Asynchronous Federated Learning on Heterogeneous Devices: A Survey

Chenhao Xu\textsuperscript{a}, Youyang Qu\textsuperscript{b}, Yong Xiang\textsuperscript{a}, Longxiang Gao\textsuperscript{b}

\textsuperscript{a} Deakin Blockchain Innovation Lab, School of Information Technology, Deakin University, Geelong, VIC, Australia
\textsuperscript{b} Shandong Computer Science Center, Qilu University of Technology, China

Abstract

Federated learning (FL) is a kind of distributed machine learning framework, where the global model is generated on the centralized aggregation server based on the parameters of local models, addressing concerns about privacy leakage caused by the collection of local training data. With the growing computational and communication capacities of edge and IoT devices, applying FL on heterogeneous devices is becoming a prevailing trend. Nonetheless, the synchronous aggregation strategy in the classic FL paradigm, particularly on heterogeneous devices, encounters limitations in resource utilization due to the need to wait for slow devices before aggregation in each training round. Furthermore, the uneven distribution of data across devices (i.e., data heterogeneity) in real-world scenarios adversely impacts the accuracy of the global model. Consequently, many asynchronous FL (AFL) approaches have been introduced across various application contexts to enhance efficiency, performance, privacy, and security. This survey comprehensively analyzes and summarizes existing AFL variations using a novel classification scheme, including device heterogeneity, data heterogeneity, privacy, and security on heterogeneous devices, as well as applications on heterogeneous devices. Finally, this survey reveals rising challenges and presents potentially promising research directions in this under-investigated domain.

Keywords: Asynchronous Federated Learning, Device Heterogeneity, Data Heterogeneity, Privacy, Security

1. Introduction

In recent years, the rapid expansion of computational capabilities coupled with the swift evolution of communication infrastructures has directly contributed to the flourishing of machine learning (ML), a pivotal driving force behind numerous contemporary technologies \cite{1,2}. Nonetheless, the training of ML models necessitates a substantial volume of high-quality data, a crucial requirement for model trainers operating in real-world scenarios \cite{3}. Notably, the emphasis on preserving privacy during data sharing continues to persist, with newly enacted legislation and regulations further complicating the process of data acquisition \cite{4,5,6}. Furthermore, industries exhibit hesitancy in sharing their local data due to competitive pressures, privacy concerns, and other potential considerations \cite{7,8,9}. All these factors jointly give rise to the challenge of isolated data islands. As a result, gathering data from diverse reliable sources becomes a near-impossible task, often accompanied by prohibitively high costs \cite{10,11}.

Federated learning (FL) presents an innovative framework facilitating collaborative ML model training among multiple entities without requiring direct access to their respective local training data. Initially introduced by Google in 2016, FL emerges as a promising ML approach that effectively addresses the imperatives of data privacy and communication efficiency \cite{12,13}. The fundamental objective of FL revolves around ensuring personal data privacy and engendering robust ML models across multiple participants or computational nodes while upholding legal mandates \cite{14,15}. Consequently, FL has found its way into numerous research papers, employing a central server to collect parameters of local models from nodes (referred to as “local models” henceforth) prior to aggregating them into a global model during each round of training \cite{16}.

Amidst the extensive rollout of the 5G network and the swift evolution of hardware capabilities, heterogeneous devices, encompassing both edge and IoT devices, are experiencing augmented communication and computational capabilities, paving the way for an expanded array of applications \cite{17}. Compared with classic ML approaches, FL presents a range of merits specifically tailored to edge applications \cite{14,18}: (1) Enhanced preservation of local data privacy, facilitated by the gradients-based aggregation of the global model; (2) Reduced network transmission latency, as the training data remains localized instead of being transmitted to cloud servers; (3) Elevated model quality, owing to the incorporation of learned features from other devices. Consequently, FL serves as a catalyst for collaborative ML model training across heterogeneous devices, a phenomenon well-documented in numerous research publications.

When employing classical FL on devices with limited resources, several drawbacks become apparent \cite{21}: (1) Device Unreliability. The presence of heterogeneous devices introduces a challenge as the aggregation server must wait for updated local gradients from chosen heterogeneous devices. These devices, however, might unexpectedly go offline due to their inherent unreliability. (2) Aggregation Efficiency Reduc-
tion. In each training round, faster devices are forced to wait for stale local models from slower devices (stragglers). This delay results from the dual factors of device heterogeneity (resource variation among devices) and data heterogeneity (uneven training data distribution across devices). (3) Low Resource Utilization. The current inefficiencies in node selection algorithms often cause multiple competent devices to be rarely chosen for participation. (4) Security and Privacy Vulnerabilities. The classic FL approach is susceptible to various security threats, such as poisoning and backdoor attacks. Moreover, concerns about privacy arise due to potential data leaks during the training process.

To address the challenges of device unreliability, aggregation efficiency reduction, and low resource utilization, asynchronous federated learning (AFL) emerges as a promising solution. In AFL, the central server promptly initiates global model aggregation upon the reception of a local model. As the devices unexpectedly going offline are ignorable to AFL, the concerns about device unreliability are mitigated. By removing the necessity to await slow devices for local model uploads before aggregation, AFL enhances aggregation efficiency. AFL also improves the utilization of computing resources across heterogeneous devices by allowing devices with varying operational efficiency to train their respective local models at their own pace.

While there have been survey papers on the subject of FL, none of them have undertaken an exhaustive investigation, classification, or summary of AFL. Consequently, the primary significance of this study lies in its comprehensive classification, summarization, and analysis of AFL. A comparative overview between this survey paper and other relevant surveys is provided in Table 1.

The contributions of this survey paper are summarized as follows. Firstly, this comprehensive survey reviews and analyzes 125 research papers spanning the years 2019 to 2022, including 7 relative survey papers. Secondly, the existing papers of AFL are categorized and summarized innovatively from the perspective of device heterogeneity, data heterogeneity, privacy and security, and application scenarios. Thirdly, the survey identifies a number of promising research topics that deserve more investigation and discussion.

The subsequent sections of this paper are organized as follows: In Section 2, the preliminary knowledge requisite for this survey is briefly introduced. Then, the AFL approaches addressing various challenges, including device heterogeneity, data heterogeneity, as well as privacy and security on heterogeneous devices, are then summarized and analyzed in Section 3, Section 4 and Section 5 respectively. Following this, Section 6 offers a comprehensive portrayal of the diverse applications of AFL with heterogeneous devices. Building upon the comprehensive analysis and discourse, Section 7 outlines potential avenues for promising research directions, followed by a conclusion in Section 8.

2. Background Knowledge

This section provides an explanation of the foundational knowledge essential for this survey, covering three key perspectives: federated learning, blockchain, and differential privacy.

2.1. Federated Learning

Distributed ML (DML) is a research topic that investigates different structures or topologies of the computer cluster for better training machine learning models [22]. Typically, DML can be categorized as centralized, decentralized, and fully distributed, along with various types of communication protocols. FL is a kind of DML where local models are trained on distributed nodes, and the global model is generated on an aggregation server by averaging the local models [23, 24]. Nevertheless, there is a difference between FL and DML. FL is designed for scenarios where data resides on multiple devices or servers and is not centralized. It is typical in environments where data privacy is crucial, such as on mobile devices. By contrast, DML often assumes cloud or data center environments. While data can be distributed across multiple servers or nodes, these nodes are typically co-located or part of a single infrastructure, and there is more flexibility in data sharing. Therefore, FL serves as an ML framework designed to dismantle the barrier of the data silo [25], primarily attributed to its privacy-preserving characteristic for local training data within each node. The conventional FL process encompasses the following key steps.

1. Initialization: Once the training task is defined according to the specific application scenario, the aggregation server prepares the initial global model $w_G^0$ alongside training parameters like learning rate, batch size, and iteration count. Following the selection process, the aggregation server broadcasts $w_G^0$ to a designated number of nodes (denoted as $K$).
2. Local Model Training: Assume $t$ stands for the current iteration number. Based on global model $w_G^t$, each node trains its respective local model $w_k^t$, where $k \in [1, K]$. These local models, $w_k^t$, are subsequently transmitted back to the aggregation server.

3. Global Model Aggregation: Assuming the training samples on node $k$ amount to $n_k$, and the total of training samples is represented as $n$, the server generates a new global model by taking a weighted average of the local models, as given in Eq. (1).

$$w_G^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t$$

Subsequently, $w_G^{t+1}$ is sent back to the nodes in preparation for the next training iteration.

Generally, datasets across nodes in FL are expected to be independent and identically distributed (IID). This entails an identical distribution of samples among nodes, with each training round independently selecting training samples. In practice, however, the training data samples collected by nodes usually deviate from this IID condition, named as non-independent and identically distributed (non-IID), posing challenges for both classic FL and AFL. Taking the smart hospital scenario as an example, IID data implies that disease cases across various hospitals exhibit similarity. Non-IID data, on the other hand, implies the diversity of disease cases among different hospitals, a portrayal more aligned with real-world scenarios. Under non-IID conditions, gradients learned and updated in one hospital provide limited utility in predicting patient conditions within another hospital.

In several application scenarios, the datasets kept at different parties include diverse feature sets but the same entities. For instance, the investment details and deposit records of a single user are usually held by a financial institution and a bank, respectively. Identifying the credit risk of an investor proves to be challenging for the financial institution due to the absence of crucial user-specific data such as deposit information. The datasets, featuring common entities but differing feature sets, fall under the category of vertically partitioned (VP) datasets. Vertical FL is the method employed to train models across such VP datasets, as shown in Fig. 1. Through vertical FL, financial institutions can leverage updated gradients from banks to assess investment risks without the need for access to the user’s sensitive information.

The proliferation of 5G networks and the integration of advanced hardware have led to enhanced computation and communication capabilities for IoT and edge devices. Researchers progressively deploy ML tasks on heterogeneous devices to enable intelligent human interactions while aiming to curtail communication costs. Nevertheless, the diversity in computational and communication capabilities among these heterogeneous devices remains inescapable. Moreover, the discrepant data sizes across nodes introduce notable discrepancies in the time required for training on each node. This often results in the generation of stale local models on slower nodes (stragglers), reducing the accuracy of the global model after aggregating.

AFL is proposed to alleviate the impact of stale nodes and improve the efficiency of FL. Within the AFL framework, global model aggregation takes place immediately once the aggregation server receives a new local model. A comparison of workflows in asynchronous and synchronous FL on heterogeneous devices is depicted in Fig. 2. The principal steps of AFL are outlined as follows:

1. Initialization: Similar to classic FL, the aggregation server broadcasts the initial global model $w_G^0$ to all $K$ nodes.

2. Local Model Training: Nodes undertake the training of their respective local models based on the most recent global model. Due to the heterogeneity in computing capabilities among devices, the completion of local model training ($\{w_1^t, w_2^t, \ldots, w_K^t\}$) does not occur concurrently. The local models are then sent back to the aggregation server separately.

3. Global Model Aggregation: The server aggregates the newly collected local model with the latest global model by using Eq. (2).

$$w_G^{t+1} = w_G^{t,k} - \frac{w_G^{t,k-1} + w_G^{t,k}}{2}$$

Following this process, the global model $w_G^{t+1}$ becomes available to the nodes and serves for their next iteration of local model training.

The iteration count in AFL, denoted as $t$, increases by 1 upon the completion of one iteration of local training by a device. The immediate model aggregation strategy in AFL reduces the waiting time for aggregation and thereby improves overall efficiency.

2.2. Blockchain

Blockchain, the backbone of Bitcoin, is a distributed ledger technology (DLT) that sustains the uniformity and immutability of transactional data across various nodes. In blockchain, nodes are responsible for maintaining the shared ledger and executing a globally unified program referred to as the smart contract. The self-verifiability and tamper-resistant attributes of the smart contract ensure the security and reliability of the shared ledger. Every node carries out the validation and execution of received transactions via the smart contract. Upon consensus attainment through the consensus algorithm, all nodes arrange transactional data into blocks and append these blocks to the shared ledger. Proof of Work (PoW) stands as the three most prevalent consensus algorithms. Generally, a consensus algorithm with better security or fault tolerance tends to have diminished efficiency.

Blockchain is typically treated as a distributed database for saving the models generated during the training process. Some researchers also utilize the reputation system of blockchain to motivate nodes to contribute their local models. In the context of AFL, blockchain yields multiple advantages. Firstly,
the immutability of shared ledgers prevents malicious node behavior, such as uploading plagiarized updated gradients [25]. Secondly, the consensus algorithm fosters trust between unfamiliar devices, given the decentralized and unmanipulable nature of the aggregation process [15]. Thirdly, the smart contract validates the authenticity of models and nodes, thus deterring malicious nodes from introducing poisoned gradients [24]. Fourthly, the decentralized aggregation strategy on the blockchain helps the aggregation server to resist DDoS attacks and reduces the risk of single-point failures [35]. However, the adoption of blockchain in AFL entails trade-offs in scalability and efficiency to a certain degree.

2.3. Differential Privacy

Differential privacy is a privacy-preserving technique that has experienced fast growth for over ten years. Originating from the concept of differential attack, differential privacy entails concealing a solitary sensitive data point within a particular dataset [36, 37]. The central objective of differential privacy is to render each data point non-discriminatory while upholding specific statistical attributes required for data analysis [38].

Various differential privacy mechanisms have been created, each serving a crucial role in its specific application context. Prominent among these are the Laplace mechanism [37], exponential mechanism [39], and Gaussian mechanism [40]. By infusing controllable randomized noise, differential privacy is able to return sanitized and privacy-preserving responses to data requesters [36]. However, this data sanitization process in differential privacy comes at the cost of diminishing data utility [7]. Thus, a parameter known as the privacy budget (ε) is introduced to measure the balance between privacy protection and data utility.

To cater to the requirements of flexible privacy protection in practical scenarios, personalized privacy protection models have been introduced. From this perspective, an index (e.g., social distance within social networks) is introduced to determine the level of personalized privacy protection [41]. By fine-tuning personalized parameters, the data utility could be improved further [42].

While being efficient and scalable, differential privacy encounters the challenge of diminished data utility, especially when the introduced noise is subject to randomization and lacks proper control [33]. Given that the training process in federated learning (FL) typically involves numerous local devices, sometimes numbering in the dozens or even thousands, these concerns can be mitigated by setting the mean value of the Laplace distribution to zero [44, 45]. Consequently, differential privacy has a great potential to be applied in FL or even AFL.

3. Device Heterogeneity

The primary obstacle of AFL revolves around optimizing resource utilization across heterogeneous devices to enhance training efficiency. Concurrently, there exists the obstacle of stale local models resulting from device heterogeneity, a factor that is detrimental to the performance of the global model.

3.1. Node Selection

Numerous algorithms for node selection are proposed to improve the training efficiency of AFL across heterogeneous devices. In contrast to classic FL selecting nodes with more training data, AFL leans towards prioritizing nodes with heightened resilience and computational capacity. However, it is challenging to strike a balance between robustness and overfitting of the global model.

For instance, in [46], the authors present a heuristic greedy node selection strategy that iteratively selects heterogeneous IoT nodes to participate in global learning aggregation based on their local computing and communication resources. Experiments are conducted on both IID and non-IID datasets to verify the effectiveness of their approach. Apart from that, considering the large number of edge devices involved, in [49], the authors limit the number of devices training simultaneously in the AFL network. A limit-size cache with a weighted averaging mechanism is introduced onto the server to reduce the impact of model staleness. Experiment results back up the improved convergence speed and model accuracy. These schemes are simple to improve aggregation efficiency but lack the sense of non-IID data across different nodes.
| Ref. | Detail | Data Dist. | Model & Dataset |
|------|--------|------------|-----------------|
| 46   | Heuristic greedy node selection according to local computation and communication resources. | H, I, N | CNN & MNIST \[25\], FMNIST \[35\] |
| 49   | Limit the number of devices training together. | H, I, N | CNN & FMNIST |
| 50   | A prioritized node-selection function based on computing power and accuracy change. | H, I, N | CNN & MNIST, FMNIST, CIFAR10 \[55\] |
| 51   | Assign a trust score to each node based on its activities. | H, I, N | CNN & MNIST |
| 52   | Select nodes with a lower crash probability. | H, I | CNN, SVM & Boston \[52\], MNIST, KDD Cup ’99 \[53\] |
| 53   | Random, significance-based, and frequency-based scheduling are analyzed. | H, I, N | MNIST |
| 58   | A mixing hyperparameter that balances the convergence rate with variance reduction according to the staleness. | H, N | CNN & CIFAR10, WikiText-2 \[50\] |
| 57   | Increase the weight of recently updated local models. | H, N | CNN, LSTM & MNIST, HAR \[53\] |
| 59   | The weight assigned to the updated gradients decreases as the staleness value increases. | H, I, N | MLP & MNIST |
| 60   | A caching mechanism with weighted averaging according to the staleness of the model. | H, I, N | CNN & FMNIST |
| 61   | A decay coefficient that is responsible for balancing the previous and the current model. | H, N | CNN, LSTM & FMNIST, FitRec \[61\], Air Qlt. \[62\], ExtraSensory \[63\] |
| 62   | A duel-weighted gradient update scheme. | H, I | MLP, CNN & MNIST, CIFAR10 |
| 65   | Dynamically adjust aggregation weight of branches based on accuracy. | H, I | CNN, MBNN & Bearing \[66\], Gear Fault \[67\] |
| 68   | Self-adaptive threshold computation and gradient communication compression. | H, I | MLP & MNIST |
| 69   | A double-end sparse compression based on Top-K AllReduce sparse compression. | H, I | LR, MLP & Insurance \[70\], Credit Card \[71\] |
| 72   | Three transmission scheduling algorithms for stragglers under different circumstances. | H, I | CNN & MNIST, CIFAR10 |
| 50   | Local models on unselected nodes will be cached for several iterations before uploading. | H, I, N | CNN & MNIST, FMNIST, CIFAR10 |
| 54   | Nodes are classified into three classes with tolerable nodes working asynchronously. | H, I | CNN, SVM & Boston, MNIST, KDD Cup ’99 |
| 58   | A private buffer on the aggregation server. | H, N | LSTM, CNN & Sent140 \[73\], CelebA \[74\] |
| 75   | The aggregation time interval depends on the slowest node. | H, I, N | CNN & CIFAR10, CIFAR100 |
| 59   | The local models received in a time window are cached. | H, I, N | MLP & MNIST |
| 76   | The server stores local models in a buffer of size K. | H, I | CNN & MNIST, CIFAR10 |
| 77   | Nodes are clustered into faster tiers and slower tiers. | H, I, N | CNN & CIFAR10, FMNIST, Sent140 |
| 78   | A metric for grouping nodes according to the gradient direction and the latency. | H, N | MCLR, LSTM & MNIST, FMNIST, Synthetic \[59\], Sent140 |
| 80   | A cascade training scheme including bottom subnetworks and top subnetworks. | V | MLP, CNN & MNIST, FMNIST, CIFAR10 |
| 81   | Nodes are grouped based on data distributions and physical locations. | H, I, N | SR, MLP, CNN & MNIST, FMNIST, EMNIST, CelebA |
| 82   | Adjust the aggregation frequency among groups. | H, I | MNIST |
| 52   | The parameters in shallow layers are updated more frequently than those in deep layers. | H, N | CNN, LSTM & MNIST, HAR |
| 53   | The parameters in shallow layers are updated more frequently than those in deep layers on UAVs. | H, I | CNN & Real-World Movie Ratings |
| 54   | Allow nodes to select a branch of the global model based on local data distribution. | H, I | CNN, MBNN & Bearing, Gear Fault |

1 Reference paper that belongs to the specific group.
2 Data distribution across nodes. H: horizontal, V: Vertical, I: IID, N: non-IID.
In order to select nodes more reasonably, in [50], a prioritized node-selecting function is designed according to the computing power and accuracy change of local models on each node. Other unselected nodes continue the iterations locally at the same time. As a result of the node-selecting function, the experiment results show a higher accuracy growth rate with a faster convergence speed. Nevertheless, this prioritized node-selection method does not consider the device unreliability of IoT devices. Thus, the authors of [51] propose an idea that assigns a trust score to each node based on its activities. ML tasks with resource requirements and a minimum trust score are published in the FL network. Any candidates who do not meet the task requirement are filtered out before the training round begins. Clients who complete tasks will be rewarded, while those who do not will have their trust value decreased. Similarly, in [21], a node with a lower probability to crash is more likely to be selected in an iteration. The straggling nodes that training models that are too stale will be tagged as deprecated and forced to synchronize with the server. The tolerable nodes are those training on the acceptable stale models, who work asynchronously with the server. After updated gradients from a fraction of nodes are received, the central server ends a round of training. As a result, the waste of computation resources is minimized, and communication expenses are kept at a relatively low level.

For a more comprehensive assessment of diverse device scheduling and update aggregation strategies, a study conducted by the authors in [54] involves experiments encompassing both IID and non-IID datasets. These experiments are conducted across a spectrum of computational resources and training data distributions, considering scenarios where a subset of IoT devices is permitted to upload local models. Specifically, the device scheduling policies include random scheduling, significance-based scheduling, and frequency-based scheduling; the update aggregation policies include equal weight aggregation and age-aware aggregation. The simulation results demonstrate that the random scheduling policy outperforms others while training on non-IID datasets. Besides, an appropriate age-aware aggregation policy performs better.

3.2. Weighted Aggregation

Numerous weighted aggregation algorithms have been introduced to lessen the influence of slow devices and increase learning efficiency. In conventional FL, weighted aggregation aims to amplify the influence of local models trained with more data. However, in AFL, the objective shifts toward alleviating the effects of stale local models, which does not exist in classic FL. One primary idea is introducing a parameter that accounts for staleness, which reduces the influence of stale local models and elevates the influence of the most recent local models during the aggregation procedure. There are several papers that adopt this method. For example, in [55], a mixing hyperparameter is introduced based on staleness to balance the convergence rate and variance reduction. The experiments conducted on CIFAR-10 and WikiText-2 validate both fast convergence and staleness tolerance. In [57], a temporally weighted aggregation
strategy is proposed, which increases the weight of recently updated local models when aggregating on shallow and deep layers. Experiment results on CNN and LSTM neural networks show that the global model accuracy and convergence are improved. Another time-based weighted aggregation algorithm is proposed in \cite{59}. The weight assigned to the updated gradients decreases as the staleness value increases. Similarly, in \cite{49}, a staleness-based weighted aggregation algorithm with cache is proposed. In \cite{60}, a decay coefficient is proposed with similar effects, balancing the previous and current models. With the dynamic learning step size, the nodes with more data or poor communication status are compensated. Experiments across three real-world datasets are conducted with results showing that their scheme converges faster and enables higher model accuracy.

Nevertheless, the aforementioned methods only focus on the staleness of local models, which is a one-sided view. Thus, a dual-weighted gradient updating strategy is proposed in \cite{64}, which takes into account the size of the dataset as well as the similarity between the local and global gradients. The updated gradients submitted by edge devices are aggregated after the dual-weight correction. The experiment results reveal that the model accuracy remains high even after gradient compression.

Apart from setting more factors of weighted aggregation, to enable more fine-grained weighted aggregation, an idea is to aggregate branches in a model with weights. In \cite{65}, the global model is split into branches with the aggregation procedure transformed into a branch-weighted merging process. The aggregation weight is dynamically adjusted depending on the training accuracy of all nodes to prevent the global model from overfitting to nodes that upload gradients frequently. To evaluate the effectiveness of the proposed scheme, a prototype is implemented on heterogeneous devices based on two industrial cases: (1) Fault diagnosis of motor bearings and (2) Fault diagnosis of the gearbox. The experiment results demonstrate that their scheme converges faster, achieves higher accuracy, and consumes less energy than the classic CNN model.

### 3.3. Gradient Compression

As gradient compression stands as a widely applicable tactic for improving the efficiency of FL, its incorporation into AFL commonly aims to achieve an additional reduction in communication expenses. Nevertheless, AFL introduces new challenges to gradient compression, primarily within the resource-restricted computing environments of edge and IoT devices, along with a higher frequency of aggregation operations. Specifically, the disparity in computational capabilities among nodes is much more significant. Besides, AFL incurs larger computational demand on the server side compared to classic FL due to the intensified frequency of aggregation and compression operations. To address these challenges, several efficient gradient compression algorithms tailored for AFL have been presented.

For instance, in \cite{68}, two sub-modules are presented for self-adaptive threshold gradient compression: (1) self-adaptive threshold computation and (2) gradient communication compression. The former is in charge of computing the threshold based on recent parameter changes, while the latter is in charge of compressing redundant gradient communications based on the threshold. The accuracies of the generated models after gradient compression are verified when training the MLP model on the MNIST dataset. Besides, the proposed scheme allows the node to join or quit freely, which is suitable to highly mobile edge computing scenarios. Another similar paper is published in \cite{64} by the same authors.

From the perspective of vertical FL, in \cite{69}, based on the Top-K AllReduce sparse compression technique, the authors present a double-end sparse compression algorithm \cite{64}. Specifically, the compression process happens on both the server and local sides to reduce the transmission cost. Experiment results demonstrate that 86.90% of information exchange is minimized during the training process, revealing that their scheme is suitable for edge computing scenarios with low-bandwidth or metered networks. Furthermore, the training data is protected securely against gradients leakage attacks \cite{85}.

Another approach to improve communication efficiency is to design a new communication protocol that more efficiently schedules model upload and download. For example, in \cite{72}, three transmission scheduling algorithms that account for slow nodes are proposed to improve the efficiency of AFL in wireless networks, where statistical information regarding uncertainty is known, unknown, or limited. The experiment results show their outperformance in terms of accuracy, convergence speed, and robustness.

### 3.4. Semi-Asynchronous FL

In AFL, the inclusion of stale local models from slow nodes during aggregation diminishes the accuracy of the global model to a certain extent. To alleviate the effects of these slow devices, semi-asynchronous FL schemes have been introduced. Typically, semi-asynchronous FL serves as a hybrid approach that combines elements from both classic FL and AFL, where the aggregation server captures and stores local models that arrive earlier, subsequently aggregating them following a certain timeframe. Depending on the magnitude of staleness, the subsequent arrivals of local models either take part in the following training rounds or are discarded. Notably, the aggregation frequency of semi-asynchronous FL falls between that of AFL and classic FL. Similar to classic FL, a training round is defined as the process spanning from one global aggregation to the next.

For example, in \cite{50}, a priority function is introduced to accurately select nodes with large amounts of data or high computation power. Meanwhile, local models on unselected nodes will be cached for a specific number of iterations before being submitted to the aggregation server. Besides, a restriction on the local training round number is set to prevent specific nodes from being unselected for a lengthy period of time, leading the global model to overfit certain nodes. The effectiveness of the scheme is evidenced by experiments conducted on IID and non-IID datasets. However, the restriction in this scheme will amplify the influence of stale local models after numerous aggregations.

On the contrary, a cache-based lag-tolerant mechanism on the aggregation server is introduced in \cite{21} to mitigate the
improves the scalability and efficiency of their scheme under various staleness distributions. The authors train an LSTM classifier on text and image classification tasks. The results reveal that their approach is faster than diverse distributions and converges faster than classic synchronous and asynchronous FL schemes. However, the aforementioned schemes do not use the cache to account for improved security. To account for the cache, a scheme that adopts a secure buffer on the server is proposed in [76], where a secure aggregation protocol is designed to prevent the server from learning any information about the local updates.

From the perspective of time, the authors in [75] aggregate local models at a specific time interval determined by the slowest node. More exact control over the training nodes is allowed, especially in edge computing networks with non-IID data distributions. The authors then compare classic synchronous, asynchronous, and semi-synchronous schemes across heterogeneous devices in experiments. The results show that their approach is faster and more accurate than other schemes.

For AFL, there is a good chance that a significant number of local model updates come in a short period of time. Considering this issue, in [59], after caching the first several local models received within a given time window, a synchronous aggregation strategy is adopted. The experiment results reveal that compared with the classic FL scheme, the time window enables its scheme many more nodes.

3.5. Cluster FL

Clustered FL is an approach geared towards augmenting training efficiency through the formation of clusters comprising devices exhibiting similar performance, functionalities, or datasets. The asynchronous update strategy has the potential to bring advantageous to inner-group updates, inter-group updates, or both.

For instance, an idea is grouping nodes into tiers based on their response latency [77]. Faster tiers are responsible for faster convergence, while slower tiers aid in the model accuracy improvement. Furthermore, a polyline-encoding-based compression algorithm is adopted in their scheme to improve communication efficiency. Experiments are conducted across multiple datasets and models, confirming that their scheme has a low communication cost and high prediction accuracy. However, grouping nodes by only considering the factor of response latency is circumscribed. By contrast, in [78], a grouping metric is proposed, where the gradient direction and the latency of model update are taken into account. The local update latency is composed of computation latency and communication latency. Experiments conducted on four imbalanced non-IID datasets assess the improvement in test accuracy.

From the aspect of grouping architecture, in [80], a cascade training scheme with bottom and top subnetworks is proposed to fully exploit all horizontally partitioned labels. Specifically, the bottom subnetworks are responsible for extracting embedding vectors from features, while the top subnetworks are for prediction. The nodes in FL are classified into three types, including active party, passive party, and collaborator. Each active party is connected to other passive parties so that it is able to gather embedding vectors and return gradients to them. The collaborator is connected to all active parties in order to aggregate the returned gradients. The experiment results reveal that their scheme effectively addresses the straggler problem with minimum performance loss.

Another advantage of cluster FL is improving communication efficiency. For example, in [81], nodes are grouped based on data distributions and physical locations to reduce global model loss and communication delay. The authors designed a control algorithm that reduces communication costs while examining the convergence of the proposed scheme in IID settings. The outperformed accuracy and efficiency of their scheme are evidenced by the experiment results. Additionally, Cluster FL allows different groups of nodes to aggregate at different frequencies, which also reduces communication costs. For example, in [82], the authors adaptively modify the aggregation frequency among groups to minimize the loss of FL. Under an environment with limited resources, a dynamic trade-off between computation and communication cost is formulated by Markov Decision Process (MDP) and optimized by deep reinforcement learning (DRL). Numerical results validate the accuracy, convergence, and energy-saving features of their proposed scheme.

3.6. Model Splitting

Following the splitting of the deep neural network model, each node is responsible for training a certain part rather than the whole model. Thus, the model-splitting strategy curtails the number of parameters necessitating transmission, consequently leading to an enhancement in communication efficiency. Upon integrating the model-splitting strategy into AFL, nodes bypass the need to await other nodes and fully utilize their computing resources to train the model for subsequent rounds. Therefore, the model-splitting strategy expedites the convergence of the global model to a certain extent.

For instance, in [57], a layerwise asynchronous model updating strategy is proposed, in which shallow layer parameters are updated more frequently than deep layer parameters. When aggregating, the most recently updated local models have the highest weight with the help of timestamps. The experiment results support the improved communication cost and model accuracy of the proposed scheme. In [83], a similar idea is achieved by using cache and communication capabilities on
UAVs and terrestrial base stations. The parameters in shallow layers are updated more frequently than those in deep layers. To predict the content caching placement, the proposed scheme employs a two-stage AFL algorithm. The efficiency of the proposed scheme is validated by experiments conducted on real-world datasets and numerical analysis.

Apart from splitting models into deep and shallow layers, an approach is to divide the global model into branches according to the sample category [65]. The splitting process involves acquiring a branch from the entire model. The aggregation process is performed on branches with different weights dynamically adjusted by the aggregation server. Besides, it allows nodes to select parts of the model according to local data distribution and update asynchronously reduces calculation and communication costs, enhancing FL efficiency.

In contrast to the node selection strategy, the model-splitting strategy reduces the computational demands placed upon nodes, affords greater flexibility in updating distinct layers of the global model, and alleviates biases present within the global model. Nonetheless, the extendability of this strategy to different models is constrained, primarily due to the requisite implementation of customized splitting and aggregation algorithms for every model across various datasets.

4. Data Heterogeneity

In practice, the data across nodes is usually non-IID. Besides, the amount of data distributed on each node is always imbalanced. Consequently, the frequent uploading of models on particular nodes has the potential to attract divergence to the global model and result in overfitting to specific datasets.

4.1. Non-Independent and Identically Distributed Data

The presence of non-IID data among nodes tends to cause a biased global model in AFL. To tackle the issues posed by this non-IID data, the research domain primarily encompasses four avenues of exploration, including constraint terms for aggregation, clustered FL, distributed validation strategy, and mathematically optimizing parameters.

A typical example of constraint terms for aggregation is [77], where a constraint term is presented to limit local updates to be closer to the global model. Besides, nodes with similar updating frequencies are grouped into the same tier through synchronous and asynchronous training strategies to prevent local models from diverging. The effectiveness of the scheme is supported by mathematical analysis and experiment results.

Clustered FL is also a strategy to alleviate the effects of divergent data distributions by grouping training nodes. In [85], the geometric properties of the FL loss surface are used to group nodes into clusters. The quality of clusters is ensured by math and validated by experiments. In [81], the data distribution on nodes in a group is optimized to be closer to the global data distribution. To better group nodes in general situations, the authors in [78] propose a spectral clustering approach, where nodes are grouped based on an affinity matrix derived from model update latency and direction. Non-IID dataset settings are also applied in their experiments, with the results showing that their scheme enhances test accuracy and convergence speed.

An example of distributed validation strategy is [87], where the authors propose a distributed validation scheme that evaluates model performance across nodes. A small percentage (5%) of local training data samples is reserved on each node to evaluate models from other nodes. As a result, a better generalized global model is obtained. By adopting both synchronous and asynchronous communication protocols, models trained on heterogeneous data and compute environments demonstrate the superior performance of the proposed scheme.

From the perspective of mathematical analysis, in [88], a training strategy with pre-determined initial weight parameters is proposed to mitigate the global model divergence. By using the Taylor Expansion formula, higher precision gradients are achieved on their AFL scheme, which is validated by experiments conducted across many real-world datasets. Another mathematical solution of non-IID data is choosing optimal hyper-parameters for the novel two-stage training strategy in AFL [89].

In addition to the aforementioned methods, a number of research papers analyze the effects of stale or imbalanced non-IID data on AFL. For example, in [90], an AFL scheme is proposed, where the authors focus on how staleness and data imbalance affect AFL by performing various levels of experiments. The results reveal that AFL works effectively on balanced data distribution when the server update frequency is unequal. Considering the effects of smooth strongly convex and smooth nonconvex functions when the data distribution is non-IID, the authors in [91] investigate the convergence theoretically and conduct several experiments. The results show that AFL has the same convergence rate as traditional FL while lowering communication requirements. By implementing the AFL scheme and conducting experiments on six Raspberry Pi 3B+ devices, the authors in [92] investigate the impact of heterogeneous devices. The results of experiments conducted on the MNIST dataset with non-IID data distribution reveal that AFL outperforms classic FL, especially when computing resources and input data sizes are disparate.

4.2. Vertical Distributed Data

In contrast to horizontal FL, vertical FL focuses on datasets where distinct subsets of features are spread across various nodes, as elaborated upon in Section 2.1. Given that the creation of the global model depends on the aggregation of local models, there exists a requirement for collaborative updating of these local models. Such skewed feature distribution and heightened model interdependence consequently present challenges to vertical AFL. To tackle these challenges, the state-of-the-art schemes mainly focus on improving communication and training efficiency.

One research direction is to improve the communication efficiency of vertical FL. In [93], apart from the flexible FL algorithm that allows random client participation, the authors utilize a local embedding model for each client to convert raw input to compact features, reducing the communication parameters in
The feasibility and effectiveness of the proposed scheme are confirmed by rigorous convergence analysis and numerical experiments on multiple datasets.

In addition to communication efficiency, the training efficiency also needs to be optimized in vertical FL. For example, the authors in [25] propose an asynchronous federated stochastic gradient descent (AFSGD-VP) algorithm with two variance reduction variants: stochastic variance reduced gradient (SVRG) and SAGA [24]. When the objective function is strongly convex, the convergence rate of AFSGD-VP is derived. Besides, the security and complexity of the proposed algorithm are provided. Experiment results on several vertical distributed datasets verify the theoretical analysis and prove the efficiency of their proposed scheme. Apart from the stochastic gradient descent, backward updating is also a key stage of training. A vertical AFL scheme with a backward updating mechanism and a bilevel asynchronous parallel architecture is proposed in [25]. Specifically, the backward updating mechanism enables all parties to update the model in a secure manner. The bilevel asynchronous parallel architecture improves the efficiency of the backward updating process. As the name implies, the parallel architecture is divided into two levels: the inner-party parallel between active parties and the intra-party parallel within each party. Both levels of the update are performed asynchronously to improve efficiency and scalability. The authors demonstrate the feasibility and security of the proposed strategy through theoretical and security analysis. Experiments on real financial datasets are conducted, whose results demonstrate efficiency, scalability, and losslessness.

Upon the aforementioned strategies, a hybrid approach that improves both training and communication efficiency is proposed. In [69], the authors propose a vertical AFL scheme with gradient prediction and double-end sparse compression algorithm. In particular, the gradient prediction presents the timely renewal of participants by using second-order Taylor expansion, reducing training time while retaining a sufficient degree of accuracy. The double-end sparse compression algorithm reduces the amount of data exchanged across the network during the training process. Experiment results obtained by training models on two public datasets reveal the outperformed efficiency of the scheme without degrading the accuracy and convergence speed.

5. Privacy and Security on Heterogeneous Devices

While FL is initially introduced to protect the privacy of local training data, new attack vectors have emerged, causing privacy concerns, such as membership inference attack [96], property inference attack [97], model inversion attack [98], and deep leakage from gradients attack [85]. Several attacks, like poisoning attacks or backdoor attacks, pose a significant threat to the integrity of the global model. Remedial measures for privacy and security concerns within FL include leveraging techniques like differential privacy and blockchain. Since AFL operates as a variant of FL, it is vulnerable to these attacks when training models across heterogeneous devices. However, the current solutions addressing privacy and security issues are often computationally demanding, rendering them less viable for deployment on resource-constrained heterogeneous devices and time-sensitive application scenarios. To confront these obstacles, many studies have come forward, presenting flexible differential-privacy models or highly efficient blockchain-based solutions tailored to AFL. A comparative analysis of these papers is presented in Table 3.

5.1. Privacy on Heterogeneous Devices

Differential privacy stands as a promising methodology embraced within various AFL schemes to protect the privacy of local models, consequently mitigating the risk of local training data leakage on heterogeneous devices.

For instance, in [93], a flexible FL scheme with differential privacy is proposed to avoid disclosing local training data. Each node employs Gaussian differential privacy to achieve a better trade-off between data privacy and data utility. However, a universal differential privacy setting lacks flexibility across heterogeneous devices. Consequently, in [99], the authors proposed an AFL scheme adopting local differential privacy for secure resource sharing in vehicular networks. Particularly, a tree-based gradient descent model is adopted on nodes to achieve high global model accuracy in a short amount of time. To protect the privacy of local models, a distributed local model updating approach with Gaussian noise is introduced to nodes in the regression tree. By offering rewards, nodes are encouraged to provide good models, thereby accelerating the convergence process. The experiment carried out on three real-world datasets demonstrates the high accuracy and efficiency of the scheme.

Apart from investigating personalized differential privacy, some researchers focus on the balance between model utility and privacy protection. For example, in [100], the convergence of AFL while adopting differential privacy is analyzed. Based on the analysis, a multi-stage adjustable algorithm is proposed to optimize the trade-off between model utility and privacy by dynamically changing the noise size and the learning rate. Experiments are conducted on edge servers and a cloud server with three different ML models, including logistic regression (LR), support vector machine (SVM), and convolutional neural network (CNN). The results reveal that MAPA achieves high model utilities and accuracy at the same time. Furthermore, in [101], differential privacy is introduced into AFL by adding Gaussian noise. The authors begin AFL training with a high learning rate and gradually reduce it to achieve optimum accuracy. The theoretical analysis and simulation results prove that their scheme reduces the network communication cost on heterogeneous devices.

5.2. Security on Heterogeneous Devices

When training ML models on heterogeneous devices, the blockchain is typically leveraged as a secure distributed database to ensure the secure storage or transmission of local models within AFL. The advantages of the integration include privacy protection and trust promotion among heterogeneous devices. For example, in [102], blockchain and digital twin
Table 3: Attack Resistance Comparison on Heterogeneous Devices

| Ref. | BKG 2 Attack | Collusion Attack | Inference Attack | Poisoning Attack | Byzantine Attack | DDoS Attack |
|------|--------------|-----------------|-----------------|-----------------|-----------------|-------------|
| [29] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [29] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [105] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ×               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |
| [106] | ✓            | ✓               | ✓               | ×               | ×               | ✓           |

✓ Fully resistant to the attack; ○ Partially resistant to the attack; × Not resistant to the attack.
1 Reference paper that belongs to the specific group.
2 Background Knowledge Attack.

edge network are integrated to store all local gradient updates in AFL. Specifically, the blockchain is adopted to track the aggregation progress by maintaining a global iteration index in AFL. A lightweight DPoS-based verification mechanism is developed, where stakes are earned based on the computing contribution to the global model. The mechanism is accomplished through the verification algorithm, which verifies the quality of the models against the historical model. In addition, a reinforcement learning-based algorithm is designed for efficient user scheduling and bandwidth allocation. A series of experiments are conducted to evaluate the performance of the scheme in terms of learning accuracy and resource cost. Another similar idea proposed by the same authors in [103] is to integrate blockchain, FL, and an asynchronous model update scheme in digital twin edge networks. The objective of lowering communication costs includes two parts: reducing transmission data size and optimizing communication resource allocation. Finally, the communication resource allocation approach is implemented by using deep neural networks. Numerical results reveal that this scheme improves communication efficiency and reduces the cost of resources. Apart from improving communication efficiency, some researchers focus on improving learning efficiency when integrating AFL with blockchain. The authors of [104] present an AFL scheme coupled with the Directed Acyclic Graph (DAG) blockchain for the Internet of Vehicles (IoV). The participating nodes are selected by Deep Reinforcement Learning (DRL) to improve the learning efficiency. Besides, a two-stage verification mechanism is developed, which comprises periodic validation of blockchain transactions and validation of local model quality. Experimental results show the excellent learning accuracy and rapid convergence speed of the scheme.

Another advantage of the integration is mitigating the risk of single-point failure caused by the centralized aggregation server. For instance, a blockchain-based AFL scheme with a staleness coefficient is proposed in [29]. Specifically, the staleness coefficient reduces the contribution from the latency device to the global model by comparing the version of the global model with the stale local model. The Proof-of-Work (PoW) consensus algorithm is adopted, where the miners are responsible for generating candidate blocks that include trained models. The block generation rate is positively correlated with the forking frequency. The experiments carried out on a variety of IoT devices demonstrate high accuracy on both horizontal and vertical FL frameworks. Another similar idea with a staleness coefficient is proposed in [107, 108]. Instead of using PoW, a committee-based consensus algorithm is adopted to improve efficiency further. The convergence speed and model accuracy are both validated by experiments on heterogeneous devices. Instead of staleness coefficient, some researchers pay attention to the architecture of the integration. For instance, to mitigate the risk of single-point failure and malicious nodes attacks, the authors in [106] propose a two-layer blockchain-driven FL framework composed of multiple Raft-based shard networks (layer-1) and a DAG-based main chain (layer-2). Layer-1 is a small group for information exchange, while layer-2 is responsible for storing and sharing models trained by layer-1 asynchronously. Furthermore, to avoid the impact of stale models, a virtual pruning procedure with a specific waiting time is presented. Models not approved by other models for a long time or with low accuracy will be pruned from the DAG blockchain. The experiment results show that this scheme is resilient against malicious nodes while maintaining acceptable convergence rates.

Additionally, the reputation of nodes is an important factor to be considered to improve the stability and security of AFL, which is easily handled by the blockchain by its built-in reputation and reward systems. For example, in [105], a blockchain-based AFL scheme is proposed, where an entropy weight method determines the participant rank by the proportion of local models trained on nodes. The metrics are all maintained in the blockchain, including the training time, training
sample size, local update correlation, and global update cheating times. The resource cost and training efficiency are well balanced by optimizing local training delays and the block generation rate. The experiment results show the superiority of the scheme in terms of efficiency and preventing poisoning attacks.

Moreover, some researchers investigate the effect of the gradient compression algorithm on the integration of blockchain and AFL, aiming at improving security and efficiency simultaneously. For example, in [109], the authors propose a SignSGD-based asynchronous federated learning paradigm (BASS) that only uploads the signs of the gradients of local models to the aggregation server, mitigating the risks of poisoning attacks. Theoretically, their paradigm is able to resist both privacy and security attacks at the cost of model convergence speed or accuracy. However, they fail to validate the performance of their paradigm in privacy preserving.

6. Applications on Heterogeneous Devices

AFL is adopted in various application scenarios, offering an efficient and adaptable training procedure on heterogeneous devices while upholding the privacy of local training data. The application scenarios of AFL and correlated research endeavors are summarized and compared in Table 4.

Smart transportation is a viable situation for AFL due to its efficient utilization of computing resources that bridges the gap between training delay and time-sensitive requirements to a certain extent. For instance, in [104], AFL is introduced to enhance the reliability and efficiency of data sharing among vehicles. The experiments conducted in a vehicular network evaluate their scheme, including one MBS and 10 RSUs covered. The results reveal that the DAG blockchain architecture in the scheme ensures both performance and security. Similarly, in [99, 102], AFL is adopted in urban vehicular networks to allocate resources more efficiently and securely. The experiment results verify the effectiveness of their scheme in terms of distributed data sharing and resource caching in urban vehicular networks. In [110], a real-time end-to-end AFL scheme is applied in IoV and focuses on steering wheel angle prediction for autonomous driving. To conduct angle prediction, the authors utilize a two-stream deep CNN model with two separate neural branches that consume spatial and temporal information, respectively. To consume real-time streaming data, a sliding training window is introduced to reduce computation and communication latency. The experiments are carried out on real-world datasets, with the results showing that their scheme improves model prediction accuracy while reducing computation and communication latency. AFL is also adopted in cameras to monitor, predict, and adjust traffic by controlling signal lights in the smart transportation scenario [68, 64]. By adjusting the hyper-parameter in the scheme, an optimal balance between the model accuracy and convergence speed is achieved. In [111], AFL is adopted in unmanned aerial vehicle (UAV) networks. In order to improve the convergence speed and model accuracy, an actor-critic-based AFL scheme is proposed, including equipment selection, drone placement, resource management, local training, and global aggregation. Specifically, to prevent low-quality devices from compromising learning efficiency and model accuracy, a device selection strategy is proposed, in which nodes with high processing capability, communication capabilities, and model accuracy are selected. The selection problem is modeled as a Markov Decision Process and optimized through reinforcement learning. The scheme is evaluated by experiments, whose results show a higher learning accuracy and lower time cost. Similarly, in [83], an intelligent content caching system in UAV networks based on AFL is proposed to extend the service coverage and reduce the communication delay of the 6G network. In the scheme, UAVs collaborate to forecast where content caching should be placed by taking real-time traffic distribution into account. In [51], AFL is applied to mobile robots that collect real-time data and perform training in a distributed and resource-constrained environment to reduce communication costs. Experiments conducted on 12 mobile robots with limited resources demonstrate that the performance of the model is guaranteed by selecting competent and reliable mobile robots.

Fault diagnosis is another application scenario for AFL. For instance, in [112], AFL is utilized to identify the local modes in real-time. To completely track actual system changes in real-time and increase the diagnostic rate of the nodes, each node turns private data into local models using an Extended Kalman Filter before transmitting. A sequential filter approach based on Sequential Kalman Filter is adopted to perform the asynchronous aggregation for uploaded local models. Experiments conducted on real-world collected fault datasets demonstrate high accuracy compared with benchmarks. Similarly, in order to improve model accuracy and convergence speed in anomaly detection while preserving privacy, the authors in [88] train the denoising autoencoder model based on labeled benign samples in AFL. Asynchronous update strategy improves the accuracy and stability of the model by reducing the impacts from the stragglers. In [113], the authors adopt AFL in the sensitive code review field to address privacy concerns. On leaks gathered from the code-sharing network Github, a prototype is developed and tested. When compared with local and centralized training, the proposed scheme improves model accuracy while preserving the privacy of the local training data. Another AFL-based fault diagnosis scheme proposed in [65] allows nodes to adaptively select branches of the model for further training according to their local datasets. Their scheme creates an effective diagnostic model for detecting potential defects while reducing resource requirements and communication overhead. Experiments conducted across heterogeneous devices verify the feasibility of their scheme.

The AFL paradigm is also applied to IIoT environments for real-time analysis and decision-making. For example, in [82], the authors break down the barrier of data island in IIoT with the help of AFL. In their scheme, the effect of slow devices is mitigated by adaptively adjusting aggregation frequency. The experiment results validate the feasibility and efficiency of their scheme. Similarly, in [103], AFL is utilized to preserve data privacy and improve the quality of services in IIoT. Besides, by adopting digital-twin technology, real-time interactions requirements in Industry 4.0 are fulfilled. The communication
### 7. Research Challenges and Future Directions

Emerging as a trending research topic, recent studies have brought to light a collection of challenges within AFL. These challenges encompass device heterogeneity, data heterogeneity, privacy and security on heterogeneous devices, as well as applications on heterogeneous devices. To deal with these challenges, in this section, several potential research directions are identified and summarized.

#### 7.1. Device Heterogeneity

**Optimization towards balanced time cost and performance improvement:** As summarized in section 3 for AFL, the existing performance improvement strategies on heterogeneous devices, such as node selection, weighted aggregation, and cluster FL, are effective in various ways. Some of the works even adopt several strategies at the same time to improve the efficiency of AFL. However, utilizing too many strategies in AFL results in a decline in efficiency to a certain extent. For instance, if selecting a range of nodes and then compressing the gradients on resource-limited devices takes longer than uploading local gradients, it is preferable to skip one of them. So far, there has been no comprehensive analysis of the balance between multiple performance improvement strategies and time consumption, which is a potential research direction.

To derive the optimized trade-off, it is possible to establish a dynamic gaming model by Markov Decision Process, which can adapt to various scenarios based on the constraints. Moreover, other lightweight convex optimization methods can be considered, such as quadratic minimization with convex quadratic constraints, semidefinite programming, and convex quadratic minimization with linear constraints.

**Optimization towards generalized AFL solution:** Usually, different performance improvement strategies have different application scenarios. For example, when the disparity in

---

| Ref. | Use Case | FL Client & Server | Key Contributions |
|------|----------|--------------------|-------------------|
| [104] | IoV | Vehicle & MBS | Improve reliability and efficiency of data sharing and traffic prediction among vehicles. |
| [92] | IoV | Vehicle & MBS, RSU | Efficiently and securely allocate resources for vehicles. |
| [112] | IoV | IoT & Base Station | Provide high-quality services with optimized network and resources allocation. |
| [110] | IoV | Vehicle & Server | Predict real-time steering wheel angle for autonomous driving. |
| [68] | IoV | Camera & Server | Monitor, predict, and adjust traffic by controlling signal lights. |
| [64] | IoV | Camera & Server | Monitor, predict, and adjust traffic by controlling signal lights. |
| [114] | UAV | Mobile & UAV | Provide efficient communication and computation services for ground mobile devices in outdoor events. |
| [53] | UAV | Mobile & UAV | Provide a content caching system in UAV networks to extend the service coverage and reduce the communication delay of the 6G network. |
| [51] | Mobile Robot | Robot & Server | Improve the performance of the ML models on mobile robots with low communication costs. |
| [112] | Fault Diagnosis | Edge & Server | Track actual system changes in real-time and improve the diagnostic rate of the devices. |
| [83] | Fault Diagnosis | Edge & Server | Train the denoising autoencoder model for anomaly detection. |
| [113] | Code Security | Edge & Server | Effectively and securely review and identify sensitive information in code before publishing. |
| [65] | Fault Diagnosis | Edge & Server | Efficiently identify possible faults in edge nodes while reducing the resource requirements and communication overhead. |
| [52] | IIoT | IoT & Edge | Address the data island problem in IIoT for dynamic perception and intelligent decision. |
| [103] | IIoT | IoT & Edge | Improve the quality of services and implement real-time interactions in IIoT while reducing the communication cost. |
| [114] | Concept Drift | Edge & Server | Speed up model convergence for detecting and dealing with the concept drift on edge. |
| [115] | Geo-location | Edge & Server | Predict the position and orientation of the camera for end-to-end localization. |

---

1 Reference paper that belongs to the specific group.
computing capabilities between heterogeneous devices is extremely high, semi-asynchronous FL with suitable weighted aggregation strategies could be an optimal solution. If the dataset distribution is IID across nodes, the local models from fast nodes should be selected and compressed, while those from slow devices should be discarded. The local models deserve higher weight if they bring a positive effect to the global model. Therefore, designing a generalized and flexible optimization framework for AFL for diverse application scenarios is a viable research field. It is expected to achieve this by integrating existing and future techniques minimally.

**Optimization towards dynamic resource allocation:** Intuitively, AFL requires more communication resources when compared with classic FL due to more global model aggregation operations. Therefore, it is expected to consider dynamic resource allocation algorithms, including transmit power, computation frequency for model training, and model selection strategy, to maximize the long-term time average (LTA) training data size with an LTA energy consumption constraint. Specifically, a possible solution is to first define the Lyapunov drift by converting the LTA energy consumption to a queue stability constraint. Then, a Lyapunov drift-plus-penalty ratio function can be constructed to decouple the original stochastic problem into multiple deterministic optimizations along the timeline. The construction is capable of dealing with uneven durations of communication rounds. To make the one-shot deterministic optimization problem of combinatorial fractional form tractable, the fractional problem is reformulated into a subtractive-form one by the Dinkelbach method, which leads to the asymptotically optimal solution in an iterative way. By doing so, there is a potential for both higher learning accuracy and faster convergence with limited time and energy consumption.

### 7.2. Data Heterogeneity

**Optimization towards heterogeneous data distribution:** Since the data distribution across nodes is usually non-IID in the real world [19], it is meaningful to obtain a generalized model while maintaining the accuracy for each local data in AFL. There are several solutions for non-IID data challenges in classic FL, such as localized independent training [116], personalized local model training [25], and cluster training [117]. However, it is hard to transplant these solutions into AFL, since the global model prefers to convergence to nodes with higher model upload frequency (i.e. fast nodes) in AFL and result in a biased global model. Such a biased global model decreases the effects of localized independent training and personalized local model training to a certain extent. Although cluster training has been utilized in several AFL schemes, it is hard to arrange a general cluster strategy for all application scenarios. For instance, a location-based cluster strategy is not ideal for traffic prediction among smart vehicles with non-IID datasets due to the randomness of vehicle movement. Cluster training based on data distribution similarity is a potential research topic [81], but it requires the development of an appropriate similarity evaluation algorithm. Besides, based on transfer learning or meta-learning, asynchronous personalized local model training is potentially an effective and accurate solution.

**Optimization towards heterogeneous data size:** Dataset sizes among nodes are usually unequal since each node gathers its own local data independently in most AFL application scenarios. Even all nodes have identical computing resources, the imbalanced datasets across nodes lead to varying update frequencies of local models. The weighted aggregation strategy based on local dataset size is a possible solution as the work in [64], but how to verify the validity of the dataset size on each node is another security problem. The smart contract in the blockchain offers self-verifying and self-executing capabilities [31], alleviating data fraud in AFL to some extent at a cost of low efficiency.

**Optimization towards vertical data distribution:** Vertical data distribution is prevalent in economic scenarios, where each node possesses different feature sets of the dataset. In heterogeneous computing environment, the lack of local models uploaded from some slow nodes causes the global model to be biased and unable to predict certain features, unlike the accuracy decline in horizontal FL. Therefore, the lagging local models are non-ignorable in vertical AFL. So far, only relatively few researches have been conducted in the vertical AFL area [80, 93, 26, 95, 69] compared with horizontal AFL. Moreover, none of these works analyzed the effects of extreme stragglers caused by computing resources or communication resources. A possible research direction is semi-asynchronous FL. With a server-side cache, it is possible to store stale local models and increase their effectiveness while keeping other local models up to date. Besides, another potential research direction is model splitting, which splits the global model according to the feature distribution on nodes and transforms to clustered horizontal AFL, as the work in [80]. However, without knowing the local dataset on each node, it is hard to identify the distribution of features among nodes by the local models.

### 7.3. Privacy and Security on Heterogeneous Devices

**Optimization towards privacy protection using differential privacy:** Differential privacy prevents AFL from a variety of privacy attacks, including background knowledge, collusion, and inference attacks. However, as the utility of local models falls, differential privacy leads global model accuracy to decline. Therefore, the trade-off between privacy and utility is hard to achieve in AFL. There are several strategies for optimizing the trade-off in classic differential privacy, such as Static Bayesian Games [13], Markov Decision Processes [7], and Generative Adversarial Nets [119]. However, in AFL, the publishing process for local models is dynamic and distributed, which is hard to balance through a trusted third party. Local differential privacy (LDP) is an approach that users randomly perturb their inputs without the necessity for a trusted third party and is treated as a solution for the privacy issues in FL [44]. Nevertheless, the trade-off between the privacy of local models and the utility of the global model is hard to achieve. Especially in AFL, it necessitates asynchronous macro control for the LDP in a distributed manner. The smart contract in the blockchain is a viable approach for manipulating LDP in a distributed manner. However, an asynchronous consensus algorithm needs to be de-
signed to balance the privacy of local models and the utility of the global model.

**Optimization towards security enhancement using blockchain:** Blockchain can be utilized to address security challenges in AFL, such as the single point of failure, Byzantine attacks, and poisoning attacks. At the same time, blockchain declines the efficiency of AFL to some extent due to its low communication efficiency and high computing resource consumption [25]. Therefore, the trade-off between security and efficiency of blockchain-based AFL is also challenging. To improve the scalability of blockchain, several improved consensus algorithms are designed to replace PoW, such as Proof-of-Stamp (PoS) [31], Proof-of-Repuation (PoR) [120], PBFT, and RAFT [121]. However, generally, the higher the performance of the consensus algorithm reaches, the worse the security level is. For instance, compared with PBFT, RAFT is not resistant to Byzantine attacks but has higher data throughput. A promising solution is to develop an efficient and secure consensus algorithm. For example, Algorand [122] is a byzantine-tolerant consensus algorithm with excellent scalability while maintaining a high level of security. A group of committees is randomly selected in each iteration to verify and ensure the security of the transactions. During training, the committees verify the local models in each iteration. However, it is difficult to select committees asynchronously without compromising security. In this situation, it is possible to separate the consensus process and the training process with a tailor-made blockchain structure that records the training models periodically.

**Optimization towards security enhancement using lightweight distributed cryptography:** Another research direction is to apply lightweight distributed cryptography to AFL to protect security. Traditional cryptography approaches, such as public-key encryption [123], homomorphic encryption [124], and attribute-based encryption [125], have several limitations in this case. Public-key encryption and homomorphic encryption are resource-consuming and unsuitable for resource-limited devices in AFL. Attribute-based encryption is not flexible enough, due to its necessity for a trusted third-party authority. A possible research area is to design a flexible and efficient attribute-based encryption algorithm for AFL with dynamical attribute adjustment that allows participants to join or leave freely.

### 7.4. Applications on Heterogeneous Devices

**Expansion of real-world applications:** As summarized in section 5, there are few real-world application scenarios for AFL for the time being, including IoV, fault diagnosis, IIoT, and so on. Compared with synchronous FL, AFL is more efficient and is better suited to time-sensitive scenarios with limited computing resources. Therefore, a possible applied research area is to apply dedicated AFL systems to a wide range of real-world scenarios. For example, in a smart hospital, ML models trained by AFL based on electronic healthcare records (EHR) predict the situation of patients. In a smart grid, AFL trains ML models on heterogeneous devices to anticipate the energy consumption in various areas and accomplish smart power dispatch. In a smart farm, the growth situation of plants is well monitored, diagnosed, and predicted by IoT with the help of ML models trained by AFL.

**Development of real-world evaluation testbeds:** As summarized in Table 2, most of the experiments of AFL are conducted in simulation mode, without demonstrating the feasibility of AFL in the real world. More experiments are expected to be conducted on IoT or edge devices to evaluate the efficiency, security, and privacy of AFL schemes. Thus, it would be a promising research direction to develop scalable and flexible testbeds deployed on heterogeneous devices and accessible from a standardized interface. The development of testbeds includes the issues of architecture design, inclusiveness of heterogeneous devices, structure-wise efficiency and performance fine-tuning, etc.

### 8. Conclusion

AFL has been attracting increasing attention due to its multiple advantageous features. To mitigate the drawbacks of existing works, three fundamental challenges in AFL are primarily studied, including device heterogeneity, data heterogeneity, as well as security and privacy issues on heterogeneous devices. By conducting an in-depth exploration of state-of-the-art research, corresponding application scenarios of AFL that potentially increase its impact and popularization on heterogeneous devices are summarized. It is pleasing to observe that the number of novel AFL schemes grows by the month. But even so, it is believed this survey is sufficiently comprehensive that new schemes can be appended and categorized correspondingly. This survey provides legible insights into the picture of AFL from a brand-new perspective, which is helpful to the community by providing potentially promising directions, and simplifying future designs, including but not limited to motivating coherent compositions uncovered by the proposed categorization and analysis.

### References

[1] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature 521 (7553) (2015) 452–459.
[2] H. Xu, X. Liu, W. Yu, D. Griffith, N. Golmie, Reinforcement learning-based control and networking co-design for industrial internet of things, IEEE Journal on Selected Areas in Communications 38 (5) (2020) 885–898.
[3] W. Nie, T. Karras, A. Garg, S. Debnath, A. Patney, A. Patel, A. Anandkumar, Semi-supervised stylegan for disentanglement learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 7360–7369.
[4] P. Voigt, A. Von dem Bussche, The eu general data protection regulation (gdpr), A Practical Guide, 1st Ed., Cham: Springer International Publishing 10 (2017) 3152676.
[5] J. Andrew, M. Baker, The general data protection regulation in the age of surveillance capitalism, Journal of Business Ethics 168 (3) (2021) 565–578.
[6] S. Wachter, B. Mittelstadt, A right to reasonable inferences: re-thinking data protection law in the age of big data and ai, Colum. Bus. L. Rev. (2019) 494.
J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, arXiv preprint arXiv:1610.05492 (2016).

J. Konečný, H. B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, arXiv preprint arXiv:1610.02527 (2016).

W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, C. Xiao, Federated learning in mobile edge networks: A comprehensive survey, IEEE Communications Surveys & Tutorials 22 (3) (2020) 2031–2063.

X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning: A taxonomy, review and future directions, ACM Computing Surveys (CSUR) 54 (6) (2021) 1–36.

O. A. Wahab, A. Mourad, H. Otrok, T. Taleb, Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems, IEEE Communications Surveys & Tutorials 23 (2) (2021) 1342–1397.

A. Imteaj, U. Thakker, S. Wang, J. Li, M. H. Amini, A survey on federated learning for resource-constrained iot devices, IEEE Internet of Things Journal (2021).

L. U. Khan, W. Saad, Z. Han, E. Hossain, C. S. Hong, Federated learning for internet of things: Recent advances, taxonomy, and open challenges, IEEE Communications Surveys & Tutorials (2021).

L. Luy, H. Yu, Q. Yang, Threats to federated learning: A survey, arXiv preprint arXiv:2003.02133 (2020).

M. Abdel-Baset, N. Moustafa, H. Hawash, W. Ding, Federated learning for privacy-preserving internet of things, in: Deep Learning Techniques for Exponentially Scaling Artificial Intelligence, Springer, 2021, pp. 215–228.

W. Wu, L. He, W. Lin, R. Mao, C. Maple, S. Jarvis, Safa: A semi-asynchronous protocol for fast federated learning with low overhead, IEEE Transactions on Computers 70 (5) (2020) 655–668.

J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J. S. Rellermeyer, A survey on distributed machine learning, ACM computing surveys (csur) 53 (2) (2020) 1–33.

B. Mcmahon, E. Moore, D. Ramage, S. Lampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, D. Dou, From distributed machine learning to federated learning: A survey, Knowledge and Information Systems 64 (4) (2022) 885–917.

C. Xu, J. Ge, Y. Li, Y. Deng, L. Gao, M. Zhang, Y. Xiang, X. Zheng, Sce: A smart-contract driven edge intelligence framework for iot systems, IEEE Transactions on Mobile Computing (2023).

B. Gu, A. Xu, Z. Hao, C. Deng, H. Huang, Privacy-preserving asynchronous vertical federated learning algorithms for multi-party collaborative learning, IEEE Transactions on Neural Networks and Learning Systems (2021).

S. Trindade, L. F. Bettencourt, N. L. da Fonseca, Management of resource at the network edge for federated learning, arXiv preprint arXiv:2107.07270 (2021).

J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, D. Huba, Federated learning with buffered asynchronous aggregation, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 3581–3607.

Y. Liu, Y. Qu, C. Xu, Z. Hao, B. Gu, Blockchain-enabled asynchronous federated learning in edge computing, Sensors 21 (10) (2021) 3335.

S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review (2008) 21260.

C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, L. Gao, A lightweight and attack-proof bidirectional blockchain paradigm for internet of things, IEEE Internet of Things Journal 9 (6) (2021) 4371–4384.

A. Kiyas, A. Russell, B. David, R. Oliynyk, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: Annual International Cryptology Conference, Springer, 2017, pp. 357–388.

H. Sukhwani, J. M. Martinez, X. Chang, K. S. Trivedi, A. Rindos, Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric), in: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), IEEE, 2017, pp. 253–255.

Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, Y. Xiang, A Blockchain-enabled federated learning framework for cognitive computing in industry 4.0 networks, IEEE Transactions on Industrial Informatics 17 (4) (2020) 2964–2973.

Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, G. Zheng, Decentralized privacy using blockchain-enabled federated learning in fog computing, IEEE Internet of Things Journal 7 (6) (2020) 5171–5183.

C. Dwork, Differential privacy: A survey of results, in: International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.

C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3-4) (2014) 211–407.

X. Cao, J. Jia, N. Z. Gong, Data poisoning attacks to local differential privacy protocols, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 947–964.

J. Dong, D. Durfee, R. Rogers, Optimal differential privacy composition for exponential mechanisms, in: International Conference on Machine Learning, PMLR, 2020, pp. 2597–2606.

Y. Zhu, X. Yu, M. Chandraker, Y.-X. Wang, Private-knn: Practical differential privacy for computer vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11854–11862.

Y. Qu, M. R. Nosouhi, L. Cui, S. Yu, Personalized privacy protection in big data (2021).

Y. Qu, S. Yu, W. Zhou, S. Chen, J. Wu, Customizable reliable privacy-preserving data sharing in cyber-physical social networks, IEEE Transactions on Network Science and Engineering 8 (1) (2020) 269–281.

J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, D. Megías, Individual differential privacy: A utility-preserving formulation of differential privacy guarantees, IEEE Transactions on Information Forensics and Security 12 (6) (2017) 1418–1429.

K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, H. V. Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Transactions on Information Forensics and Security 15 (2020) 3454–3469.

Y. Liu, J. Peng, J. Kang, A. M. Iyyasu, D. Niyato, A. A. Abd El-Latif, A secure federated learning framework for 5g networks, IEEE Wireless Communications 27 (4) (2020) 24–31.

Z. Chen, W. Liao, K. Hua, C. Lu, W. Yu, Towards asynchronous federated learning for heterogeneous edge-powered internet of things, Digital Communications and Networks (2021).

H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

G. Cohen, S. Afshar, J. Tapos, A. Van Schaik, Emnist: Extending mnist to handwritten letters, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2921–2926.

C. Zhou, H. Tian, H. Zhang, J. Zhang, M. Dong, J. Jia, Tea-fed: time-efficient asynchronous federated learning for edge computing, in: Proceedings of the 18th ACM International Conference on Computing Frontiers, 2021, pp. 30–37.

J. Hao, Y. Zhao, J. Zhang, Time efficient federated learning with semi-asynchronous communication, in: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2020, pp. 156–163.

A. Imteaj, M. H. Amini, Fedar: Activity and resource-aware federated
sium on Security and Privacy (SP), IEEE, 2019, pp. 691–706.

[98] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1322–1333.

[99] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Differentially private asynchronous federated learning for mobile edge computing in urban informatics, IEEE Transactions on Industrial Informatics 16 (3) (2019) 2134–2143.

[100] Y. Li, S. Yang, X. Ren, C. Zhao, Asynchronous federated learning with differential privacy for edge intelligence, arXiv preprint arXiv:1912.07902 (2019).

[101] M. van Dijk, N. V. Nguyen, T. N. Nguyen, L. M. Nguyen, Q. Tran-Dinh, P. H. Nguyen, Asynchronous federated learning with reduced number of rounds and with differential privacy from less aggregated gaussian noise, arXiv preprint arXiv:2007.09208 (2020).

[102] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Communication-efficient federated learning and permissioned blockchain for digital twin edge networks, IEEE Internet of Things Journal 8 (4) (2020) 2276–2288.

[103] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Communication-efficient federated learning for digital twin edge networks in industrial iot, IEEE Transactions on Industrial Informatics 17 (8) (2020) 5709–5718.

[104] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles, IEEE Transactions on Vehicular Technology 69 (4) (2020) 4298–4311.

[105] L. Feng, Y. Zhao, S. Guo, X. Qiu, W. Li, P. Yu, Blockchain-based asynchronous federated learning for internet of things, IEEE Transactions on Computers (2021).

[106] S. Yuan, B. Cao, M. Peng, Y. Sun, ChainSFL: Blockchain-driven federated learning from design to realization, in: 2021 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2021, pp. 1–6.

[107] C. Xu, Y. Qu, P. W. Eklund, Y. Xiang, L. Gao, Buff: An efficient blockchain-based asynchronous federated learning framework, in: 2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021, pp. 1–6.

[108] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, L. Gao, An efficient and reliable asynchronous federated learning scheme for smart public transportation, IEEE Transactions on Vehicular Technology (2022).

[109] C. Xu, Y. Qu, Y. Xiang, L. Gao, D. Smith, S. Yu, Bass: Blockchain-based asynchronous signsgd for robust collaborative data mining, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2022, pp. 1–7.

[110] H. Zhang, J. Bosch, H. H. Olsson, Real-time end-to-end federated learning: An automotive case study, arXiv preprint arXiv:2103.11879 (2021).

[111] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, L. Xiao, Privacy-preserving federated learning for uav-enabled networks: Learning-based joint scheduling and resource management, IEEE Journal on Selected Areas in Communications (2021).

[112] X. Ma, C. Wen, T. Wen, An asynchronous and real-time update paradigm of federated learning diagnosis for fault, IEEE Transactions on Industrial Informatics (2021).

[113] S. Kall, S. Trabelsi, An asynchronous federated learning approach for a security source code scanner, in: ICISSP, 2021, pp. 572–579.

[114] Y. Chen, Z. Chai, Y. Cheng, H. Rangwala, Asynchronous federated learning for sensor data with concept drift, arXiv preprint arXiv:2109.00151 (2021).

[115] M. R. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, M. Kopp, Asynchronous federated learning for geospatial applications, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 21–28.

[116] M. Khodak, M.-F. F. Balcan, A. S. Talwalkar, Adaptive gradient-based meta-learning methods, Advances in Neural Information Processing Systems 32 (2019).

[117] Y. Smith, C.-C. Chi, M. Sanjabi, A. S. Talwalkar, Federated multitask learning, Advances in neural information processing systems 30 (2017).

[118] Y. Qu, L. Cui, S. Yu, W. Zhou, J. Wu, Improving data utility through game theory in personalized differential privacy, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.