Qin, R. Liu, X. Zhao, Communication-efficient distributed deep learning with merged gradient sparsification on GPUs, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 406–415.

[79] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 2(7) (2015).

[80] D. Li, J. Wang, FedMD: heterogenous federated learning via model distillation, arXiv preprint arXiv:1910.03581 (2019).

[81] T. Lin, L. Kong, S.U. Stich, M. Jaggi, Ensemble distillation for robust model fusion in federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 2351–2363.

[82] F. Sattler, A. Marban, R. Rischke, W. Samek, CFD: communication-efficient federated distillation via soft-label quantization and delta coding, IEEE Trans. Netw. Sci. Eng. 9 (2021) 2025–2038.

[83] S. Itahara, T. Nishio, Y. Koda, M. Morikura, K. Yamamoto, Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data, IEEE Trans. Mob. Comput. 22 (2021) 191–205.

[84] F. Sattler, T. Korjakow, R. Rischke, W. Samek, FEDAUX: leveraging unlabeled auxiliary data in federated learning, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–13.

[85] X. Li, Y. Gong, Y. Liang, L.-E. Wang, Personalized federated learning with semisupervised distillation, Secur. Commun. Netw. 2021 (2021).

[86] S.P. Sturluson, S. Trew, L. Muñoz-González, M. Grama, J. Passerat-Palmbach, D. Rueckert, A. Alansary, FedRAD: federated robust adaptive distillation, arXiv preprint arXiv:2112.01405 (2021).

[87] L. Liu, J. Zhang, S. Song, K.B. Letaief, Communication-efficient federated distillation with active data sampling, arXiv preprint arXiv:2203.06900 (2022).

[88] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, S.-L. Kim, Communication-efficient on-device machine learning: federated distillation and augmentation under non-iid private data, arXiv preprint arXiv:1811.11479 (2018).

[89] D. Jiang, C. Shan, Z. Zhang, Federated learning algorithm based on knowledge distillation, in: 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), IEEE, 2020, pp. 163–167.
[90] C. Wu, F. Wu, L. Lyu, Y. Huang, X. Xie, Communication-efficient federated learning via knowledge distillation, Nat. Commun. 13 (1) (2022) 1–8.
[91] Z. Zhu, J. Hong, J. Zhou, Data-free knowledge distillation for heterogeneous federated learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 12878–12889.
[92] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, H. Jin, Z. Xu, L. Sun, Local-global knowledge distillation in heterogeneous federated learning with non-iid data, arXiv preprint arXiv:2107.00051 (2021).
[93] L. Zhang, L. Shen, L. Ding, D. Tao, L.-Y. Duan, Fine-tuning global model via data-free knowledge distillation for non-iid federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10174–10183.
[94] Y.J. Cho, A. Manoel, G. Joshi, R. Sim, D. Dimitriadis, Heterogeneous ensemble knowledge transfer for training large models in federated learning, arXiv preprint arXiv:2204.12703 (2022).
[95] Z. Zhu, J. Hong, S. Drew, J. Zhou, Resilient and communication efficient learning for heterogeneous federated systems, in: International Conference on Machine Learning, PMLR, 2022, pp. 27504–27526.
[96] L. Zhang, D. Wu, X. Yuan, FedZKT: zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models, in: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), IEEE, 2022, pp. 928–938.
[97] K. Ozkara, N. Singh, D. Data, S. Diggavi, QuPeD: quantized personalization via distillation with applications to federated learning, Adv. Neural Inf. Process. Syst. 34 (2021) 3622–3634.
[98] P. Kairouz, S. Oh, P. Viswanath, Extremal mechanisms for local differential privacy, Adv. Neural Inf. Process. Syst. 27 (2014) 2879–2887.
[99] W. Du, M.J. Atallah, Secure multi-party computation problems and their applications: a review and open problems, in: Proceedings of the 2001 Workshop on New Security Paradigms, 2001, pp. 13–22.
[100] L. Sun, L. Lyu, Federated model distillation with noise-free differential privacy, arXiv preprint arXiv:2009.05537 (2020).
[101] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis, S.-L. Kim, Mix2FLD: downlink federated learning after uplink federated distillation with two-way mixup, IEEE Commun. Lett. 24 (10) (2020) 2211–2215.
[102] H. Shi, Y. Zhang, Z. Shen, S. Tang, Y. Li, Y. Guo, Y. Zhuang, Towards communication-efficient and privacy-preserving federated representation learning, arXiv preprint arXiv:2109.14611 (2021).
[103] C. Wu, S. Zhu, P. Mitra, Federated unlearning with knowledge distillation, arXiv preprint arXiv:2201.09441 (2022).
[104] H. Chang, V. Shejwalkar, R. Shokri, A. Houmansadr, Cronus: robust and heterogeneous collaborative learning with black-box knowledge transfer, arXiv preprint arXiv:1912.11279 (2019).
[105] H. Cha, J. Park, H. Kim, S.-L. Kim, M. Bennis, Federated reinforcement distillation with proxy experience memory, arXiv preprint arXiv:1907.06536 (2019).
[106] H. Cha, J. Park, H. Kim, M. Bennis, S.-L. Kim, Proxy experience replay: federated distillation for distributed reinforcement learning, IEEE Intell. Syst. 35 (4) (2020) 94–101.
[107] D. Sui, Y. Chen, J. Zhao, Y. Jia, Y. Xie, W. Sun, Feded: federated learning via ensemble distillation for medical relation extraction, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 2118–2128.
[108] C. Sun, T. Jiang, S. Zonouz, D. Pompili, Fed2KD: heterogeneous federated learning for pandemic risk assessment via two-way knowledge distillation, in: 2022 17th Wireless On-Demand Network Systems and Services Conference (WONS), IEEE, 2022, pp. 1–8.
[109] J.-H. Ahn, O. Simeone, J. Kang, Wireless federated distillation for distributed edge learning with heterogeneous data, in: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE, 2019, pp. 1–6.
[110] J.-H. Ahn, O. Simeone, J. Kang, Cooperative learning via federated distillation over fading channels, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 8856–8860.
[111] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, J.J. Rodrigues, V.H.C. de Albuquerque, Industrial cyber-physical systems-based cloud IoTedge for federated heterogeneous distillation, IEEE Trans. Ind. Inf. 17 (8) (2020) 5511–5521.
[112] T. Li, M. Sanjabi, A. Beirami, V. Smith, Fair resource allocation in federated learning, arXiv preprint arXiv:1905.10497 (2019).

[113] N. Hyeon-Woo, M. Ye-Bin, T.-H. Oh, FedPara: low-rank hadamard product for communication-efficient federated learning, arXiv preprint arXiv:2108.06098 (2021).

[114] J. Sun, T. Chen, G.B. Giannakis, Q. Yang, Z. Yang, Lazily aggregated quantized gradient innovation for communication-efficient federated learning, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2020) 2031–2044.

[115] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: strategies for improving communication efficiency, arXiv preprint arXiv:1610.05492 (2016).

[116] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular value decomposition and principal component analysis, in: A Practical Approach to Microarray Data Analysis, Springer, 2003, pp. 91–109.

[117] S.S. Azam, S. Hosseinalipour, Q. Qiu, C. Brinton, Recycling model updates in federated learning: are gradient subspaces low-rank? in: International Conference on Learning Representations, 2021.

[118] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[119] O. Marfoq, C. Xu, G. Neglia, R. Vidal, Throughput-optimal topology design for cross-silo federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 19478–19487.

[120] Y. Guo, Y. Sun, R. Hu, Y. Gong, Hybrid local SGD for federated learning with heterogeneous communications, in: International Conference on Learning Representations, 2022.

[121] M.M. Amiri, D. Gündüz, Federated learning over wireless fading channels, IEEE Trans. Wirel. Commun. 19 (5) (2020) 3546–3557.

[122] W. Wu, L. He, W. Lin, R. Mao, C. Maple, S. Jarvis, SAFA: a semi-asynchronous protocol for fast federated learning with low overhead, IEEE Trans. Comput. 70 (5) (2020) 655–668.

[123] M. Salehi, E. Hossain, Federated learning in unreliable and resource-constrained cellular wireless networks, IEEE Trans. Commun. 69 (8) (2021) 5136–5151.

[124] W. Wu, L. He, W. Lin, R. Mao, Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2020) 1539–1551.

[125] C. Yu, H. Tang, C. Renggli, S. Kassing, A. Singla, D. Alistarh, C. Zhang, J. Liu, Distributed learning over unreliable networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7202–7212.

[126] X. Zhang, X. Zhu, J. Wang, H. Yan, H. Chen, W. Bao, Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks, Inf. Sci. 540 (2020) 242–262.

[127] H. Ye, L. Liang, G.Y. Li, Decentralized federated learning with unreliable communications, IEEE J. Sel. Top. Signal Process. 16 (3) (2022) 487–500.

[128] G. Zhu, Y. Wang, K. Huang, Broadband analog aggregation for low-latency federated edge learning, IEEE Trans. Wirel. Commun. 19 (1) (2019) 491–506.

[129] H. Yang, P. Qiu, J. Liu, A. Yener, Over-the-air federated learning with joint adaptive computation and power control, arXiv preprint arXiv:2205.05867 (2022).

[130] N. Zhang, M. Tao, Gradient statistics aware power control for over-the-air federated learning in fading channels, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–6.

[131] H. Guo, A. Liu, V.K. Lau, Analog gradient aggregation for federated learning over wireless networks: customized design and convergence analysis, IEEE Internet Things J. 8 (1) (2020) 197–210.

[132] T. Sery, K. Cohen, On analog gradient descent learning over multiple access fading channels, IEEE Trans. Signal Process. 68 (2020) 2897–2911.

[133] H. Hellström, V. Fodor, C. Fischione, Over-the-air federated learning with retransmissions, in: 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2021, pp. 291–295.

[134] Z. Lin, H. Liu, Y.-J. A. Zhang, Relay-assisted cooperative federated learning, IEEE Trans. Wirel. Commun. (2022).

[135] S. Xia, J. Zhu, Y. Yang, Y. Zhou, Y. Shi, W. Chen, Fast convergence algorithm for analog federated learning, in: ICC 2021-IEEE International Conference on Communications, IEEE, 2021, pp. 1–6.

[136] M.M. Amiri, D. Gündüz, S.R. Kulkarni, H.V. Poor, Convergence of federated learning over a noisy downlink, IEEE Trans. Wirel. Commun. 21 (3) (2021) 1422–1437.

[137] M.B. Mashhadi, N. Shlezinger, Y.C. Eldar, D. Gündüz, FedRec: federated learning of universal receivers over fading channels, in: 2021 IEEE Statistical Signal Processing Workshop (SSP), IEEE, 2021, pp. 576–580.

[138] X. Wei, C. Shen, Federated learning over noisy channels, in: ICC 2021-IEEE International Conference on Communications, IEEE, 2021, pp. 1–6.

[139] M.M. Amiri, S.R. Kulkarni, H.V. Poor, Federated learning with downlink device selection, in: 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2021, pp. 306–310.

[140] L. Li, L. Yang, X. Guo, Y. Shi, H. Wang, W. Chen, K.B. Letaief, Delay analysis of wireless federated learning based on saddle point approximation and large deviation theory, IEEE J. Sel. Areas Commun. 39 (12) (2021) 3772–3789.

[141] Y. Mao, Z. Zhao, M. Yang, L. Liang, Y. Liu, W. Ding, T. Lan, X.-P. Zhang, Safari: sparsity enabled federated learning with limited and unreliable communications, arXiv preprint arXiv:2204.02321 (2022).

[142] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, J. Liu, Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent, Adv. Neural Inf. Process. Syst. 30 (2017) 5336–5346.

[143] M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, G. Pekhimenko, Moshpit SGD: communication-efficient decentralized training on heterogeneous unreliable devices, Adv. Neural Inf. Process. Syst. 34 (2021) 18195–18211.

[144] W. Shi, S. Zhou, Z. Niu, M. Jiang, L. Geng, Joint device scheduling and resource allocation for latency constrained wireless federated learning, IEEE Trans. Wirel. Commun. 20 (1) (2020) 453–467.

[145] C. Yu, S. Shen, K. Zhang, H. Zhao, Y. Shi, Energy-aware device scheduling for joint federated learning in edge-assisted internet of agriculture things, in: 2022 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2022, pp. 1140–1145.

[146] A. Mahmoudi, H.S. Ghadikolaei, J.M.B.D.S. Júnior, C. Fischione, FedCau: a proactive stop policy for communication and computation efficient federated learning, arXiv preprint arXiv:2204.07773 (2022).

[147] S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221.

[148] M. Chen, H.V. Poor, W. Saad, S. Cui, Convergence time optimization for federated learning over wireless networks, IEEE Trans. Wirel. Commun. 20 (4) (2020) 2457–2471.

[149] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, H.V. Poor, S. Cui, Delay minimization for federated learning over wireless communication networks, arXiv preprint arXiv:2007.03462 (2020).

[150] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks, IEEE Trans. Ind. Inf. 17 (7) (2020) 5098–5107.

[151] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019-2019 IEEE international conference on communications (ICC), IEEE, 2019, pp. 1–7.

[152] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, H.V. Poor, Federated learning over wireless IoT networks with optimized communication and resources, IEEE Internet Things J. 9 (2022) 16592–16605.

[153] Z. Zhao, J. Xia, L. Fan, X. Lei, G.K. Karagiannidis, A. Nallanathan, System optimization of federated learning networks with a constrained latency, IEEE Trans. Veh. Technol. 71 (1) (2021) 1095–1100.

[154] Q. Zeng, Y. Du, K. Huang, K.K. Leung, Energy-efficient radio resource allocation for federated edge learning, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–6.

[155] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, IEEE Trans. Wirel. Commun. 20 (3) (2020) 1935–1949.

[156] A. Imteaj, M.H. Amini, FedAR: activity and resource-aware federated learning model for distributed mobile robots, in: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2020, pp. 1153–1160.

[157] C.W. Zaw, S.R. Pandey, K. Kim, C.S. Hong, Energy-aware resource management for federated learning in multi-access edge computing systems, IEEE Access 9 (2021) 34938–34950.

[158] K. Mishchenko, B. Wang, D. Kovalev, P. Richtárik, IntSGD: adaptive floatless compression of stochastic gradients, in: International Conference on Learning Representations, 2022.

[159] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, VerifyNet: secure and verifiable federated learning, IEEE Trans. Inf. Forensics Secur. 15 (2019) 911–926.

[160] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: attacks and defenses, arXiv preprint arXiv:1705.07204 (2017).

[161] A.A. Abd EL-Latif, B. Abd-El-Atty, S.E. Venegas-Andraca, W. Mazurczyk, Efficient quantum-based security protocols for information sharing and data protection in 5G networks, Future Gener. Comput. Syst. 100 (2019) 893–906.

[162] White paper for federated learning in mobile communication networks, China Mobile Communications Research Institute, 2021.

[163] Study on enablers for network automation for the 5G System (5GS), 3GPP TR 23.700-91, 2020.

[164] Y. Xiao, G. Shi, M. Krunz, Towards ubiquitous ai in 6G with federated learning, arXiv preprint arXiv:2004.13563 (2020).

[165] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, D. Niyato, Federated learning for 6G communications: challenges, methods, and future directions, China Commun. 17 (9) (2020a) 105–118.

[166] Y. Liu, J. James, J. Kang, D. Niyato, S. Zhang, Privacy-preserving traffic flow prediction: afederated learning approach, IEEE Internet Things J. 7 (8) (2020b) 7751–7763.