electronics

MDPI

*Article*

# Computation and Communication Efficient Adaptive Federated Optimization of Federated Learning for Internet of Things

**Zunming Chen [1], Hongyan Cui [2,*], Ensen Wu [2] and Xi Yu [2]**

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; czm@bupt.edu.cn
[2] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; wuensen@bupt.edu.cn (E.W.); yusy@bupt.edu.cn (X.Y.)
* Correspondence: cuihy@bupt.edu.cn

**Abstract:** The proliferation of the Internet of Things (IoT) and widespread use of devices with sensing, computing, and communication capabilities have motivated intelligent applications empowered by artificial intelligence. Classical artificial intelligence algorithms require centralized data collection and processing, which are challenging in realistic intelligent IoT applications due to growing data privacy concerns and distributed datasets. Federated Learning (FL) has emerged as a privacy-preserving distributed learning framework, which enables IoT devices to train global models through sharing model parameters. However, inefficiency due to frequent parameter transmissions significantly reduces FL performance. Existing acceleration algorithms consist of two main types including local update and parameter compression, which considers the trade-offs between communication and computation/precision, respectively. Jointly considering these two trade-offs and adaptively balancing their impacts on convergence have remained unresolved. To solve the problem, this paper proposes a novel efficient adaptive federated optimization (FedEAFO) algorithm to improve the efficiency of FL, which minimizes the learning error via jointly considering two variables including local update and parameter compression. The FedEAFO enables FL to adaptively adjust two variables and balance trade-offs among computation, communication, and precision. The experiment results illustrate that compared with state-of-the-art algorithms, the FedEAFO can achieve higher accuracies faster.

**Keywords:** federated learning; distributed machine learning; communication efficiency; privacy protection

check for updates

**Citation:** Chen, Z.; Cui, H.; Wu, E.; Yu, X. Computation and Communication Efficient Adaptive Federated Optimization of Federated Learning for Internet of Things. *Electronics* **2023**, *12*, 3451. https://doi.org/10.3390/electronics12163451

Academic Editor: Imran

Received: 21 July 2023
Revised: 11 August 2023
Accepted: 13 August 2023
Published: 15 August 2023

## 1. Introduction

The explosive growth in the amount of data from devices has witnessed the rapid development of the Internet of Things (IoT), which provides ubiquitous sensing, computing, and communication capabilities to connect things to the Internet [1]. To provide deep analysis for data from IoT devices, artificial intelligence (AI) algorithms have been adopted to enable intelligent IoT applications such as smart transportation and smart city [2]. Traditionally, AI algorithms that need the centralized collection and processing of data are deployed on a centralized cloud/edge server or data center for data mining. However, the offloading of massive amounts of IoT data to remote servers and the processing of data in remote servers induces significant delays. Furthermore, the third-party server also raises data privacy concerns [3]. In this context, integrating privacy-preservation and distributed AI into IoT becomes an important topic.

Recently, Federated Learning (FL) has emerged as a privacy-preserving distributed learning framework that enables intelligent IoT applications by allowing distributed IoT devices to collaboratively train machine learning models [4]. FL enables multiple devices to train a joint global model via Stochastic Gradient Descent (SGD) and shares local model parameters instead of raw data [5]. FL has seen recent successes in several applications. For example, Genetic Clustered Federated Learning (CFL) proposed in [6] has been applied to detect COVID-19 patients in a privacy-preserving manner. An FL-assisted cooperative perception task in [7] has been applied to vehicular networks where vehicles fuse sensor data in order to obtain an extended vision of the driving environment.

However, the existing federated learning methods suffer from inefficiency in model training performance when applied in resource-constrained IoT environments due to several reasons. (i) Unbalanced Data: Training data at IoT devices are different in size and distribution because of different sensing environments [8]. Thus, communications in the uplink and downlink in FL are highly sensitive subject to non-independent and identically distributed (IID) data. (ii) Limited Bandwidth: IoT devices often have limited communication bandwidth due to their constrained network capabilities. Transmitting large model updates in traditional federated learning setups can be inefficient and slow in such environments. (iii) High Latency: IoT devices might have high communication latency, making frequent model updates impractical. This can lead to delays in aggregating updates and hinder the training process. (iv) Unreliable Connectivity: IoT devices might experience intermittent or unreliable connectivity. This can disrupt the communication process and lead to incomplete or delayed updates. The convergence of FL for IoT cannot be guaranteed all the time due to the intermittent connections [9–11]. The situation of federated learning on mobile devices (e.g., sensors and Unmanned Aerial Vehicles (UAVs)) gets even worse, as they communicate via wireless channels and suffer from lower bandwidth, higher latency, and intermittent connections [12]. Thus, the inefficiency training performance problem becomes an important bottleneck for scaling up FL.

It is necessary to solve the inefficiency training performance problem of federated learning. Several works have been undertaken to speed up federated learning convergence via local update and parameter compression [13,14]. The approach of a local update aims to reduce the frequent transmissions of model parameters via making full use of the computing capability of devices. The local update algorithms characterize a trade-off between computation and communication via a proposed concept of a local update coefficient that determines the ratio of the local update to bandwidth [15]. The approach of parameter compression aims at reducing the amount of data to be transmitted through data compression techniques such as quantization and sparsification. The parameter compression algorithms characterize a trade-off between communication and model precision via the parameter compression coefficient (compression budget), which determines the ratio between bandwidth and accuracy [16]. These methods have been individually studied to improve the efficiency of federated learning. However, jointly considering these two trade-offs and adaptively balancing their impacts on the convergence of federated learning from both mathematical estimation and theoretical analysis perspectives have remained unresolved. This significant problem motivated our research. This paper proposes a novel efficient adaptive federated optimization (FedEAFO) algorithm to speed up the convergence of federated learning for IoT, which minimizes the learning error via jointly considering two methods including local update and parameter compression. The FedEAFO enables federated learning to adaptively adjust two variables and balance trade-offs among computation, communication, and accuracy.

The proposed efficient adaptive federated optimization algorithm stands apart from traditional optimization approaches in several significant ways. These distinctions underscore the algorithm's advancements and its tailored suitability for the complexities of IoT environments. (i) Joint Communication and Computation Optimization: One of the key differentiators of the FedEAFO algorithm is its holistic approach toward optimization. Unlike traditional methods that often focus on either communication or computation aspects, FedEAFO takes a pioneering step by seamlessly integrating both model compression techniques and multiple local training strategies. This integrated optimization effectively addresses the intricate interplay between computation and communication efficiency within the federated learning framework. (ii) Adaptive Optimization: FedEAFO introduces an adaptive dimension that sets it apart from conventional techniques. This adaptability empowers the algorithm to dynamically and intelligently adjust its optimization parameters, facilitating a fine-tuned balance between various trade-offs. This adaptiveness is in stark contrast to traditional methods that often employ fixed or pre-determined parameters, making FedEAFO particularly responsive to the dynamic nature of IoT environments. (iii) Enhanced Convergence Speed: Another distinctive feature lies in FedEAFO's aim to expedite the convergence speed of federated learning. Unlike some conventional methods that might struggle to adapt to the specific challenges posed by IoT settings, FedEAFO strategically integrates model compression and local training. This synergy fosters quicker convergence by optimizing learning updates and conserving communication resources simultaneously. (iv) Tailoring to IoT Constraints: Traditional optimization methods might not fully account for the unique constraints and intricacies of IoT environments. FedEAFO, on the other hand, is meticulously crafted to align with the limitations of IoT devices, such as constrained bandwidth, high latency, and energy considerations. Its ability to harmonize these factors with optimization objectives marks a substantial departure from generic optimization paradigms.

This research introduces a novel approach that addresses the unique inefficiency training performance challenge faced by federated learning when applied to IoT environments, which is raised by limitations including limited bandwidth of devices, unreliable network connectivity, device heterogeneity, and unbalanced data. This research proposes innovative model compression techniques tailored to IoT devices, which aim to reduce the size of the model updates that need to be communicated between IoT devices and the central server. By compressing the model effectively, this research enables efficient utilization of the limited communication bandwidth of IoT devices. In addition, recognizing the high latency and limited energy resources of IoT devices, this research suggests the incorporation of multiple local training methods, which enable IoT devices to perform multiple rounds of training on their local data before transmitting updates to the central server. This can minimize the need for frequent communication, addressing the challenge posed by high latency and unreliable connectivity. Furthermore, this research takes into account the inherent heterogeneity of IoT devices in terms of hardware capabilities, network conditions, and data distributions. By jointly considering model compression and multiple local training, the proposed algorithm adapts to the diverse IoT system and ensures compatibility with devices of varying capabilities. Lastly, IoT devices often exhibit a wide range of computational capabilities and data distribution. The proposed efficient adaptive optimization method addresses these heterogeneities by incorporating adaptive multiple local training. The proposed algorithm can adjust the local update frequencies of devices to mitigate the impact of device heterogeneity and data imbalance. This method allows each IoT device to perform several rounds of training using its available computational resources. Devices with higher computational capacities can perform more local training rounds, contributing more effectively to the federated learning process. In addition, devices with smaller datasets might be allowed to perform more local updates before communicating with the central server, which enables devices with sparse data to catch up with those that have more data, reducing the disparities caused by data imbalance. The key contributions of the FedEAFO are summarized as the following:

- This paper investigates the federated learning problem with a practical formulation of minimizing the error of the global model in terms of local update coefficient and compression budget, which characterizes trade-offs between communication and computation/model precision, respectively.
- This paper proposes a novel efficient adaptive federated optimization algorithm using a derived error upper bound considering two variables including a local update and compression coefficient, which adaptively adjusts these two variables to improve the efficiency of federated learning.
- Besides theoretical analysis of the proposed algorithm, we demonstrate strong empirical performance on two datasets of FedEAFO compared with other state-of-the-art methods, which achieve higher accuracies faster.

The proposed efficient adaptive federated optimization algorithm is a promising approach that addresses the challenges of training efficiency, heterogeneity, non-IID data, and privacy concerns in a distributed machine learning environment. It offers several practical applications across various industries: (i) Autonomous Vehicles: Autonomous vehicles generate a wealth of data during operation. With efficient adaptive federated learning, vehicles can collectively learn from each other's experiences to improve safety and navigation while respecting user privacy. (ii) Financial Services: Banks and financial institutions often face data privacy regulations, making data sharing difficult. With efficient adaptive federated learning, banks can collectively analyze customer behavior and detect fraudulent activities without directly sharing sensitive financial information. It enables improved risk assessment, personalized financial recommendations, and fraud detection while preserving customer privacy. (iii) Smart Grids: In the energy sector, smart grids consist of numerous energy-consuming and energy-producing devices. Federated learning can help optimize energy consumption, predict energy demand, and manage grid stability while maintaining data privacy and decentralized decision making. (iv) Manufacturing: In smart factories, different machines and sensors generate data with varying characteristics. Efficient adaptive federated learning enables predictive maintenance, process optimization, and quality control without the need for centralized data collection, improving manufacturing efficiency and reducing downtime.

## 2. Related Work

### 2.1. Local Update in Federated Learning

The approaches to improve the efficiency of federated learning and overcome communication bottlenecks can be categorized into local update and parameter compression. The approach of the local update aims at taking full advantage of the computing capability of devices to reduce the frequent transmissions of model parameters. For example, Nenghai et al. proposed the Asynchronous Stochastic Gradient Descent (ASGD) algorithm, which derives the bound of convergence of distributed gradient descent and only allows one step of the local update before the aggregation of the global model [17]. Virginia et al. proposed Fedprox, which puts forward the concept of a local model update coefficient determining the ratio of computation to communication, and it performs multiple local updates with a fixed local update coefficient [18]. Joshi et al. proposed ADACOMM, where an adaptive communication strategy is adopted to adjust the local update coefficient and dynamically adjust the trade-off between computation and communication to solve the problem of heterogeneous computing and communication capabilities of devices [19]. Kevin et.al proposed an AFD that adopts an adaptive strategy to determine the best local update coefficient under a given resource budget in order to speed up federated learning in non-IID settings [20].

### 2.2. Parameter Compression in Federated Learning

The approach of parameter compression adopts data compression algorithms such as sparsification and quantization to significantly reduce the amount of data to be transmitted. The parameter compression algorithm characterizes the trade-off between communication and model precision by a compression coefficient determining the ratio of communication to precision. The algorithms of parameter compression reduce communication overheads via uploading the quantized version or sparse representation of the model parameters. For instance, Wen et al. proposed Terngrad, which quantizes each parameter to 2 bits [21]. Alistarh et al. proposed the Quantification Stochastic Gradient Descent (QSGD) algorithm, which uses 2 bits and 4 bits to quantize various layers of model networks [22]. A 1-bit SGD that even quantizes the parameter to 1 bit was studied in [23]. The sparsification technique aims to sparsify parameters to send the significant parameters rather than all parameters. The sparsification techniques can be categorized into two categories according to the domain the sparsity is sought for, which include the raw domain and the transformed domain. For instance, Dally et al. proposed a parameter sparsification strategy to set unimportant elements to zero via a threshold, which defines values of unimportant elements as being between top 0.05% and bottom 0.05% [24]. Wright et al. proposed ATOMO, which aims at reducing the communication overheads via transforming parameters to a domain of Singular Value Decomposition (SVD) to exploit the low-dimensional structure to obtain more sparsity [25].

The summary of related works is listed in Table 1. The aforementioned methods have been individually studied to overcome communication bottlenecks and improve the efficiency of federated learning. It is expected that integrating the two approaches would be more effective in speeding up federated learning. However, integrating the two approaches and jointly considering the trade-offs between communication and computation/precision to adaptively balance and adjust their impacts on the convergence of federated learning have remained unresolved. Thus, there is an urgent necessity to design an efficient adaptive federated optimization algorithm via jointly balancing the trade-offs among communication, computation, and precision to supplement existing approaches.

**Table 1.** Summary of the approaches of efficient federated learning.

| Ref. | Challenge | Technique | Key Idea |
|---|---|---|---|
| [18] | Communication frequency reduction | Local update | Enable devices to perform multiple local updates with a fixed local update coefficient |
| [19] | Communication frequency reduction | Local update | An adaptive strategy is adopted to adjust local update coefficient dynamically |
| [20] | Communication frequency reduction | Local update | Similar to [19], but with convergence guarantees for non-IID setting |
| [21] | Model updates size reduction | Compression (quantization) | Compress the local model updates to a finite number of bits to reduce the amount of data transmitted between server and devices |
| [24] | Model updates size reduction | Compression (sparsification) | A model updates sparsification strategy to set unimportant elements to zero |
| [25] | Model updates size reduction | Compression (sparsification) | Transform model updates into Singular Value Decomposition (SVD) domain and sparsify model updates into low-dimensional structure |

## 3. Preliminaries and Problem Formulation

The system overview of the proposed algorithm is presented in Figure 1. Before introducing details of the proposed algorithm, this paper first presents a mathematical analysis on how coefficients of local update and parameter compression affect federated learning in order to further describe the impact of local update and parameter compression. Additionally, we theoretically formulate our problem and derive the error upper bound of federated learning, which jointly considers two variables including local update and parameter compression.

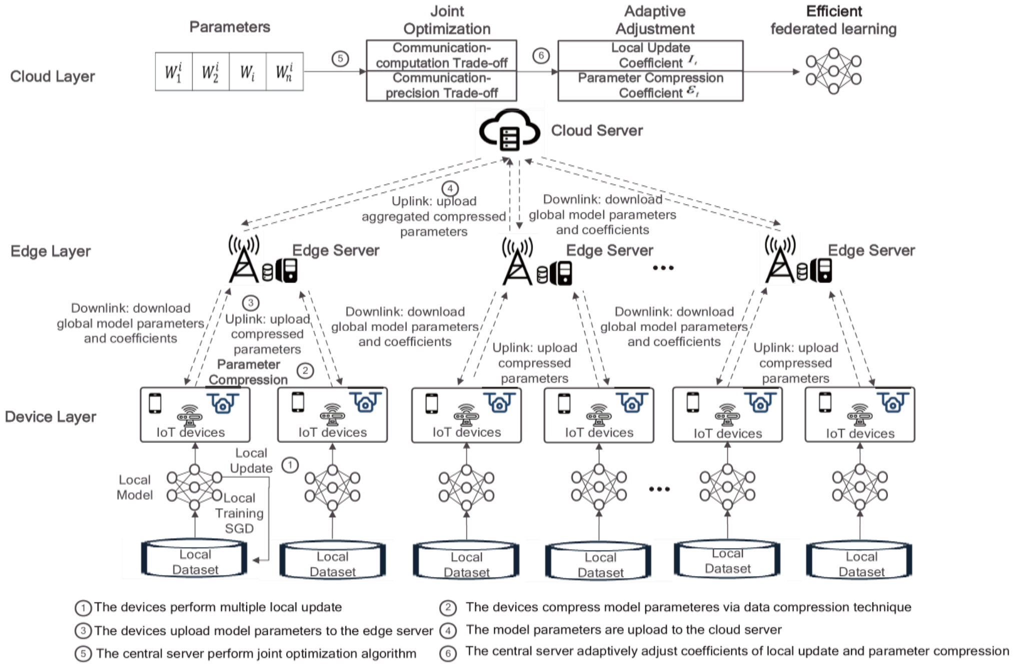

Parameters
Joint Optimization
Adaptive Adjustment
Efficient federated learning
Cloud Layer
$W_1^i$ $W_2^i$ $W_i$ $W_n^i$
⑤
Communication-computation Trade-off
Communication-precision Trade-off
⑥
Local Update Coefficient $I_t$
Parameter Compression Coefficient $\varepsilon_t$
Cloud Server
④
Uplink: upload aggregated compressed parameters
Downlink: download global model parameters and coefficients
Edge Layer
Edge Server
Edge Server
Edge Server
Downlink: download global model parameters and coefficients
③
Uplink: upload compressed parameters
Parameter Compression
②
Downlink: download global model parameters and coefficients
Uplink: upload compressed parameters
Downlink: download global model parameters and coefficients
Uplink: upload compressed parameters
Device Layer
IoT devices
Local Update
①
Local Model
Local Training SGD
Local Dataset
① The devices perform multiple local update
② The devices compress model parameteres via data compression technique
③ The devices upload model parameters to the edge server
④ The model parameters are upload to the cloud server
⑤ The central server perform joint optimization algorithm
⑥ The central server adaptively adjust coefficients of local update and parameter compression

**Figure 1.** The system overview of FedEAFO scheme.

### *3.1. Federated Learning*

We consider a federated learning system in resource-constrained IoT environments, which consists of a total of $N$ IoT devices. Each device performs model training on its local dataset. The devices aim to jointly solve the following optimization problem.

$$w = \min_{w}\left\{ F(w) \Leftrightarrow \sum_{n=1}^{N} p^n F^n(w) \right\} \tag{1}$$

where $w$ represents global model weights, $p^n$ corresponds to weight of the $n$-th device, and $F^n$ corresponds to local objective of the $n$-th device. Equation (1) can be optimized by iterative exchange of model parameters between devices and the central server. Particularly, at the $t$-th round, the local model of $n$-th device $w^n_{t,i}$ can be denoted as

$$w^n_{t,i+1} \leftarrow w^n_{t,i} - \eta_{t,i} \nabla F^n\left(w^n_{t,i}, \varphi^n_{t,i}\right) \tag{2}$$

where $\varphi^n_{t,i}$ corresponds to the data samples, $\eta_{t,i}$ represents the learning rate, and $i$ represents the $i$-th local updates.

In practical applications of the IoT, different IoT devices will be used and different kinds of data will be collected. FL is well-suited for handling different types of data. In IoT scenarios, various devices with diverse sensors and functionalities generate data, and these data can be highly heterogeneous. FL provides a distributed learning framework that allows these devices to collaboratively learn a shared model without sharing their raw data with a central server. Here's how FL handles different types of data in IoT applications: (i) Adaptive Hyperparameter Tuning: Some FL algorithms incorporate adaptive optimization techniques that allow the model to adjust its hyperparameters based on the data characteristics of each device. This adaptivity helps in efficiently utilizing devices with diverse data distributions. (ii) Decentralized Data Processing: In FL, each IoT device retains control over its data locally. The devices preprocess and transform their data to ensure consistency and compatibility with the chosen model architecture. This decentralization allows the devices to handle their specific data types and formats independently. (iii) Model Personalization: FL supports model personalization, where the global model can be adapted to different devices' data characteristics. When devices have unique data distributions, they can personalize the shared model using their local data. Model aggregation mechanisms, like Federated Averaging, ensure that the personalized models are combined to improve the overall performance.

As soon as the $n$-th device performs local updates based on Equation (2), the aggregated updates of local models $\ell(w^n_t)$ can be obtained by

$$\ell(w^n_t) \leftarrow \sum_{i=1}^{I_t} \ell\left(w^n_{t,i}, \varphi^n_{t,i}\right) \tag{3}$$

where $\ell\left(w^n_{t,i}, \varphi^n_{t,i}\right) = \nabla F^n\left(w^n_{t,i}, \varphi^n_{t,i}\right)$. $I_t$ represents the number of consecutive local updates, which can be adjusted to balance the trade-off between computation and communication of FL.

The devices then compress their locally aggregated updates $\ell(w^n_t)$ to $\hat{\ell}(w^n_t)$ with the sparsity budget denoted as $\varepsilon_t$, which can be adjusted to balance trade-offs between the communication and model precision of FL. The server aggregates all compressed locally aggregated updates from devices to obtain a compressed global model given by

$$\hat{\ell}(w_t) = \frac{1}{N} \sum_{n=1}^{N} \hat{\ell}(w^n_t) \tag{4}$$

The latest global model $w_{t+1}$ can be updated via the stochastic gradient descent algorithm as

$$w_{t+1} \leftarrow w_t - \eta \hat{\ell}(w_t) \tag{5}$$

The central server forwards the updated global model $w_{t+1}$ to devices involved in federated learning for the next training round. The local model of devices can be updated via the received latest global weights. This is the complete process of federated learning with joint consideration of both local update and parameter compression.

For the analysis of the latency of the training process of our system, we consider a circular small-cell network in which devices are uniformly distributed within the coverage of central base stations. Let $d^n$ represent the between the base station and the device $n$, and $L_{path}(d)$ denote the path loss. Let $P^n$ and $P$ denote the transmit power of the device $n$ and the base station, respectively. Assume that the uplink channel adopts the orthogonal frequency division multiple access (OFDMA) technology. The devices compete for a resource block to upload local models under the rule of the Slotted ALOHA protocol in the uplink channel. And the base station occupies all the bandwidth to forward global models in downlink time slots. Let $B$ denote the total bandwidth of our system and $B_U$ denote the resource block bandwidth. Thus, the signal-to-noise ratio of the uplink and downlink channel can be denoted by $\rho_U^n = P^n - L_{\text{path}}(d^n) - N_0 B_U$ and $\rho_D^n = P - L_{\text{path}}(d^n) - N_0 B$, where $N_0$ is the spectral power density of the noise.

Based on the above definition, the communication latency $T_U^n$ for the local model upload and the global model download $T_D^n$ of the $n$-th device can be given by

$$T_U^n = \frac{H}{B_U \log_2(1+\rho_U^n)}, \quad T_D^n = \frac{H}{B \log_2(1+\rho_D^n)} \tag{6}$$

where $H$ denotes the total quantization bits for transmitting the model.

The computation latency for training and updating the local model of the $n$-th device can be calculated as

$$T_{C,t}^n = \frac{ECD^n}{f^n}, \quad T_{L,t}^n = \frac{Q}{f^n} \tag{7}$$

where $E$ is the number of local updates, $C$ and $Q$ respectively denote the number of floating-point operations required for training a data sample and updating the local model, and $f^n$ denotes the CPU capability of the $n$-th device.

Thus, the end-to-end delay of the FL training process at the $t$-th round can be given by

$$T_t = \underbrace{\max\{T_{U,t}^n + T_{D,t}^n\}}_{T_{comm}} + \underbrace{\max\{T_{C,t}^n + T_{L,t}^n\}}_{T_{comp}} \tag{8}$$

### *3.2. Problem Formulation*

Our goal is to jointly consider two variables including $I_t$ and $\varepsilon_t$ and adaptively adjust them to speed up federated learning. Theoretically, the problem is to find the optimal solution of $I_t$ and $\varepsilon_t$ at different training rounds, which minimizes the error of federated learning in a given time. The problem can be mathematically formulated as

$$\begin{aligned} &\min_{\{I_t\},\{\varepsilon_t\}} E_{\{\varphi_{t,i}^n\}}[\min F(w_t)] \\ &\text{s.t.} \sum_{t=1}^{U_t} (T_{comp} + T_{comm}) = T \end{aligned} \tag{9}$$

where $F(w_t)$ corresponds to the global learning objective defined in Equation (1). $T$ represents the given time constraint. To solve the problem in Equation (9), the key is to derive expression of the error of federated learning describing the interdependence of $I_t$ and $\varepsilon_t$. However, such expression is almost impossible to obtain. Additionally, it is also hard to find the error upper bound, which jointly considers $I_t$ and $\varepsilon_t$ in FL.

### *3.3. Learning Error Upper Bound*

To derive the error upper bound of federated learning, which jointly considers two variables including local update and parameter compression, this subsection first derives error upper bound in terms of $I_t$. Additionally, we derive error upper bound with consideration of local update described by $I_t$ and parameter compression described by $\varepsilon_t$.

We first make the following Assumptions 1–3 to present the analysis of error upper bound without parameter compression inspired by [19].

- The global loss function is differentiable, and L-smooth: $|\nabla F(\mathbf{V}) - \nabla F(\mathbf{W})|| \leq L||\mathbf{V} - \mathbf{W}|$ and there is a lower bound $\mathrm{F_{inf}}$.
- The global weights variance in mini-batch is bounded by: $E_{\{\varphi_{t,i}^{t}\}}||\ell(w_t) - \nabla F(w_t)||^2 \leq \lambda||\nabla F(w_n)||^2 + \delta$. $\delta$ corresponds to variance between $\ell(w_n)$. $\lambda$ and $\delta$ are constants inversely proportional to mini-batch size.
- The SGD is the unbiased estimator of FGD: $E_{\{\varphi_{t,i}^{n}\}}[\ell(w_t)] = \nabla F(w_t)$.

**Theorem 1.** *Let Assumptions 1–3 hold. Choose the learning rate that satisfies* $\eta L + \eta^2 L^2 I_t (I_t - 1) \leq 1$. Thus, the learning error after n rounds within given time T in Equation (9) is bounded by:

$$\frac{2[F(w_t) - \mathrm{F_{inf}}]}{\eta T}\left(T_{comp} + \frac{T_{comm}}{I_t}\right) + \frac{\eta L \delta}{N} + \eta^2 L^2 \delta (I_t - 1) \tag{10}$$

**Proof.** See Appendix in [19]. □

The above Theorem 1 describes the trade-off between computation and communication to minimize learning error. Equation (10) illustrates that the local update coefficient $I_t$ is in the numerator and the denominator of expression of error upper bound, which means error upper bound will decrease, where either the values of $I_t$ is too small or too large. Thus, it is necessary to achieve balance. Additionally, the trade-off needs to be dynamically adjusted over various rounds of federated learning due to the dynamically varying loss function $F(w_k)$, which is in Equation (10).

Apart from the local update, the approach of parameter compression, which introduces compression into locally aggregated weights, will complicate the analysis of error upper bound in two aspects: (i) $T_{comm}$ will be affected by the parameter compression coefficient $\varepsilon_t$ [26,27]. (ii) The variance $\delta$ in Equation (10) will depend on the parameter compression coefficient $\varepsilon_t$ [28,29].

As mentioned before, the compressed parameters are approximated by basic components in parameter compression. Because of sparsity, several components are more significant than others in the aspects of approximating the raw parameters. Thus, the problem is to select basic components unbiasedly to minimize the variance $\delta$. The locally aggregated weights from the $n$-th device can be rewritten as:

$$\ell(w_t^n) = \sum_{k=1}^{K} d^k(w_t^n)\alpha^k(w_t^n) \tag{11}$$

where $K$ is the number of basic components, $\alpha^k(w_n^t)$ corresponds to the $k$-th basic component, and $d^k(w_t^n)$ is the corresponding weight. Our analysis is based on the fact that a matrix can be denoted as a combination of basic matrices, which is the atomic decomposition for sparse representation in compressed sensing. Thus, our analysis can be extended to nearly all unbiased compressions. For example, TernGrad [21] and QSGD [22] are special cases of Equation (8). Additionally, sparsification algorithms including ATOMO [25] also follow

Equation (8). The formulation presents a problem on the selection of $d^k(w_t^n)$. To meet the requirement of unbiased selection, we propose an estimator as follows, inspired by [25]:

$$\hat{\ell}(w_t^n) = \sum_{k=1}^{K} \frac{d^k(w_t^n)e^k(w_t^n)}{p^k(w_t^n)} \alpha^k(w_t^n) \tag{12}$$

where $p^k(w_t^n)$ corresponds to the probability characterizing the Bernoulli distribution, $p^k(w_t^n) \in (0,1]$ and $e^k(w_t^n)$ obeys the Bernoulli distribution. We provide two significant properties for the estimator via Lemmas 1 and 2.

**Lemma 1.** *The variance of the estimator given by Equation (12) can be denoted as:* $E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] = \sum\limits_{k=1}^{K} d^k(w_t^n)^2 \left(\frac{1}{p^k\left(w_t^n\right)} - 1\right)$.

**Proof.** The variance of the estimator in Equation (9) is defined as follows:

$$\begin{aligned}
&E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] \\
&= E_{\{e^k(w_t^n)\}}\left[\left(\sum_{k=1}^{K} d^k(w_t^n)a^k(w_t^n) \left(\frac{e^k(w_t^n)-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)\right)^T \right.\\
&\left.\times\left(\sum_{k=1}^{K} d^k(w_t^n)a^k(w_t^n) \left(\frac{e^k(w_t^n)-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)\right)\right] \\
&= \sum_{k=1}^{K} d^k(w_t^n)^2||a^k(w_t^n)||^2 \times E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^k(w_t^n)-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)^2\right] \\
&+ \sum_{x,y;x\neq y}^{k} d^x(w_t^n)d^y(w_t^n)\langle a^x(w_t^n), a^y(w_t^n)\rangle \\
&\times E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^x(w_t^n)-p^x(w_t^n)}{p^x\left(w_t^n\right)}\right) \times \left(\frac{e^y(w_t^n)-p^y(w_t^n)}{p^y\left(w_t^n\right)}\right)\right].
\end{aligned} \tag{13}$$

where $E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^k(w_t^n)-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)^2\right]$ can be denoted as:

$$\begin{aligned}
&E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^k(w_t^n)-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)^2\right] \\
&= \left(1 - p^k(w_t^n)\right) \times \left(\frac{0-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)^2 + p^k(w_t^n) \times \left(\frac{1-p^k(w_t^n)}{p^k\left(w_t^n\right)}\right)^2 \\
&= \left(\frac{1}{p^k\left(w_t^n\right)} - 1\right)
\end{aligned} \tag{14}$$

and $E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^x(w_t^n)-p^x(w_t^n)}{p^x\left(w_t^n\right)}\right)\right]$ can be obtained by:

$$\begin{aligned}
&E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^x(w_t^n)-p^x(w_t^n)}{p^x\left(w_t^n\right)}\right)\right] \\
&= (1 - p^x(w_t^n))\left(\frac{0-p^x(w_t^n)}{p^x\left(w_t^n\right)}\right) + p^x(w_t^n)\left(\frac{1-p^x(w_t^n)}{p^x\left(w_t^n\right)}\right) \\
&= 0
\end{aligned} \tag{15}$$

and in the same way, $E_{\{e^k(w_t^n)\}}\left[\left(\frac{e^y(w_t^n)-p^y(w_t^n)}{p^y\left(w_t^n\right)}\right)\right] = 0$.

Thus, based on Equations (13)–(15) and $\left|\left|a^k(w_t^n)\right|\right|_2 = 1$, the variance of the estimator in Equation (12) can be obtained as follows:

$$E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] = \sum_{k=1}^{K} d^k(w_t^n)^2\left(\frac{1}{p^k(w_t^n)} - 1\right) \tag{16}$$

□

**Lemma 2.** *The estimator in Equation (12) is unbiased, which means:* $E_{\{e^k(w_t^n)\}}\left[\hat{\ell}(w_t^n)\right] = \ell(w_t^n)$.

**Proof.** This can be proved simply by the definition of the expectation. □

In order to minimize the variance, we formulate the optimization problem. The reason why we try to minimize the variance is that the compressed parameters are closer to the original parameters when the variance decreases. The problem of minimizing the variance can be given as:

$$\min\sum_{k=1}^{K}\frac{d^k(w_t^n)^2}{p^k(w_t^n)} \quad \text{s.t.} \quad 0 < p^k(w_t^n) \le 1 \quad \textit{and} \quad \sum_{k=1}^{K} p^k(w_t^n) = \varepsilon_t \tag{17}$$

Before solving the optimization problem, we first provide the following Definition 1 of $\varepsilon_t$-balancedness:

**Definition 1.** $\ell(w_\mathrm{t}^n) = \sum\limits_{k=1}^{K} d^k(w_\mathrm{t}^n)\alpha^k(w_\mathrm{t}^n)$ *is* $\varepsilon_t$*-unbalanced if* $d^k(w_t^n)\varepsilon_t > \left|\left|dw_t^n\right|\right|_1$. $\varepsilon_t$*-balanced corresponds to the case that no element of* $\ell(w_\mathrm{t}^n)$ *is* $\varepsilon_t$*-unbalanced.*

Thus, the theorem for the solution of the optimization problem can be given by:

**Theorem 2.** *When* $\ell(w_\mathrm{t}^n)$ *is* $\varepsilon_t$*-balanced, the solution for the aforementioned problem can be obtained by:*

$$p^k(w_t^n) = \frac{d^k(w_t^n)\varepsilon_t}{\|d(w_t^n)\|_1} \tag{18}$$

**Proof.** Theorem 2 can be proved by the Lagrangian multiplier [30]. □

**Lemma 3.** *The difference between uncompressed parameters* $\ell(w_t^n)$ *and compressed parameters* $\hat{\ell}(w_t^n)$ *and can be given by:*

$$E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] = \frac{\delta_{1,n}^t}{\varepsilon_t} + \delta_{2,n}^t \tag{19}$$

*where* $\delta_{1,n}^t = \sum\limits_{k=1}^{K}\left(d^k(w_t^n)\|d(w_t^n)\|_1\right)$ *and* $\delta_{2,n}^t = -\sum\limits_{k=1}^{K} d^k(w_t^n)^2$.

**Proof.** Based on Lemma 1 and Theorem 2, $E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right]$ can be denoted by:

$$\begin{aligned} &E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] \\ &= \sum_{k=1}^{K} d^k(w_t^n)^2\left(\tfrac{1}{p^k(w_t^n)} - 1\right) \\ &= \tfrac{1}{\varepsilon_t}\sum_{k=1}^{K}\left(d^k(w_t^n)\|d(w_t^n)\|_1\right) - \sum_{k=1}^{K} d^k(w_t^n)^2 = \tfrac{\delta_{1,n}^t}{\varepsilon_t} + \delta_{2,n}^t \end{aligned} \tag{20}$$

where $\delta_{1,n}^{t} = \sum_{k=1}^{K}\left(d^k(w_t^n)\|d(w_t^n)\|_1\right)$ and $\delta_{2,n}^{t} = -\sum_{k=1}^{K} d^k(w_t^n)^2$. □

After deriving the probability of the variance of the estimator, we are able to access the effects of compression on variance $\delta$ and communication time $T_{comm}$. Assuming that the model parameters to be uploaded are denoted as $\Upsilon$. Rather than sending original parameters, the devices can upload the compressed approximated parameters denoted as $\overset{\wedge}{\Upsilon}$ with sparsity representation, which only uploads the $\varepsilon_t$ number of basic components. Thus, the communication time $T_{comm}$ can be rewritten as $T_{comm} = \gamma\varepsilon_t$, where $\gamma$ corresponds to the communication time of each device. As to variance $\delta$, the theorem for the variance of compressed locally aggregated parameters is based on Theorem 2, which can be given by:

**Theorem 3.** *The bound for the variance of compressed locally aggregated parameters is as follows:*

$$E_{\{\varphi_{t,i}^n\},\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t) - \nabla F(w_t)\|^2\right] \le \lambda\|\nabla F(w_t)\|^2 + \frac{\delta_1}{\varepsilon_t} + \delta_2 \tag{21}$$

*where $\lambda$, $\delta_1$, and $\delta_2$ are constants inversely proportional to the size of the mini-batch.*

**Proof.** The $E_{\{\varphi_{t,i}^n\},\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t) - \nabla F(w_t)\|^2\right]$ can be described as:

$$\begin{aligned}
&E_{\{\varphi_{t,i}^n\},\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t) - \nabla F(w_t)\|^2\right] \\
&= E_{\{\varphi_{t,i}^n\},\{e^k(w_t^n)\}}\left[\left|\left|\hat{\ell}(w_t) - \ell(w_t)\ +\ell(w_t) - \nabla F(w_t)\right|\right|^2\right] \\
&\le E_{\{e^k(w_t^n)\}}\left[\left|\left|\hat{\ell}(w_t) - \ell(w_t)\right|\right|^2\right] + E_{\{\varphi_{t,i}^n\}}\left[\|\ell(w_t) - \nabla F(w_t)\|^2\right]
\end{aligned} \tag{22}$$

For the first term in Equation (22), the $E_{\{e^k(w_t^n)\}}\left[\left|\left|\hat{\ell}(w_t) - \ell(w_t)\right|\right|^2\right]$ can be obtained by:

$$\begin{aligned}
&E_{\{e^k(w_t^n)\}}\left[\left|\left|\hat{\ell}(w_t) - \ell(w_t)\right|\right|^2\right] \\
&\le \tfrac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} d^k(w_t^n)^2\left(\tfrac{1}{p^k(w_t^n)} - 1\right) = \tfrac{1}{N}\left(\sum_{n=1}^{N}\tfrac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n}\right)
\end{aligned} \tag{23}$$

where $E_{\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t^n) - \ell(w_t^n)\|^2\right] = \sum_{k=1}^{K} d^k(w_t^n)^2\left(\tfrac{1}{p^k(w_t^n)} - 1\right)$ is from Lemma 1, and $\sum_{k=1}^{K} d^k(w_t^n)^2\left(\tfrac{1}{p^k(w_t^n)} - 1\right) = \tfrac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n}$ is from Lemma 3.

For the second term in Equation (22), $E_{\{\varphi_{t,i}^n\}}\left[\|\ell(w_t) - \nabla F(w_t)\|^2\right]$ can be obtained by $\lambda\|\nabla F(w_t)\|^2 + \delta$ following by [19].

Thus, based on Equations (22)–(23), $E_{\{\varphi_{t,i}^n\},\{e^k(w_t^n)\}}\left[\|\hat{\ell}(w_t) - \nabla F(w_t)\|^2\right]$ can be described by:

$$\begin{aligned}
&E_{\{e^k(w_t^n)\}}\left[\left|\left|\hat{\ell}(w_t) - \ell(w_t)\right|\right|^2\right] \\
&\le \tfrac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} d^k(w_t^n)^2\left(\tfrac{1}{p^k(w_t^n)} - 1\right) = \tfrac{1}{N}\left(\sum_{n=1}^{N}\tfrac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n}\right)
\end{aligned} \tag{24}$$

where $\delta_1 = \max_k\left(\tfrac{1}{N}\sum_{n=1}^{N}\delta_{1,n}^{k}\right)$ and $\delta_2 = \max_k\left(\tfrac{1}{N}\sum_{n=1}^{N}\delta_{2,j}^{k}\right) + \delta$. □

With the derived new communication time and variance, Assumption 2 in Theorem 1 can be rewritten via Equation (25). To derive the error upper bound with parameter compression, we adapt Theorem 1 via new variance and communication time, which considers the impact of parameter compression on communication time and variance on the convergence of federated learning. The updated Theorem is as follows:

**Theorem 4.** *The bound error upper bound with parameter compression is as follows:*

$$v_t(I_t, \varepsilon_t) = \frac{2[F(w_t) - F_{\inf}]}{\eta T}\left(T_{comp} + \frac{\gamma\varepsilon_t}{I_t}\right) + \frac{\eta L(\frac{\delta_1}{\varepsilon_t} + \delta_2)}{N} + \eta^2 L^2(\frac{\delta_1}{\varepsilon_t} + \delta_2)(I_t - 1) \tag{25}$$

*where $v_t(I_t, \varepsilon_t)$ corresponds to the error upper bound considering local update and parameter compression.*

**Proof.** See Appendix in [19], except for the new communication time $T_{comp}$ replaced by $\gamma\varepsilon_t$ and variance $\delta$ replaced by $\frac{\delta_1}{\varepsilon_t} + \delta_2$. □

$v_t(I_t, \varepsilon_t)$ indicates the dynamics of trade-offs between communication and computation/precision, which are determined by $I_t$ and $\varepsilon_t$. The first term in $v_t(I_t, \varepsilon_t)$ expression shows that the $v_t(I_t, \varepsilon_t)$ decreases as $I_t$ increases because $I_t$ is in the denominator, and the $v_t(I_t, \varepsilon_t)$ decreases as $\varepsilon_t$ decreases due to $\varepsilon_t$ is in the numerator. Thus, the $I_t$ and $\varepsilon_t$ need to be reduced based on the analysis of the first term in Equation (22). However, the third term of Equation (22) requires $I_t$ to remain small because $I_t$ is in the numerator. The second and the third terms of Equation (22) require to remain large because $\varepsilon_t$ is in the denominator. The above analysis indicates that either $I_t = 1$ or $I_t >> 1$ is not an optimal choice as the former results in unnecessary communication overheads, while the latter suffers from a prolonged convergence due to large discrepancies among local models caused by less communication. Both $\varepsilon_t = 1$ and $\varepsilon_t >> 1$ are not optimal choices because the former sends imprecise model parameters causing the prolonged convergence, whereas the latter results in frequent communication. Thus, this paper aims at finding the optimal balance to adjust trade-offs between communication and computation/precision.

## 4. The Proposed FedEAFO Algorithm

The above theoretical analyses illustrate that the error upper bound is ruled by $I_t$ and $\varepsilon_t$. This paper proposes an efficient adaptive federated optimization (FedEAFO) algorithm, which minimizes the learning error via jointly considering two variables including local update and parameter compression. The FedEAFO adaptively adjusts and balances trade-offs between communication and precision/computation. Figure 1 presents an overview of FedEAFO scheme. Mathematically, FedEAFO finds the optimal balance to minimize the error upper bound in Equation (25), which can be denoted as:

$$I_t^*, \varepsilon_t^* = \mathrm{argmin}_{I_t, \varepsilon_t} v_t(I_t, \varepsilon_t) \tag{26}$$

This paper presents Theorem 5 to provide the theoretical analysis for the proof of convexity of $v_t(I_t, \varepsilon_t)$, followed by Theorem 6, which finds optimal solutions to solve the problem of Equation (26).

**Theorem 5.** *Let L, T, and $\eta$ be defined therein. Choose Assumptions (4) $I_t \geq 2$, (5) $\eta^5 \approx 0$, (6) $\left(L^4 T\delta_1/2\alpha(F(w_t) - F_{\inf})\varepsilon_t^4\right) < \infty$, (7) $2\eta^2 LT\delta_1 I_t \geq \alpha N\varepsilon_t^2(F(w_t) - F_{\inf})$, thus the $v_t(I_t, \varepsilon_t)$ is convex.*

**Proof.** Hessian matrix of $v_t(I_t, \varepsilon_t)$ must be positive semidefinite is the condition of $v_t(I_t, \varepsilon_t)$ to be convex. The Hessian matrix of $v_t(I_t, \varepsilon_t)$ is derived as the following:

$$H(v_t(I_t, \varepsilon_t)) = \begin{bmatrix} 2X\frac{\alpha\varepsilon_t}{I_t^3} & -X\frac{\alpha}{I_t^2} - P\frac{\delta_1}{\varepsilon_t^2} \\ -X\frac{\alpha}{I_t^2} - P\frac{\delta_1}{\varepsilon_t^2} & 2Z\frac{\delta_1}{\varepsilon_t^3} + 2P(I_t - 1)\frac{\delta_1}{\varepsilon_t^3} \end{bmatrix} \tag{27}$$

where $X = \frac{2[F(w_0) - F_{\text{inf}}]}{\eta T}$, $Z = \frac{\eta L}{N}$, and $P = \eta^2 L^2$. The positive diagonal elements and the determinant is the condition of the Hessian matrix to be positive semidefinite, which can provide proof of the convexity of $v_t(I_t, \varepsilon_t)$. Obviously, the diagonal elements and the determinant are positive based on Assumptions 4–7. □

**Theorem 6.** *The optimal solutions to minimize error upper bound*$v_t(I_t, \varepsilon_t)$ *can be given by:*

$$I_t = \sqrt{\frac{2\alpha[F(w_t) - F_{\text{inf}}]\varepsilon_t^2}{\eta^3 L^2(\delta_1 + \delta_2 \varepsilon_t)}}, \quad \varepsilon_t = \sqrt{\frac{\delta_1 \eta^2 LT(1 - \eta L(I_t - 1))I_t}{2\alpha[F(w_t) - F_{\text{inf}}]}} \tag{28}$$

With Assumption 8 ($\delta_1 << \delta_2 \varepsilon_t$), Assumption 9 ($\eta L(I_t - 1) << 1$) and Assumption 10 ($F_{\text{inf}} = 0$), the $I_t$ and $\varepsilon_t$ can be approximated by:

$$\frac{I_{t+1}}{I_t} \approx \sqrt{\frac{F(w_{t+1})}{F(w_t)}} \sqrt{\frac{\varepsilon_{t+1}}{\varepsilon_t}} \tag{29}$$

$$\frac{\varepsilon_{t+1}}{\varepsilon_t} \approx \sqrt{\frac{F(w_t)}{F(w_{t+1})}} \sqrt{\frac{I_{t+1}}{I_t}} \tag{30}$$

**Proof.** This can be proven via adopting Assumptions 8–10 and setting the partial derivatives of $v_t(I_t, \varepsilon_t)$ as 0. □

In Equations (29) and (30), the values of $I_t$ and $\varepsilon_t$ are interdependent, which can be decoupled by substituting Equation (29) in Equation (30) as follows:

$$\frac{I_{t+1}}{I_t} = \sqrt[3]{\frac{F(w_{t+1})}{F(w_t)}}, \quad \frac{\varepsilon_{t+1}}{\varepsilon_t} = \sqrt[3]{\frac{F(w_t)}{F(w_{t+1})}} \tag{31}$$

with the initial values $I_0$, $\varepsilon_0$, $F(w_0)$, the Equation (32) can be rewritten as:

$$I_t = \sqrt[3]{\frac{F(w_t)}{F(w_0)}} I_0, \quad \varepsilon_t = \sqrt[3]{\frac{F(w_0)}{F(w_t)}} \varepsilon_0 \tag{32}$$

Equation (32) illustrates that as the loss value $F(w_t)$ decreases during the training process of federated learning, $I_t$ needs to decrease and $\varepsilon_t$ should increase.

Algorithm 1 describes the details of the training process when performing the FedEAFO algorithm. The full flow of Algorithm 1 can be described as the following steps: (i) The server broadcasts the calculated local update coefficient $I_t$, compression coefficient $\varepsilon_t$, and latest weights to the selected devices. The coefficient of local update determines the number of parameter computations in local training, whereas the coefficient of parameter compression determines the rate of parameter compression. (ii) The devices perform multiple local training based on the received local update coefficient $I_t$ and upload the compressed locally aggregated model parameters at the certain ratio of compression determined by the compression coefficient $\varepsilon_t$. (iii) The server aggregates all received compressed model parameters to update the global model. (iv) The server jointly optimizes and adjusts the two variables including the local update coefficient and parameter compression coefficient according to the latest value of loss via Theorem 6. The latest local update coefficient, parameter compression coefficient, and latest weights are broadcast by the server to the selected devices for the next iteration.

**Algorithm 1:** Efficient Adaptive Federated Optimization

```
Server Executes:
the initialization of the global model parameter: $w_0$
for round $t = 1$ to $I$ do
    The server jointly optimizes $I_t$ and $\varepsilon_t$ via Equation (32)
    The server broadcasts $I_t$ and $\varepsilon_t$ to selected devices
    for device $n = 1$ to $N$ do
        Devices receive $I_t$ and $\varepsilon_t$
        for local training $i = 1$ to $I_t$ do
            Devices compute $\ell\left(w^n_{t,i}, \varphi^n_{t,i}\right)$
            Devices update $w^n_{t,i}$ via Equation (2)
        end for
        Devices compute $\ell(w^n_{t,i})$ via Equation (3)
        Devices compress $\ell(w^n_{t,i})$ via Equations (12) and (17)
        Devices upload compressed $\hat{\ell}(w_t)$ to the server
    end for
    The server aggregates compressed updates via Equation (4)
    The server updates the global model via Equation (5)
    The latest global model $w_{t+1}$ are broadcast to the devices
end for
```

## 5. Experimental Evaluation

This section performs experiments to evaluate the performance of FedEAFO in speeding up federated learning and compares it with state-of-the-art algorithms on Fashion-MNIST and CIFAR-10 image classification tasks. In our experiment, this paper implements the algorithm via a centos7 server with Nvidia TITAN RTX GPU.

### *5.1. Experiment Setup*

Training Datasets: Following [31,32], this paper constructs both IID and Non-IID datasets from Fashion-MNIST and CIFAR-10. The Fashon-MNIST dataset consists of 10 categories of images of fashion items including 70,000 28 × 28 images. The CIFAR-10 dataset is comprised of 60,000 examples with 32 × 32 color pixels in 10 classes such as airplanes, birds, and racing cars.

Implementation settings: For settings of training, this paper initializes the iteration time as 200, local epoch as 10, batch size as 32, and learning rate as 0.01. The proposed design is examined on image classification tasks on Fashion-MNIST and CIFAR-10 datasets. For model architectures, this paper adopts the Convolutional Neural Network (CNN) with two convolutional layers with 32 and 64 channels, respectively, and the fully connected and softmax output layer on Fashion-MNIST, and a light-weight ResNet18 on CIFAR-10.

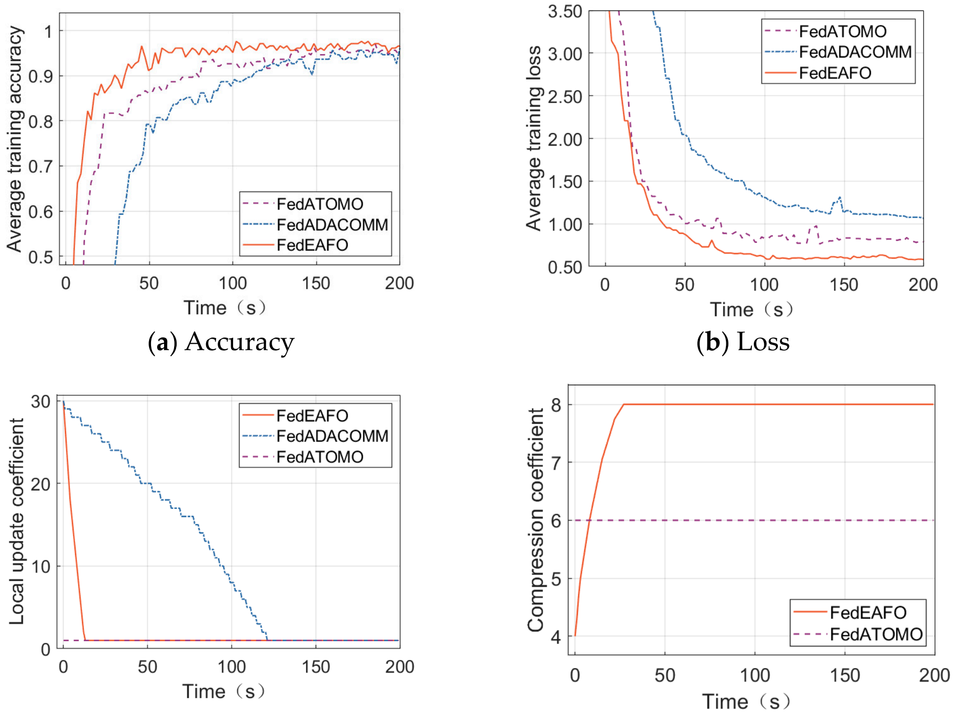
Comparison baselines: This paper compares the proposed FedEAFO with two categories of the state-of-the-art methods: (i) Local update that aims at taking full advantage of the computing capability of devices to reduce the frequent transmissions of model parameters. This paper considers ADACOMM [19], which dynamically adjusts local update coefficients with adaptive strategies as baselines. (ii) Parameter compression that reduces the amount of data to be transmitted via data compression techniques. We adopt ATOMO [25], which exploits the sparsity of parameters to compress the parameters for comparisons.

Evaluation metrics: This paper considers different dimensions to analyze the experiment results accurately: (1) Round completion time (efficiency metric). This paper defines the round completion time as the time needed to be spent in one round of federated learning. The reason why we adopt round completion time instead of iteration/communication round is that unlike traditional machine learning where the time of different iterations are nearly the same; thus, the iteration round reflects the convergence speed and the iteration time of different rounds are different. The key factors for the convergence speed of FL are the round completion time and iteration round. The round completion time varies over the round due to three time-varying delay components: (i) The download delays of global model parameters from the devices with different downlink speeds. (ii) The transmission delays of compressed local model parameters from the devices with different compression ratios and uplink speeds. (iii) The Computation delays of local model update from the devices with different local update coefficients and hardware constraints. (2) Global Learning accuracy (utility metric). This paper considers global learning accuracy as the average accuracy of the global model to evaluate the effectiveness of the FedEAFO in accelerating the FL. (3) Traffic Consumption (communication efficiency metric). Communication efficiency quantifies the amount of data exchanged between IoT devices and the central server during the federated learning process. In the context of IoT, where communication resources are often limited, this metric becomes crucial. A more communication-efficient algorithm reduces the amount of data transmitted, leading to lower bandwidth consumption and energy expenditure for devices. It is essential to assess how the adaptive optimization technique impacts communication efficiency while maintaining or improving other performance aspects.

### *5.2. Results and Analysis*

This paper compares the proposed FedEAFO with two categories of state-of-the-art algorithms including ADACOMM [19] of the local updates category and ATOMO [25] of the parameter compression category. The comparative experiment results in Figure 2 present the values of accuracy, loss, local update coefficient $I_t$, and parameter compression coefficient $\varepsilon_t$ over round completion time for 32 devices on the Fashion-MNIST dataset. Figure 2a illustrates the superiority of the proposed FedEAFO in terms of efficiency and high accuracy compared with ATOMO and ADACOMM, which achieves higher accuracies faster over round completion time. Figure 2b illustrates the training loss of the algorithms. Figure 2c shows the values of $I_t$ over training time, where the local update of ATOMO is assigned as one because it does not perform multiple local updates. Figure 2d shows the values of $\varepsilon_t$ except for ADACOMM because it does not adopt parameter compression, whereas ATAMO adopts parameter compression with a fixed value of $\varepsilon_t$. The reason why the proposed EAFO is more efficient is that the FedEAFO dynamically adjusts the values of $I_t$ and $\varepsilon_t$ during federated learning. For $I_t$, the proposed FedEAFO begins with high values to enable more local updates to be performed, but ends with lower values and high accuracy. For $\varepsilon_t$, the proposed FedEAFO begins with low values due to coarse parameters and it can still contribute to improving the accuracy of the model effectively with low accuracy. The proposed FedEAFO ends with high values providing finer-grained parameters and high accuracy in Figure 2d. The communication/computation trade-off and communication/precision trade-off can be viewed as two exploration–exploitation trade-offs. Both adjustments of the local update coefficient and the parameter compression coefficient can be used to resolve the corresponding exploration–exploitation trade-off. The proposed FedEAFO starts with high exploration via large $I_t$ and small $\varepsilon_t$, and FedEAFO ends with pure exploitation via small $I_t$ and large $\varepsilon_t$, which means the steady state is reached. The data rates of both uplink and downlink are set to 100 Kbps. The dataset is IID. Note that $I_t \in \{1, \cdots, 40\}$ and $\varepsilon_t \in [4, 8]$.

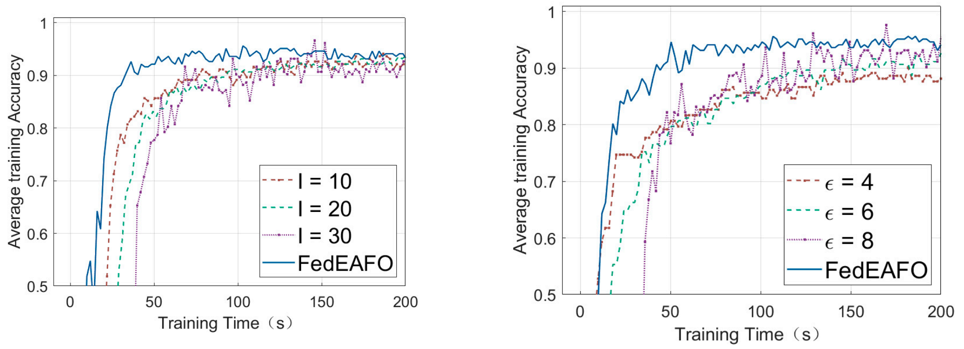
The second simulation presents the comparison of the performance of our proposed FedEAFO algorithm with the fixed local update coefficient schemes and the fixed parameter compression coefficient schemes. Figure 3a illustrates the training accuracy over time under different values of the local updates coefficient $I_t$ set to 10, 20, and 30. Figure 3b illustrates the training accuracy over time under different values of parameter compression coefficient $\varepsilon_t$ set to 4, 6, and 8. As shown in Figure 3, our proposed FedEAFO algorithm converges faster and smoother than the fixed local update and parameter compression schemes, which verifies the effectiveness of the proposed algorithm.



(**a**) Accuracy (**b**) Loss

(**c**) Local update coefficient $I_t$ (**d**) Parameter compression coefficient $\varepsilon_t$

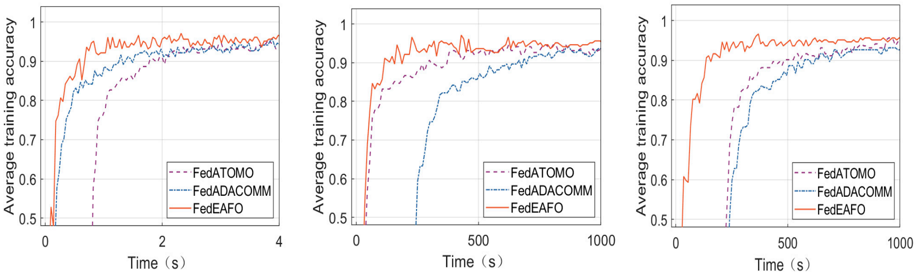
**Figure 2.** The values of training accuracy, training loss, $I_t$, and $\varepsilon_t$ over round completion time.



(**a**) Local update coefficient $I_t$ (**b**) Parameter compression coefficient $\varepsilon_t$

**Figure 3.** The values of training accuracy with different values of local updates coefficient $I_t$ and parameter compression coefficient $\varepsilon_t$ over round completion time.

This paper evaluates performance with different uplink/downlink data rates of the proposed FedEAFO compared with ADACOMM and ATOMO. The results in Figure 4a–c present the accuracy for 10/10 Mbps, 10/100 Kbps, and 100/10 Kbps uplink/downlink data rates, respectively. This paper considers high uplink/downlink data rates as 10 Mbps in Figure 4a, where parameter compression is less important than computation with relatively high data rates. In this case, computation has become a bigger bottleneck than communication for scaling up federated learning. Figure 4a presents that the ADACOMM outperforms the ATOMO, which adopts parameter compression in 10/10 Mbps data rates due to multiple local updates, and the proposed FedEAFO outperforms the ADACOMM because of benefits from the parameter compression. In Figure 4b considering the 10/100 Kbps data rates where the uplink communication has become the bottleneck, the ATOMO wins the race with ADACOMM because it compresses the parameters in the uplink. Additionally, the proposed FedEAFO outperforms the ATOMO due to the benefit from multiple local updates. On the contrary, in Figure 4c, considering the 100/10 Kbps data rates, the downlink communication has become the bottleneck. Thus, the performance of ATOMO is similar to the performance of ADACOMM despite parameter compression in the uplink.



(**a**) 10/10 Mbps (**b**) 10/100 Kbps (**c**) 100/10 Kbps

**Figure 4.** The values of accuracy with different uplink/downlink data rates over time.

Figure 5 shows the network traffic consumption of different algorithms when they achieve different target accuracy. We can find that the network traffic consumption of all algorithms for all datasets increases with the increasing accuracy. However, FedEAFO can always consume the minimum network traffic, which demonstrates the communication efficiency of FedEAFO. In addition, the algorithms with model compression can save much more network traffic than the algorithms without model compression.

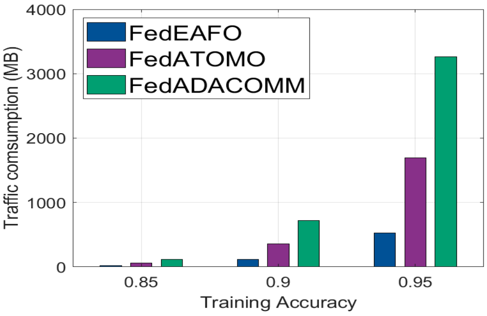


**Figure 5.** The network traffic consumption of algorithms when achieving the target accuracy.

This paper further evaluates the performance of the FedEAFO and the baselines on non-IID data. Concretely, we create non-IID data of the Fashon-MNIST dataset with different partition schemes. Each device has p of a unique class in 10 classes and other devices shared the remaining samples uniformly. Figure 6 shows the impacts of non-IID levels on the test accuracy of different algorithms, where the value of p is set to 0.1, 0.2, 0.4, 0.6, and 0.8. The result indicates that all algorithms suffer from a loss of accuracy with the increase in the non-IID level, while FedEAFO is more robust than other baselines.

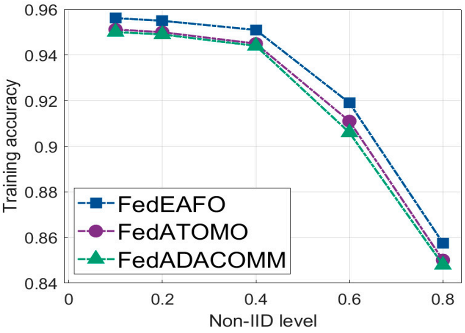


**Figure 6.** The values of accuracy of the algorithms on non-IID data.

## 6. Conclusions

FL has received considerable attention in IoT because of its capability of enabling devices to train machine learning models collaboratively via sharing local model parameters rather than privacy-sensitive data. However, FL-IoT implementation suffers from inefficiency and limited convergence performance in resource-constrained IoT environments. We propose a novel efficient adaptive federated optimization algorithm to improve the efficiency of FL training. The proposed FedEAFO minimizes the learning error by the joint consideration of two variables consisting of the coefficient of local update and parameter compression. Additionally, the proposed FedEAFO enables federated learning to adaptively and dynamically adjust the trade-offs among communication, computation, and model precision. The evaluation results demonstrate the superiority of the FedEAFO in terms of efficiency and accuracy. In future work, we will study interactions of FedEAFO with message-passing protocols that enable distributed optimization in unstable environments.

**Author Contributions:** Conceptualization, Z.C. and E.W.; methodology, Z.C. and E.W.; software, Z.C. and X.Y.; validation, Z.C. and H.C.; formal analysis, Z.C.; investigation, Z.C. and X.Y.; resources, H.C.; data curation, Z.C. and X.Y.; writing—original draft preparation, Z.C.; writing—review and editing, E.W. and H.C.; visualization, Z.C. and X.Y.; supervision, H.C.; project administration, Z.C. and H.C.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant No. 62171049 and the National Key Research and Development Program of China under Grant No. 2020YFB1807805.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Haddad Pajouh, H.; Dehghantanha, A.; Parizi, R.M.; Aledhari, M.; Karimipour, H. A survey on internet of things security: Requirements, challenges, and solutions. *IEEE Internet Things J.* **2021**, *14*, 100129.
2. Liu, Y.; Wang, J.; Li, J.; Niu, S.; Song, H. Machine Learning for the Detection and Identification of Internet of Things Devices: A Survey. *IEEE Internet Things J.* **2021**, *9*, 298–320.
3. Ghosh, A.M.; Grolinger, K. Edge-cloud computing for Internet of Things data analytics: Embedding intelligence in the edge with deep learning. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2191–2200.
4. Khan, L.U.; Saad, W.; Han, Z.; Hossain, E.; Hong, C.S. Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1759–1799.
5. Zhang, T.; Gao, L.; He, C.; Zhang, M.; Krishnamachari, B.; Avestimehr, A.S. Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities. *IEEE Internet Things Mag.* **2022**, *5*, 24–29.
6. Kandati; Reddy, D.; Gadekallu, T.R. Genetic Clustered Federated Learning for COVID-19 Detection. *Electronics* **2022**, *11*, 2714.
7. Posner, J.; Tseng, L.; Aloqaily, M.; Jararweh, Y. Federated Learning in Vehicular Networks: Opportunities and Solutions. *IEEE Netw.* **2021**, *35*, 152–159.
8. Mills, J.; Hu, J.; Min, G. Client-side optimization strategies for communication-efficient federated learning. *IEEE Commun. Mag.* **2022**, *60*, 60–66.
9. Nguyen, C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Vincent Poor, H. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1622–1658.
10. Sun, H.; Li, S.; Yu, F.R.; Qi, Q.; Wang, J.; Liao, J. Toward Communication-Efficient Federated Learning in the Internet of Things with Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 11053–11067.
11. Ullo, S.L.; Sinha, G.R. Advances in IoT and smart sensors for remote sensing and agriculture applications. *Remote Sens.* **2021**, *13*, 2585.
12. Qin, Z.; Li, G.Y.; Ye, H. Federated learning and wireless communications. *IEEE Wirel. Commun.* **2021**, *28*, 134–140.
13. Nguyen, H.T.; Sehwag, V.; Hosseinalipour, S.; Brinton, C.G.; Chiang, M.; Vincent Poor, H. Fast-Convergent Federated Learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 201–218.
14. Shah, S.M.; Lau, V.K.N. Model Compression for Communication Efficient Federated Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *3*, 1–15.

15. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. In Proceedings of the International Conference on Artificial Intelligence and Statistics (PMLR), Virtual Conference, 26–28 August 2020; pp. 2021–2031.
16. Mitra, A.; Jaafar, R.; Pappas, G.J.; Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14606–14619.
17. Zheng, S.; Meng, Q.; Wang, T.; Chen, W.; Yu, N.; Ma, Z.M.; Liu, T.Y. Asynchronous Stochastic Gradient Descent with Delay Compensation. In Proceedings of the International Conference on Artificial Intelligence and Statistics (PMLR), Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 4120–4129.
18. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
19. Wang, J.; Joshi, G. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *Proc. Mach. Learn. Syst.* **2019**, *1*, 212–229.
20. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221.
21. Wen, W.; Xu, C.; Yan, F.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *Adv. Neural Inf. Process. Syst.* **2017**, *4*, 30–43.
22. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 47–60.
23. Seide, F.; Fu, H.; Droppo, J.; Li, G.; Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014; pp. 20–27.
24. Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, B. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 40–48.
25. Wang, H.; Sievert, S.; Liu, S.; Charles, Z.; Papailiopoulos, D.; Wright, S. Atomo: Communication-efficient learning via atomic sparsification. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 10–24.
26. Xu, J.; Du, W.; Jin, Y.; He, W.; Cheng, R. Ternary Compression for Communication-Efficient Federated Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1162–1176. [CrossRef] [PubMed]
27. Haijian, S.; Ma, X.; Hu, R.Q. Adaptive federated learning with gradient compression in uplink NOMA. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16325–16329.
28. Zha, Z.; Yuan, X.; Wen, B.; Zhang, J.; Zhou, J.; Zhu, C. Image Restoration Using Joint Patch-Group-Based Sparse Representation. *IEEE Trans. Image Process.* **2020**, *29*, 7735–7750.
29. Sattler, F.; Müller, K.R.; Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3710–3722.
30. Ananduta, W.; Nedich, A.; Ocampo-Martinez, C. Distributed Augmented Lagrangian Method for Link-Based Resource Sharing Problems of Multi-Agent Systems. *IEEE Trans. Autom. Control.* **2021**, *8*, 10–23.
31. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
32. Zheng, Y.; Lai, S.; Liu, Y.; Yuan, X.; Yi, X.; Wang, C. Aggregation service for federated learning: An efficient, secure, and more resilient realization. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 988–1001.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.