Robust Federated Learning for Edge Intelligence

First Author[1][0000−1111−2222−3333], Second Author[2,3][1111−2222−3333−4444], and Third Author[3][2222−−3333−4444−5555]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Artificial intelligence (AI) has revolutionized daily life and greatly benefited human society. AI-powered systems are now used in diverse domains, ranging from entertainment, e-commerce, and social media to healthcare, finance, and defense. However, as AI is increasingly used in critical and sensitive areas, such as medical diagnosis, financial fraud detection, and military surveillance, the trustworthiness and reliability of the AI models become crucial. Ensuring the transparency, accountability, and fairness of AI systems is essential to increase their social acceptance and adoption, reduce their risks and harms, and enhance their benefits and opportunities.

One of the emerging and promising solutions to achieve trustworthy AI is edge federated learning (EFL). EFL combines edge computing and federated learning (FL) to enable more efficient and collaborative machine learning (ML) at the edge of the network [36]. In EFL, multiple edge devices (e.g., smartphones, tablets, laptops, IoT devices, etc.) and edge servers (e.g., base stations, access points, routers, cloudlets, etc.) collaboratively train a global model without sharing their local data. EFL has some unique features compared with traditional FL, such as:

- Offloading part of the computation from edge devices to edge servers, which can reduce the computational cost and battery consumption of the devices.
- Using edge servers to aggregate the local model updates and reduce the communication cost and frequency between the devices and the cloud server.
- Benefiting from over-the-air computation (OTA-C), which is a technique that simultaneously uploads and updates model parameters via analog beamforming, avoiding sophisticated baseband signal processing and reducing communication delays and implementation costs.

– Enabling bi-directional knowledge transfer (BKT) between the edge and the cloud, sharing feature embeddings and prediction logits, which can enhance personalization, enable model heterogeneity, tolerate training asynchronization, and relieve communication burdens.

EFL has many promising applications in diverse real-world scenarios, such as autonomous driving, smart cities, and healthcare. For example, in autonomous driving, EFL can enable the sharing of the driving behavior data among multiple cars and improve the accuracy and safety of the driving model. In smart cities, EFL can enable the collaboration of multiple sensors and devices to monitor and predict traffic, pollution, and energy consumption. In healthcare, EFL can enable the sharing of patient data among multiple hospitals and clinics, and improve the accuracy and fairness of the medical diagnosis and treatment.

However, EFL also faces significant challenges in ensuring the trustworthiness of the learning process and the quality of the learned model. In particular, EFL needs to deal with *communication bottleneck*, *byzantine resilience*, *privacy preservation*, and *heterogeneity*. These challenges can affect the convergence speed, accuracy, and security of EFL, and limit its potential benefits and applications. Therefore, developing trustworthy edge federated learning (TEFL) is a critical research direction that can enhance the transparency, accountability, and fairness of AI systems, and enable more efficient and sustainable AI.

In this chapter, we provide a comprehensive overview of TEFL, which aims to address the aforementioned challenges and achieve trustworthy and efficient EFL. We review the state-of-the-art techniques and solutions for TEFL from four perspectives: communication bottleneck, byzantine resilience, privacy preservation, and heterogeneity. By providing a detailed and structured analysis of TEFL, we hope to contribute to the advancement of trustworthy AI and the development of practical TEFL systems. Our review aims to assist researchers, practitioners, and policy-makers in understanding the state-of-the-art techniques and solutions for TEFL and their respective strengths, weaknesses, and applicability. Our review also highlights some open problems and future research directions in TEFL, which can inspire further research and innovation in this exciting and important field.

The rest of this chapter is organized as follows. In Section II, we provide a brief overview of the FL and edge computing paradigms, and introduce the basic concepts and framework of EFL. In Section III, we discuss the communication bottleneck issue in EFL, and present some techniques and solutions to mitigate its impact. In Section IV, we address the byzantine resilience issue in EFL, and introduce some approaches to detect and tolerate malicious edge devices. In Section V, we focus on the privacy preservation issue in EFL, and describe some methods to protect the sensitive data of the edge devices while enabling collaborative learning. In Section VI, we deal with the heterogeneity issue in EFL, and explore some strategies to handle the differences in computation capability, communication resource, availability, and data distribution among the edge devices.

2 Background and Framework of Edge Federated Learning

In this section, we provide a brief background of the FL and edge computing paradigms, and introduce the basic concepts and framework of EFL.

2.1 Federated Learning

Federated Learning (FL) is a distributed machine learning paradigm that enables multiple clients to collaboratively train a global model without sharing their local data. The idea of FL was first proposed by McMahan et al. [**?**] in 2016, and has since gained increasing attention from both academia and industry. In FL, each client (also known as a participant, device, or node) has its own local dataset, and performs local model training using its data. The local model updates are then sent to a central server (also known as an aggregator, coordinator, or parameter server), which aggregates the updates and updates the global model. The updated global model is then sent back to the clients, and the process is repeated iteratively until the model converges.

FL has several advantages over traditional centralized machine learning, such as preserving data privacy, reducing communication costs, and enabling better scalability and generalization. However, FL also faces several challenges, such as communication bottleneck, heterogeneity, and security and privacy issues.

2.2 Edge Computing

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the edge of the network, where the data is generated and consumed. The idea of edge computing is motivated by the increasing demand for real-time and low-latency applications, as well as the growing amount of data generated by the Internet of Things (IoT) devices. In edge computing, the computation and storage resources are distributed across multiple edge devices (such as smartphones, tablets, laptops, and IoT devices) and edge servers (such as base stations, access points, and routers), which form a decentralized network.

Edge computing has several advantages over traditional cloud computing, such as reducing the network latency, improving the data privacy and security, and enabling real-time and context-aware applications. However, edge computing also faces several challenges, such as resource constraints, heterogeneity, and scalability and management issues.

2.3 Edge Federated Learning

Edge Federated Learning (EFL) is a combination of FL and edge computing, which aims to enable more efficient and privacy-preserving machine learning at the edge of the network. In EFL, multiple edge devices and edge servers collaboratively train a global model without sharing their local data. EFL can offload

part of the computation from edge devices to edge servers, which can reduce the computational cost and battery consumption of the devices. EFL can also use edge servers to aggregate the local model updates and reduce the communication cost and frequency between the devices and the cloud server. EFL can benefit from over-the-air computation (OTA-C), which is a technique that simultaneously uploads and updates model parameters via analog beamforming, avoiding sophisticated baseband signal processing and reducing communication delays and implementation costs. EFL can enable bi-directional knowledge transfer (BKT) between the edge and the cloud, sharing feature embeddings and prediction logits, which can enhance personalization, enable model heterogeneity, tolerate training asynchronization, and relieve communication burdens.

The basic framework of EFL consists of three phases: local training, global aggregation, and global updating. In the local training phase, each edge device performs local model training using its local data. In the global aggregation phase, the local model updates are sent to the edge server, which aggregates the updates and computes a new global model. In the global updating phase, the updated global model is sent back to the edge devices, which use the global model to perform the next round of local training. The three phases are repeated iteratively until the global model converges.

EFL has several advantages over both FL and edge computing. EFL can leverage the benefits of both paradigms, such as preserving data privacy, reducing communication costs, improving the scalability and generalization, reducing the computation and battery consumption, improving the real-time and low-latency performance, and enabling better context-awareness and personalization. EFL can also address the challenges of both paradigms, such as communication bottleneck, heterogeneity, security and privacy issues, resource constraints, heterogeneity, and scalability and management issues.

However, EFL also faces several research challenges and open issues. For example, how to design efficient and scalable algorithms for global aggregation and updating in a heterogeneous and dynamic network, how to address the security and privacy issues of EFL in a decentralized and untrusted environment, how to balance the trade-off between the computation and communication costs of EFL, how to enable model heterogeneity and adaptive model selection in EFL, and how to integrate EFL with other emerging technologies, such as blockchain, 5G, and edge AI.

In the next section, we will review the state-of-the-art EFL research and discuss the existing solutions and limitations.

3 Communication Bottleneck in Edge Federated Learning

In this section, we aim to provide a comprehensive overview of communication-efficient EFL. We first define the concept and characteristics of communication bottleneck in edge federated learning. We then review the existing solutions and techniques for reducing communication overhead and latency in edge federated learning. We also compare and evaluate the performance and trade-

offs of different solutions and techniques. Next, we present a classical paper on communication-efficient edge federated learning and critically analyze its contribution and limitation. Finally, we discuss some open challenges and future directions for communication-efficient edge federated learning.

The organization of this section are as follows:

- We provide a clear definition and analysis of communication bottleneck in edge federated learning, and highlight its main challenges and impacts on the learning performance and scalability.
- We categorize the existing solutions and techniques for communication-efficient edge federated learning based on some criteria (e.g., architecture, protocol, algorithm, etc.), and describe their main idea, design, implementation, and evaluation in detail.
- We analyze the performance and trade-offs of different solutions and techniques in terms of communication overhead, latency, accuracy, robustness, privacy, etc., and identify their strengths and weaknesses.
- We select a classical paper on communication-efficient edge federated learning that has high impact and relevance to the topic of this section, and introduce its motivation, problem formulation, main contribution, methodology, results, insights, implications, novelty, soundness, clarity, significance, and limitations.
- We conclude this section, and identify some open challenges and research gaps that need to be addressed in communication-efficient edge federated learning. We also provide some future directions and research opportunities for communication-efficient edge federated learning.

3.1 Communication Bottleneck in Edge Federated Learning

Communication bottleneck is one of the major challenges in edge federated learning (EFL), which refers to the high communication cost and latency caused by the limited bandwidth and unreliable connections between edge devices and servers [28]. Communication bottleneck not only affects the efficiency and scalability of EFL, but also impacts the accuracy and convergence of the global model [36]. The communication bottleneck in EFL is influenced by several factors, such as the number and distribution of edge devices, the size and frequency of model updates, the network topology and congestion, etc. [49]. For example, increasing the number of edge devices can improve the diversity and coverage of data, but also increase the communication overhead and complexity. Similarly, increasing the size or frequency of model updates can improve the model accuracy and convergence, but also increase the communication bandwidth and latency. Moreover, different network topologies and congestion levels can affect the reliability and stability of communication in EFL. Therefore, it is important to understand the causes and effects of communication bottleneck in EFL and to explore possible solutions that can overcome this challenge.

3.2 Communication-Efficient Solutions and Techniques for Edge Federated Learning

Communication-efficient solutions and techniques for edge federated learning aim to reduce the communication overhead and latency between the edge nodes and the central aggregator, which can affect the learning performance and scalability. We categorize these solutions and techniques into four groups based on some criteria: architecture-based, protocol-based, algorithm-based, and hybrid-based.

3.2.1 Architecture-Based Solutions and Techniques Architecture-based solutions and techniques focus on designing different architectures for edge federated learning that can distribute the aggregation tasks among multiple edge servers or devices. These architectures can be classified into three types: hierarchical, decentralized, and hybrid.

Hierarchical architectures adopt a multi-level structure that divides the edge nodes into different clusters or groups, each with a local aggregator or leader. The local aggregators or leaders are responsible for aggregating the model parameters from their cluster members or group members, and then sending them to the central aggregator or a higher-level aggregator. This way, the communication overhead and latency between the edge nodes and the central aggregator can be reduced [42]. However, hierarchical architectures may introduce additional communication overhead and latency within each cluster or group, as well as increase the complexity of coordination and synchronization [73].

Decentralized architectures adopt a peer-to-peer structure that eliminates the central aggregator or any intermediate aggregators. The edge nodes directly communicate with each other to exchange their model parameters or gradients. This way, the communication overhead and latency between the edge nodes and the central aggregator can be eliminated [32]. However, decentralized architectures may require more communication rounds or messages among the edge nodes, as well as face challenges of consensus and robustness [73].

Hybrid architectures adopt a combination of hierarchical and decentralized structures that can balance the trade-offs between them. The edge nodes are divided into different clusters or groups, each with a local aggregator or leader. The local aggregators or leaders communicate with each other in a peer-to-peer manner to exchange their model parameters or gradients. This way, the communication overhead and latency between the edge nodes and the central aggregator can be reduced, while avoiding additional communication overhead and latency within each cluster or group [10]. However, hybrid architectures may still suffer from complexity of coordination and synchronization, as well as challenges of consensus and robustness [73].

3.2.2 Protocol-Based Solutions and Techniques Protocol-based solutions and techniques focus on designing different protocols for edge federated learning that can optimize the communication process between the edge nodes and the central aggregator. These protocols can be classified into two types: model download protocol (MDP) and model upload protocol (MUP).

MDP is responsible for distributing the global model parameters from the central aggregator to the edge nodes. MDP can reduce the communication overhead by compressing, encoding, or quantizing the model parameters before sending them [87]. MDP can also reduce the communication latency by using multicast, broadcast, or anycast techniques to deliver the model parameters in parallel [1].

MUP is responsible for collecting the local model parameters or gradients from the edge nodes to the central aggregator. MUP can reduce the communication overhead by compressing, encoding, or quantizing the model parameters or gradients before sending them [87]. MUP can also reduce the communication latency by using aggregation, coding, or scheduling techniques to combine or coordinate the model parameters or gradients in parallel [1].

3.2.3 Algorithm-Based Solutions and Techniques Algorithm-based solutions and techniques focus on designing different algorithms for edge federated learning that can improve the communication efficiency by reducing the frequency or amount of communication between the edge nodes and the central aggregator. These algorithms can be classified into two types: local update algorithm (LUA) and global update algorithm (GUA).

LUA is responsible for updating the local model parameters or gradients at each edge node based on its local training data. LUA can reduce the frequency of communication by increasing the number of local iterations or epochs before sending the model parameters or gradients to the central aggregator [28]. LUA can also reduce the amount of communication by selecting a subset of model parameters or gradients that have large changes or high importance to send to the central aggregator [74].

GUA is responsible for updating the global model parameters at the central aggregator based on the received model parameters or gradients from the edge nodes. GUA can reduce the frequency of communication by decreasing the number of global iterations or epochs before distributing the model parameters to the edge nodes [54]. GUA can also reduce the amount of communication by aggregating only a fraction of model parameters or gradients from a subset of edge nodes that have high quality or diversity [33].

3.2.4 Hybrid-Based Solutions and Techniques Hybrid-based solutions and techniques focus on combining different solutions and techniques from architecture-based, protocol-based, and algorithm-based groups to achieve better communication efficiency and performance for edge federated learning. These solutions and techniques can be classified into two types: cross-group hybrid and within-group hybrid.

Cross-group hybrid solutions and techniques integrate different solutions and techniques from different groups to complement each other. For example, FedEdge [73] combines hierarchical architecture, gradient compression, and adaptive local update to reduce both inter-cluster and intra-cluster communication overhead and latency. FedZip [87] combines decentralized architecture, gradient

encoding, and adaptive global update to eliminate central aggregator bottleneck and improve communication efficiency.

Within-group hybrid solutions and techniques integrate different solutions and techniques from the same group to enhance each other. For example, FedHealth [10] combines hierarchical architecture, peer-to-peer architecture, and cluster-based architecture to balance communication overhead, latency, accuracy, robustness, and privacy. FedGG [72] combines adaptive quantization, adaptive loss weight, and global update guidance to balance communication efficiency and accuracy trade-off.

3.3 A Classical Paper on Communication-Efficient Edge Federated Learning

In this section, we introduce a classical paper on communication-efficient edge federated learning that has high impact and relevance to the survey topic. We summarize its motivation, problem formulation, main contribution, methodology, results, insights, implications, novelty, soundness, clarity, significance, and limitations.

The paper we select is [72], titled "Global Update Guidance for Communication-Efficient Federated Learning". This paper was published in the Proceedings of the 38th International Conference on Machine Learning (ICML) in 2021.

3.3.1 Motivation The motivation of this paper is to address the communication efficiency and accuracy trade-off in federated learning. The paper observes that existing communication-efficient solutions or techniques often suffer from accuracy degradation due to the lossy compression or quantization of model parameters or gradients. The paper argues that the key to achieving both communication efficiency and accuracy is to balance the trade-off between the global model quality and the local model diversity. The paper proposes a novel global update guidance (GUG) framework that can guide the edge devices to update their local models towards a high-quality global model while preserving their local diversity.

3.3.2 Problem Formulation The paper formulates the problem of communication-efficient edge federated learning as follows. Consider a network of N edge devices and a central server. Each device n has a local dataset D_n of size m_n that follows an unknown distribution p_n. The goal is to train a global model w that minimizes the following empirical risk:

$$\min_{w} F(w) = \sum_{n=1}^{N} p_n F_n(w) = \frac{1}{m_n} \sum_{(x,y)\in D_n} f(w; x, y), \tag{1}$$

where $f(w; x, y)$ is the loss function for a single data point (x, y). The paper assumes that the edge devices have limited communication bandwidth and computation resources, and that the local datasets are non-i.i.d. and unbalanced.

The paper adopts the federated averaging (FedAvg) algorithm as the baseline method, which consists of the following steps:

1. The server randomly selects K devices to participate in each communication round and broadcasts the current global model w_t to them.
2. Each selected device n performs E epochs of local stochastic gradient descent (SGD) on its own dataset with a learning rate η, and obtains a local model w_{t+1}^n.
3. Each selected device n uploads its local model w_{t+1}^n to the server.
4. The server aggregates the received local models using a weighted average, and updates the global model as follows:

$$w_{t+1} = \sum_{n=1}^{K} \frac{m_n}{M} w_{t+1}^n, \tag{2}$$

where $M = \sum_{n=1}^{K} m_n$ is the total number of data points from the selected devices.

The paper identifies two main challenges for FedAvg: (1) the communication overhead caused by uploading the full-precision local models in each round; and (2) the accuracy degradation caused by the inconsistency between the global model and the local models due to the non-i.i.d. and unbalanced data distributions. The paper aims to address these challenges by proposing a novel global update guidance framework that can reduce the communication cost and improve the accuracy simultaneously.

3.3.3 Main Contribution The main contribution of this paper is the global update guidance (GUG) framework, which consists of two novel techniques: (1) global update acceleration (GUA), which estimates the global gradient using the local gradients and accelerates the global model update; and (2) global update alignment (GUA), which aligns the local model updates with the accelerated global model using a regularization term. The paper claims that GUG can achieve both communication efficiency and accuracy by balancing the trade-off between the global model quality and the local model diversity.

3.3.4 Methodology The paper introduces the GUG framework as follows. In each communication round, the server first performs GUA to estimate the global gradient g_t using the local gradients from the previous round:

$$g_t = \sum_{n=1}^{K} \frac{m_n}{M} g_{t-1}^n, \tag{3}$$

where g_{t-1}^n is the local gradient of device n at round $t-1$. Then, the server accelerates the global model update using a momentum term μ_t:

$$w_t = w_{t-1} - \eta g_t + \mu_t (w_{t-1} - w_{t-2}), \tag{4}$$

where μ_t is a hyperparameter that controls the acceleration rate. The paper shows that GUA can improve the convergence speed and stability of FedAvg.

Next, the server performs GUA to align the local model updates with the accelerated global model. The server broadcasts the accelerated global model w_t to the selected devices, and each device n updates its local model w_t^n using a regularized local SGD:

$$w_{t+1}^n = w_t^n - \eta(\nabla f(w_t^n; x, y) + \lambda(w_t^n - w_t)), \tag{5}$$

where λ is a hyperparameter that controls the alignment strength. The paper shows that GUA can reduce the inconsistency between the global model and the local models, and improve the accuracy of FedAvg.

Finally, to reduce the communication cost, the paper applies gradient sparsification and quantization to compress the local gradients before uploading them to the server. The paper also proposes a dynamic sparsification scheme that adapts the sparsity level according to the gradient norm. The paper shows that these compression techniques can achieve significant communication reduction without sacrificing accuracy.

3.3.5 Results The paper evaluates its proposed method on various datasets and models, and compares it with several baseline methods, including FedAvg, FedProx, QFedAvg, and SCAFFOLD. The paper uses two metrics to measure the performance: (1) test accuracy, which reflects how well the global model generalizes to unseen data; and (2) communication rounds, which reflects how many rounds of communication are needed to reach a target accuracy.

The paper reports that its proposed method outperforms all the baseline methods in terms of both test accuracy and communication rounds on most of the datasets and models. The paper also conducts ablation studies to analyze the effects of different components of its method, such as GUA, GUA, sparsification, quantization, and dynamic sparsity. The paper finds that each component contributes to improving either communication efficiency or accuracy or both. The paper also provides some insights into how GUG balances the trade-off between the global model quality and the local model diversity.

3.3.6 Insights Some insights that can be drawn from this paper are:

- Communication efficiency and accuracy are two important aspects of federated learning, and they are often in conflict with each other due to the heterogeneous and distributed nature of edge devices and data.
- Balancing the trade-off between communication efficiency and accuracy requires considering both the global model quality and the local model diversity, and designing techniques that can guide or align the local model updates towards a high-quality global model while preserving their local characteristics.

- Accelerating the global model update using a momentum term can improve the convergence speed and stability of federated learning, especially when there is high variance or noise in the local gradients.
- Aligning the local model updates with an accelerated global model using a regularization term can reduce the inconsistency between them and improve their generalization performance.
- Applying gradient compression techniques such as sparsification and quantization can significantly reduce the communication cost without sacrificing accuracy, especially when combined with dynamic sparsity schemes that adapt to the gradient norm.

3.3.7 Implications and Novelty The implications and novelty of this paper are:

- The paper provides a novel and effective framework for communication-efficient edge federated learning that can balance the trade-off between communication efficiency and accuracy by leveraging global update guidance techniques.
- The paper introduces two novel techniques, namely global update acceleration and global update alignment, that can improve the stability and quality of the global model update and the local model update, respectively.
- The paper demonstrates the benefits of its proposed method on various datasets and models, and shows that it can achieve significant improvements over the state-of-the-art methods in terms of both test accuracy and communication rounds.
- The paper also provides some theoretical analysis and empirical insights into how its proposed method balances the trade-off between the global model quality and the local model diversity, and how it adapts to different scenarios and settings.

3.3.8 Soundness and Clarity The soundness and clarity of this paper are:

- The paper is sound in terms of its problem formulation, methodology, results, and analysis. The paper provides sufficient details and explanations for its proposed method, and supports its claims with rigorous theoretical analysis and extensive empirical evaluation.
- The paper is clear in terms of its presentation, organization, and writing. The paper follows a logical structure that introduces the motivation, problem formulation, main contribution, methodology, results, insights, implications, novelty, soundness, clarity, significance, and limitations of its work. The paper uses clear and concise language, mathematical notations, figures, tables, and algorithms to illustrate its ideas and results.

3.3.9 Significance and Limitations The significance and limitations of this paper are:

- The paper is significant in terms of its impact and relevance to the survey topic. The paper addresses a fundamental and practical problem of communication efficiency in edge federated learning, which is one of the key challenges and research directions in this field. The paper proposes a novel and effective solution that can achieve both communication efficiency and accuracy by balancing the trade-off between the global model quality and the local model diversity. The paper also provides some valuable insights and implications for future research in this area.
- The paper has some limitations in terms of its assumptions, applicability, and scalability. The paper assumes that the edge devices have sufficient computation resources to perform local SGD with a fixed number of epochs. The paper also assumes that the edge devices have reliable network connections to communicate with the server. The paper does not consider some practical issues such as device failures, malicious attacks, or data drifts. The paper also does not evaluate its method on large-scale or real-world datasets or scenarios.

3.3.10 Summary To summarize, this paper proposes a novel global update guidance framework for communication-efficient edge federated learning that can balance the trade-off between communication efficiency and accuracy by leveraging global update acceleration and global update alignment techniques. The paper demonstrates the benefits of its proposed method on various datasets and models, and shows that it can achieve significant improvements over the state-of-the-art methods in terms of both test accuracy and communication rounds. The paper also provides some theoretical analysis and empirical insights into how its proposed method balances the trade-off between the global model quality and the local model diversity. The paper is sound, clear, significant, novel, but has some limitations in terms of its assumptions, applicability, and scalability.

3.4 Open Challenges and Future Directions

We now summarize this section and identify some open challenges and research gaps that need to be addressed in communication-efficient edge federated learning. We provide some future directions and research opportunities for communication-efficient edge federated learning. We discuss some potential applications and impacts of communication-efficient edge federated learning in various fields.

3.4.1 Summary and Conclusions We have provided a comprehensive overview of communication-efficient edge federated learning, which is a novel paradigm that integrates edge computing and federated learning to enable distributed and collaborative machine learning at the network edge. We have defined the concept and characteristics of communication bottleneck in edge federated learning, and highlighted its main challenges and impacts on the learning performance and scalability. We have categorized the existing solutions and tech-

niques for communication-efficient edge federated learning based on some criteria: architecture-based, protocol-based, algorithm-based, and hybrid-based. We have described their main idea, design, implementation, and evaluation in detail. We have analyzed their performance and trade-offs in terms of communication overhead, latency, accuracy, robustness, privacy, etc. We have identified their strengths and weaknesses and suggested some possible improvements or extensions. We have selected a classical paper on communication-efficient edge federated learning that has high impact and relevance to the survey topic. We have introduced its motivation, problem formulation, main contribution, methodology, results, insights, implications, novelty, soundness, clarity, significance, and limitations.

3.4.2 Open Challenges and Research Gaps Despite the significant progress made in communication-efficient edge federated learning, there are still some open challenges and research gaps that need to be addressed. Some of them are:

- How to design more effective and efficient compression, encoding, or quantization techniques for model parameters or gradients that can preserve the accuracy and robustness of edge federated learning?
- How to design more adaptive and dynamic local update or global update algorithms that can adjust the number of iterations or epochs, the subset of model parameters or gradients, or the subset of edge nodes according to the data distribution or network condition?
- How to design more scalable and flexible architectures or protocols that can handle large-scale or heterogeneous networks with varying number of edge nodes, model parameters, or network bandwidth?
- How to design more secure and privacy-preserving solutions or techniques that can protect the model parameters or gradients from malicious attacks or eavesdropping during the communication process?
- How to design more comprehensive and rigorous theoretical analysis or empirical evaluation that can capture the communication efficiency and performance trade-offs of different solutions or techniques under realistic settings?

3.4.3 Future Directions and Research Opportunities Based on the open challenges and research gaps identified above, we provide some future directions and research opportunities for communication-efficient edge federated learning. Some of them are:

- Exploring more advanced compression, encoding, or quantization techniques based on deep neural networks (DNNs), sparse coding (SC), low-rank approximation (LRA), etc., that can achieve higher compression ratio or lower reconstruction error for model parameters or gradients.
- Exploring more adaptive and dynamic local update or global update algorithms based on reinforcement learning (RL), meta-learning (ML), online learning (OL), etc., that can learn the optimal number of iterations or epochs,

the subset of model parameters or gradients, or the subset of edge nodes from data or feedback.
- Exploring more scalable and flexible architectures or protocols based on blockchain (BC), distributed ledger technology (DLT), software-defined networking (SDN), etc., that can enable decentralized or self-organized aggregation without relying on a central aggregator or any intermediate aggregators.
- Exploring more secure and privacy-preserving solutions or techniques based on homomorphic encryption (HE), secure multi-party computation (SMPC), differential privacy (DP), etc., that can encrypt or perturb the model parameters or gradients without compromising their utility or quality.
- Exploring more comprehensive and rigorous theoretical analysis or empirical evaluation based on optimization theory (OT), information theory (IT), game theory (GT), etc., that can provide convergence guarantees, complexity bounds, incentive mechanisms, or fairness measures for communication-efficient edge federated learning.
- Exploring more novel and practical applications and impacts of communication-efficient edge federated learning in various fields such as healthcare, smart city, Internet of Things (IoT), etc., that can benefit from distributed and collaborative machine learning at the network edge.

4 Byzantine Fault Tolerance in Federated Learning

Federated learning [55] has emerged as a promising paradigm for distributed machine learning, enabling the collaborative training of models across multiple devices without the need to share raw data. This decentralized approach not only mitigates privacy concerns but also reduces communication overhead, making it particularly suitable for edge scenarios where devices have limited computational and communication resources. However, the distributed nature of federated learning also exposes the system to a wide range of potential failures and malicious behaviors [14, 8]. In this context, incorporating Byzantine fault tolerance into federated learning becomes crucial for ensuring the resilience, security, and reliability of these systems. In this section, we provide an overview of Byzantine fault-tolerant federated learning in edge scenarios and outline the key challenges and opportunities in this emerging research area.

Edge scenarios encompass a broad range of settings, from Internet of Things (IoT) networks [71] and autonomous vehicles [86] to mobile devices [63] and edge computing platforms [3]. In these environments, devices are often heterogeneous in terms of their computational capabilities, communication resources, and data distributions. Furthermore, edge devices may be prone to faults, adversarial attacks, or even physical damage due to their distributed and exposed nature. These unique characteristics of edge scenarios pose several challenges for the design and implementation of Byzantine fault-tolerant federated learning systems.

Despite these challenges, edge-scenario Byzantine fault-tolerant federated learning presents numerous opportunities for innovation and research. By leveraging the unique characteristics of edge devices and environments, researchers

can develop novel algorithms and protocols that not only enhance the resilience, security, and privacy of federated learning systems but also enable new applications and use cases that were previously infeasible or impractical.

For instance, edge devices' distributed and localized nature can be exploited to develop decentralized and privacy-preserving solutions for real-time anomaly detection, predictive maintenance, and context-aware recommendation systems. Additionally, the integration of Byzantine fault-tolerant federated learning with other emerging technologies, such as blockchain, edge computing, and 5G networks, can open up new avenues for secure, resilient, and high-performance distributed intelligence systems.

In this section, we aim to provide a comprehensive overview of the state-of-the-art in Byzantine fault-tolerant federated learning for edge scenarios. Our objective is to offer insights into the current state of the field, identify potential avenues for future research, and foster a better understanding of the unique requirements and constraints of edge-scenario Byzantine fault-tolerant federated learning.

4.1 Foundations of Byzantine Fault Tolerance

4.1.1 The Byzantine Generals Problem The Byzantine Generals Problem [29] is a thought experiment that illustrates the challenges of achieving consensus and coordination in a distributed system with potentially malicious nodes. The problem is framed in the context of a group of Byzantine generals who must decide whether to attack or retreat from a besieged city. The generals are situated at different locations and can only communicate through messengers. Some generals may be traitors, who may send incorrect or conflicting messages to disrupt the decision-making process.

The objective of the Byzantine Generals Problem is to design a communication protocol that allows the loyal generals to reach a consensus on their strategy (attack or retreat), despite the presence of traitorous generals.

The Byzantine Generals Problem exemplifies the challenges of achieving consensus in distributed systems with potentially malicious nodes, where communication channels may be unreliable and nodes may exhibit unpredictable behavior.

4.1.2 Byzantine Fault Tolerance Byzantine fault tolerance is the ability of a distributed system to function correctly and reach consensus, even when some of its nodes are compromised or fail in arbitrary ways. These arbitrary failures, known as Byzantine faults, may manifest as nodes sending incorrect or conflicting information, refusing to participate in the consensus process, or otherwise behaving maliciously.

The principles of Byzantine fault tolerance revolve around the development of algorithms and protocols that ensure the correct operation of a distributed system in the presence of Byzantine faults. These techniques typically involve the use of redundancy, consensus algorithms, and cryptographic primitives. Redundancy provides fault tolerance by replicating critical components, while consensus algorithms enable nodes to agree on a single, consistent view of the system

state. Cryptographic primitives, such as digital signatures and hash functions, help ensure the authenticity and integrity of messages exchanged between nodes.

Achieving Byzantine fault tolerance in a distributed system often requires satisfying specific conditions. According to the well-known result [29] by Leslie Lamport, Robert Shostak, and Marshall Pease, a system with n nodes can tolerate up to $(n-1)/3$ Byzantine faults, provided that nodes use a suitable consensus algorithm and communicate through authenticated channels.

4.2 Overview of Byzantine Fault-Tolerant Federated Learning Algorithms

Byzantine fault-tolerant federated learning (BFTFL) seeks to develop robust algorithms and protocols that ensure the resilience, security, and reliability of federated learning systems in the presence of arbitrary failures or malicious behavior. This section provides an in-depth exploration of various Byzantine fault-tolerant federated learning algorithms, their underlying principles, and their advantages and limitations.

4.2.1 Robust Gradient Descent Robust gradient descent is a fundamental approach to mitigating the impact of Byzantine failures in federated learning. The idea is to use robust aggregation techniques that reduce the influence of malicious or faulty updates during the model aggregation process.

We consider a general federated learning system, consisting of a server and n nodes, where f nodes may be Byzantine. We represent the set of all nodes as $N = \{1, \cdots, n\}$, and the set of Byzantine nodes as $B \subset N$, where $|B| = f$. In round t, the server broadcasts its parameter $x^t \in \mathbb{R}^d$ to all n nodes. Then each correct node $p \in N \backslash B$ will compute an estimate $V_p^t = G_p(x_t, \xi_p^t)$ of the gradient $\nabla F_p(x_t)$ of the loss function F_p based on x^t and local data, while each Byzantine node $b \in B$ will generate an arbitrary form vector V_b^t. All nodes will send V_i^t, $i \in N$ to the server (If the Byzantine node b does not actually send any information to the server, it is considered to have sent $V_b^t = 0$ in theory), and then the server will update the model according to some aggregation rule $Agg(V_1^t, \cdots, V_n^t)$:

$$x^{t+1} = x^t - \eta Agg(V_1^t, \cdots, V_n^t).$$

In round t, each V_p^t sent by the correct node is assumed to be an unbiased estimate of the gradient $\nabla F_p(x_t)$, i.e., $\mathbb{E}G_p(x_t, \xi_p^t) = \nabla F_p(x_t)$. The Byzantine nodes have full knowledge of the entire federated system, including the aggregation rule $Agg(\cdot)$ and the vectors sent to the server [7]. Furthermore, they can also collaborate with each other.

4.2.1.1 Krum Krum [7] is a robust aggregation algorithm that identifies and discards Byzantine updates by computing a score for each local model update based on its pairwise distances to other updates. The algorithm selects the update with the lowest score, which corresponds to the minimum sum of distances to other updates.

Definition 1 (Krum). *The aggregation rule of Krum can be represented as* $KR(V_1^t, \cdots, V_n^t) = V_*^t$. *For any* $i, j \in N (i \neq j)$, $V_*^t = \arg\min_{i \in N} \sum_j \parallel V_i^t - V_j^t \parallel^2$, *where* V_j^t *is the* $n - f - 2$ *closest ones to* V_i^t.

Krum has been shown to provide strong resilience against Byzantine attacks, but its computational complexity may limit its scalability.

4.2.1.2 Coordinate-wise Median Coordinate-wise Median aggregation [80] is a straightforward technique that replaces the mean aggregation [55] with the coordinate-wise median of local model updates.

Definition 2 (Coordinate-wise Median). *The aggregation rule of Coordinate-wise Median can be denoted as* $MED(V_1^t, \cdots, V_n^t) = V_*^t$, *where* V_*^t *is a vector with its k-th coordinate* V_*^t *being* $med\{V_i^t[k] : i \in N\}$.

By selecting the median, this method effectively reduces the influence of extreme values introduced by Byzantine nodes. However, the median-based approach may not be suitable for high-dimensional data, as it can lead to suboptimal model convergence.

4.2.1.3 Coordinate-wise Trimmed Mean Coordinate-wise trimmed mean [80] is another robust aggregation technique that computes the average of local model updates after discarding the highest and lowest values for each coordinate.

Definition 3 (Coordinate-wise Trimmed Mean). *The aggregation rule of Coordinate-wise* β*-Trimmed Mean can be written as* $TRMean_\beta(V_1^t, \cdots, V_n^t) = V_*^t$, *for* $\beta \in [0, \frac{1}{2})$. V_*^t *is a vector with its k-th coordinate being* $\frac{1}{(1-2\beta)f} \sum_{u \in U^t[k]} u$. *Here* $U^t[k]$ *is a subset of* $\{V_1^t[k], \cdots, V_n^t[k]\}$ *obtained by removing the largest and smallest* β *fraction of its elements, where* $V_i^t[k]$ *is the k-th coordinate of* V_i^t.

This method is more resilient to Byzantine attacks than median-based aggregation and has lower computational complexity than Krum. However, it may still be vulnerable to more sophisticated attacks that introduce carefully crafted adversarial updates.

4.2.2 Secure Aggregation and Communication Secure communication channels are essential for protecting federated learning systems against Byzantine failures resulting from message tampering or eavesdropping. Cryptographic techniques can be employed to ensure the confidentiality, integrity, and authenticity of exchanged messages.

4.2.2.1 Secure Multi-Party Computation (SMPC) SMPC is a cryptographic technique that allows multiple parties to jointly compute a function over their inputs while keeping the inputs private. In the context of BFTFL, SMPC can be used to securely aggregate local model updates without revealing the individual updates to other nodes. Fitzi et al. [15] have shown that n nodes can achieve

unconditional broadcast if the number of Byzantine nodes $f < n/2$ with the help of secure multi-party computation. This approach enhances both privacy and resilience against Byzantine attacks but may introduce additional computational overhead.

4.2.2.2 Homomorphic Encryption Homomorphic encryption is a form of encryption that enables computations to be performed directly on encrypted data without decrypting it first. This technique can be used to aggregate encrypted local model updates in federated learning, ensuring privacy and security while mitigating the impact of Byzantine failures. Tian et al. [69] propose a lattice based distributed threshold addition homomorphic encryption scheme to cope with the situation where the nodes quit midway or the adversary interrupts the nodes or the parameter server in the federated learning. However, the current state-of-the-art homomorphic encryption schemes may not be efficient enough for large-scale federated learning scenarios.

4.2.3 Trust Management Establishing trust among nodes in a federated learning system is crucial for mitigating the impact of Byzantine failures. Reputation-based or trust-based mechanisms can be employed to evaluate the trustworthiness of participating nodes, enabling the system to identify and isolate potentially malicious or faulty nodes.

4.2.3.1 Reputation-based Systems Reputation-based systems assign a trust score to each node based on its past behavior, such as the quality of its local model updates or its adherence to the communication protocol. Nodes with higher trust scores have a greater influence on the model aggregation process, reducing the impact of Byzantine nodes. Wei et al. [75] present a general framework for detecting malicious peers in Reputation-based P2P systems. Kang et al. [25] set a reputation value according to the performance of each participant in the federated learning, and the server preferentially selects the participants with good reputation for aggregation. However, maintaining and updating trust scores can introduce additional overhead and may be susceptible to Sybil attacks [64, 67], where a malicious node creates multiple fake identities to manipulate trust scores. Bankovic et al. [4] propose to couple reputation systems with agents based on self-organizing map algorithm to detect and confine Sybil attack.

4.2.3.2 Blockchain-based Trust Management Blockchain technology can be employed to establish a decentralized trust management system for federated learning. By maintaining an immutable record of nodes' contributions and behaviors, blockchain-based systems can enhance transparency and trust among participating nodes. Bao et al. [5] propose FLChain to choose the most reliable participant as the leader, and use its aggregated model as a new global model to reach a consensus. Li et al. [38] propose BFLC, which utilizes two kinds of blocks to hold the global model and local model updates respectively. Liu et al. [43] design a secure federated learning framework for 5G networks, combining federated learning

with Ethereum. Although blockchain-based trust management can improve the security and resilience of BFTFL systems, it may introduce additional overhead due to the complexity of consensus mechanisms and the need for maintaining the blockchain.

4.2.4 Adaptive Consensus Algorithms Adaptive consensus algorithms play a critical role in determining the global model's state based on local model updates from participating nodes. Implementing adaptive consensus algorithms that can tolerate Byzantine failures can help ensure the reliability and accuracy of the federated learning process.

4.2.4.1 Practical Byzantine Fault Tolerance (PBFT) PBFT [9] is a Byzantine fault-tolerant consensus algorithm designed for distributed systems. In the context of federated learning, PBFT can be employed to achieve consensus on the aggregated model update, ensuring that the learning process continues even in the presence of Byzantine nodes. PBFT provides strong consistency guarantees and has been shown to be effective in small to moderately-sized networks. However, its performance may degrade in large-scale settings due to its communication complexity.

4.2.4.2 Federated Byzantine Agreement System (FBAS) FBAS [53] is another consensus algorithm designed to tolerate Byzantine failures in distributed systems. In contrast to PBFT, FBAS allows for more flexible trust configurations, enabling nodes to form quorums based on their trust relationships. This approach can improve the scalability and adaptability of the consensus process in federated learning systems, ensuring resilience against Byzantine failures. However, the performance of FBAS may be sensitive to the choice of quorum configurations, and achieving optimal configurations may be challenging.

In summary, Byzantine fault-tolerant federated learning algorithms seek to enhance the resilience, security, and reliability of federated learning systems by addressing the challenges posed by arbitrary failures and malicious behavior. These algorithms encompass robust gradient descent techniques, secure aggregation and communication methods, trust management mechanisms, adaptive consensus algorithms, and privacy-preserving approaches. Each of these techniques offers unique advantages and limitations, and selecting the appropriate combination of algorithms for a given federated learning scenario may require careful consideration of the trade-offs between performance, fault tolerance, privacy, and scalability. As the fields of machine learning and artificial intelligence continue to evolve, the development and refinement of Byzantine fault-tolerant federated learning algorithms will play a crucial role in advancing collaborative, decentralized, and privacy-preserving intelligence systems.

4.3 An interesting Paper on Byzantine Fault-Tolerant Edge Federated Learning

Here we introduce an interesting paper [68] on Byzantine-robust edge federated learning, with the title of "Byzantine-Resilient Federated Learning At Edge". We will summarize its motivation, contributions, problem setup, and main results.

4.3.1 Motivation This paper points out that there are some practical issues such as unreliability of edge devices and communication overhead that hinder the successful implementation of edge FL. In addition, heavy-tailed data widespread at edge devices may further degrade the performance of learning algorithms. However, existing works on Byzantine resilience in FL all make strong assumptions on the distribution of loss gradients. The authors design an edge FL framework that is robust to heavy-tailed data as well as satisfies the requirement of Byzantine resilience and communication efficiency under the standard assumptions.

4.3.2 Problem Setup This paper considers the stochastic convex and non-convex optimization problem. Let $\mathcal{W} \subseteq \mathbb{R}^d$ be the parameter space containing all the possible model parameters and $\mathcal{D}$ be an unknown distribution over the data universe $\mathcal{Z}$. Given a loss function $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}$, where $\ell(w, z)$ measures the risk induced by data z under the model parameter choice w, and a dataset $D = z_1, z_2, \cdots, z_N$, where z_i's are i.i.d. samples from the distribution $\mathcal{D}$ over $\mathcal{Z}$, the goal is to learn an optimal parameter choice $w^* \in \mathcal{W}$ that minimizes the population risk $R_{\mathcal{D}}(w)$, i.e.,

$$w^* \in \arg\min_{w \in \mathcal{W}} R_{\mathcal{D}}(w) \triangleq \mathbb{E}[\ell(w, z)] \tag{6}$$

This paper assumes that the total N training data are evenly distributed across the m devices such that each worker machine holds $n = \frac{N}{m}$ data. And the details of Byzantine devices are the same as described in Section 3.2.1. The only limit on Byzantine devices is that these devices cannot contaminate the local dataset.

4.3.3 Main Results

4.3.3.1 Byzantine-Resilient Heavy-tailed Gradient Descent (BHGD)

Definition 4 (Robust estimator). *Considering a one-dimensional random variable $x \sim \mathcal{X}$ and assuming that $x_1, x_2, \cdots, x_n$ are i.i.d. samples of x. The robust estimator consists of three steps:*

*1) **Scaling and Truncation** For each sample x_i, the authors re-scale it by dividing s and apply a soft truncation function ϕ on the re-scaled one. Then they calculate the empirical mean of the altered samples and put the mean back to the original scale.*

$$\frac{s}{n} \sum_{i=1}^{n} \phi(\frac{x_i}{s}) \approx \mathbb{E}[x].$$

2) ***Noise Multiplication*** *Let $\epsilon_1, \cdots, \epsilon_n$ be independent random noise generated from a common distribution ν with $\mathbb{E}[\epsilon_i] = 0$ for each. The authors multiply each sample x_i by $(1 + \epsilon_i)$, and then perform the scaling and truncation step on $x_i \cdot (1 + \epsilon_i)$.*

$$\tilde{x}(\epsilon) = \frac{s}{n} \sum_{i=1}^{n} \phi(\frac{x_i + \epsilon_i x_i}{s}).$$

3) ***Noise Smoothing*** *The authors smooth the multiplicative noise via taking the expectation with respect to the noise distribution ν.*

$$\hat{x} = \mathbb{E}[\tilde{x}(\epsilon)] = \frac{s}{n} \sum_{i=1}^{n} \int \phi(\frac{x_i + \epsilon_i x_i}{s}) d\nu(\epsilon_i)$$

The main idea of BHGD is that, instead of using empirical mean as the local estimator which may be subject to the heavy-tailed outliers, the authors let each device apply the one-dimensional robust mean estimator described above to each coordinate of its local loss gradients so that a more accurate local estimator $g_i(\cdot)$ for $\nabla R_{\mathcal{D}}(\cdot)$ can be obtained. The server then uses the coordinate-wise trimmed mean to aggregate these local estimators and obtain a global estimator $g(\cdot)$ for $\nabla R_{\mathcal{D}}(\cdot)$.

4.3.3.2 Byzantine-Resilient Heavy-tailed Gradient Descent with Compression (BHGD-C) In order to reduce the communication cost, the authors adopt the gradient compression technique to further improve BHGD. Specifically, each device sends a compressed version $\mathcal{Q}(g_i(\cdot))$ of its local gradient estimator $g_i(\cdot)$ to the server, where $\mathcal{Q}(\cdot)$ is an δ-approximate compressor defined as follows.

Definition 5 (δ-Approximate Compressor). *An operator $\mathcal{Q}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is said to be an δ-approximate compressor on a set $\mathcal{S} \subseteq \mathbb{R}^d$ if $\forall x \in \mathcal{S}$,*

$$\| \mathcal{Q}(x) - x \|_2^2 \leq (1 - \delta) \| x \|_2^2,$$

where $\delta \in (0, 1]$ is the compression factor.

4.3.4 Contributions The contributions of this paper can be summarized as follows.

- This paper presents a comprehensive study of Byzantine-tolerant distributed gradient descent with heavy-tailed data under standard assumptions. In particular, with the assumption that the distribution of loss gradients has only coordinate-wise bounded second-order raw moment for heavy-tailed data, the authors establish the high-probability guarantees of statistical error rate for strongly convex, general convex and non-convex population risk functions respectively. Specifically, for all the cases, their algorithm achieves the following statistical error rate:

$$\tilde{\mathcal{O}}\big(d^2[\frac{\alpha^2}{n} + \frac{1}{mn}]\big),$$

where $\alpha \in (0, \frac{1}{2})$ is the fraction of Byzantine devices, n is the size of local dataset on each edge device and m is the number of edge devices. The error rate above matches the error rate given in [79], which implies that the algorithm still achieve order-wise optimality in terms of (α, n, m), even in the presence of heavy-tailed data.

– To achieve the communication efficiency, the authors adopt the technique of gradient compression and propose a communication-efficient and Byzantine resilient distributed gradient descent algorithm with heavy-tailed data. In this case, the statistical error rates becomes:

$$\tilde{\mathcal{O}}\big(d^2[\frac{\alpha^2}{n} + \frac{1-\delta}{n} + \frac{1}{mn}]\big),$$

where δ is the compression factor, and when $\delta = 1$, the error becomes $\tilde{\mathcal{O}}\big(d^2[\frac{\alpha^2}{n} + \frac{1}{mn}]\big)$, which means that the compression term has no order-wise contribution to the error rate.

4.4 Applications of Byzantine Fault-Tolerant Federated Learning in Edge Scenarios

Byzantine fault-tolerant federated learning (BFTFL) has the potential to revolutionize a variety of applications in edge scenarios by enabling secure, resilient, and privacy-preserving collaboration among distributed devices. In this part, we briefly discuss several promising applications of BFTFL in edge environments, illustrating its potential impact on various domains.

– **Smart Cities and Transportation** BFTFL can be employed in smart cities and transportation systems to develop joint machine learning models on distributed data from traffic sensors, connected vehicles, and public transit systems. These models can help optimize traffic management, enhance public transportation efficiency, and improve urban planning. BFTFL ensures that the system remains robust against potential attacks or failures while preserving the privacy of users' location data and travel patterns.
– **Healthcare and Wearable Devices** In the healthcare domain, BFTFL can be applied to create collaborative models for medical diagnosis, treatment planning, and drug discovery using data from distributed wearable devices and medical equipment. This approach enables the development of personalized and data-driven healthcare solutions while maintaining patient privacy and ensuring system resilience against malicious attacks or faulty devices.
– **Industrial IoT and Predictive Maintenance** BFTFL can be employed in Industrial IoT settings to enable collaborative learning among distributed sensors, machines, and control systems for predictive maintenance and process optimization. By jointly training machine learning models on distributed data sources, BFTFL can help identify potential failures or inefficiencies in real-time, ensuring the reliability and security of industrial systems while preserving the privacy of proprietary data.

– **Environmental Monitoring and Agriculture** In environmental monitoring and agriculture, BFTFL can be used to develop joint machine learning models on distributed data from sensor networks, remote sensing systems, and research institutions. These models can help predict crop yields, optimize irrigation strategies, or monitor environmental changes. BFTFL guarantees the resilience and security of the system, even in the presence of faulty or malicious nodes, while preserving the privacy of individual data sources.

These examples showcase the potential of Byzantine fault-tolerant federated learning in edge scenarios, enabling secure, resilient, and privacy-preserving collaborative learning across a wide range of applications and domains. As research in this area continues to advance, we can expect to see even more innovative applications of BFTFL that capitalize on the unique characteristics of edge environments.

4.5 Future Directions

Byzantine fault-tolerant federated learning has many advantages, prompting researchers to focus on exploration in this field, but there are still some challenges:

– **Scalability**: BFTFL algorithms require a large number of participants to function correctly, making it difficult to deploy them in real-world scenarios.
– **Adversarial attacks**: Attackers may use sophisticated strategies to evade detection and disrupt the learning process, leading to poor model performance and accuracy.
– **Incentive misalignment**: Participants may not have the same goals or incentives, leading to strategic behavior that can undermine the performance of the model and BFT algorithm.
– **Real-world deployment**: Deploying BFTFL algorithms in real-world applications requires overcoming technical, regulatory, and ethical challenges.

To address these challenges, researchers can explore the following solutions and research directions:

– **Multi-layered security**: Developing algorithms that can detect and prevent adversarial attacks at multiple levels, such as data poisoning attacks, model inversion attacks, and backdoor attacks.
– **Incentive mechanisms**: Designing incentive mechanisms that can align the interests of participants and encourage them to contribute truthful updates, even in the presence of Byzantine faults.
– **Decentralized architecture**: Exploring decentralized architectures that can enable BFTFL without relying on a central server, reducing the risk of single-point failures and improving scalability.
– **Real-world deployment**: Conducting more case studies and pilot tests to evaluate the feasibility, effectiveness, and ethical implications of BFT Federated Learning in real-world scenarios.

Byzantine fault-tolerant federated learning is a promising research area with great application potential in edge scenarios. However, it also faces several challenges that require further research and development. Addressing these challenges requires collaboration between researchers, practitioners, and policymakers to ensure that BFTFL can deliver its full potential while preserving the privacy, security, and ethics of participants and users.

5 Privacy Issues in Edge Federated Learning

Federated learning at edge is a distributed machine learning paradigm that enables multiple clients to collaboratively train a machine learning model without sharing their data with each other or with a central server. This approach has gained significant attention in recent years due to its potential to address privacy concerns and data ownership issues in machine learning [78, 41].

In traditional machine learning approaches, data is collected from multiple sources and centralized in a single location for training. This approach raises several privacy concerns as the data owners lose control over their data once it is shared with the central server. Moreover, centralized machine learning models are vulnerable to adversarial attacks and can compromise the privacy of the training data and the clients' private information [77].

Federated learning at edge addresses these issues by enabling clients to train a machine learning model collaboratively without sharing their data with each other or with a central server. In this approach, the clients' data remains on their devices, and only the model updates are shared with other clients or with a central aggregator. This approach preserves the privacy of the training data and the clients' private information while enabling collaborative machine learning tasks.

Privacy-preserving federated learning at edge [44, 94, 46] is an emerging research area that focuses on developing techniques and algorithms that can enable collaborative machine learning while preserving privacy. It ensures that sensitive data remains private and secure while still allowing for collaboration and model training. This research area has gained significant attention in recent years due to its potential to enable distributed machine learning tasks in various domains such as healthcare, finance, and smart cities.

While Federated Learning at Edge offers many benefits, it also introduces new challenges related to data privacy [65, 90, 66]. In a typical Federated Learning scenario, each device trains a local model on its own data and shares updates with a central server. However, even if the raw data is not shared, the model updates may still reveal sensitive information about the data used for training [57].

Addressing privacy concerns in Edge Federated Learning is crucial for ensuring the trust and adoption of this technology [91]. If users do not trust that their data will be kept private, they may be hesitant to participate in Federated Learning at Edge. Furthermore, failure to adequately address privacy concerns may result in legal and regulatory issues.

5.1 Privacy Threats in Edge Federated Learning

Federated Learning at Edge offers many benefits, it also introduces new challenges related to data privacy. In particular, there are several types of attacks that can threaten the privacy of data used in Edge Federated Learning, including data poisoning attacks, model inversion attacks, and membership inference attacks. These attacks can compromise the privacy of users' data and undermine the trust in Federated Learning at Edge. Therefore, it is important to develop techniques for mitigating these threats and ensuring the privacy of data used in Edge Federated Learning.

5.1.1 Data Poisoning Attacks Data poisoning attacks are a type of adversarial attack where an attacker injects malicious data into the training dataset to manipulate the learning process and compromise the model's performance. In privacy-preserving edge federated learning, data poisoning attacks can be more challenging to detect and mitigate due to the distributed nature of the learning process and the lack of access to the clients' data. Several techniques have been proposed to detect and mitigate data poisoning attacks in federated learning [48, 76, 88], such as robust aggregation methods and outlier detection techniques.

5.1.2 Model Inversion Attacks Model inversion attacks are a type of adversarial attack where an attacker tries to infer sensitive information about the training data or the model's parameters by exploiting the model's output. In privacy-preserving edge federated learning, model inversion attacks [47, 93, 30] can be more challenging to detect and mitigate due to the distributed nature of the learning process and the lack of access to the clients' data. Several techniques have been proposed to detect and mitigate model inversion attacks in federated learning, such as differential privacy and adversarial training.

5.1.3 Membership Inference Attacks Membership inference attacks are a type of privacy attack where an attacker tries to infer whether a particular client has contributed to the training dataset or not by analyzing the model's output. In privacy-preserving edge federated learning, membership inference attacks [26, 45] can be more challenging to prevent due to the distributed nature of the learning process and the lack of access to the clients' data. Several techniques have been proposed to prevent membership inference attacks in federated learning, such as differential privacy and secure aggregation methods.

5.2 Techniques in Privacy-preserving Edge Federated Learning

Federated Learning at Edge introduces new challenges related to data privacy. In particular, there are several techniques that can be used to preserve the privacy of data used in Edge Federated Learning, including differential privacy, homomorphic encryption, and secure multi-party computation. These techniques can help ensure that users' data remains private while still enabling collaborative

training of machine learning models. Therefore, it is important to develop and implement effective privacy-preserving techniques in Edge Federated Learning. Therefore, it is important to develop techniques for preserving privacy in Edge Federated Learning.

5.2.1 Differential Privacy Differential privacy [12] is a well-known privacy-preserving technique that adds noise to the training data to mask sensitive information while preserving the overall statistical properties of the dataset. Differential Privacy is a mathematical framework for quantifying the privacy of data used in statistical analysis. It provides a formal definition of privacy and a set of techniques for ensuring that the output of analysis does not reveal too much information about any individual data point. In privacy-preserving edge federated learning, differential privacy can be used to add noise to the local model updates before sharing them with other clients or with a central aggregator. This approach can help preserve the privacy of the training data and the clients' private information while enabling collaborative machine-learning tasks.

There are several techniques for implementing Differential Privacy in Edge Federated Learning [19, 81, 2]. One approach is to add noise to the model updates before sharing them with other devices. This can be done using techniques such as the Laplace mechanism or the Exponential mechanism. Another approach is to use local differential privacy, where each device adds noise to its own data before training a local model. This can help ensure that the model updates do not reveal too much information about any individual data point.

[19] discusses the framework of differentially private FL in edge networks from the perspective of noise reduction. It summarizes three noise reduction methods based on the intrinsic factors influencing the added noise scale, including privacy amplification, model sparsification, and sensitivity reduction. [81] proposes the Dynamic Local Differential Privacy Federated Learning (DLDP-FL) framework suitable for edge computing based on the characteristics of mesh network structure. [2] discusses how federated learning has been shown as a promising approach in paving the last mile of artificial intelligence due to its great potential of solving the data isolation problem in large-scale machine learning.

5.2.2 Homomorphic Encryption Homomorphic encryption [16] is another privacy-preserving technique that enables computations on encrypted data without decrypting it first. This approach can be used to perform machine learning tasks on encrypted data while preserving privacy. This means that it is possible to perform operations on encrypted data and obtain an encrypted result, which can then be decrypted to reveal the true result of the computation. In privacy-preserving edge federated learning, homomorphic encryption can be used to encrypt the local model updates before sharing them with other clients or with a central aggregator. This approach can help preserve the privacy of the training data and the clients' private information while enabling collaborative machine-learning tasks.

There are several techniques for implementing homomorphic encryption in Edge Federated Learning [51, 24, 61]. One approach is to use a partially homomorphic encryption scheme, such as the Paillier cryptosystem, to encrypt the data used for training. This allows devices to perform certain operations on the encrypted data, such as addition or multiplication, without the need to decrypt it first. Another approach is to use a fully homomorphic encryption scheme, which allows arbitrary computations to be performed on encrypted data. However, fully homomorphic encryption is currently computationally expensive and may not be practical for use in Edge Federated Learning.

[51] proposes xMK-CKKS, a multi-key homomorphic encryption protocol to design a novel privacy-preserving federated learning scheme. In this scheme, model updates are encrypted via an aggregated public key before sharing with a server for aggregation. [24] presents FedML-HE, the first practical system for efficient HE-based secure federated aggregation that provides a user/device-friendly deployment platform. FL-HE utilizes a novel universal overhead optimization scheme, significantly reducing both computation and communication overheads during deployment while providing customizable privacy guarantees. [61] evaluates the performance of federated learning from two perspectives which are computational costs of cryptosystem and performance of VPFL on MNIST dataset.

5.2.3 Secure Multi-Party Computation Secure Multi-Party Computation (SMPC) is a cryptographic technique [27, 92] that allows multiple parties to jointly compute a function on their private inputs without revealing their inputs to each other. In privacy-preserving edge federated learning, secure multi-party computation can be used to enable clients to collaboratively train a machine learning model without sharing their data with each other or with a central aggregator. This approach can help preserve the privacy of the training data and the clients' private information while enabling collaborative machine-learning tasks.

There are several techniques for implementing SMPC in Edge Federated Learning [17, 18, 20]. One approach is to use secret sharing, where each device splits its data into multiple shares and distributes them among the other devices. The devices can then perform computations on the shares without revealing the original data. Another approach is to use garbled circuits, where each device constructs a circuit that represents the computation to be performed and garbles it to hide the inputs and outputs. The garbled circuit can then be evaluated by the other devices without revealing the original data.

[17] proposes a privacy-preserving federated learning framework that adopts secure multiparty computation (SMC). In this method, the hospitals are divided into clusters. After local training, each hospital splits its model weights among other hospitals in the same cluster such that no single hospital can retrieve other hospitals' weights on its own. [18] proposes a novel blockchain-empowered decentralized secure multiparty learning system with heterogeneous local models called BEMA. Particularly, it considers two types of Byzantine attacks and carefully

designs "off-chain sample mining" and "on-chain mining" schemes to protect the security of the proposed system. [20] discusses how multi-party computation (MPC) allows distributed machine learning to be performed in a privacy-preserving manner so that end hosts are unaware of the true model.

5.3 Representative Algorithms in Privacy-Preserving Edge Federated Learning

5.3.1 Federated Averaging FedAvg [56] is a representative algorithm in federated learning that enables clients to collaboratively train a machine learning model without sharing their data with each other or with a central aggregator. In this approach, the clients' data remains on their devices, and only the model updates are shared with other clients or with a central aggregator.

It consists of alternating between a few local stochastic gradient updates at client nodes, followed by a model averaging update at the server. The amount of computation performed at each client node can be controlled by a hyperparameter that determines the number of local stochastic gradient updates performed at each client node.

The model averaging update at the server can be computed using the following formula:

$$w_{t+1} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} w_{t+1}^i$$

where w_{t+1} is the updated model parameter vector at time $t+1$, w_{t+1}^i is the updated model parameter vector at time $t+1$ for client i, $\mathcal{S}$ is the set of clients selected to participate in the current round of federated learning, and $|\mathcal{S}|$ is the size of $\mathcal{S}$.

The local stochastic gradient update performed at each client node can be computed using the following formula:

$$w_{t+1}^i = w_t - \eta \nabla f_i(w_t)$$

where w_t is the current model parameter vector at time t, w_{t+1}^i is the updated model parameter vector at time $t+1$ for client i, η is the learning rate hyperparameter, $\nabla f_i(w_t)$ is the stochastic gradient of the loss function with respect to the model parameters for client i, and $f_i(w_t)$ is the loss function for client i.

The federated averaging algorithm works by aggregating the local model updates from multiple clients and computing the average of these updates to obtain a global model update. This approach can help preserve the privacy of the training data and the clients' private information while enabling collaborative machine learning tasks.

5.3.2 Differential Privacy [83] addresses the concern of preserving data privacy of users during the learning process in vehicular networks. With the prosperity of vehicular networks and intelligent transport systems, a vast amount

of data can be easily collected by vehicular devices from their users and widely spread in vehicular networks for the purpose of solving large-scale machine learning problems.

Definition 6. *Given a dataset with domain $\mathcal{D}$ and range $\mathcal{R}$, a randomized mechanism $\mathcal{M}$ preserves (ϵ, δ)-DP if for any two adjacent datasets $d, d' \in \mathcal{D}$ and any subset of outputs $\mathcal{S} \subset \mathcal{R}$ it holds that*

$$Pr(\mathcal{M}(d) \in \mathcal{S}) \leq e^{\epsilon} Pr(\mathcal{M}(d') \in \mathcal{S}) + \delta,$$

where $\epsilon \geq 0$ is a constant and δ is the probability of breaking this lower bound.

The sensitivity of the query function is widely used to analyze DP, which is formally defined as follows.

Definition 7. *For vector-valued function $f : \mathcal{D} \to \mathbb{R}^N$, the L2-sensitivity of f is*

$$\Delta_2 f = \max_{d_1, d_2 \in \mathcal{D}} \|f(d_1) - f(d_2)\|,$$

where d_1 and d_2 differ in at most one element.

Lemma 1. *Let $\delta \in (0, 1)$ be arbitrary. For $c_2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2 f/\epsilon$ is (ϵ, δ)-differentially private.*

To address this concern, under the celebrated framework of DP, the authors present a decentralized parallel stochastic gradient descent (D-PSGD) algorithm, called DP^2-SGD, which can offer protection for privacy of users in vehicular networks. The privacy preservation process in this algorithm is achieved through the use of differential privacy. Specifically, each vehicle adds random noise to its local model parameters before communicating them to other vehicles. This random noise ensures that individual vehicles cannot be identified from their local model parameters. The amount of noise added to each local model parameter can be computed by the Gaussian Mechanism in Lemma 1.

With thorough analysis, they show that DP^2-SGD satisfies (ϵ, δ)-DP while the learning efficiency is the same as D-PSGD without privacy preservation. EC-SGD is a refined version of DP^2-SGD that introduces an error-compensate strategy. Extensive experiments show that EC-SGD can further improve the convergence efficiency over DP^2-SGD in reality.

5.3.3 Homomorphic encryption Homomorphic encryption can be used to protect the privacy of data and model updates in federated learning (FL), where multiple edge devices collaboratively train a machine learning model without sharing their local data. However, HE introduces significant computation and communication overheads, which limit its scalability and efficiency.

To address this challenge, [24] propose a novel algorithm based on multi-key homomorphic encryption (MK-HE), which allows multiple parties to encrypt their data with different keys and perform homomorphic operations on

the ciphertexts. The algorithm, called FLASHE, leverages the additive symmetric property of MK-HE to design a secure and efficient federated aggregation scheme. FLASHE reduces the computation and communication costs compared to existing HE-based FL methods, while preserving the accuracy and privacy of the model.

FLASHE consists of four main steps: encryption, aggregation, partial decryption, and final decryption.

FLASHE achieves privacy-preserving FL by ensuring that no party can learn any information about other parties' data or model updates from the ciphertexts or the aggregated model. FLASHE also reduces the computation and communication overheads by using xMK-CKKS, which has lower complexity and smaller ciphertext size than other HE schemes. FLASHE can handle large-scale models such as ResNet-50 and BERT with reasonable overheads, as shown by the experimental evaluation.

5.3.4 MPC A malicious party may infer information about the local data from the model updates. To address this challenge, [58] introduce SMPAI, a secure multi-party computation (SMPC) framework for federated learning. SMPAI leverages additive secret sharing, homomorphic encryption, and differential privacy to protect the data privacy and model accuracy in federated learning.

SMPAI consists of four main components: (1) a central server that coordinates the federated learning process and aggregates the model updates from the edge devices; (2) a set of edge devices that participate in the federated learning and hold their own local data; (3) a set of trusted third parties (TTPs) that help with the secret sharing and reconstruction; and (4) a set of cryptographic primitives that enable secure computation and communication among the parties.

The SMPAI algorithm has several advantages over existing SMPC algorithms for federated learning. First, it reduces the communication overhead by only sending shares of the model updates instead of the whole models. Second, it improves the computation efficiency by using homomorphic encryption instead of more expensive cryptographic protocols such as garbled circuits or oblivious transfer. Third, it preserves the data privacy by keeping the local data on the edge devices and only sharing random shares with other parties.

5.4 Applications of Privacy-Preserving Edge Federated Learning

- **Healthcare** Healthcare is one of the most promising applications of privacy-preserving edge federated learning. In this application, multiple hospitals or clinics can collaborate to train a machine learning model on their patients' data without sharing the data with each other or with a central aggregator. This approach can help preserve the privacy of the patients' data and the healthcare providers' private information while enabling collaborative machine learning tasks such as disease diagnosis and drug discovery.
- **Smart Grids** Smart grids are another promising application of privacy-preserving edge federated learning. In this application, multiple energy providers

can collaborate to train a machine learning model on their energy consumption data without sharing the data with each other or with a central aggregator. This approach can help preserve the privacy of the energy consumption data and the energy providers' private information while enabling collaborative machine learning tasks such as energy forecasting and demand response.
– **Autonomous Driving** Autonomous driving is another promising application of privacy-preserving edge federated learning. In this application, multiple car manufacturers or suppliers can collaborate to train a machine learning model on their driving data without sharing the data with each other or with a central aggregator. This approach can help preserve the privacy of the driving data and the car manufacturers' private information while enabling collaborative machine learning tasks such as object detection and path planning.

6 Heterogeneity Issues in Federated Learning

Federated learning is a machine learning approach where multiple clients collaborate to train a model without sharing their data directly. This technique is gaining popularity because it can overcome data silos and privacy concerns, among other benefits. However, Federated learning in edge environments has the challenges of data heterogeneity, model heterogeneity, and system heterogeneity because of the unique characteristics of edge computing environments. Edge computing environments are characterized by their distributed nature, which can lead to data heterogeneity and system heterogeneity. Additionally, edge computing environments are characterized by their limited resources, which can lead to model heterogeneity. These challenges can lead to poor performance of federated learning models. Therefore, in this section, we will explore these challenges and their potential solutions.

6.1 data heterogeneity

Data heterogeneity is a significant challenge in federated learning in edge environments. Data heterogeneity refers to the differences in data distribution across different devices. In edge environments, devices may have different types of sensors or different types of data sources, which can lead to significant differences in data distribution across devices. This can make it difficult to train a model that performs well across all devices.Specifically, data heterogeneity can lead to significant challenges in edge federated learning, including:

Class Imbalance Class imbalance occurs when some classes in the data distribution have significantly more samples than others, which can lead to biased models. One solution to this challenge is to use techniques like data sampling, data weighting, and oversampling/undersampling to balance the class distribution across different clients.

Non-IID Data Non-IID (Non-Independent and Identically Distributed) data occurs when the data distribution across different clients is significantly

different, which can affect the performance of the model. One solution to this challenge is to use techniques like data augmentation and transfer learning to make the data distribution more similar across different clients.

Label Noise Label noise refers to the incorrect labeling of data samples. This can occur due to various reasons, such as human error or machine error. Label noise can lead to incorrect model training and poor model performance. Therefore, it is essential to address label noise in federated learning.

Data Distribution Shift Data distribution shift refers to the change in the data distribution over time. This can occur due to various reasons, such as user behavior, device upgrades, and system updates. Data distribution shift can lead to poor model performance and the need to retrain the model. Therefore, it is essential to address data distribution shift in federated learning.

Data Quality The quality of the data can vary across different clients, which can affect the performance of the model. Therefore, it is essential to address Data Quality in federated learning.

To mitigate data heterogeneity in edge federated learning, kinds of approaches has been proposed.

Data selection Data selection techniques can be used to select a subset of data that is more representative of the overall data distribution. For example, a data selection technique called active learning can be used to select data points that are most informative [50].

Transfer learning Transfer learning techniques can be used to transfer knowledge from one device to another. For example, a transfer learning technique called fine-tuning can be used to fine-tune a pre-trained model on each device's data.

regular optimization Regular optimization can be used in FL to control the complexity of the model and prevent overfitting to the data from individual devices. By penalizing complex model structures, regular optimization can help to ensure that the model is generalizable across different types of data. [34] presented a novel framework, named FedProx, which addressed the issue of heterogeneity in federated networks. The proposed approach can be considered as an extension and re-parameterization of the state-of-the-art method for federated learning, FedAvg. Theoretically, [34] provided rigorous guarantees for the convergence of our framework, even when dealing with data from non-identical distributions . [37] proposed a Teacher-Student mechanism, which involves the integration of a regularization term into the objective function in order to adjust the gradients from clients among JointCloud that possess different data distributions. Regular optimization is important because it helps to prevent overfitting in machine learning models and improves the accuracy and generalizability of machine learning models.Overall, regular optimization is an essential tool for improving the performance and generalizability of machine learning models.

Meta learning Meta learning can help in FL by enabling the model to learn how to adapt to different types of data from different clients. By learning how to learn, the model can quickly adapt to new data types and extract relevant features from them. This can improve the overall performance of the model on het-

erogeneous data. [21] presented FL as a natural source of practical applications for MAML algorithms, and observed that the popular FL algorithm,Federated Averaging (McMahan et al., 2017), can be interpreted as a meta learning algorithm. Careful fine-tuning can yield a global model with higher accuracy,which is at the same time easier to personalize. [39] proposed a platform-aided collaborative learning framework. In order to mitigate the susceptibility of meta-learning algorithms to potential adversarial attacks, they have also proposed a robust version of the federated meta-learning algorithm that relies on distributionally robust optimization. [89] proposed a new algorithm design strategy from the primal-dual optimization perspective. Their strategy yields algorithms that can deal with non-convex objective functions, achieved the best possible optimization and communication complexity, and accommodated full-batch and mini-batch local computation models. [35] proposed a privacy-preserving spatial-temporal prediction technique via federated learning and proposed the personalized federated learning methods based on meta-learning. We automatically construct the global spatial-temporal pattern graph under a data federation. This global pattern graph incorporates and memorizes the local learned patterns of all of the clients, and each client leverages those global patterns to customize its own model by evaluating the difference between global and local pattern graph. meta learning is important in edge Federated Learning (FL) because it enables the model to learn how to adapt to different types of data from different clients. By learning how to learn, the model can quickly adapt to new data types and extract relevant features from them. This can improve the overall performance of the model on heterogeneous data, which is a common challenge in FL due to the diversity of data across different client devices. Overall, meta learning is an essential tool for improving the efficiency, performance, and generalizability of FL models.

Efficient communication is also essential in edge FL to ensure that the central server is able to aggregate the model updates from different clients effectively. This involves clear documentation of the data from different clients, as well as ensure that the model is generalizable across different client devices. [13] proposed a novel decentralized learning algorithm known as Cross-Gradient Aggregation (CGA). The CGA algorithm operates by having each agent aggregate cross-gradient information, which refers to the derivatives of its model with respect to its neighboring datasets. Furthermore, the model is updated using a projected gradient based on quadratic programming (QP). The CGA algorithm maintains the improved performance under information compression to reduce peer-to-peer communication overhead.

In addition, there are many literatures that alleviate the data heterogeneity problem from the perspectives of network topology, momentum technology, heuristic algorithm, etc. [82] proposed a hybrid learning mechanism, named Hybrid-FL, addressed both the client- and data-selection problems through heuristic algorithms. [70] proposed approach involves a method for distributedly selecting relevant data in a federated learning setting. [31] proposed comprehensive data partitioning strategies to cover non-IID data scenarios commonly en-

countered in the field and indicated that non-IID data presents considerable obstacles to achieving high accuracy in FL algorithms. [62] revealed that the primary factor contributing to the poor performance of the global model was a biased classifier. To address this issue, they proposed a novel privacy-preserving FL approach, Classifier Re-training with Federated Features (CReFF), designed for heterogeneous and long-tailed data. [40] proposed a novel momentum-based method to alleviate the difficulty of decentralized training . [6] proposed a novel topology called D-Cliques, which addresses the issue of gradient bias by organizing nodes into interconnected cliques. This arrangement ensured the local joint distribution within a clique accurately reflects the global class distribution.

In conclusion, data heterogeneity in federated learning refer to the difficulties that arise when attempting to train a model using data from multiple sources that have different distributions, formats, and quality levels. These challenges can be particularly pronounced in edge environments, where the data may be distributed across many different devices and networks with varying computational resources, but there are various solutions that can be used to mitigate its effects. These solutions involve using techniques like data sampling, data weighting, oversampling/undersampling, data augmentation, transfer learning, label smoothing, label filtering, label correction, data cleaning, data normalization, and data filtering to address class imbalance, non-IID data, label noise, data quality, and data privacy concerns.

6.2 model heterogeneity

Model heterogeneity is another significant challenge in federated learning in edge environments. Model heterogeneity refers to the differences in model architectures and hyperparameters across different devices. In edge environments, devices may have different hardware configurations and software environments (e.g., device capabilities and network connectivity among the participating clients), which can lead to significant differences in model architectures and hyperparameters across devices. This can make it difficult to train a model that performs well across all devices.Specifically, the model heterogeneity problem can lead to significant challenges in federated learning, including:

Model Compatibility Model compatibility refers to the ability of different models to work together seamlessly. In federated learning, model compatibility is critical because the models used to train the data sources are distributed across different devices or servers. Therefore, it is essential to ensure that the models are compatible with each other.

Model Drift Model drift refers to the change in the model's performance over time. This can occur due to various reasons, such as data distribution shift, model architecture changes, and hyperparameter updates. Model drift can lead to poor model performance and the need to retrain the model. Therefore, it is essential to address model drift in federated learning.

Model Compression/Pruning Model compression refers to the process of reducing the size of the model without losing significant performance. Model compression is critical in federated learning because the models used to train

the data sources are distributed across different devices or servers. Therefore, it is essential to compress the models to reduce communication overhead and improve model training efficiency.

Model Interpretability Model interpretability refers to the ability to understand and explain the model's behavior. In federated learning, model interpretability is critical because the models used to train the data sources are distributed across different devices or servers. Therefore, it is essential to ensure that the models are interpretable.

To mitigate model heterogeneity in edge federated learning, kinds of approaches has been proposed.

Efficient edge FL with Adaptive Model Compression and Pruning Efficient FL in edge environment with Model Compression and Pruning offers several benefits over traditional centralized training approaches. It enables the training of large-scale machine learning models on edge devices with limited computational resources and network bandwidth constraints. It also provides a privacy-preserving approach to machine learning, as the data remains on the edge devices and is not sent to a central server for processing. [22] proposed PruneFL, a novel FL approach that incorporates adaptive and distributed parameter pruning. PruneFL dynamically adjusts the model size during FL, reducing both communication and computation overhead while minimizing overall training time. [23] had developed and executed the FedMP framework, which utilizes adaptive model pruning to enhance efficiency. The approach includes a Multi-Armed Bandit online learning algorithm that can determine pruning ratios for heterogeneous edge nodes without prior knowledge of their computation and communication capabilities. [52] presented a new grow-and-prune methodology called scheduled GaP,which addresses previous shortcomings by iteratively growing a subset of layers to a dense state, followed by pruning them back to a sparse state after some training.

Efficient edge FL with Heterogeneous Network In traditional federated learning, all devices contribute to the training of the model, but this can be slow and inefficient due to the heterogeneity of the edge devices and the varying processing power and network connectivity. Efficient federated learning with heterogeneous networks is a method of federated learning that allows for more efficient and effective training of machine learning models across multiple devices and networks. And efficient federated learning with heterogeneous networks has been shown to outperform traditional federated learning methods in terms of training efficiency and model accuracy, especially useful in edge environment. [84] proposed a resource management approach for federated learning that incorporates module-based techniques and neural-structure-aware strategies. This approach assigns mobile clients with subnetworks of the global model based on the available resources of each client, thus optimizing the allocation of computational resources. [11] proposed HeteroFL, a novel federated learning framework designed to handle heterogeneous clients with varying computation and communication capabilities. The framework adaptively distributes subnetworks based on each client's capabilities, resulting in an efficient use of both computation

and communication resources. [96] introduced a unifying framework for heterogeneous federated learning algorithms featuring arbitrary adaptive online model pruning, accompanied by a general convergence analysis. They examined two critical factors that impact convergence: pruning-induced noise and minimum coverage index. Furthermore, [23] introduced a novel parameter synchronization system, R2SP, which recovers residuals synchronously in parallel. [85] proposed method for distributed fully connected neural network learning is called independent subnet training (IST). This approach naturally employs a "model parallel" strategy by limiting memory usage to only store a portion of network parameters on each device,which reduces communication volume and frequency and eliminats the need for data sharing between workers.

In conclusion, model heterogeneity is a significant challenge in federated learning, especially in edge environments where devices may have different hardware configurations, network bandwidths and data distributions. Model heterogeneity can lead to issues such as slow convergence, poor accuracy and privacy violations, but there are various solutions that can be used to mitigate its effects. These solutions involve using techniques like data augmentation, transfer learning, compression, quantization, selective aggregation, meta-learning, and secure and privacy-preserving techniques.

6.3 system heterogeneity

System heterogeneity is also a significant challenge in federated learning in edge environments. System heterogeneity refers to the differences in capabilities and resources across different devices. In edge environments, devices may have different processing power, memory capacity, and battery power, which can lead to significant differences in capabilities and resources across devices. This can make it difficult to train a model that performs well across all devices.Specifically, System heterogeneity can lead to significant challenges in federated learning, including:

Network Latency Network latency refers to the delay in communication between different devices or servers. In edge federated learning, network latency is critical because the data sources and models are distributed across different devices or servers. Therefore, it is essential to reduce network latency to improve model training efficiency.

Device Heterogeneity Device heterogeneity refers to the differences in the hardware configurations of different devices used to train the model. In federated learning, device heterogeneity can lead to significant challenges in model training, such as varying processing speeds, memory limitations, and battery life. Therefore, it is essential to consider device heterogeneity in federated learning.

System Security System security refers to the protection of the federated learning system from malicious attacks. In federated learning, system security is critical because the data sources and models are distributed across different devices or servers. Therefore, it is essential to ensure that the federated learning system is secure from attacks.

Resource Constraints Clients may have limited resources, such as computational power and memory, which can affect the performance of the federated learning algorithm. To address this challenge, federated learning algorithms can be designed to use lightweight models and compression techniques to reduce the computational and memory requirements.

Communication Overhead Communication overhead refers to the amount of data exchanged between different devices or servers during the model training process. In federated learning, communication overhead is critical because the data sources and models are distributed across different devices or servers. Therefore, it is essential to reduce communication overhead to improve model training efficiency.

To mitigate system heterogeneity in federated learning, kinds of approaches has been proposed. [59] proposed a new FL protocol, Fed CS, addressed the challenge of resource management and facilitates efficient FL by actively managing clients based on their resource conditions. FedCS tackles a client selection problem with resource constraints, enabling the server to aggregate a maximum number of client updates and expedite performance improvement in ML models. [60] proposed federated learning method is designed to be resilient to stragglers and incorporates statistical characteristics of clients' data to adaptively select them and expedite the learning process.To address the difficulties of resource limitation and edge heterogeneity. [23] designed and implemented FedMP, an efficient FL framework through adaptive model pruning.

In conclusion, system heterogeneity is a significant challenge in federated learning in edge environment. Edge devices have different hardware specifications, such as CPU, memory, and storage, which can affect the performance and convergence of the federated learning model, but there are various solutions that can be used to mitigate its effects. These solutions involve using techniques like adaptive communication strategies, device-aware algorithms, standardized communication protocols and interfaces, lightweight models, compression techniques, and secure and privacy-preserving techniques to address network connectivity, device heterogeneity, software and framework compatibility, resource constraints, and security and privacy concerns.

Future developments in federated learning in the edge environment include the integration of edge intelligence, which offloads some of the computation and decision-making to the edge devices, reducing the communication overhead and improving the model's response time. Another development is the use of federated reinforcement learning, which enables the edge devices to learn from their own experiences and from the experiences of other devices, improving the overall performance of the federated learning model.

6.4 Representative algorithm

In this section, we introduce two representative algorithms that have made significant breakthroughs in data and model heterogeneity, respectively.

6.4.1 Cross-Gradient Aggregation for Decentralized Learning from Non-IID Data This paper [13] proposes a novel decentralized learning algorithm called Cross-Gradient Aggregation (CGA) that enables collaborative agents to learn models using a distributed dataset without the need for a central parameter server. The key contribution of this paper is to overcome the issue of handling non-IID data distributions in a decentralized learning setting.

Specifically, the CGA algorithm proposes a series of steps to achieve collaborative learning among agents in a decentralized system. Firstly, each agent calculates the gradients of the model parameters based on its own data set. Secondly, agents share their model parameters with their neighbors. Thirdly, agents compute gradients of their neighbors' models on their own data set and send back the cross gradients to the respective neighbors. Fourthly, cross gradients and local gradients are projected into an aggregated gradient using Quadratic Programming. Lastly, this aggregated gradient is utilized to update the model parameter for improved performance.

The cross gradient received by each client is calculated from its own local model and neighbor data.

Definition 8. *cross-gradient: For agents j, l, consider the model parameter x_j of agent j, the dataset D_l and objective function f_l in agent l. The cross-gradient denote as:*

$$g^{jl} := \nabla f_l(D_l; x_j) \tag{7}$$

In addition, the authors provide theoretical analysis of the convergence characteristics of CGA and achieve $O(\frac{1}{\sqrt{NK}})$ convergence rate (the number of agents: N and the communication rounds: K). Moreover, the authors demonstrate its efficiency on non-IID data distributions sampled from the MNIST and CIFAR-10 datasets. They compare the effectiveness of their algorithm with other baseline decentralized algorithms.

In conclusion, this paper provides an effective solution for handling non-IID data distributions in decentralized learning settings. By aggregating cross-gradient information from neighboring agents, the algorithm achieves state-of-the-art results on benchmark datasets while eliminating the need for a central parameter server.

6.4.2 Federated Learning with Online Adaptive Heterogeneous Local Models hanhan zhou et al. [95] recently proposed heterogeneous FL algorithm with theoretical convergence analysis. The key idea is arbitrary adaptive online model pruning. By arbitrary adaptive pruning strategies, different clients form heterogeneous local models according to their local resource constraints. And during training process, the local model is change continuously due to the changing local resources.

More formally, in training round q, this paper denotes the initial global model by θ_q. Each client n generates a pruning mask $m_{q,n} \in \{0,1\}^{|\theta_q|}$ by adaptive online pruning strategy, then let $\odot$ be element-wise product and the local model

on client n is defined by $\theta_{q,n,0} = \theta_q \odot m_{q,n}$. As mentioned previously, the heterogeneous local models are formed by arbitrary adaptive online model pruning. Thus, Assume that each global model parameter i is contained in the subset N_q^i of clients (N_q^i clients whose local models contain global model parameter i).

In addition, the authors give convergence analysis with some common assumptions in federated learning and the proposed algorithm achieves $O(\frac{1}{\sqrt{Q}})$ convergence rate. Significantly, the analysis illuminates two key factors impacting the convergence in heterogeneous federated learning: pruning-induced noise and minimum coverage index.

Assumption 1 *pruning-induced noise: Assume that for some $\delta^2 \in [0, 1)$, pruning-induced error on client n and any round q is bounded by*

$$\|\theta_q - \theta_q \odot m_{q,n}\| \leq \delta^2 \|\theta_q\|^2 \tag{8}$$

Assumption 1 is widely applied to communication compression and model pruning. And the convergence result in algorithm 1 implies that more pruning level may cause a larger error, deviating from standard federated learning at a speed quantified by δ^2.

Definition 9. *minimum coverage index: minimum coverage index Γ^* denote as:*

$$\Gamma^* = \min_{q,i} |N_q^i|, i \in K, \forall q \tag{9}$$

where $|N_q^i|$ is the number of clients whose local models contain global parameters i. Thus, minimum coverage index Γ^* represents the minimum occurrence of any parameters contained in the local models in all rounds. For any parameter, it is at least contained in one local model which ensures that all parameters can be updated. And the convergence result indicates that the more times a parameter is contained by local models, the faster it will converge to the desired target.

In conclusion, [95] investigates the convergence properties of federated learning (FL) with heterogeneous local models that may vary over time and across clients and achieves a convergence rate of $\frac{1}{\sqrt{Q}}$ (the communication rounds: Q). The optimality gap is characterized by two factors: pruning-induced noise and minimum coverage index. The result encompasses various important FL algorithms as special cases and provides new insights into optimized pruning strategies in heterogeneous FL, emphasizing the minimum coverage index (Γ_{min}) and pruning-induced noise (δ^2). The work contributes to a theoretical understanding of heterogeneous FL with adaptive local model pruning and presents valuable insights for future algorithm design.

6.4.3 Summary To summarize, Cross-Gradient Aggregation [13] enables collaborative agents to learn models from both independently and identically distributed (IID) and non-IID data distributions. CGA algorithm differs from other decentralized learning algorithms in that it aggregates cross-gradient information, which are derivatives of an agent's model with respect to its neighbors'

datasets. This allows agents to learn from non-IID data distributions, which is a common issue in decentralized learning settings. Additionally, CGA uses quadratic programming to update models based on the aggregated information, which is different from other decentralized algorithms that use stochastic gradient descent or variants thereof.

Literature [95] focuses on Federated Learning with Online Adaptive Heterogeneous Local Models, which addresses the challenge of training models with varying computation and communication resources. The authors provide theoretical analysis for the convergence of these algorithms and characterize the optimality gap, which depends on pruning-induced noise and a new notion of minimum coverage index. The result recovers a number of important Federated Learning algorithms as special cases and provides new insights on designing optimized pruning strategies in heterogeneous Federated Learning. The novelty of this paper lies in its analysis of convergence conditions for Federated Learning with heterogeneous local models, which can vary over time and across clients. It also introduces a new notion of minimum coverage index to address the challenges presented by this approach. Therefore, the two representative algorithms have made significant breakthroughs in data and model heterogeneity.

6.5 Future Directions

Edge Federated Learning (EFL) is a rapidly evolving research area that has gained significant attention in recent years. It aims to combine the benefits of edge computing with federated learning to enable distributed machine learning across a network of edge devices while maintaining privacy, security, and efficiency. However, this domain still faces several research challenges and directions concerning data heterogeneity, model heterogeneity, and system heterogeneity. This section outlines the key issues and future research directions in these areas.

6.5.1 Data Heterogeneity Data heterogeneity is a prominent issue in EFL, as edge devices often generate and store diverse types of data. The main challenges and future research directions related to data heterogeneity include:

- Data Sampling and Preprocessing: Developing robust methods for data sampling and preprocessing to address the non-IID (independent and identically distributed) nature of edge data, which can improve model accuracy and convergence.

- Data Augmentation and Transformation: Investigating novel data augmentation and transformation techniques to deal with imbalanced and diverse datasets, which can lead to better learning outcomes.

- Privacy-Preserving Data Sharing: Designing secure and privacy-preserving mechanisms for data sharing among edge devices, which can enable better collaboration and learning without compromising user privacy.

- Heterogeneous Data Fusion: Developing methods to effectively fuse and utilize heterogeneous data types, such as structured, semi-structured, and unstructured data, in the learning process to improve the overall model performance.

6.5.2 Model Heterogeneity Model heterogeneity refers to the diversity of machine learning models used across edge devices. It is essential to address model heterogeneity to ensure that EFL can be applied effectively across various domains and applications. Key challenges and future research directions include:

- Adaptive Model Selection: Investigating methods for adaptively selecting the most appropriate model for each edge device, based on factors such as computational capacity, data availability, and application requirements.

- Meta-Learning: Exploring meta-learning techniques that can learn to learn from diverse models, which can help improve the performance of EFL systems.

- Model Compression and Quantization: Developing techniques for model compression and quantization to reduce the communication overhead and improve the efficiency of EFL.

- Model Personalization: Investigating approaches for personalized model training and adaptation, which can enhance the user experience and model performance by accounting for individual preferences and requirements.

6.5.3 System Heterogeneity System heterogeneity refers to the varying characteristics of edge devices and communication networks in EFL systems. Addressing system heterogeneity is crucial for ensuring the scalability, robustness, and efficiency of EFL. Key challenges and future research directions include:

- Resource Allocation and Optimization: Developing methods for efficient resource allocation and optimization among edge devices, taking into account factors such as computation power, storage, and communication capabilities.

- Heterogeneous Communication Protocols: Investigating and designing communication protocols that can support diverse networking technologies and standards, ensuring seamless collaboration among edge devices.

- Fault Tolerance and Robustness: Developing fault-tolerant and robust EFL mechanisms that can handle device failures, communication errors, and other system anomalies.

- Security and Trust Management: Designing security mechanisms and trust management schemes that can address potential attacks and malicious behaviors in EFL systems, ensuring the overall integrity and reliability of the learning process.

In conclusion, addressing the challenges and future research directions related to data heterogeneity, model heterogeneity, and system heterogeneity is crucial for the successful implementation and deployment of edge federated learning. These areas offer exciting opportunities for researchers and practitioners to contribute to the development of more efficient, secure, and privacy-preserving EFL systems.

References

1. Abreha, A.G., Liang, X., Liang, Y.C.: Management of communication resources for federated learning over wireless networks: A survey. IEEE Communications Surveys & Tutorials **24**(1), 1–1 (2022)

2. arXiv: Asynchronous federated learning with differential privacy for edge intelligence. arXiv preprint arXiv:1912.07902 (2019)
3. Bai, Y., Chen, L., Abdel-Mottaleb, M., Xu, J.: Automated ensemble for deep learning inference on edge computing platforms. IEEE Internet Things J. **9**(6), 4202–4213 (2022). https://doi.org/10.1109/JIOT.2021.3102945, https://doi.org/10.1109/JIOT.2021.3102945
4. Bankovic, Z., Fraga, D., Moya, J.M., Vallejo, J.C., Araujo, Á., Malagón, P., de Goyeneche, J., Villanueva, D., Romero, E., Blesa, J.: Detecting and confining sybil attack in wireless sensor networks based on reputation systems coupled with self-organizing maps. In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds.) Artificial Intelligence Applications and Innovations - 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010. Proceedings. IFIP Advances in Information and Communication Technology, vol. 339, pp. 311–318. Springer (2010). https://doi.org/10.1007/978-3-642-16239-8_41, https://doi.org/10.1007/978-3-642-16239-8_41
5. Bao, X., Su, C., Xiong, Y., Huang, W., Hu, Y.: Flchain: A blockchain for auditable federated learning with trust and incentive. In: 5th International Conference on Big Data Computing and Communications, BIGCOM 2019, QingDao, China, August 9-11, 2019. pp. 151–159. IEEE (2019). https://doi.org/10.1109/BIGCOM.2019.00030, https://doi.org/10.1109/BIGCOM.2019.00030
6. Bellet, A., Kermarrec, A., Lavoie, E.: D-cliques: Compensating noniidness in decentralized federated learning with topology. CoRR **abs/2104.07365** (2021), https://arxiv.org/abs/2104.07365
7. Blanchard, P., Mhamdi, E.M.E., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 119–129 (2017), https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html
8. Cao, D., Chang, S., Lin, Z., Liu, G., Sun, D.: Understanding distributed poisoning attack in federated learning. In: 25th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2019, Tianjin, China, December 4-6, 2019. pp. 233–239. IEEE (2019). https://doi.org/10.1109/ICPADS47876.2019.00042, https://doi.org/10.1109/ICPADS47876.2019.00042
9. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: Seltzer, M.I., Leach, P.J. (eds.) Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999. pp. 173–186. USENIX Association (1999), https://dl.acm.org/citation.cfm?id=296824
10. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W.: Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems **35**(4), 83–93 (2020)
11. Diao, E., Ding, J., Tarokh, V.: Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=TNkPBBYFkXg

12. Dwork, C.: Differential privacy. Automata, Languages and Programming pp. 1–12 (2006)
13. Esfandiari, Y., Tan, S.Y., Jiang, Z., Balu, A., Herron, E., Hegde, C., Sarkar, S.: Cross-gradient aggregation for decentralized learning from non-iid data. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 3036–3046. PMLR (2021), http://proceedings.mlr.press/v139/esfandiari21a.html
14. Fang, M., Cao, X., Jia, J., Gong, N.Z.: Local model poisoning attacks to byzantine-robust federated learning. In: Capkun, S., Roesner, F. (eds.) 29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020. pp. 1605–1622. USENIX Association (2020), https://www.usenix.org/conference/usenixsecurity20/presentation/fang
15. Fitzi, M., Gisin, N., Maurer, U.M., von Rotz, O.: Unconditional byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In: Knudsen, L.R. (ed.) Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2332, pp. 482–501. Springer (2002). https://doi.org/10.1007/3-540-46035-7_32, https://doi.org/10.1007/3-540-46035-7_32
16. Gaid, M., Salloum, S.A.: Homomorphic encryption (2022), https://www.researchgate.net/publication/351945616_Homomorphic_Encryption
17. Hosseini, S.M., Sikaroudi, M., Babaei, M., Tizhoosh, H.: Cluster based secure multiparty computation in federated learning for histopathology images. arXiv preprint arXiv:2208.10919 (2022)
18. IEEE: Ai at the edge: Blockchain-empowered secure multiparty learning with heterogeneous models. IEEE Internet of Things Journal **7**(10), 9600–9610 (2020)
19. IEEE: Differentially private federated learning in edge networks: The perspective of noise reduction. IEEE Network **36**(5), 167–172 (2022)
20. IEEE: Partially encrypted multi-party computation for federated learning. IEEE Conference Publication (2022)
21. Jiang, Y., Konečný, J., Rush, K., Kannan, S.: Improving federated learning personalization via model agnostic meta learning. CoRR **abs/1909.12488** (2019), http://arxiv.org/abs/1909.12488
22. Jiang, Y., Wang, S., Ko, B.J., Lee, W., Tassiulas, L.: Model pruning enables efficient federated learning on edge devices. CoRR **abs/1909.12326** (2019), http://arxiv.org/abs/1909.12326
23. Jiang, Z., Xu, Y., Xu, H., Wang, Z., Qiao, C., Zhao, Y.: Fedmp: Federated learning through adaptive model pruning in heterogeneous edge computing. In: 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022. pp. 767–779. IEEE (2022). https://doi.org/10.1109/ICDE53745.2022.00062, https://doi.org/10.1109/ICDE53745.2022.00062
24. Jin, W., Yao, Y., Han, S., Joe-Wong, C., Ravi, S., Avestimehr, S., He, C.: Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system. arXiv preprint arXiv:2303.10837 (2023)
25. Kang, J., Xiong, Z., Niyato, D., Zou, Y., Zhang, Y., Guizani, M.: Reliable federated learning for mobile networks. IEEE Wirel. Commun. **27**(2), 72–80 (2020). https://doi.org/10.1109/MWC.001.1900119, https://doi.org/10.1109/MWC.001.1900119

26. Khan, M.B., Khan, M.U., Khan, M.K., Khan, M.A.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity **4**(1) (Dec 2021). https://doi.org/10.1186/s42400-021-00105-6, https://cybersecurity.springeropen.com/articles/10.1186/s42400-021-00105-6
27. Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., van der Maaten, L.: Crypten: Secure multi-party computation meets machine learning. arXiv preprint arXiv:2109.00984 (2021)
28. Konečn'y, J., McMahan, H.B., Yu, F.X., Richt'arik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
29. Lamport, L., Shostak, R.E., Pease, M.C.: The byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982). https://doi.org/10.1145/357172.357176, https://doi.org/10.1145/357172.357176
30. Li, J., Wang, X., Zhang, Y., Liu, Y., Li, T., Zhu, Y.: Do gradient inversion attacks make federated learning unsafe? arXiv preprint arXiv:2202.06924 (2022)
31. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. In: 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022. pp. 965–978. IEEE (2022). https://doi.org/10.1109/ICDE53745.2022.00077, https://doi.org/10.1109/ICDE53745.2022.00077
32. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine **37**(3), 50–60 (2019)
33. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning systems: Vision, hype and reality for data privacy and protection. arXiv preprint arXiv:2007.07258 (2020)
34. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Dhillon, I.S., Papailiopoulos, D.S., Sze, V. (eds.) Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020. mlsys.org (2020), https://proceedings.mlsys.org/book/316.pdf
35. Li, W., Wang, S.: Federated meta-learning for spatial-temporal prediction. Neural Comput. Appl. **34**(13), 10355–10374 (2022). https://doi.org/10.1007/s00521-021-06861-3, https://doi.org/10.1007/s00521-021-06861-3
36. Li, X., Wang, S., Zhang, X., Liang, W.: Edgefed: An edge computing based federated learning framework for privacy-preserving iot applications. IEEE Internet of Things Journal **7**(11), 11179–11190 (2020)
37. Li, X., Liu, N., Chen, C., Zheng, Z., Li, H., Yan, Q.: Communication-efficient collaborative learning of geo-distributed jointcloud from heterogeneous datasets. In: 2020 IEEE International Conference on Joint Cloud Computing. pp. 22–29 (2020). https://doi.org/10.1109/JCC49151.2020.00013
38. Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., Yan, Q.: A blockchain-based decentralized federated learning framework with committee consensus. IEEE Netw. **35**(1), 234–241 (2021). https://doi.org/10.1109/MNET.011.2000263, https://doi.org/10.1109/MNET.011.2000263
39. Lin, S., Yang, G., Zhang, J.: A collaborative learning framework via federated meta-learning. In: 40th IEEE International Conference on Distributed Computing Systems, ICDCS 2020, Singapore, November 29 - December 1, 2020. pp. 289–299. IEEE (2020). https://doi.org/10.1109/ICDCS47774.2020.00032, https://doi.org/10.1109/ICDCS47774.2020.00032

40. Lin, T., Karimireddy, S.P., Stich, S.U., Jaggi, M.: Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 6654–6665. PMLR (2021), http://proceedings.mlr.press/v139/lin21c.html
41. Liu, W., Yang, Y., Chen, J.: Federated learning at the edge: A comprehensive review. In: 2021 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2021)
42. Liu, Y., Chen, M., Liu, X., Qin, T., Chen, E., Song, S.: Communication-efficient federated learning with sketching and adaptive round numbers. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 1752–1759 (2019)
43. Liu, Y., Peng, J., Kang, J., Iliyasu, A.M., Niyato, D., El-Latif, A.A.A.: A secure federated learning framework for 5g networks. IEEE Wirel. Commun. **27**(4), 24–31 (2020). https://doi.org/10.1109/MWC.01.1900525, https://doi.org/10.1109/MWC.01.1900525
44. Liu, Y., Zhang, J., Li, Z.: Privacy-preserving federated edge learning: Modeling and optimization. IEEE Journal on Selected Areas in Communications **39**(8), 2384–2396 (2021). https://doi.org/10.1109/JSAC.2021.3081023
45. Liu, Y., Zhang, J., Li, Z.: Source inference attacks in federated learning (2021)
46. Liu, Y., Zhang, J., Li, Z.: Privacy-preserving and verifiable federated learning framework for edge computing. IEEE Transactions on Industrial Informatics **18**(3), 1745–1754 (2022). https://doi.org/10.1109/TII.2021.3118885
47. Liu, Y., Li, T., Zhu, Y., Lin, Z.: Broadening differential privacy for deep learning against model inversion attacks. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 10–13. IEEE (2020)
48. Liu, Y., Li, T., Zhu, Y., Lin, Z.: Data poisoning attacks against federated learning systems (2020)
49. Liu, Z., Yang, K., Chen, M., Alouini, M.S.: Edge federated learning with over-the-air computation: A communication-efficient approach to decentralized machine learning. In: 2022 IEEE International Conference on Communications (ICC). IEEE (2022)
50. de Luca, A.B., Zhang, G., Chen, X., Yu, Y.: Mitigating data heterogeneity in federated learning with data augmentation. CoRR **abs/2206.09979** (2022). https://doi.org/10.48550/arXiv.2206.09979, https://doi.org/10.48550/arXiv.2206.09979
51. Ma, J., Naas, S.A., Sigg, S., Lyu, X.: Privacy-preserving federated learning based on multi-key homomorphic encryption. arXiv preprint arXiv:2104.06824 (2021)
52. Ma, X., Qin, M., Sun, F., Hou, Z., Yuan, K., Xu, Y., Wang, Y., Chen, Y., Jin, R., Xie, Y.: Effective model sparsification by scheduled grow-and-prune methods. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=xa6otUDdP2W
53. Mazieres, D.: The stellar consensus protocol: A federated model for internet-level consensus. Stellar Development Foundation **32**, 1–45 (2015)
54. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
55. McMahan, H.B., Moore, E., Ramage, D., y Arcas, B.A.: Federated learning of deep networks using model averaging. CoRR **abs/1602.05629** (2016), http://arxiv.org/abs/1602.05629

56. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629 (2016)
57. McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Federated learning: Challenges, methods, and future directions (2019)
58. Mugunthan, V., Polychroniadou, A., Byrd, D., Balch, T.H.: Smpai: Secure multiparty computation for federated learning. In: Advances in Neural Information Processing Systems. p. 11877–11887 (2019)
59. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: 2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019. pp. 1–7. IEEE (2019). https://doi.org/10.1109/ICC.2019.8761315, https://doi.org/10.1109/ICC.2019.8761315
60. Reisizadeh, A., Tziotis, I., Hassani, H., Mokhtari, A., Pedarsani, R.: Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity. IEEE J. Sel. Areas Inf. Theory **3**(2), 197–205 (2022). https://doi.org/10.1109/JSAIT.2022.3205475, https://doi.org/10.1109/JSAIT.2022.3205475
61. ScienceDirect: Vpfl: A verifiable privacy-preserving federated learning scheme for edge devices. ScienceDirect (2022)
62. Shang, X., Lu, Y., Huang, G., Wang, H.: Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022. pp. 2218–2224. ijcai.org (2022). https://doi.org/10.24963/ijcai.2022/308, https://doi.org/10.24963/ijcai.2022/308
63. Shi, D., Li, L., Chen, R., Prakash, P., Pan, M., Fang, Y.: Toward energy-efficient federated learning over 5g+ mobile devices. IEEE Wirel. Commun. **29**(5), 44–51 (2022). https://doi.org/10.1109/MWC.003.2100028, https://doi.org/10.1109/MWC.003.2100028
64. Singh, A., Ngan, T., Druschel, P., Wallach, D.S.: Eclipse attacks on overlay networks: Threats and defenses. In: INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain. IEEE (2006). https://doi.org/10.1109/INFOCOM.2006.231, https://doi.org/10.1109/INFOCOM.2006.231
65. Singh, S., Kumar, A., Kumar, R.: Federated edge learning: Design issues and challenges. In: 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). pp. 1–6 (2020). https://doi.org/10.1109/ANTS49819.2020.9343292
66. Singh, S., Kumar, A., Kumar, R.: Safelearn: Secure aggregation for private federated learning. In: 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER). pp. 1–6 (2021). https://doi.org/10.1109/DISCOVER53268.2021.9474309
67. Sit, E., Morris, R.T.: Security considerations for peer-to-peer distributed hash tables. In: Druschel, P., Kaashoek, M.F., Rowstron, A.I.T. (eds.) Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2429, pp. 261–269. Springer (2002). https://doi.org/10.1007/3-540-45748-8_25, https://doi.org/10.1007/3-540-45748-8_25

68. Tao, Y., Cui, S., Xu, W., Yin, H., Yu, D., Liang, W., Cheng, X.: Byzantine-resilient federated learning at edge. IEEE Transactions on Computers pp. 1–14 (2023). https://doi.org/10.1109/TC.2023.3257510
69. Tian, H., Wen, Y., Zhang, F., Shao, Y., Li, B.: A distributed threshold additive homomorphic encryption for federated learning with dropout resiliency based on lattice. In: Chen, X., Shen, J., Susilo, W. (eds.) Cyberspace Safety and Security - 14th International Symposium, CSS 2022, Xi'an, China, October 16-18, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13547, pp. 277–292. Springer (2022). https://doi.org/10.1007/978-3-031-18067-5_20, https://doi.org/10.1007/978-3-031-18067-5_20
70. Tuor, T., Wang, S., Ko, B.J., Liu, C., Leung, K.K.: Overcoming noisy and irrelevant data in federated learning. In: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021. pp. 5020–5027. IEEE (2020). https://doi.org/10.1109/ICPR48806.2021.9412599, https://doi.org/10.1109/ICPR48806.2021.9412599
71. Venkatasubramanian, M., Lashkari, A.H., Hakak, S.: Iot malware analysis using federated learning: A comprehensive survey. IEEE Access **11**, 5004–5018 (2023). https://doi.org/10.1109/ACCESS.2023.3235389
72. Wang, S., Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Global update guidance for communication-efficient federated learning. In: Proceedings of the 38th International Conference on Machine Learning. p. 10764–10774 (2021)
73. Wang, S., Tuor, T., Salonidis, T., Leung, K.K., Makaya, C., He, T., Chan, K.: Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications **37**(6), 1205–1221 (2019)
74. Wang, S., Tuor, T., Salonidis, T., et al.: When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In: Proceedings of IEEE INFOCOM 2018 - IEEE Conference on Computer Communications (2018)
75. Wei, X., Fan, J., Chen, M., Zhang, G.: A general framework for detecting malicious peers in reputation-based peer-to-peer systems. In: 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. pp. 463–468 (2014). https://doi.org/10.1109/3PGCIC.2014.95
76. Xie, Y., Zhang, W., Pi, R., Wu, F., Chen, Q., Xie, X., Kim, S.: Robust federated learning against both data heterogeneity and poisoning attack via aggregation optimization (2022)
77. Xu, J., Hu, Y., Wang, X., Ma, J.: Sok: Training machine learning models over multiple sources with privacy preservation (2020)
78. Yang, Y., Liu, W., Chen, J.: Federated learning for edge intelligence: A comprehensive survey. In: Proceedings of the IEEE. vol. 108, pp. 1297–1330. IEEE (2020)
79. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 5650–5659. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/yin18a.html
80. Yin, D., Chen, Y., Ramchandran, K., Bartlett, P.L.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 5636–5645. PMLR (2018), http://proceedings.mlr.press/v80/yin18a.html

81. Yin, K., Wu, B., Zhu, R., Xiao, L., Tan, Z., He, G., Wang, Z., Yin, G.: Dldp-fl: Dynamic local differential privacy federated learning method based on mesh network edge devices. Journal of Computational Science **63**, 101789 (2022)
82. Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., Yonetani, R.: Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data. In: 2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020. pp. 1–7. IEEE (2020). https://doi.org/10.1109/ICC40277.2020.9149323, https://doi.org/10.1109/ICC40277.2020.9149323
83. Yu, D., Zou, Z., Chen, S., Tao, Y., Tian, J., Jia, X.: Decentralized parallel sgd with privacy preservation in vehicular networks. IEEE Transactions on Vehicular Technology **70**(6), 5211–5220 (2021)
84. Yu, R., Li, P.: Toward resource-efficient federated learning in mobile edge computing. IEEE Netw. **35**(1), 148–155 (2021). https://doi.org/10.1109/MNET.011.2000295, https://doi.org/10.1109/MNET.011.2000295
85. Yuan, B., Wolfe, C.R., Dun, C., Tang, Y., Kyrillidis, A., Jermaine, C.: Distributed learning of fully connected neural networks using independent subnet training. Proc. VLDB Endow. **15**(8), 1581–1590 (2022), https://www.vldb.org/pvldb/vol15/p1581-wolfe.pdf
86. Zeng, T., Semiari, O., Chen, M., Saad, W., Bennis, M.: Federated learning on the road autonomous controller design for connected and autonomous vehicles. IEEE Trans. Wirel. Commun. **21**(12), 10407–10423 (2022). https://doi.org/10.1109/TWC.2022.3183996, https://doi.org/10.1109/TWC.2022.3183996
87. Zhang, J., Wang, X., Wang, H., Zhang, Q., Ren, K.: Fedzip: An efficient compression framework for federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. p. 14477–14486 (2020)
88. Zhang, S., Liu, Y., Li, T., Zhu, Y.: Untargeted poisoning attack detection in federated learning via behavior attestation (2021)
89. Zhang, X., Hong, M., Dhople, S.V., Yin, W., Liu, Y.: Fedpd: A federated learning framework with adaptivity to non-iid data. IEEE Trans. Signal Process. **69**, 6055–6070 (2021). https://doi.org/10.1109/TSP.2021.3115952, https://doi.org/10.1109/TSP.2021.3115952
90. Zhang, Y., Li, X., Ji, X.: A survey of federated learning for edge computing: Research problems and solutions. Journal of Systems Architecture **117**, 102098 (2021). https://doi.org/https://doi.org/10.1016/j.sysarc.2021.102098, https://www.sciencedirect.com/science/article/pii/S138376212100009X
91. Zhang, Y., Li, X., Ji, X.: Ppefl: An edge federated learning architecture with privacy-preserving mechanism. Wireless Communications and Mobile Computing **2022**, 1–14 (2022)
92. Zhang, Y., Li, X., Zhang, X., Yang, X., Liu, X.: Partially encrypted multi-party computation for federated learning. In: 2021 IEEE International Conference on Communications (ICC). pp. 1–6 (2021). https://doi.org/10.1109/ICC42927.2021.9500984
93. Zhang, Y., Liu, Y., Li, T., Zhu, Y.: Privacy leakage of adversarial training models in federated learning systems. arXiv preprint arXiv:2202.10546 (2022)
94. Zhao, Y., Liu, Y., Zhang, J., Li, Z.: Secure and privacy-preserving federated learning via co-utility (2021)
95. Zhou, D., Zhang, Y., Sonabend-W, A., Wang, Z., Lu, J., Cai, T.: Federated offline reinforcement learning. arXiv preprint arXiv:2206.05581 (2022)

96. Zhou, H., Lan, T., Venkataramani, G., Ding, W.: On the convergence of heterogeneous federated learning with arbitrary adaptive online model pruning. CoRR **abs/2201.11803** (2022), https://arxiv.org/abs/2201.11803