Semi-Synchronous Hierarchies and Credibility Management for Robust Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In synchronous Federated Learning (FL) architectures, three major operational challenges persistently emerge: the stragglers' effect, which significantly impedes aggregate computation efficiency; network congestion that compromises communication efficacy; and the vulnerability to poisoning attacks that endangers model integrity. In response to these critical issues, this paper introduces a novel FL framework named Clustered Semi-synchronous Hierarchical Federated Learning (CSS-HFL). It utilizes edge servers to synchronously train models with their respective clustered clients, which are clustered based on their computational capabilities and network conditions. As for the cloud server, a semi-synchronous training scheme is adopted to defend cloud aggregation against adversarial attacks. To bolster the robustness of CSS-HFL against poisoning attacks, we propose a new algorithm, Fusion Credibility (FusCred), which leverages a credibility scoring system and a small clean dataset on the cloud server to filter out potentially malicious updates. We provide a theoretical convergence guarantee and efficiency analysis for CSS-HFL and extensive experiments on MNIST, FMNIST, and CIFAR-10 datasets under various attack scenarios to demonstrate its effectiveness. Our results show that CSS-HFL with FusCred significantly enhances model accuracy and robustness compared to state-of-the-art FL algorithms. For example, on the Non-IID CIFAR-10 dataset, FusCred showcased an improvement in accuracy of 17.7%, 17.8% and 10.4%, respectively, over the state-of-the-art algorithm when exposed to three types of model poisoning attacks in experiments with 40% attackers.

1 Introduction

Federated Learning (FL) [21] is a decentralized machine learning paradigm that enables end devices to train models locally and share only the parameter updates, thereby alleviating concerns regarding data privacy [30] and legal compliance [41]. A typical FL framework, such as federated averaging (FedAvg) [30], trains a global model by iteratively aggregating local updates from many clients synchronously. This framework is widely adopted in various applications, yet it faces several challenges that hinder its practical implementation.

Stragglers effect: firstly, because of the computational capacity and the network constraints, slow clients (*i.e.*, stragglers) require more time to train local models. This makes normal clients waste a great deal of time to wait stragglers, which is called *stragglers effect* [4, 25, 33]. To mitigate the impact of stragglers, [30, 55, 20] aggregate local updates only from a delicately selected subset of clients. Nevertheless, due to the Non-IID (not identically and independently distributed) distribution, the absence of excluding clients can greatly reduce the global model performance. Additionally, Xie *et. al.* [51] propose Asynchronous FL framework, where the server can aggregate with the first received local update without waiting for the lagging devices.

Network congestion: second, in practical applications, the network condition becomes a bottleneck when a significant amount of end devices collaborate to train the global model under Synchronous FL framework [11]. Liu *et. al.* [26] introduced Hierarchical Federated Learning (HFL) to relieve the congestion on the backbone network. A client-edge-cloud HFL architecture can greatly decrease the model training time and the energy consumption of the clients compared to traditional FL [27].

Poisoning attacks: third, due to its special framework, traditional FL faces some severe security problems if some clients are malicious. For instance, malicious clients could upload modified parameters (*i.e.*, model poisoning) [39, 56] or dirty training data (*i.e.*, data poisoning) [17, 2]. The global model performance would be degraded even though only one single malicious client in traditional FL [12]. Actually, some robust algorithms [13, 31, 43, 58] are proposed to protect global models against adversarial attacks. Nevertheless, all of them are based on Synchronous FL which means they cannot fit perfectly with Asynchronous FL.

However, the existing FL frameworks and FL algorithms can only address part of drawbacks of typical FL framework. The summarization of the limitation is shown in Table 1. In the real-world FL system implementation, we should comprehensively deal with stragglers effect, network constraints, and malicious attacks. An urgent need thus arises to propose a new FL framework to simultaneously address the above problems.

Table 1: The limitations of existing FL frameworks.

Limitations	Framework		
	Synchronous FL	Asynchronous FL	HFL
Stragglers effect	✕	✓	✕
Network congestion	✕	✓	✓
Poisoning attacks	✓	✕	✓

In this paper, we introduce a novel FL framework termed **C**lustered **S**emi-**s**ynchronous **H**ierarchical **F**ederated **L**earning (CSS-HFL). Within this framework, clients are organized into distinct clusters according to their computational capacities and network conditions to mitigate the stragglers effect. Then, the edge servers engage in training with their respective clients utilizing Synchronous FL methods (*e.g.* Fed-Credit [10], Median [58], GeoMed [13]). Subsequently, the cloud server strategically determines the timing for aggregating edge models, ensuring that each edge server has recently completed its aggregation. Noticeably there is no requirement for uniform epochs across all edge servers. Either the Semi-synchronous FL in the cloud layer or the synchronous FL in the edge layer can be accessed to apply robust Synchronous FL algorithms to resist malicious attacks. Additionally, under CSS-HFL framework, we propose a robust algorithm named **Fus**ion **Cred**ibility (FusCred). This algorithm leverages Fed-Credit [10] in the edge layer and maintains a small clean dataset on the cloud server. The updated models of each edge cluster are assigned a credit score by comparing it with the model trained on the cloud dataset. Subsequently, only the top k edge parameters are aggregated with the cloud parameter. We provide both the convergence guarantee and efficiency analysis of CSS-HFL, followed by the efficiency simulation and comprehensive experiments conducted on MNIST, FMNIST, and CIFAR-10 datasets. Our experiments encompassed various attack types, ratios of malicious clients, and dataset distributions. The empirical findings unequivocally showcase that our proposed *FusCred* not only preserves high test accuracies but also exhibits exceptional robustness against adversarial attacks. Our main contributions can be summarized as follows:

- To the best of our knowledge, this is the first work to explore both the robustness and efficiency of HFL. By leveraging semi-synchronized aggregation and adaptive clustering, CSS-HFL framework is proposed to comprehensively address the limitations of existing FL frameworks, when dealing with stragglers effect, network congestion and poisoning attacks.
- We derive the efficiency analysis of CSS-HFL in comparison with famous FL frameworks and prove the convergence guarantee in CSS-HFL framework. We also conducted an efficiency simulation to show that our CSS-HFL can significantly enhance efficiency by involving few edge servers.
- Within the CSS-HFL, we design a novel defense algorithm named *FusCred*. FusCred utilizes Fed-Credit on edge servers and maintains a small clean dataset on the cloud server to assign credit scores to edge model updates. This ensures that only top k edge parameters are aggregated with the cloud parameter to mitigate attack effects passed over the edge. Notably,

FusCred demonstrates superior performance across various scenarios, outperforming state-of-the-art algorithms.

- The extensive comparative experiments between *FusCred* and various prior algorithms validate its effectiveness. Specifically, on the Non-IID CIFAR-10 dataset, our algorithm exhibited performance enhancements of 17.7%, 17.8%, and 10.4%, respectively, in comparison to the state-of-the-art algorithm when facing three types of model poisoning attacks in experiments involving 40% attackers.

2 Observation And Threat Model

In this section, we first briefly introduce the observation of extant FL frameworks. Then we will describe the threat model of Hierarchical Federated Learning (HFL) system considered in this work.

2.1 Observation

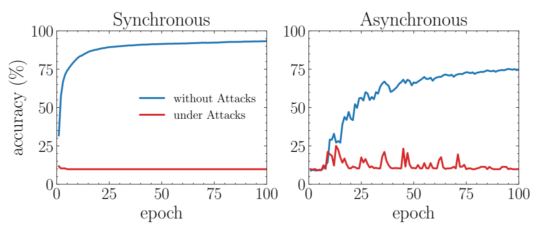


Figure 1: The accuracy of Synchronous FL and Asynchronous FL under no attacks or 20% attacks in Three-tier HFL on the Non-IID Mnist dataset.

In Figure 1, we briefly investigate the resilience to malicious attacks in HFL, using both Synchronous FL and Asynchronous FL. In the experiment, we set the attack as Sign-Flip (SF), in which the malicious clients upload the local updates by flipping the sign of each number. We assess the accuracy of Synchronous FedAvg and Asynchronous FedAsync in both attacks-free and 20% attacks scenarios under the Three-tier FL framework, employing the Non-IID MNIST dataset. A glance at the Figure 1 reveals the same trends between attacks-free and 20% attacks. All of them see a plunge in accuracy compared to no attacks scenario, which is not acceptable in practical application, when facing attacks. More comparisons about existing FL framework can be found in Appendix A.

2.2 Threat Model

In this section, we present a comprehensive threat model for poisoning attacks within the context of CSS-HFL.

Poisoning Attacker's Goal: Aligned with numerous prior studies on poisoning attacks [15, 50], the primary objective of the poisoning attacker in CSS-HFL is to deliberately manipulate the local training process. Their ultimate aim is to compromise the aggregation process of the global model.

Types of Poisoning Attacks: The strategies employed in our attacks align with those detailed in the work of Fed-Credit [10], encompassing data poisoning attacks [17, 2] and model poisoning attacks [39, 56].

Poisoning Attacker's Knowledge: The poisoning attackers are indeed components of CSS-HFL, possessing specific knowledge within the framework. As clients, they have access to important information including the training data, model structure, learning algorithms, and the global model. This knowledge equips them to conduct their attacks effectively within the system.

Poisoning Attacker's Assumptions: *1)* Poisoning attackers are capable of collaborating with one another, thereby enabling them to coordinate and execute the same type of attack collectively. *2)* Poisoning attackers are constrained to conducting their operations solely on the client side, implying that both the edge and cloud components are deemed trustworthy. *3)* It is assumed that the number of malicious clients does not surpass half of the total [15]. *4)* We assume that the network communication in CSS-HFL is reliable.

3 Clustered Semi-synchronous HFL Framework (CSS-HFL)

In this section, we introduce the CSS-HFL framework (Figure 2). CSS-HFL mainly addresses three goals: *1)* **To mitigate the waiting time of clients**. *2)* **To relieve the network congestion**. *3)* **To provide an interface for robust FL algorithms**.

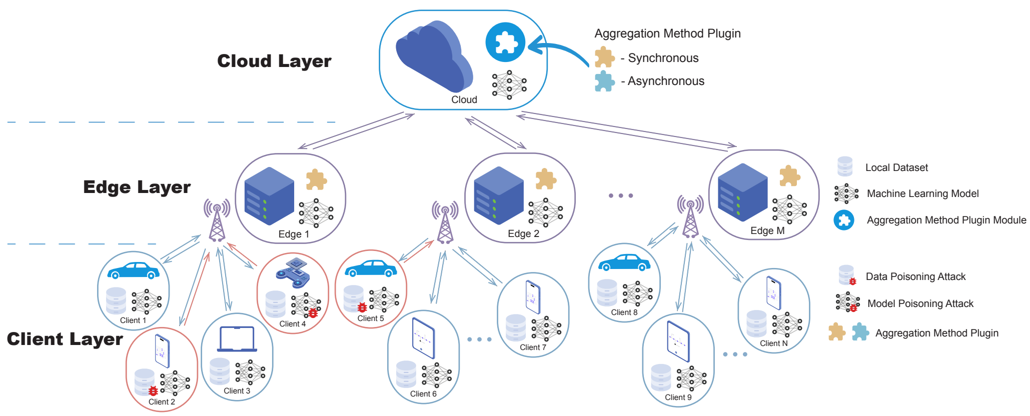


Figure 2: The CSS-HFL Framework.

We will begin by describing the components of the CCS-HFL framework. Overall, we adopt hierarchical federated learning to enhance communication efficiency and release network congestion [26]. *1)* **At the client layer**, since we have N edge servers, we cluster all the clients into N clusters, each belonging to one of N edges. The cluster criterion is based on the computation capacity and network condition [34, 45, 7]. Our objective in this stage is to reduce waiting time of clients within respective clusters. *2)* **At the edge layer**, each edge server conducts synchronous federated learning with its participating clients. During the edge aggregation stage, the edge server can select a robust aggregation algorithm to protect the edge model from the attacks of malicious clients. Training at the edge server resembles the traditional federated learning. *3)* **At the cloud server**, the cloud can choose either a different or the same secure aggregation algorithm used by the edges. It is significant for cloud server to carefully determine the timing, when each edge server has recently completed an edge aggregation, (note: edge servers are not mandated to go through the same number of communication rounds with respective clients), to aggregate edge models. It is noteworthy that the semi-synchronous aggregation scheme provides interfaces to different robustness algorithms, where the users have the flexibility to choose the appropriate algorithm. The overall algorithm of our proposed CSS-HFL framework can be found in Algorithm 1.

3.1 Fusion Credibility (FusCred) in CSS-HFL

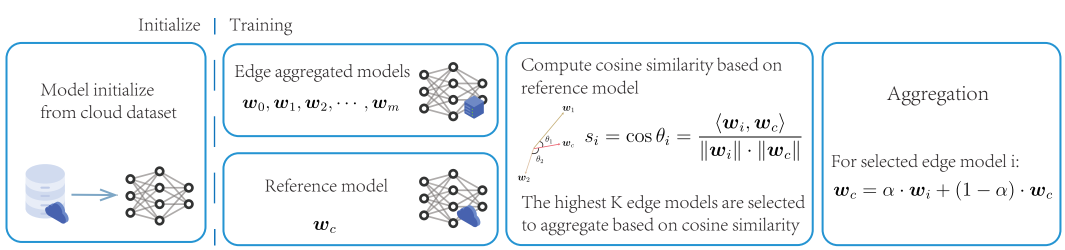


Figure 3: Cloud Aggregation Algorithm

Under the CSS-HFL framework, we propose a more robust aggregation algorithm named *FusCred*, which comprises both edge aggregation method and cloud aggregation method. This algorithm can maintain the efficiency of CSS-HFL while offering a high level of resilience against attacks.

In a macroscopic view, we use a non-discriminatory aggregation algorithm at each edge and a discriminatory one in the cloud. For the edges, with their narrow perspective limited to the few client models they can observe, this non-discriminatory aggregation preserves data diversity and some resistance to attack. The cloud, with access to all edge gradients and cloud dataset references, uses a

Algorithm 1: CSS-HFL Training Process

Input : n clients with local training datasets, $C_1, C_2, C_3, \ldots, C_n$; N edge servers, $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \ldots, \mathcal{E}_N$; learning rate lr; batch size B; number of local training iterations E; number of cloud communication rounds R.

Output : Convergent cloud model $\boldsymbol{w}$.

The cloud server utilizes the Balanced Clustering Algorithm[45] to form N clusters by grouping clients based on their computation capacities and network conditions. The number of clients in the λ^{th} group is denoted as N_λ.

The cloud server assigns an edge server to each cluster and defines the communication rounds for each edge server denoted as $E_1, E_2, E_3, \ldots, E_N$.

Cloud server excutes:

    $\boldsymbol{w} \leftarrow$ pre-train model.
    for i_R *in* R **do**
        The cloud sends cloud model $\boldsymbol{w}$ to $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \ldots, \mathcal{E}_N$.
        Receive edge models $\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots \boldsymbol{w_N}$.
        $\boldsymbol{w} \leftarrow$ **Cloud aggregation**($\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots \boldsymbol{w_N}$).
    end

Edge server excutes:

    for $\lambda = 1$ *to* N ***parallel*** **do**
        Receive cloud model $\boldsymbol{w}$.
        $\boldsymbol{w_\lambda} \leftarrow \boldsymbol{w}$.
        `// Semi-synchronous lies in varied edge communication rounds` E_λ.
        for i_λ *in* E_λ **do**
            for $k = 1$ *to* N_λ ***parallel*** **do**
                /* Client executes */
                $\boldsymbol{w_{\lambda,k}} \leftarrow \boldsymbol{w_\lambda}$.
                for $i_E = 1$ *to* E **do**
                    $\boldsymbol{w_{\lambda,k}} \leftarrow$ **SGD**($\boldsymbol{w_{\lambda,k}}$, local dataset).
                end
                The k^{th} client in the λ^{th} edge server uploads its local model $\boldsymbol{w_{\lambda,k}}$ to the λ^{th} edge server.
            end
            $\boldsymbol{w_\lambda} \leftarrow$ **Edge aggregation**($\boldsymbol{w_{\lambda,1}}, \boldsymbol{w_{\lambda,2}}, \boldsymbol{w_{\lambda,3}}, \ldots, \boldsymbol{w_{\lambda,k}}$).
        end
        The λ^{th} edge server uploads its edge model $\boldsymbol{w_\lambda}$ to the cloud server.
    end

discriminatory aggregation algorithm to filter out compromised edges. Together, these two methods allow the global model to converge stably.

For edge aggregation algorithm, we employ Fed-Credit [10], the recently proposed robust algorithm that currently works well at Two-tier FL. Next, we will delve into cloud aggregation algorithm in detail.

As illustrated in Figure 3, the Cloud Aggregation Algorithm is primarily divided into two sections. In the Initialization phase, the cloud-side dataset is utilized to train the global model for a specified number of epochs. In the Training phase, the cloud sends the global model to each edge, where it is used to train the edge-side models, $\boldsymbol{w_i}$. Concurrently, the cloud trains for a specified number of epochs on the cloud dataset based on the global model to obtain a reference model, denoted by $\boldsymbol{w_c}$. Thereafter, the cloud calculates the credibility of each edge's updated model according to the following equation: $s_i = \cos\theta_i = \frac{\langle \boldsymbol{w_i}, \boldsymbol{w_c} \rangle}{\|\boldsymbol{w_i}\| \cdot \|\boldsymbol{w_c}\|}$. Subsequently, the most credible edges are selected for aggregation. Finally, the updates from the selected K edges are aggregated according to the following equation: $\boldsymbol{w_c} = \alpha \cdot \boldsymbol{w_j} + (1 - \alpha) \cdot \boldsymbol{w_c}$. It should be noted that the aggregation order is randomised. The training process is repeated until the global model converges or reaches a preset number of epochs. The pseudo-code for this algorithm can be found in Algorithm 2.

Algorithm 2: Cloud Aggregation Method

Input : Cloud dataset, cloud model $\boldsymbol{w}$, N edge aggregated models $\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_N}$, aggregate proportion p, initial epochs E_i, cloud reference epochs E_r, cloud communications rounds R.

Output : Convergent cloud model $\boldsymbol{w}$.

for $epoch = 1$ *to* E_i **do**
  $\boldsymbol{w} \leftarrow$ **SGD**($\boldsymbol{w}$, cloud dataset).
end
for r *in* R **do**
  The cloud sends cloud model $\boldsymbol{w}$ to all edge.
  Each edge aggregates their clients' updates and returns new edge models $\boldsymbol{w_1}$, $\boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_N}$ to the cloud.
  $\boldsymbol{w_c} = \boldsymbol{w}$.
  for $epoch = 1$ *to* E_r **do**
    $\boldsymbol{w_c} \leftarrow$ **SGD**($\boldsymbol{w_c}$, cloud dataset).
  end
  for $i = 1$ *to* N **do**
    Compute cosine similarity $s_i = \cos\theta_i = \frac{\langle \boldsymbol{w_i}, \boldsymbol{w_c} \rangle}{\|\boldsymbol{w_i}\| \cdot \|\boldsymbol{w_c}\|}$.
  end
  for $j = 1$ *to* N **do**
    Compute the $rank$ of s_j in $s_1, s_2, s_3, \ldots, s_N$.
    if $rank \geq p \times N$ **then**
      $\boldsymbol{w_c} \leftarrow \alpha \cdot \boldsymbol{w_j} + (1 - \alpha) \cdot \boldsymbol{w_c}$.
    end
  end
  $\boldsymbol{w} \leftarrow \boldsymbol{w_c}$.
end
return *Cloud model* $\boldsymbol{w}$.

3.2 Efficiency Analysis

We introduce a metric called *Average Waiting Time* (AWT), which aims to calculate the average waiting time of all end devices, to assess the efficiency of the framework. The less value of AWT indicates higher efficiency of framework. We neglect the time taken for edge aggregation, cloud aggregation, and communication between the edge server and the cloud server. AWT calculates the average waiting time across all end devices during one cloud aggregation.

Let $t_{\lambda,k}$ denotes the total training time, including local training and communication overhead, of k^{th} client in the λ^{th} edge. We designate $T = \max\{t_{\lambda,k}\}$ for all $1 \leq \lambda \leq N$ and $1 \leq k \leq N_\lambda$ as the slowest client among all n clients, and $T_\lambda = \max\{T_{\lambda,k}\}$ for all $1 \leq k \leq N_\lambda$ as the slowest client in the λ^{th} edge. Δ_λ represents the idle time between training of λ^{th} edge server in FedAT and HiFlash.

Table 2: Average waiting time (AWT) comparisons of various FL frameworks.

Framework	AWT Expression	Framework	AWT Expression
FedAsync [51]	0	FedSync [26]	$\frac{1}{n}\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda}(T - t_{\lambda,k})$
FedAT [9], HiFlash [46]	$\frac{1}{n}\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda}(T_\lambda - t_{\lambda,k} + \Delta_\lambda)$	CSS-HFL	$\frac{1}{n}\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda}(T_\lambda - t_{\lambda,k})$

The AWT comparison among various frameworks is shown in Table 2. Since $T_\lambda \leq T$ and $\Delta \geq 0$, we have $AWT_{\text{Asynchronous}} \leq AWT_{\text{CSS}} \leq AWT_{\text{Synchronous}}$ and $AWT_{\text{CSS}} \leq AWT_{\text{FedAT}}, AWT_{\text{HiFlash}}$. It's important to note that as N approaches 1, CSS-HFL behaves like Synchronous FL, while as N approaches n, it behaves like Asynchronous FL. In summary, the efficiency of CSS-HFL lies between Synchronous FL and Asynchronous FL, with the choice of N playing a significant role, and is higher than FedAT and HiFlash. Experimental results about efficiency can be found in Appendix F.1.

4 Evaluation

4.1 Experimental Setup

In our experiments, we evaluate CSS-HFL with **FedAvg** [30], **A-Krum** [43], **Median** [58], **GeoMed** [13], and on MNIST, Fashion-MNIST, and CIFAR-10 datasets under both IID and Non-IID settings. We utilized Dirichlet distribution to model Non-IID distribution [59]. For each scenario, we take the average of results of three seeds (2023, 2024, 3047). Experiments are conducted on a server comprises the AMD EPYC 7742 64-Core Processor and the NVIDIA Tesla A100 40G computing accelerator.

Datasets and Networks. A detailed description of the dataset can be found in the Appendix F.2. For MNIST, we adopt a Multi-Layer Perceptron (MLP) network with two hidden layers and one output layer to train the model. For Fashion-MNIST, a 7-layer LeNet [22] with convolutional layers is employed for model training. For CIFAR-10, we opt for a lightweight model Compact Convolutional Transformers (CCT) [18] due to its compact design and effectiveness, which holds promise for mitigating the resource constraints in onboard FL end devices.

Attacks. Our model poisoning attacks are implemented in three distinct forms: Constant Parameter (CP), where all model parameters remain identical; Normal Parameter (NP), which generates model parameters following a normal distribution; and Sign-Flip Parameter (SF), producing a model with parameters opposite to those obtained during training. As for the data poisoning attack, we choose the based on pairwise (PW) and symmetric (SM) matrices to flip the training labels. Additionally, 20%, 30%, and 40% attack ratios are adopted to evaluate the resilience of algorithms and model the attackers distribution by Dirichlet ($G \sim DP(\alpha, G_0)$) with three αs (0.2, 0.5, 0.8). A larger α corresponds to a distribution closer to uniform, while a smaller α indicates a more concentrated distribution.

Evaluation metric. To assess the performance of the multiple defense algorithms under CSS-HFL framework, as many prior studies [8, 54], we employ *accuracy* as a key criterion. Higher accuracy signifies better defense.

4.2 Results And Analysis

We demonstrate the partial accuracy results of our experiments in Table 3 and Table 4. Table 3 presents the average accuracy of different attack types under varying attack ratios. Table 4 displays the accuracy of various attack types in the presence of 30% attacks.

Impact of Ratio of Malicious Clients. Firstly, as illustrated in Table 3, A-Krum exhibits lower accuracy compared to other methods with the absence of attacks. This distinction is particularly obvious when the dataset distribution is Non-IID. The other methods achieve relatively higher accuracy. This phenomenon might be because A-Krum tend to heavily rely on few local updates to update the global update, which lead to the global model cannot fitting the overall dataset well. A discernible pattern emerges in the results: an increase in the ratio of malicious clients corresponds to a noticeable decline in accuracy and a growth in bias. Our *FusCred* outperforms other approaches. Notably, the *FusCred* consistently achieves higher accuracy levels and maintains fewer instances of extreme variability. This reinforces the assertion that *FusCred* adeptly preserves both accuracy and stability, even in the presence of an escalating ratio of adversarial entities.

Impact of Attack Types. Table 4 illustrates the varying effectiveness of different aggregation approaches against a range of attack techniques on the FMNIST dataset with diverse distributions. Notably, distinct patterns emerge, particularly in scenarios with high ratios of attackers. For example, GeoMed performs well under 20% and 30% ratio attacks, similar to *FusCred*, but experiences a significant decline when facing 40% ratio attacks. Another finding is the compared methods do not exhibit comprehensive robustness across various attack types. For instance, both FedAvg and Median perform poorly when facing SF attacks compared to other attack types, while A-Krum only demonstrates better tolerance for SF attacks. Furthermore, our *FusCred* consistently perform well, effectively mitigating all types of attacks with higher accuracy compared to alternative methods considered.

Table 3: Comparing accuracies (%) under various attack ratios. Gold, silver, and bronze respectively denote the top three winners.

Dataset	Attack ratio	FedAvg	Median	GeoMed	A-Krum	FusCred
IID MNIST	0%	98.35±0.04	98.15±0.03	98.31±0.02	93.55±0.24	97.49±0.13
	20%	95.64±2.85	96.35±1.33	97.08±0.86	75.80±32.98	97.41±0.09
	30%	89.14±12.67	93.95±3.56	96.44±1.21	75.92±33.11	97.47±0.15
	40%	82.15±16.81	88.02±10.17	90.92±5.78	63.11±34.34	97.40±0.16
Non-IID MNIST	0%	98.20±0.01	97.98±0.01	98.11±0.01	74.26±0.60	96.86±0.11
	20%	92.14±8.29	95.08±2.94	97.05±0.56	61.78±15.97	96.86±0.28
	30%	89.25±8.52	89.45±9.78	94.60±2.79	49.30±32.25	96.06±1.26
	40%	78.55±16.12	76.33±18.19	75.24±17.68	47.79±31.57	95.81±1.50
IID FMNIST	0%	90.07±0.07	89.70±0.08	90.20±0.06	84.33±0.46	89.08±0.17
	20%	86.27±2.40	86.82±2.64	88.98±0.47	77.94±14.74	88.55±0.29
	30%	79.05±11.18	79.83±14.84	87.97±0.58	73.11±24.00	88.76±0.21
	40%	72.47±16.75	69.66±20.52	79.03±9.32	52.64±29.84	88.53±0.40
Non-IID FMNIST	0%	89.57±0.09	89.18±0.03	89.75±0.10	75.94±0.06	87.04±0.48
	20%	81.51±6.95	82.31±6.76	88.17±0.59	73.47±7.05	86.92±0.84
	30%	73.69±12.96	73.38±17.78	85.87±2.46	56.97±20.62	86.91±0.75
	40%	63.17±19.55	58.91±25.20	72.48±12.05	37.54±27.79	85.95±1.34
IID CIFAR-10	0%	66.07±0.03	65.63±0.02	65.98±0.10	51.79±0.35	62.79±0.20
	20%	48.12±11.74	47.17±13.55	58.95±4.38	51.68±0.96	62.37±0.47
	30%	43.35±12.21	41.56±15.51	47.91±12.52	51.11±1.12	62.15±0.46
	40%	36.89±12.52	34.60±16.37	34.65±18.05	47.94±6.04	61.62±0.46
Non-IID CIFAR-10	0%	66.08±0.07	65.58±0.05	65.87±0.09	51.97±0.01	62.53±0.12
	20%	48.37±11.36	48.60±12.44	60.48±3.96	51.60±0.93	62.73±0.29
	30%	42.61±12.73	41.57±15.97	48.79±11.89	51.57±1.46	62.38±0.40
	40%	36.21±12.24	34.88±16.33	36.33±17.55	47.48±5.89	61.72±0.39

5 Related Work

Poisoning attacks. According to Xia *et al.* [47], poisoning attacks can be classified into two categories: *data poisoning attacks* and *model poisoning attacks*. In data poisoning attacks, malicious clients have the ability to inject poisoned information into training data or labels. [40, 17, 5, 32, 35, 48] propose label flipping to attack the models by manipulating labels. Specifically, symmetric flipping [40] and pairwise flipping [17] are introduced to flip each label to other labels via a specific transition matrix, significantly enhancing the efficiency of label flipping attacks. [61, 60, 38, 37, 2] focus on the training data poisoning. They carefully craft the training data or generate fake data with aim of undermining the performance of the global model. On the other hand, Model poisoning [3, 56, 39, 49, 19, 23] directly manipulates clients to upload arbitrary or counterfeit local updates which poses significant threats to the global model.

Robust FL. In recent years, multiple robust FL algorithms in Synchronous FL have emerged. Broadly, these algorithms can be categorized into the following two groups. *1)* **Discarding rules** detect and exclude potential attackers when aggregating global models. Krum [6], Multi-Krum [6], Bulyan [31], A-Krum [43], Trimmed-Mean [58], and MAB-RFL [42] are represent of discarding algorithms. While discarding algorithms excel in defending against attacks, the removal of partial clients can be detrimental, especially in cases of non-IID data distribution or when the number of clients is limited. *2)* **Non-discarding rules** aim to leverage all the information of local updates instead of directly dropping out potential threat clients. Zeno [52], Zeno++ [53], Fed-Credit [10], and FLTrust [8] assign weights to each candidate local updates. Suspicious clients are assigned lower weights, while benign clients receive higher weights. GeoMed [13], Median [58], RFA [36] and FoolsGold [16] utilize statistical characteristics of updates to update the global parameters.

Efficient FL. Traditional FL is susceptible to the stragglers effect and network congestion. Algorithms such as FedAsync [51], FedSA [29], ASO-Fed [14], Async-FedED [44], and DP-AFL [28] have been introduced to mitigate the first issue. These algorithms enable the aggregator to update without waiting for lagging or lost clients, thereby saving training time. Asynchronous FL, however, can not effectively address peak network congestion. Additionally, Multi-tier FL (*e.g.* HFL [26], FedAT [9],

Table 4: Comparing robust accuracies (%) under 30% attacks of various types.

Dataset	Attack Type		FedAvg	Median	GeoMed	A-krum	FusCred
IID MNIST	Model poison	CP	95.41±0.12	95.73±0.22	95.93±0.22	92.06±0.52	97.42±0.15
		NP	95.18±0.12	94.40±0.02	94.55±0.27	9.70±0.22	97.47±0.07
		SF	89.23±3.95	87.55±2.18	96.33±0.28	92.78±0.11	97.28±0.09
	Data poison	PW	70.96±18.57	95.21±2.11	97.63±0.10	92.77±0.55	97.56±0.03
		SM	94.91±1.68	96.85±0.10	97.78±0.06	92.28±0.49	97.61±0.09
Non-IID MNIST	Model poison	CP	89.55±0.24	91.93±1.45	91.90±2.60	9.80±0.00	96.73±0.08
		NP	91.70±0.68	91.82±1.98	93.37±2.20	10.16±0.26	96.80±0.07
		SF	74.89±8.93	71.96±8.50	93.07±1.04	75.72±4.21	96.41±0.10
	Data poison	PW	95.43±1.12	95.13±1.46	97.32±0.17	73.55±4.21	94.86±2.16
		SM	94.69±1.07	96.38±0.02	97.37±0.09	77.27±1.77	95.51±0.61
IID FMNIST	Model poison	CP	83.59±0.49	85.46±0.62	88.21±0.07	83.93±0.81	88.85±0.11
		NP	83.73±0.65	85.13±0.61	87.79±0.14	26.26±11.51	88.91±0.10
		SF	64.42±9.73	55.05±18.14	88.51±0.28	86.46±0.09	88.47±0.22
	Data poison	PW	76.32±14.11	86.65±1.11	87.57±0.92	84.38±0.26	88.85±0.08
		SM	87.19±0.53	86.84±0.51	87.78±0.37	84.51±0.56	88.71±0.18
Non-IID FMNIST	Model poison	CP	73.97±2.23	76.65±2.71	85.24±1.15	37.11±9.12	87.21±0.20
		NP	75.40±1.62	75.26±3.14	84.91±1.20	32.20±0.40	87.31±0.24
		SF	51.01±9.66	42.47±16.05	83.70±3.54	78.22±0.57	85.82±0.84
	Data poison	PW	83.18±2.39	86.02±1.92	87.42±1.29	75.60±2.36	87.08±0.36
		SM	84.91±0.74	86.51±0.26	88.08±0.07	61.74±14.38	87.15±0.60
IID CIFAR-10	Model poison	CP	36.78±1.63	35.45±3.01	34.36±3.80	51.73±0.23	62.07±0.52
		NP	35.99±1.63	34.07±2.19	33.38±3.52	51.72±0.22	61.87±0.52
		SF	28.61±2.77	19.98±1.19	49.39±2.19	51.78±0.27	62.71±0.24
	Data poison	PW	56.00±0.45	57.16±0.89	60.46±0.92	50.02±1.34	62.08±0.05
		SM	59.40±0.33	61.13±0.59	61.93±0.42	50.32±1.12	62.01±0.23
Non-IID CIFAR-10	Model poison	CP	35.82±1.12	35.94±1.77	35.37±5.61	51.96±0.02	62.42±0.20
		NP	35.01±0.84	33.92±1.50	37.25±3.49	51.67±0.15	62.42±0.20
		SF	26.99±2.38	18.92±3.61	48.74±6.03	52.62±0.49	62.71±0.11
	Data poison	PW	56.10±0.66	57.65±1.00	60.71±0.52	51.11±1.48	62.41±0.13
		SM	59.13±0.17	61.43±0.29	61.89±0.25	50.51±2.38	61.93±0.61

FedEdge [43], HiFlash [46]) combined with cluster algorithms (*e.g.*, FL+HC [7], ClusterFL [34], FedCH [45]) have been proposed to address both the stragglers effect and network congestion. To our best of knowledge, nevertheless, few works have considered the robustness in efficient FL.

6 Discussion

The *FusCred* consistently outperforms alternative methods across various scenarios. In real-world implementation, however, it may be necessary for servers to change aggregation methods for some concerns. The edge servers are advised to apply non-discarding robustness algorithms. Given their limited visibility, edge servers can only observe their own client models, making it challenging to determine whether an outlier client model is due to model diversity or malicious attacks. In contrast, the cloud server is recommended to implement discarding robustness algorithms. Since it can perceive more information from edge models and it is easier for cloud server to discern whether an outlier model indicates a malicious attack.

7 Conclusion

In this paper, we propose CSS-HFL, a novel FL framework that can simultaneously handle with stragglers effect, network congestion, and poisoning attacks. The theoretical proofs demonstrating the efficiency and convergence guarantee of CSS-HFL are provided. Additionally, within CSS-HFL, we design a robust aggregation algorithm, named *FusCred*, outperforming alternative methods in defending against adversarial attacks, as exhibited in extensive experiments.

References

[1] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin. Hierarchical federated learning across heterogeneous cellular networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8866–8870, 2020.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 2020.

[3] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8632–8642, 2019.

[4] Amir Behrouzi-Far and Emina Soljanin. On the effect of task-to-worker assignment in distributed computing systems with stragglers. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 560–566, 2018.

[5] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

[6] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 119–129, 2017.

[7] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[8] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

[9] Zheng Chai, Yujing Chen, Ali Anwar, Liang Zhao, Yue Cheng, and Huzefa Rangwala. Fedat: a high-performance and communication-efficient federated learning system with asynchronous tiers. In Bronis R. de Supinski, Mary W. Hall, and Todd Gamblin, editors, *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*, page 60. ACM, 2021.

[10] Jiayan Chen, Zhirong Qian, Tianhui Meng, Xitong Gao, Tian Wang, and Weijia Jia. Fed-credit: Robust federated learning with credibility management. *CoRR*, abs/2405.11758, 2024.

[11] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H. Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20(1):269–283, 2021.

[12] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In Konstantinos Psounis, Aditya Akella, and Adam Wierman, editors, *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018*, page 96. ACM, 2018.

[13] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In Konstantinos Psounis, Aditya Akella, and Adam Wierman, editors, *Abstracts of the 2018 ACM International Conference on Measurement and*

Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018, page 96. ACM, 2018.

[14] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In Xintao Wu, Chris Jermaine, Li Xiong, Xiaohua Hu, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, Chengxiang Zhai, Eyhab Al-Masri, Zhiyuan Chen, and Jeff Saltz, editors, *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020*, pages 15–24. IEEE, 2020.

[15] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 1605–1622. USENIX Association, 2020.

[16] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, San Sebastian, October 2020. USENIX Association.

[17] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546, 2018.

[18] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *CoRR*, abs/2104.05704, 2021.

[19] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5311–5319. PMLR, 2021.

[20] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020.

[21] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[23] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1544–1551. AAAI Press, 2019.

[24] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[25] Xingyu Li, Zhe Qu, Bo Tang, and Zhuo Lu. Stragglers are not disasters: A hybrid federated learning framework with delayed gradients. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 727–732, 2022.

[26] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pages 1–6. IEEE, 2020.

[27] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Hierarchical federated learning with quantization: Convergence analysis and system design. *IEEE Trans. Wirel. Commun.*, 22(1):2–18, 2023.

[28] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Trans. Ind. Informatics*, 16(3):2134–2143, 2020.

[29] Qianpiao Ma, Yang Xu, Hongli Xu, Zhida Jiang, Liusheng Huang, and He Huang. Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE J. Sel. Areas Commun.*, 39(12):3654–3672, 2021.

[30] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[31] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3518–3527. PMLR, 2018.

[32] Konda Reddy Mopuri, Vaisakh Shaj, and R. Venkatesh Babu. Adversarial fooling beyond "flipping the label". In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 3374–3382. Computer Vision Foundation / IEEE, 2020.

[33] Diego Cardoso Nunes, Bruno Loureiro Coelho, Ricardo Parizotto, and Alberto Schaeffer-Filho. Serene: Handling the effects of stragglers in in-network machine learning aggregation. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–10, 2023.

[34] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Guoliang Xing, and Jianwei Huang. Clusterfl: A clustering-based federated learning system for human activity recognition. *ACM Trans. Sen. Netw.*, 19(1), dec 2022.

[35] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In Carlos Alzate, Anna Monreale, Haytham Assem, Albert Bifet, Teodora Sandra Buda, Bora Caglayan, Brett Drury, Eva García-Martín, Ricard Gavaldà, Stefan Kramer, Niklas Lavesson, Michael Madden, Ian M. Molloy, Maria-Irina Nicolae, and Mathieu Sinn, editors, *ECML PKDD 2018 Workshops - Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings*, volume 11329 of *Lecture Notes in Computer Science*, pages 5–15. Springer, 2018.

[36] Venkata Krishna Pillutla, Sham M. Kakade, and Zaïd Harchaoui. Robust aggregation for federated learning. *CoRR*, abs/1912.13445, 2019.

[37] Dazhong Rong, Shuai Ye, Ruoyan Zhao, Hon Ning Yuen, Jianhai Chen, and Qinming He. Fedrecattack: Model poisoning attack to federated recommendation. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pages 2643–2655. IEEE, 2022.

[38] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6106–6116, 2018.

[39] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

[40] Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. Learning with symmetric label noise: The importance of being unhinged. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 10–18, 2015.

[41] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[42] Wei Wan, Shengshan Hu, Jianrong Lu, Leo Yu Zhang, Hai Jin, and Yuanyuan He. Shielding federated learning: Robust aggregation with adaptive client selection. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 753–760. ijcai.org, 2022.

[43] Kaibin Wang, Qiang He, Feifei Chen, Hai Jin, and Yun Yang. Fededge: Accelerating edge-assisted federated learning. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2895–2904, New York, NY, USA, 2023. Association for Computing Machinery.

[44] Qiyuan Wang, Qianqian Yang, Shibo He, Zhiguo Shi, and Jiming Chen. Asyncfeded: Asynchronous federated learning with euclidean distance based adaptive weight aggregation. *CoRR*, abs/2205.13797, 2022.

[45] Zhiyuan Wang, Hongli Xu, Jianchun Liu, Yang Xu, He Huang, and Yangming Zhao. Accelerating federated learning with cluster construction and hierarchical aggregation. *IEEE Transactions on Mobile Computing*, 22(7):3805–3822, 2023.

[46] Qiong Wu, Xu Chen, Tao Ouyang, Zhi Zhou, Xiaoxi Zhang, Shusen Yang, and Junshan Zhang. Hiflash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association. *IEEE Transactions on Parallel and Distributed Systems*, 34(5):1560–1579, 2023.

[47] Geming Xia, Jian Chen, Chaodong Yu, and Jun Ma. Poisoning attacks in federated learning: A survey. *IEEE Access*, 11:10708–10722, 2023.

[48] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 870–875. IOS Press, 2012.

[49] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. AUAI Press, 2019.

[50] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. PMLR, 22–25 Jul 2020.

[51] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *CoRR*, abs/1903.03934, 2019.

[52] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June*

2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 6893–6901. PMLR, 2019.

[53] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous SGD. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10495–10503. PMLR, 2020.

[54] Haonan Yan, Wenjing Zhang, Qian Chen, Xiaoguang Li, Wenhai Sun, Hui Li, and Xiaodong Lin. RECESS vaccine for federated learning: Proactive defense against model poisoning attacks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[55] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-IID federated learning. In *International Conference on Learning Representations*, 2021.

[56] Ming Yang, Hang Cheng, Fei Chen, Ximeng Liu, Meiqing Wang, and Xibin Li. Model poisoning attack in differential privacy-based federated learning. *Information Sciences*, 630:158–172, 2023.

[57] Zhengjie Yang, Sen Fu, Wei Bao, Dong Yuan, and Albert Y. Zomaya. Hierarchical federated learning with momentum acceleration in multi-tier networks. *IEEE Transactions on Parallel and Distributed Systems*, 34(10):2629–2641, 2023.

[58] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR, 2018.

[59] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *J. Mach. Learn. Res.*, 24:100:1–100:7, 2023.

[60] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.*, 8(5):3310–3322, 2021.

[61] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. Poisoning attack in federated learning using generative adversarial nets. In *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, August 5-8, 2019*, pages 374–380. IEEE, 2019.

A Additional Discussions

A.1 Synchronous v.s. Asynchronous

Synchronous FL is a typical architecture in which a server distributes the global model to a selected subset of clients and does not update the global model until it receives all local updates. However the server can update global model without waiting for the lagging clients in Asynchronous FL.

Due to the synchronous nature of FL, numerous previous robust FL algorithms have been proposed which heavily rely on comparisons of local updates. However, waiting for stragglers or offline clients can lead to substantial costs. Asynchronous FL significantly enhances convergence efficiency compared to Synchronous FL. Nevertheless, defending against malicious attacks becomes challenging when updating the global model with just one local update.

A.2 Two-tier v.s. Three-tier

In Two-tier FL, multiple clients are directly connected to a remote server or cloud, which suffers from peak network congestion in both Synchronous FL and Asynchronous FL. With the development of edge computing, an edge tier is added between the local clients and remote cloud to alleviate the strain caused by peak network congestion. In a Three-tier HFL, the clients can first communicate with the edge node for edge-level aggregation. Subsequently, the edge nodes communicate with the remote cloud for cloud-level aggregation.

The Three-tier architecture presents a promising solution for real-world large-scale clients and has captivated significant attention from researchers [1, 7, 57]. To the best of our knowledge, however, none of these works have focused on security, which poses significant threats to convergence, privacy, economics, and even life security.

B Notations

Table 5: Key Notations For The Clustered Semi-synchronous HFL Framework.

Denote	Description	Denote	Description
n	The number of clients	N	The number of edges
N_λ	The number clients of λ^{th} edge	C_i	The i^{th} client ($1 \le i \le n$)
$\mathcal{E}_\lambda$	The λ^{th} edge ($1 \le \lambda \le N$)	lr	Learning rate
B	Batch size	E	Number of client training epochs
E_λ	Number of λ^{th} edge training epochs	R	Number of cloud training epochs
$\boldsymbol{w}, \boldsymbol{w_\lambda}, \boldsymbol{w_{\lambda,k}}$	The model of cloud, λ^{th} edge, k^{th} client in λ^{th} edge ($1 \le k \le N_\lambda$)		

C Convergence Analysis

Our convergence analysis is inspired by [24]. We first make the following assumptions.

Assumption 1. *The loss functions F in the cloud server, the edge server, and the client are all L-smooth:* $\forall \boldsymbol{v}, \boldsymbol{w}, F(\boldsymbol{w}) \le F(\boldsymbol{v}) + (\boldsymbol{w} - \boldsymbol{v})^\top \nabla F(\boldsymbol{v}) + \frac{L}{2}\|\boldsymbol{w} - \boldsymbol{v}\|_2^2$.

Assumption 2. *The loss functions F in the cloud server, the edge server, and the client are all μ-strongly convex:* $\forall \boldsymbol{v}, \boldsymbol{w}, F(\boldsymbol{w}) \ge F(\boldsymbol{v}) + (\boldsymbol{w} - \boldsymbol{v})^\top \nabla F(\boldsymbol{v}) + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{v}\|_2^2$.

Assumption 3. *Let $\xi_t^{\lambda,k}$ be sampled uniformly at random from local data of the k^{th} end device in the λ^{th} edge. The variance of stochastic gradients in each device is bounded as follows:* $\mathbb{E}\|\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k}) - \nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}})\|^2 \le \sigma_{\lambda,k}^2$, *for* $1 \le \lambda \le N$ *and* $1 \le k \le N_\lambda$.

Assumption 4. *The expected squared norm of stochastic gradients is uniformly bounded,* i.e., $\mathbb{E}\|\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k})\|^2 \le G^2$, *for* $1 \le \lambda \le N$, $1 \le k \le N_\lambda$, *and* $1 \le t \le T$.

We define F^* and $F_{\lambda,k*}$ as the minimum value of F and $F_{\lambda,k}$ and let $\Lambda = \sum\limits_{\lambda=1}^{N} \sum\limits_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}(F^* - F_{\lambda,k*})$ quantify the degree of Non-IID [24]. We assume the cloud server aggregation interval is T_c and the total number of rounds is T. Then, under our CSS-HFL framework, we have the following convergence guarantee for FedAvg.

Theorem 1. *Let* (1) (2) (3) (4) *hold and* $L, \mu, \sigma^2_{\lambda,k}, G$ *be defined therein. Choose* $\tau = \frac{L}{\mu}$, $\varphi = max\{8\tau, T_c\}$ *and the learning rate* $\eta_t = \frac{2}{u(\varphi+t)}$. *Then our CSS-HFL framework satisfies*

$$\mathbb{E}[F(\boldsymbol{w_t})] - F^* \leq \frac{\tau}{\varphi + t - 1}\left(\frac{2\Upsilon}{\mu} + \frac{\mu\varphi}{2}\mathbb{E}\|\boldsymbol{w_1} - \boldsymbol{w^*}\|^2\right), \tag{1}$$

where

$$\Upsilon = \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda^2 p_{\lambda,k}^2 \sigma_{\lambda,k}^2 + 6L\Lambda + 8(T_c - 1)^2 G^2.$$

According to the above result, we observe that the right-hand side of the Equation (1) consists of two terms. The first term, $\tau/(\varphi + t - 1)$, exhibits a decreasing trend concerning t. As t grows sufficiently large, the constants φ and 1 can be disregarded, leading to an approximate form of $\frac{\tau}{t}\left(\frac{2\Upsilon}{\mu} + \frac{\mu\tau}{2}\mathbb{E}\|\boldsymbol{w_1} - \boldsymbol{w^*}\|^2\right)$. This implies a convergence rate of $O(1/t)$, indicating sub-linear convergence.

D Proof of Convergence

Table 6: Table of Key Notations for Convergence Analysis.

Notation	Description
N	The number of edges
N_λ	The number clients of λ^{th} edge
p_λ	The weight of λ^{th} in the cloud aggregation
$p_{\lambda,k}$	The weight of k^{th} client in the λ^{th} edge aggregation
$\boldsymbol{w^{\lambda,k}}$	The model of k^{th} client in λ^{th} edge ($1 \leq k \leq N_\lambda$)
$\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k})$	The gradient of k^{th} client in λ^{th} edge ($1 \leq k \leq N_\lambda$)
η_t	Learning rate of t^{th} round
T_c	The aggregation interval of cloud server

We prove the Theorem 1 in this section. The key notations for convergence analysis is presented in Table 6.

D.1 Additional Denotes

Let $T_c, T_\lambda, T_{\lambda,k}$ be the aggregation interval of the cloud server, the λ^{th} edge, and the k^{th} client in the λ^{th} edge. Note that for cloud server and edges server, aggregations only occur if the remainder of t divided by the interval T_c or T_λ is 0, t is the current round.And for the k^{th} client in the λ^{th} edge, if the t^{th} round is not the aggregation round for the client (*i.e.*, $t \bmod T_{\lambda,k} \neq 0$), $\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k}) = 0$; otherwise, $\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k})$ represent the gradient of the sampled mini-batch local dataset. We also adopt the virtual sequence from [24] to represent the immediate result of t^{th} round. The above note can be described as

$$\nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k}) = \begin{cases} 0, & \text{if } t \bmod T_{\lambda,k} \neq 0, \\ \nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k}), & \text{if } t \bmod T_{\lambda,k} \neq 0. \end{cases} \tag{2}$$

$$\boldsymbol{v_{t+1}^{\lambda,k}} = \boldsymbol{w_t^{\lambda,k}} - \eta_t \nabla F_{\lambda,k}(\boldsymbol{w_t^{\lambda,k}}, \xi_t^{\lambda,k}). \tag{3}$$

$$\boldsymbol{w}_{\boldsymbol{t+1}}^{\boldsymbol{\lambda,k}} = \begin{cases} \boldsymbol{v}_{\boldsymbol{t+1}}^{\boldsymbol{\lambda,k}}, & \text{if } t+1 \bmod T_\lambda \neq 0 \text{ and } t+1 \bmod T_c \neq 0, \\ \sum_{k=1}^{N_\lambda} p_{\lambda,k} \boldsymbol{v}_{\boldsymbol{t+1}}^{\boldsymbol{\lambda,k}}, & \text{if } t+1 \bmod T_\lambda = 0 \text{ and } t+1 \bmod T_c \neq 0, \\ \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \boldsymbol{v}_{\boldsymbol{t+1}}^{\boldsymbol{\lambda,k}}, & \text{if } t+1 \bmod T_c = 0. \end{cases} \tag{4}$$

For convenience, we define $\bar{\boldsymbol{v}}_{\boldsymbol{t}} = \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \boldsymbol{v}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}$, $\bar{\boldsymbol{w}}_{\boldsymbol{t}} = \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}$, $\bar{\boldsymbol{g}}_{\boldsymbol{t}} = \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}})$, and $\boldsymbol{g}_{\boldsymbol{t}} = \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}, \xi_t^{\lambda,k})$. Therefore, $\bar{\boldsymbol{v}}_{\boldsymbol{t+1}} = \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \eta_t \boldsymbol{g}_{\boldsymbol{t}}$ and $\mathbb{E}\boldsymbol{g}_{\boldsymbol{t}} = \bar{\boldsymbol{g}}_{\boldsymbol{t}}$.

D.2 Key Lemmas

In this section, we describe and proof the key useful lemmas.

Lemma 1. *Assume* (3) *holds, we have*

$$\mathbb{E}\|\boldsymbol{g}_{\boldsymbol{t}} - \bar{\boldsymbol{g}}_{\boldsymbol{t}}\|^2 \leq \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda^2 p_{\lambda,k}^2 \sigma_{\lambda,k}^2.$$

Proof. From (3), the variance of the stochastic gradients in k^{th} client device in λ^{th} edge is bounded by $\sigma_{\lambda,k}^2$, then

$$\begin{aligned} \mathbb{E}\|\boldsymbol{g}_{\boldsymbol{t}} - \bar{\boldsymbol{g}}_{\boldsymbol{t}}\|^2 &= \mathbb{E}\left\| \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} (\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}, \xi_t^{\lambda,k}) - \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}})) \right\|^2 \\ &= \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda^2 p_{\lambda,k}^2 \mathbb{E} \left\| \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}, \xi_t^{\lambda,k}) - \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}) \right\|^2 \\ &\leq \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda^2 p_{\lambda,k}^2 \sigma_{\lambda,k}^2. \end{aligned}$$

□

Lemma 2. *Assume* (4) *holds, η_t is non-increasing, and $\eta_t \leq 2\eta_{t+T_c}$ for all $t \geq 0$. It follows that*

$$\mathbb{E}\left[\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\| \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}} \right\|^2 \right] \leq 4\eta_t^2 (T_c - 1)^2 G^2.$$

Proof. $\forall t \geq 0, \exists t_0 \leq t$, such that $\boldsymbol{w}_{\boldsymbol{t_0}}^{\boldsymbol{\lambda,k}} = \bar{\boldsymbol{w}}_{\boldsymbol{t_0}}$, *i.e.*, round t_0 is the last cloud server aggregation round. Therefore we can indicate that $t - t_0 \leq T_c - 1$. Additionally, we utilize the assumptions that η_t is non-increasing and $\eta_{t_0} \leq 2\eta_{t_0+T_c} \leq 2\eta_t$, then

$$
\begin{aligned}
\mathbb{E}\left[\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}\right\|^2\right] &= \mathbb{E}\left[\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t_0}})-(\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\bar{\boldsymbol{w}}_{\boldsymbol{t_0}})\right\|^2\right] \\
&\leq \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\mathbb{E}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t_0}}\right\|^2 \\
&\leq \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\sum_{t=t_0}^{t-1}(T_c-1)\eta_t^2\mathbb{E}\left\|\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}},\xi_t^{\lambda,k})\right\|^2 \\
&\leq \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\sum_{t=t_0}^{t-1}(T_c-1)\eta_{t_0}^2 G^2 \\
&\leq \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\eta_{t_0}^2(T_c-1)^2 G^2 \\
&\leq 4\eta_t^2(T_c-1)^2G^2.
\end{aligned}
$$

First inequality: We use the property of variance as follow

$$\mathbb{E}\left\|X-\mathbb{E}X\right\|^2 \leq \mathbb{E}\left\|X\right\|^2$$

where $X=(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t_0}})$.

Second inequality: We use the Cauchy-Schwarz inequality and $t-t_0\leq T_c-1$.

$$
\begin{aligned}
\mathbb{E}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t_0}}\right\|^2 &= \mathbb{E}\left\|\sum_{t=t_0}^{t-1}\eta_t\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}},\xi_t^{\lambda,k})\right\|^2 \\
&\leq \sum_{t=t_0}^{t-1}(t-t_0)\eta_t^2\mathbb{E}\left\|\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}},\xi_t^{\lambda,k})\right\| \\
&\leq \sum_{t=t_0}^{t-1}(T_c-1)\eta_t^2\mathbb{E}\left\|\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}},\xi_t^{\lambda,k})\right\|.
\end{aligned}
$$

Third inequality: We leverage (3) and η_t is non-increasing (*i.e.* $\eta_t\leq\eta_{t_0}$ for $t\geq t_0$).

Fourth inequality: We utilize $\eta_{t_0}\leq 2\eta_{t_0+T_c}\leq 2\eta_t$. $\square$

Lemma 3. *We assume the* (1), (2), *and* $\eta_t=\frac{\alpha}{t+\varphi}$ *for some* $\alpha>\frac{1}{\mu}$ *and* $\varphi>0$ *such that* $\eta_1\leq min\{\frac{1}{\mu},\frac{1}{4L}\}=\frac{1}{4L}$, *it follows that*

$$
\begin{aligned}
\mathbb{E}\left\|\bar{\boldsymbol{v}}_{\boldsymbol{t+1}}-\boldsymbol{w}^*\right\|^2 \leq& (1-\eta_t\mu)\mathbb{E}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^*\right\|^2+\eta_t^2\mathbb{E}\left\|\boldsymbol{g}_{\boldsymbol{t}}-\bar{\boldsymbol{g}}_{\boldsymbol{t}}\right\|^2+6L\eta_t^2\Lambda \\
&+2\mathbb{E}\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda}p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda,k}}\right\|^2,
\end{aligned}
$$

where $\Lambda=\sum\limits_{\lambda=1}^{N}\sum\limits_{k=1}^{N_\lambda}p_\lambda p_{\lambda,k}(F^*-F_{\lambda,k}^*)\geq 0$.

Proof. We first divide $\left\|\bar{\boldsymbol{v}}_{\boldsymbol{t+1}}-\boldsymbol{w}^*\right\|^2$ into following three parts.

$$
\begin{aligned}
\left\|\bar{\boldsymbol{v}}_{\boldsymbol{t+1}}-\boldsymbol{w}^*\right\|^2 &= \left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\eta_t\boldsymbol{g}_{\boldsymbol{t}}-\boldsymbol{w}^*-\eta_t\bar{\boldsymbol{g}}_{\boldsymbol{t}}+\eta_t\bar{\boldsymbol{g}}_{\boldsymbol{t}}\right\|^2 \\
&= \underbrace{\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^*-\eta_t\bar{\boldsymbol{g}}_{\boldsymbol{t}}\right\|^2}_{P_1}+\underbrace{2\eta_t\left\langle\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^*-\eta_t\bar{\boldsymbol{g}}_{\boldsymbol{t}},\bar{\boldsymbol{g}}_{\boldsymbol{t}}-\boldsymbol{g}_{\boldsymbol{t}}\right\rangle}_{P_2}+\eta_t^2\left\|\bar{\boldsymbol{g}}_{\boldsymbol{t}}-\boldsymbol{g}_{\boldsymbol{t}}\right\|^2. \qquad (5)
\end{aligned}
$$

Next, we focus on the P_1:

$$P_1 = \|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^* - \eta_t \bar{\boldsymbol{g}}_{\boldsymbol{t}}\|^2 = \|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\|^2 - \underbrace{2\eta_t \langle \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*, \bar{\boldsymbol{g}}_{\boldsymbol{t}} \rangle}_{Q_1} + \underbrace{\eta_t^2 \|\bar{\boldsymbol{g}}_{\boldsymbol{t}}\|^2}_{Q_2}. \tag{6}$$

We pay attention to Q_1:

$$\begin{aligned}
Q_1 =& 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\langle \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} + \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^*, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \\
=& 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\langle \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \\
&+ 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\langle \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^*, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \\
\geq& 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \frac{1}{2} \left(-\frac{1}{\eta_t} \left\| \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} \right\|^2 - \eta_t \left\| \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\|^2 \right) \\
&+ 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\langle \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^*, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \\
\geq& 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \frac{1}{2} \left(-\frac{1}{\eta_t} \left\| \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} \right\|^2 - \eta_t \left\| \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\|^2 \right) \\
&+ 2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left(\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) - \nabla F_{\lambda,k}(\boldsymbol{w}^*) + \frac{\mu}{2} \left\| \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^* \right\|^2 \right).
\end{aligned} \tag{7}$$

First inequality: We derive the first inequality in Equation (7) by Cauchy-Schwarz inequality and AM-GM inequality.

$$\left\langle \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \geq \frac{1}{2} \left(-\frac{1}{\eta_t} \left\| \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} \right\|^2 - \eta_t \left\| \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\|^2 \right).$$

Second inequality: By the μ-strong convexity of $F_{\lambda,k}$, we have

$$\left\langle \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^*, \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\rangle \geq \left(\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) - \nabla F_{\lambda,k}(\boldsymbol{w}^*) + \frac{\mu}{2} \left\| \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \boldsymbol{w}^* \right\|^2 \right).$$

Then, we analyze Q_2, By the convexity of $\|\cdot\|^2$ and the L-smoothness of $F_{\lambda,k}$, we have

$$\begin{aligned}
Q_2 = \eta_t^2 \|\bar{\boldsymbol{g}}_{\boldsymbol{t}}\|^2 \leq& \eta_t^2 \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left\| \nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) \right\|^2 \\
\leq& 2L\eta_t^2 \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k} \left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}) - F_{\lambda,k*} \right)
\end{aligned} \tag{8}$$

By combining Equation (7) and Equation (8), we have

$$
\begin{aligned}
P_1 \leq & \left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^{*}\right\|^2+\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(\frac{1}{\eta_t}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2+\eta_t\left\|\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})\right\|^2\right) \\
& -2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(\nabla F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-\nabla F_{\lambda,k}(\boldsymbol{w}^{*})+\frac{\mu}{2}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\boldsymbol{w}^{*}\right\|^2\right) \\
& +2L\eta_t^2 \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-F_{\lambda,k*}\right) \\
\leq & (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^{*}\right\|^2+\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 \\
& +4L\eta_t^2 \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-F_{\lambda,k}^{*}\right) \\
& -2\eta_t \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-F_{\lambda,k}(\boldsymbol{w}^{*})\right) \\
= & (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^{*}\right\|^2+\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 \\
& +4L\eta_t^2 \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left((F^{*}-F_{\lambda,k}^{*}\right) \\
& +(4L\eta_t^2-2\eta_t)\underbrace{\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-F^{*}\right)}_{S},
\end{aligned}
\tag{9}
$$

where we use the L-smoothness of $F_{\lambda,k}(\cdot)$ again and the following inequality for the second inequality,

$$
\begin{aligned}
\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}}-\boldsymbol{w}^{*}\right\|^2 &= \left\|\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\boldsymbol{w}^{*}\right)\right\|^2 \\
&\leq \| \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} \left(\sqrt{p_\lambda p_{\lambda,k}}\right)^2 \| \cdot \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda}\left(\sqrt{p_\lambda p_{\lambda,k}}\left(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\boldsymbol{w}^{*}\right)\right)^2 \\
&= \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\boldsymbol{w}^{*}\right\|^2
\end{aligned}
$$

We next focus S,

$$
\begin{aligned}
S = & \left(\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}})-F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}})\right)+\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}})-F^{*}\right)\right) \\
\geq & \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\langle\nabla F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}}), \bar{\boldsymbol{w}}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\rangle+\left(F(\bar{\boldsymbol{w}}_{\boldsymbol{t}})-F^{*}\right) \\
\geq & -\frac{1}{2} \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left[\eta_t\left\|\nabla F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}})\right\|^2+\frac{1}{\eta_t}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\|^2\right]+\left(F(\bar{\boldsymbol{w}}_{\boldsymbol{t}})-F^{*}\right) \\
\geq & -\sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left[\eta_t L(F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}})-F_{\lambda,k}^{*})+\frac{1}{2\eta_t}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}-\bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\|^2\right]+\left(F(\bar{\boldsymbol{w}}_{\boldsymbol{t}})-F^{*}\right).
\end{aligned}
\tag{10}
$$

The first inequality arises from the convexity of $F_{\lambda,k}(\cdot)$, the second inequality from the AM-GM inequality, and the third inequality from the L-smoothness of $F_{\lambda,k}$.

By combining Equation (9) and Equation (10), and utilize the notation $\Lambda = \sum\limits_{\lambda=1}^{N} \sum\limits_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}(F^* - F_{\lambda,k*})$, we have

$$
\begin{aligned}
P_1 \leq& (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\right\|^2 + \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 + (4L\eta_t^2)\Lambda \\
&+ (2\eta_t - 4L\eta_t^2)\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\Bigg[\eta_t L(F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) - F^* + F^* - F_{\lambda,k*}) \\
&+ \frac{1}{2\eta_t}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\|^2\Bigg] - (2\eta_t - 4L\eta_t^2)\left(F(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) - F^*\right) \\
=& (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\right\|^2 + \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 + (6L\eta_t^2 - 4L^2\eta_t^3)\Lambda \\
&+ \frac{2\eta_t - 4L\eta_t^2}{2\eta_t}\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\|^2 \\
&+ (2\eta_t - 4L\eta_t^2)(\eta_t L - 1)\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left(F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) - F^*\right) \\
\leq& (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\right\|^2 + \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 + 6L\eta_t^2\Lambda \\
&+ \sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}} - \bar{\boldsymbol{w}}_{\boldsymbol{t}}\right\|^2 \\
=& (1-\mu\eta_t)\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\right\|^2 + 2\sum_{\lambda=1}^{N}\sum_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{\lambda},\boldsymbol{k}}\right\|^2 + (6L\eta_t^2)\Lambda,
\end{aligned}
\tag{11}
$$

For the last inequality, we use the following facts:

1. $\Lambda \geq 0$ and $4L^2\eta_t^3 > 0$.
2. $\frac{2\eta_t - 4L\eta_t^2}{2\eta_t} \leq 1$.
3. $\eta_t L - 1 \leq 0$ and $\sum\limits_{\lambda=1}^{N}\sum\limits_{k=1}^{N_\lambda} p_\lambda p_{\lambda,k}(F_{\lambda,k}(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) - F^*) = F(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) - F^* \geq 0$.

Then, back to the Equation (5), notice that $\mathbb{E}\|\boldsymbol{g}_{\boldsymbol{t}}\| = \bar{\boldsymbol{g}}_{\boldsymbol{t}}$, *i.e.*,

$$\mathbb{E}\|P_2\| = 0. \tag{12}$$

Using the Equation (11) and Equation (12), we prove the Lemma 3. □

D.3 Proof of Theorem 1

Proof. It is evident that we always have $\bar{\boldsymbol{w}}_{\boldsymbol{t}} = \bar{\boldsymbol{v}}_{\boldsymbol{t}}$. For a non-increasing learning rate, $\eta_t = \frac{\alpha}{t+\varphi}$ for some $\alpha > \frac{1}{\mu}$ and $\varphi > 0$ such that $\eta_1 \leq min\{\frac{1}{\mu}, \frac{1}{4L}\} = \frac{1}{4L}$ and $\eta_t \leq 2\eta_{t+T_c}$ for all $t \geq 0$. From Lemma 1 Lemma 3, it follows that

$$\mathbb{E}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t+1}} - \boldsymbol{w}^*\right\|^2 \leq (1-\eta_t\mu)\mathbb{E}\left\|\bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^*\right\|^2 + \eta_t^2\Upsilon, \tag{13}$$

where

$$\Upsilon = \sum_{\lambda=1}^{N} \sum_{k=1}^{N_\lambda} p_\lambda^2 p_{\lambda,k}^2 \sigma_{\lambda,k}^2 + 6L\Lambda + 8(T_c - 1)^2 G^2.$$

Let $\Delta_t = \mathbb{E} \left\| \bar{\boldsymbol{w}}_{\boldsymbol{t}} - \boldsymbol{w}^* \right\|^2$, $\zeta = max \left\{ \frac{\alpha^2 \Upsilon}{\alpha\mu - 1}, (\varphi + 1)\Delta_1 \right\}$. We can easily find that $\Delta_1 \leq \frac{\zeta}{\varphi + t}$ holds for $t = 1$. Next, we prove $\Delta_t \leq \frac{\zeta}{\varphi + t}$ by induction.

$$\begin{aligned}
\Delta_{t+1} \leq& (1 - \eta_t \mu)\Delta_t + \eta_t^2 \Upsilon \\
\leq& \left(1 - \frac{\alpha\mu}{t + \varphi}\right) \frac{\zeta}{t + \varphi} + \frac{\alpha^2 \Upsilon}{(t + \varphi)^2} \\
=& \frac{t + \varphi - 1}{(t + \varphi)^2} \zeta + \left[\frac{\alpha^2 \Upsilon}{(t + \varphi)^2} - \frac{\alpha\mu - 1}{(t + \varphi)^2} \zeta \right] \\
\leq& \frac{\zeta}{t + \varphi + 1}.
\end{aligned}$$

If we set $\alpha = \frac{2}{\mu}$, $\varphi = \max \left\{ \frac{8L}{\mu}, T_c \right\} - 1$, and define $\tau = \frac{L}{\mu}$, then $\eta_t = \frac{2}{\mu(\varphi + t)}$. It can be verified that this choice of η_t satisfies $\eta_t \leq 2\eta_{t+T_c}$ for $t \geq 1$. Thus, we obtain

$$\zeta = max \left\{ \frac{\alpha^2 \Upsilon}{\alpha\mu - 1}, (\varphi + 1)\Delta_1 \right\} \leq \frac{\alpha^2 \Upsilon}{\alpha\mu - 1} + (\varphi + 1)\Delta_1 \leq \frac{4\Upsilon}{\mu^2} + (\varphi + 1)\Delta_1,$$

and by the L-smoothness of $F(\cdot)$, we have

$$\begin{aligned}
\mathbb{E} \left\| F(\bar{\boldsymbol{w}}_{\boldsymbol{t}}) \right\| - F^* \leq \frac{L}{2}\Delta_t \leq \frac{L\zeta}{2(\varphi + t)} \leq& \frac{\tau}{\varphi + t} \left(\frac{2\Upsilon}{\mu} + \frac{\mu(\varphi + 1)}{2} \Delta_1 \right) \\
\leq& \frac{\tau}{\varphi + t - 1} \left(\frac{2\Upsilon}{\mu} + \frac{\mu\varphi}{2} \mathbb{E} \| \bar{\boldsymbol{w}}_{\boldsymbol{1}} - \boldsymbol{w}^* \| \right).
\end{aligned}$$

□

E Limitations

The experiments were conducted on the assumption that the number of attackers would remain below 50%. Scenarios involving a higher number of attackers were not considered in the current study. Additionally, the framework was derived under the assumption that the computational power and communication capabilities among clients would not significantly differ. The performance and robustness of the framework in scenarios where there are substantial variations in computational power among clients remain areas for future research.

F Experiments

F.1 Efficiency Simulation

In this section, we evaluate the efficiency of frameworks using the Average Waiting Time (AWT) metric.

We utilize a mixed normal distribution to model the computational capacity of clients and a normal distribution to model the communication conditions between clients and edge servers. Next, we apply the *Balanced Cluster Algorithm* [45] to cluster clients into N groups. Subsequently, we calculate the AWT for various values of N.

We plot the simulation results in Figure 4. The scheduler scheme in HiFlash [46] is trained using reinforcement learning, which makes its efficiency difficult to simulate. It is evident that the AWT value of CSS-HFL decreases with an increase in the number of edge servers. Moreover, the AWT decreases rapidly when there are fewer edge servers, exhibiting a trend similar to the elbow pattern

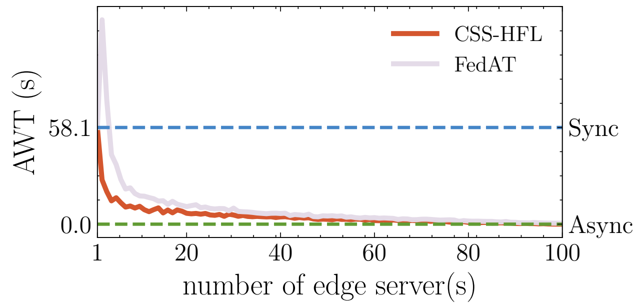


Figure 4: The AWT with different number of edge servers under various frameworks

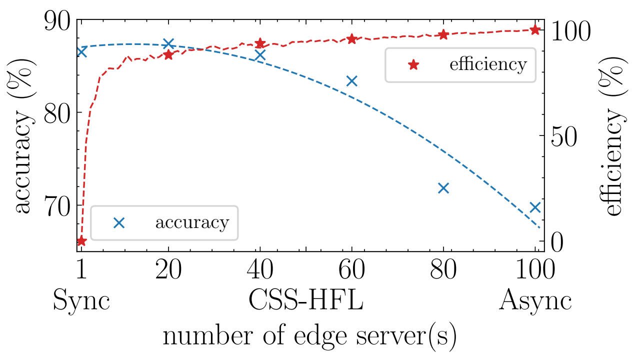


Figure 5: The accuracy and efficiency with different number of edge servers under 30% attacks on FMNIST.

often observed in K-means clustering. Another finding, which is in line with our theoretical analysis, is that when the number N of edge servers equals 1, the AWT of CSS-HFL is equivalent to that of Synchronous FL. Conversely, when N equals the number n of clients, the AWT of CSS-HFL aligns with that of Asynchronous FL Theoretically, AWT follows a strictly decreasing trend. However, our figure exhibits small fluctuations. This can be attributed to the fact that clustering task is a NP-hard problem. Additionally, we limit the maximum number of iterations in the Balanced Cluster Algorithm to 10, which may prevent us from achieving the optimal solution for clustering in each case.

To further explore the trade-off relationships of CSS-HFL, we use (14) to define the efficiency eff, which converts AWT to an efficiency metric. In this context, the efficiency of Asynchronous and Synchronous FL are respectively normalized to 100% and 0%.

$$\text{eff} = \left(1 - \frac{AWT}{AWT_{\text{Sync}}}\right) \cdot 100\%. \tag{14}$$

We investigated the accuracy and efficiency with various number of edge servers under 30% attacks on FMNIST. As shown in Figure 5, a trade-off pattern emerges between accuracy and efficiency. As the number of edge servers increases, the accuracy declines while the efficiency improves. Notably, it is possible to achieve both robustness and high efficiency by selecting a certain number of edge servers (e.g., 20 edge servers in the 100-client scenario).

In conclusion, our findings suggest that the inclusion of several edge servers can significantly decrease the AWT (*i.e.* enhance efficiency) and maintain robustness against attacks under our CSS-HFL framework.

F.2 Datasets

MNIST: The MNIST dataset is a well-known collection of handwritten digits widely used in the field of machine learning. It consists of 60,000 training examples and 10,000 testing examples. Each sample is a 28x28 image of a digit, ranging from 0 to 9. MNIST is a standard benchmark dataset in the machine learning community and is widely employed to assess the performance of various algorithms.

Fashion-MNIST: Fashion-MNIST is a dataset similar in structure to MNIST but comprises images of fashion items instead of handwritten digits. The Fashion-MNIST dataset consists of 60,000 training samples and 10,000 testing samples, which is consistent with MNIST. Fashion-MNIST stands as a benchmark dataset for image classification endeavors, specifically in the domain of fashion and clothing recognition. Each image within the dataset is a grayscale 28x28 pixel representation of a fashion item, categorized into one of 10 distinct classes, such as shirts, trousers, dresses, and shoes. Like MNIST, Fashion-MNIST has become broadly utilized in the machine learning scope for evaluating the models.

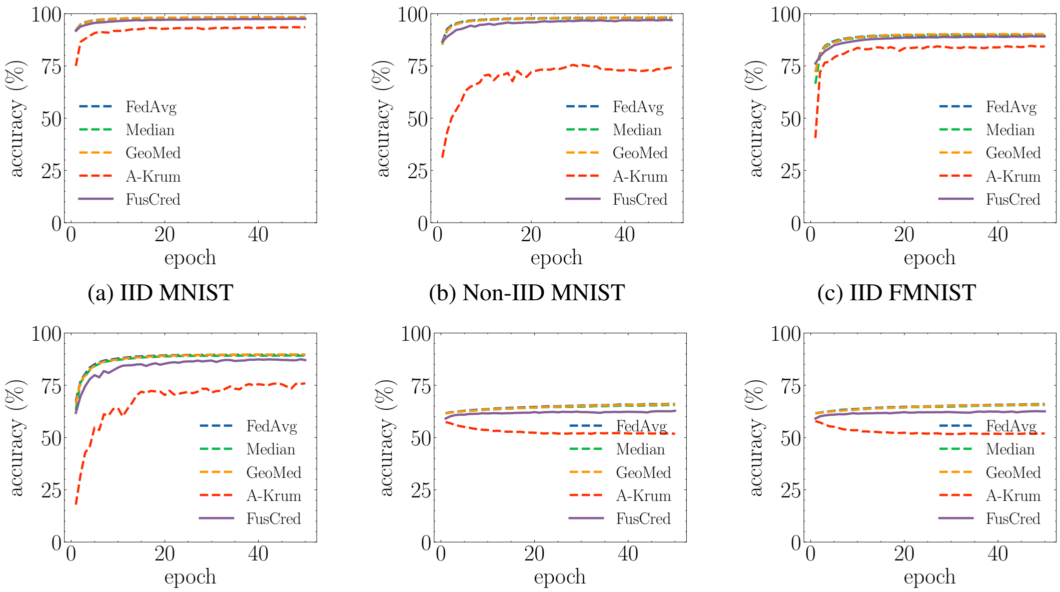
CIFAR-10: The CIFAR-10 dataset stands as a widely recognized benchmark in the domain of computer vision. It comprises 60,000 color images, each measuring 32x32 pixels, and is categorized into 10 distinct classes. Like MNIST and Fashion-MNIST, CIFAR-10 serves as a standard evaluation tool for image classification algorithms, facilitating advancements in the field of deep learning.

F.3 Experimental Results

In this section, we present the hyperparameters settings in Table 7 and the overview of experiment results in Table 8. Our *FusCred* exhibited superior robustness across various poisoning attack scenarios.

Table 7: The Hyperparameters Settings

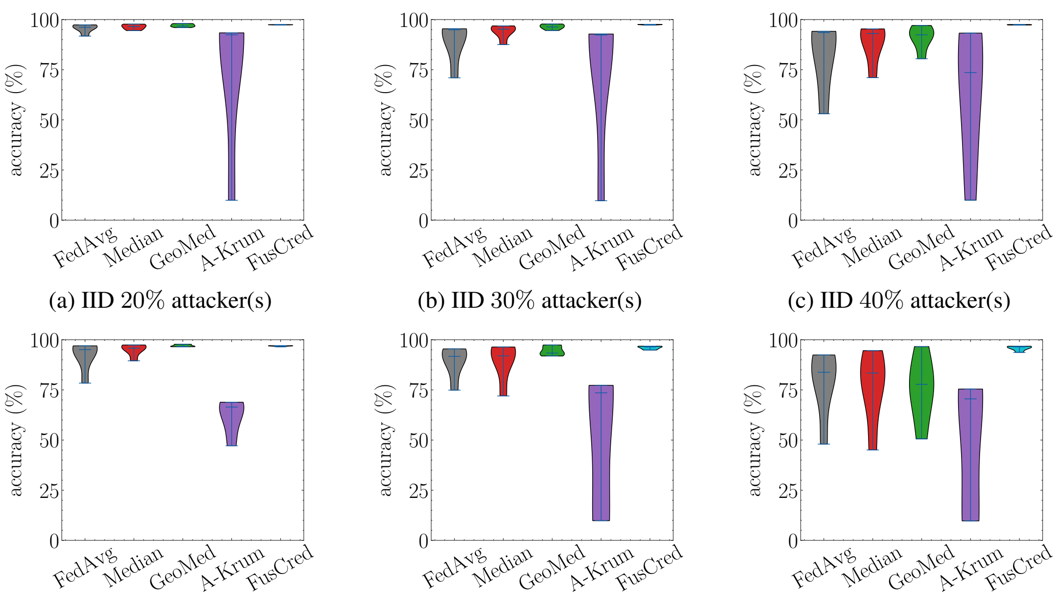
<table>
<tr><td>Parameters</td><td colspan="2">Description</td><td>Value</td></tr>
<tr><td>n</td><td colspan="2">Number of clients</td><td>100</td></tr>
<tr><td>N</td><td colspan="2">Number of edges</td><td>10</td></tr>
<tr><td>lr</td><td colspan="2">Learning rate</td><td>0.01</td></tr>
<tr><td rowspan="3">B</td><td rowspan="3">Batch size</td><td>MNIST</td><td rowspan="2">64</td></tr>
<tr><td>FMNIST</td></tr>
<tr><td>CIFAR-10</td><td>32</td></tr>
<tr><td>E</td><td colspan="2">Number of client training epochs</td><td>2</td></tr>
<tr><td>R</td><td colspan="2">Number of cloud training epochs</td><td>50</td></tr>
<tr><td>ar</td><td colspan="2">The attack ratio</td><td>0%, 20%, 30%, 40%</td></tr>
<tr><td>rs</td><td colspan="2">The random seed</td><td>2023, 2024, 3047</td></tr>
<tr><td>α</td><td colspan="2">Parameter of Dirichlet</td><td>0.2, 0.5, 0.8</td></tr>
</table>



(a) IID MNIST (b) Non-IID MNIST (c) IID FMNIST

(d) Non-IID FMNIST (e) IID CIFAT-10 (f) Non-IID CIFAR-10

Figure 6: Accuracy without malicious attacks on IID and Non-IID datasets. A-Krum is significantly lower than other methods.



(a) IID 20% attacker(s) (b) IID 30% attacker(s) (c) IID 40% attacker(s)

(d) Non-IID 20% attacker(s) (e) Non-IID 30% attacker(s) (f) Non-IID 40% attacker(s)

Figure 7: The violin plot of accuracy of various aggregation methods under 20%, 30%, 40% attacker(s) on IID MNIST and Non-IID MNIST.

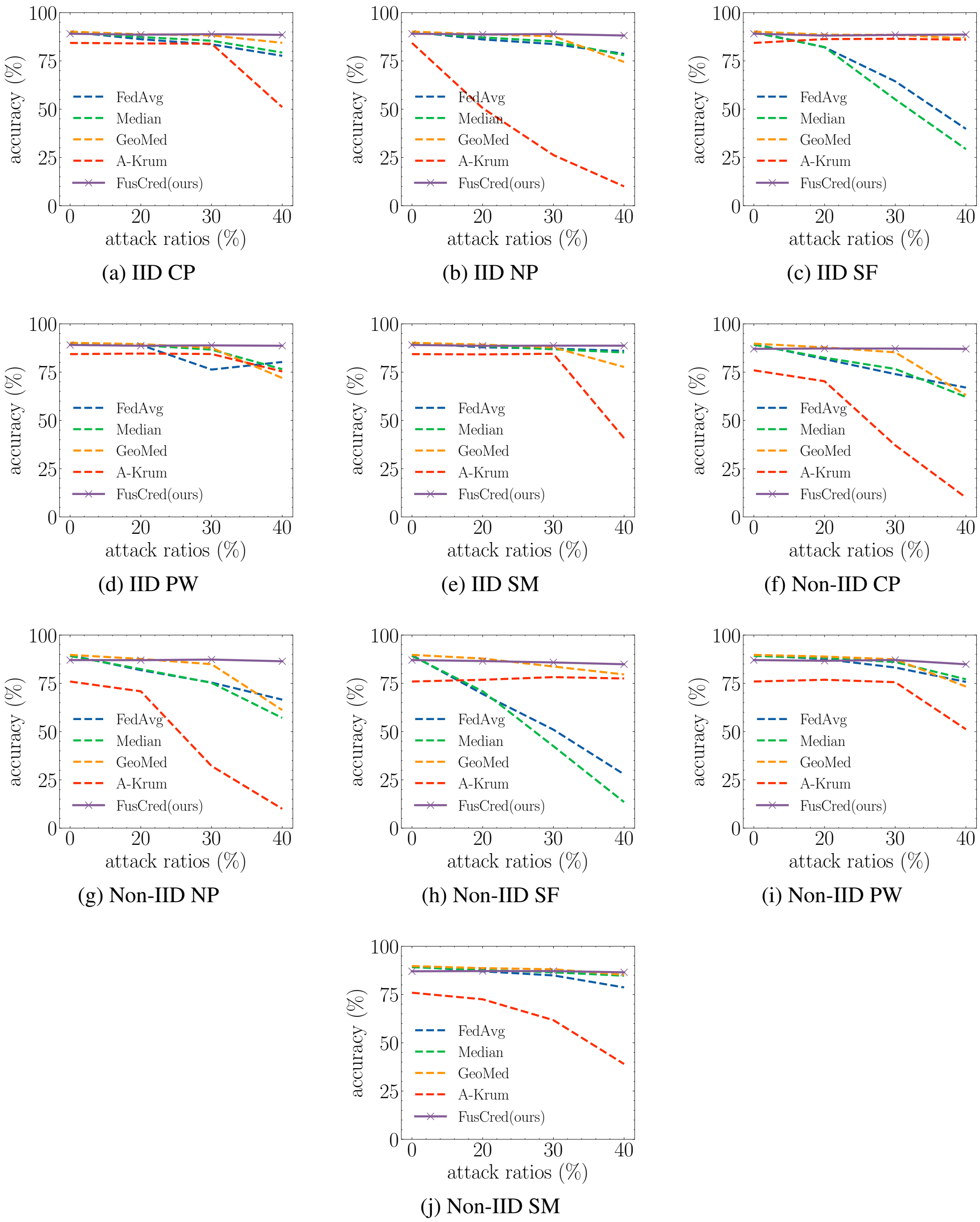


Figure 8: Impact of different attack types on accuracy for IID and Non-IID FMNIST. *FusCred* demonstrates better tolerance against various attack types than other methods.

Table 8: Experiment results overview.

Dataset	Attack ratio	Attack type		FedAvg	Median	GeoMed	A-Krum	FusCred
IID MNIST	0%	-		98.35±0.04	98.15±0.03	98.31±0.02	93.55±0.24	97.49±0.13
	20%	Model poison	CP	96.16±0.15	96.48±0.34	96.77±0.28	92.40±0.49	97.33±0.06
			NP	95.76±0.04	95.56±0.39	96.00±0.58	9.90±0.26	97.43±0.01
			SF	91.75±4.42	94.50±1.13	96.76±0.66	92.87±0.75	97.36±0.11
		Data poison	PW	97.17±0.44	97.71±0.06	97.87±0.04	93.39±0.60	97.50±0.01
			SM	97.36±0.13	97.51±0.07	98.00±0.04	90.43±2.20	97.45±0.04
	30%	Model poison	CP	95.41±0.12	95.73±0.22	95.93±0.22	92.06±0.52	97.42±0.15
			NP	95.18±0.12	94.40±0.02	94.55±0.27	9.70±0.22	97.47±0.07
			SF	89.23±3.95	87.55±2.18	96.33±0.28	92.78±0.11	97.28±0.09
		Data poison	PW	70.96±18.57	95.21±2.11	97.63±0.10	92.77±0.55	97.56±0.03
			SM	94.91±1.68	96.85±0.10	97.78±0.06	92.28±0.49	97.61±0.09
	40%	Model poison	CP	93.86±0.38	94.34±0.43	90.66±1.00	45.87±12.89	97.28±0.12
			NP	94.14±0.47	93.05±0.12	92.39±0.07	9.99±0.18	97.41±0.09
			SF	53.03±8.23	71.03±8.01	94.04±0.65	92.87±0.49	97.28±0.08
		Data poison	PW	76.22±7.13	86.31±6.47	80.46±2.59	93.25±0.11	97.49±0.15
			SM	93.50±0.94	95.35±0.49	97.05±0.14	73.58±26.61	97.55±0.14
Non-IID MNIST	0%	-		98.20±0.01	97.98±0.01	98.11±0.01	74.26±0.60	96.86±0.11
	20%	Model poison	CP	94.78±0.25	95.81±0.07	96.69±0.29	58.68±18.71	96.92±0.07
			NP	95.08±0.19	95.57±0.26	96.52±0.19	47.17±22.65	96.95±0.03
			SF	78.39±10.19	89.51±1.35	96.68±0.27	67.73±3.79	96.49±0.43
		Data poison	PW	95.48±0.76	97.38±0.13	97.63±0.05	68.87±5.36	97.01±0.12
			SM	96.95±0.29	97.11±0.06	97.74±0.03	66.44±6.18	96.92±0.03
	30%	Model poison	CP	89.55±0.24	91.93±1.45	91.90±2.60	9.80±0.00	96.73±0.08
			NP	91.70±0.68	91.82±1.98	93.37±2.20	10.16±0.26	96.80±0.07
			SF	74.89±8.93	71.96±8.50	93.07±1.04	75.72±4.21	96.41±0.10
		Data poison	PW	95.43±1.12	95.13±1.46	97.32±0.17	73.55±4.21	94.86±2.16
			SM	94.69±1.07	96.38±0.02	97.37±0.09	77.27±1.77	95.51±0.61
	40%	Model poison	CP	83.81±0.37	83.60±0.84	69.97±11.71	9.80±0.00	96.59±0.05
			NP	87.84±0.89	83.40±1.05	50.59±12.24	9.70±0.31	96.78±0.20
			SF	47.97±1.81	45.05±12.75	77.73±9.88	70.52±11.04	96.44±0.15
		Data poison	PW	80.75±6.89	75.03±8.58	81.33±6.51	75.42±2.15	93.65±2.07
			SM	92.41±1.21	94.59±0.44	96.60±0.06	73.49±4.64	95.60±0.51
IID FMNIST	0%	-		90.07±0.07	89.70±0.08	90.20±0.06	84.33±0.46	89.08±0.17
	20%	Model poison	CP	86.24±0.05	87.38±0.50	88.79±0.34	84.06±0.46	88.59±0.14
			NP	86.12±0.20	87.20±0.47	88.76±0.34	50.55±12.01	88.73±0.21
			SF	82.07±0.85	82.16±2.21	88.59±0.50	86.30±0.57	88.13±0.35
		Data poison	PW	89.06±0.18	88.92±0.08	89.51±0.07	84.61±0.17	88.67±0.09
			SM	87.85±0.17	88.46±0.28	89.26±0.07	84.19±0.24	88.62±0.09
	30%	Model poison	CP	83.59±0.49	85.46±0.62	88.21±0.07	83.93±0.81	88.85±0.11
			NP	83.73±0.65	85.13±0.61	87.79±0.14	26.26±11.51	88.91±0.10
			SF	64.42±9.73	55.05±18.14	88.51±0.28	86.46±0.09	88.47±0.22
		Data poison	PW	76.32±14.11	86.65±1.11	87.57±0.92	84.38±0.26	88.85±0.08
			SM	87.19±0.53	86.84±0.51	87.78±0.37	84.51±0.56	88.71±0.18
	40%	Model poison	CP	77.65±1.36	79.32±1.69	84.38±1.95	51.07±12.81	88.49±0.25
			NP	78.69±1.08	77.96±1.67	74.48±10.05	10.00±0.00	88.16±0.46
			SF	39.76±3.38	29.31±3.70	86.66±0.51	85.99±0.34	88.64±0.24
		Data poison	PW	80.26±3.05	76.52±2.85	71.96±6.26	75.34±12.93	88.67±0.32
			SM	86.00±0.40	85.17±0.23	77.66±11.42	40.80±23.10	88.70±0.40

Gold, silver, and bronze respectively denote the top three winners.

Table 8 continued from previous page

Dataset	Attack ratio	Attack type		FedAvg	Median	GeoMed	A-Krum	FusCred
Non-IID FMNIST	0%	-		89.57±0.09	89.18±0.03	89.75±0.10	75.94±0.06	87.04±0.48
	20%	Model poison	CP	81.74±1.03	82.49±1.42	87.83±0.32	70.29±9.57	87.24±0.46
			NP	81.82±1.32	82.36±2.12	87.64±0.35	70.86±10.19	87.00±0.79
			SF	69.45±5.15	70.89±5.39	87.83±0.41	76.84±2.85	86.54±1.06
		Data poison	PW	87.54±0.44	88.12±0.41	88.87±0.15	76.85±0.79	86.67±0.99
			SM	87.00±0.30	87.68±0.15	88.67±0.25	72.53±1.97	87.15±0.46
	30%	Model poison	CP	73.97±2.23	76.65±2.71	85.24±1.15	37.11±9.12	87.21±0.20
			NP	75.40±1.62	75.26±3.14	84.91±1.20	32.20±0.40	87.31±0.24
			SF	51.01±9.66	42.47±16.05	83.70±3.54	78.22±0.57	85.82±0.84
		Data poison	PW	83.18±2.39	86.02±1.92	87.42±1.29	75.60±2.36	87.08±0.36
			SM	84.91±0.74	86.51±0.26	88.08±0.07	61.74±14.38	87.15±0.60
	40%	Model poison	CP	67.00±1.56	62.12±5.21	63.11±10.94	10.00±0.00	87.04±0.31
			NP	66.55±1.76	57.08±5.55	61.17±9.92	10.00±0.00	86.43±0.27
			SF	27.87±15.25	13.48±4.92	79.59±2.53	77.51±0.33	84.88±0.60
		Data poison	PW	75.75±0.74	77.10±4.16	73.31±8.46	51.17±20.62	84.86±2.04
			SM	78.68±1.29	84.75±0.15	85.24±0.40	39.00±11.60	86.51±0.46
IID CIFAR-10	0%	-		66.07±0.03	65.63±0.02	65.98±0.10	51.79±0.35	62.79±0.20
	20%	Model poison	CP	42.68±0.99	42.39±0.66	54.79±0.33	51.68±0.11	62.13±0.58
			NP	41.63±0.78	41.28±0.70	53.73±0.93	51.88±0.24	62.08±0.47
			SF	32.65±0.89	27.35±0.36	58.43±0.90	51.85±0.10	62.80±0.20
		Data poison	PW	61.21±0.32	61.52±0.32	63.80±0.17	50.66±1.68	62.35±0.32
			SM	62.41±0.29	63.32±0.08	64.03±0.13	52.33±0.46	62.47±0.27
	30%	Model poison	CP	36.78±1.63	35.45±3.01	34.36±3.80	51.73±0.23	62.07±0.52
			NP	35.99±1.63	34.07±2.19	33.38±3.52	51.72±0.22	61.87±0.52
			SF	28.61±2.77	19.98±1.19	49.39±2.19	51.78±0.27	62.71±0.24
		Data poison	PW	56.00±0.45	57.16±0.89	60.46±0.92	50.02±1.34	62.08±0.05
			SM	59.40±0.33	61.13±0.59	61.93±0.42	50.32±1.12	62.01±0.23
	40%	Model poison	CP	29.96±2.27	28.11±1.35	15.09±0.42	42.74±6.72	61.46±0.39
			NP	30.00±2.05	25.87±0.51	15.06±5.00	42.82±6.58	61.37±0.20
			SF	21.83±3.19	12.69±2.02	33.82±3.61	52.14±0.34	62.34±0.07
		Data poison	PW	48.05±1.63	48.89±2.62	52.58±3.31	52.02±0.31	61.59±0.20
			SM	54.61±0.62	57.46±1.50	56.67±2.58	49.98±1.35	61.31±0.32
Non-IID CIFAR-10	0%	-		66.08±0.07	65.58±0.05	65.87±0.09	51.97±0.01	62.53±0.12
	20%	Model poison	CP	43.39±0.94	44.62±2.77	57.63±3.15	51.56±0.10	62.67±0.22
			NP	42.21±1.34	43.55±3.38	57.19±3.80	51.90±0.21	62.78±0.17
			SF	33.37±3.37	30.50±3.93	59.81±3.34	51.85±0.23	62.90±0.44
		Data poison	PW	60.87±0.54	61.36±0.48	63.80±0.20	51.93±0.20	62.60±0.26
			SM	62.03±0.60	62.99±0.26	63.98±0.12	50.74±1.77	62.70±0.15
	30%	Model poison	CP	35.82±1.12	35.94±1.77	35.37±5.61	51.96±0.02	62.42±0.20
			NP	35.01±0.84	33.92±1.50	37.25±3.49	51.67±0.15	62.42±0.20
			SF	26.99±2.38	18.92±3.61	48.74±6.03	52.62±0.49	62.71±0.11
		Data poison	PW	56.10±0.66	57.65±1.00	60.71±0.52	51.11±1.48	62.41±0.13
			SM	59.13±0.17	61.43±0.29	61.89±0.25	50.51±2.38	61.93±0.61
	40%	Model poison	CP	29.42±0.37	29.52±1.56	15.26±1.53	42.71±7.14	61.42±0.39
			NP	29.36±0.50	26.43±1.21	18.23±3.54	43.69±6.29	61.51±0.36
			SF	21.58±3.02	12.17±1.23	37.00±1.44	51.76±0.78	62.15±0.27
		Data poison	PW	46.59±2.04	48.53±0.96	53.48±0.68	49.72±2.82	61.82±0.04
			SM	54.08±1.20	57.76±1.25	57.68±0.93	49.52±3.13	61.71±0.25

Gold, silver, and bronze respectively denote the top three winners.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our work introduces a novel FL framework, Clustered Semi-synchronous Hierarchical Federated Learning (CSS-HFL), to address the limitations of existing FL frameworks, namely the stragglers effect, network congestion, and robustness against poisoning attacks. We also propose a robust algorithm, Fusion Credibility (FusCred), to enhance the robustness of the CSS-HFL framework.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix E for the limitations of existing FL frameworks.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Please refer to Appendix D.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide detailed information on the experimental settings necessary to reproduce the main experimental results in Section 4.1.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access to the data and code, along with detailed instructions to reproduce the main experimental results in the supplemental material.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Section 4.1 for the detailed experimental settings.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars in the experimental results to show the statistical significance of the experiments.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Section 4.1 provides detailed information on the computer resources used in the experiments.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work addresses the limitations of existing FL frameworks, namely the stragglers effect, network congestion, and robustness against poisoning attacks, which are crucial for the real-world FL system implementation. This work can potentially provide positive societal impacts by improving the efficiency and robustness of FL systems.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve data or models that have a high risk for misuse.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We include the license and terms of use for all assets used in the paper in the supplemental material.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our work does not introduce new assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing experiments or research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve research with human subjects.