2024 IEEE 44th International Conference on Distributed Computing Systems Workshops (ICDCSW) | 979-8-3503-5471-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCSW63686.2024.00009

FLAMINGO: Adaptive and Resilient Federated Meta-Learning against Adversarial Attacks

Md Zarif Hossain[1,2], Ahmed Imteaj[1,2], Abdur R. Shahid[1]
[1]*School of Computing, Southern Illinois University, Carbondale, IL, USA*
[2]*Security, Privacy and Intelligence for Edge Devices Laboratory (SPEED Lab)*
mdzarif.hossain@siu.edu, imteaj@cs.siu.edu, shahid@cs.siu.edu

***Abstract*—In today's data-centric world, the synergy between Meta Learning and Federated Learning (FL) signifies a new era of technological advancement, driving rapid adaptation, improved model generalization, and collaborative model training across decentralized networks. This fusion, known as Federated Meta-Learning (FML), emerges as a cutting-edge solution for resource-constrained edge devices, enabling the production of personalized models with limited training data. However, FML navigates a complex terrain, balancing efficiency with security, as adversarial attacks on edge devices pose significant threats. These attacks risk introducing bias and undermining the integrity of model training, a critical concern given the typically sparse data on edge devices. This paper explores the intricate dynamics of FML amidst such adversarial challenges, introducing a novel algorithm, FLAMINGO. FLAMINGO is designed to conduct adversarial meta-training coupling with data augmentation and consistency regularization strategies, thereby strengthening the meta-learner's defenses against adversarial attacks. This strategic approach not only protects meta-learners against adversarial threats but also prevents overfitting, striving a balance between privacy, security, and technological efficiency, all while optimizing communication costs in the FML landscape. We have released our code on GitHub[1], which is publicly accessible.**

***Index Terms*—Federated meta-learning, Adversarial attack, Meta training, Data augmentation, Consistency regularization.**

I. Introduction

A. Motivation

The rapid advancement of IoT devices and their continuous influx of data generation has led to new opportunities of Artificial Intelligence of Things (AIoT), driving AI to the forefront of edge computing. Tailoring edge intelligence can reduce down time-to-action latency, conserve bandwidth and resource costs, and offer privacy and security measures [1]. Nevertheless, a single edge device can hardly perform effective edge intelligence mainly due to data scarcity, and computational and memory constraints. Hence, collaborative edge learning is gaining widespread popularity, enabling edge devices to pool their resources to accomplish resource-intensive tasks. Building upon the fusion of FL [2] and meta-learning [3], the concept of Federated Meta-Learning (FML) has been introduced, with a shared objective of fostering collaboration at the edge. FML is particularly useful when resource-constraint edge devices possess limited quantities of data samples and aim to obtain personalized models through collaborative learning. In FML, the edge devices learn from a shared meta-model provided by the server, allowing all network edge devices to quickly produce a model through just a few gradient descent optimization steps with their respective local datasets. However, alongside its remarkable advantages, Meta-learning also presents certain pitfalls. One notable concern is meta-learners' susceptibility to overfitting issues. Additionally, when meta-learners are subjected to adversarial attacks, it can lead to substantial instability in model training, especially since each edge device is exposed to a small set of data samples during training. Furthermore, In centralized learning, if a portion of the data gets attacked, the remaining clean data can still be used to train the model. Conversely, in FML, such recovery from attacks is not as feasible due to the scarcity of data, rendering the impact of adversarial attacks far more severe. In addition, traditional centralized defense mechanisms may not suffice in the context of FL due to their distributive nature, necessitating robust and decentralized defense strategies to safeguard against adversarial incursions effectively.

To this end, we tailor a novel Adversarial Federated Meta Training technique, called FLAMINGO for facilitating few-shot learning scenarios while addressing the aforementioned challenges of meta-learning. Our proposed approach leverages *Consistency Regularization (CR)* and diverse *Data Augmentation (DA)* strategies to safeguard against adversarial attacks and avert the meta-learner from overfitting.

B. Literature Reviews

1) FL-based Adversarial Attacks and Defense Strategies: In a federated setting, malicious clients has the capacity to influence the outcome of the global model [4]. Without adequate security measures, adversaries can alter the data and model in different phases of the intelligent system development to craft adversarial models [5], [6]. Adversaries may introduce backdoor in the model through malicious updates to mislead the global model to behave incorrectly when triggered by certain inputs (backdoor attack [7]), or manipulate input data at inference time to cause the model to make wrong predictions and decisions (evasion attack [8]). This active area of research focuses on developing new attack strategies to identify vulnerabilities of federated IoT systems and devise efficient and effective defense mechanisms to make the system secure against sophisticated attack vectors. For instance, in PoisonGAN [9], authors proposed an approach to mimic the training data distribution of benign clients and generate poisoned data to compromise the global model by utilizing

[1] https://github.com/speedlab-git/Flamingo-Adversarial-FML.git
Corresponding author: Ahmed Imteaj (imteaj@cs.siu.edu)

979-8-3503-5471-3/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCSW63686.2024.00009

generative adversarial network (GAN) in the federated setting. Kim et al. [8] analyzes a new threat model of “internal evasion attacks” to highlight the vulnerability of FL due to model similarity among the clients.

2) Adversarial Attacks in Meta Learning and Defense Strategies: Recent studies [3], [10] have demonstrated that meta-learners are more vulnerable to adversarial samples than typical DNN models, due to the nature inherent to few-shot learning. Particularly, meta-learners face substantial performance drops when exposed to first-order attacks that employ the ℓ_p norm, such as Projected Gradient Descent (PGD) [11], which are imperceptible to the human vision [12]. In response, several defense strategies have been proposed to enhance the adversarial resilience of meta-learners. One such approach is Adversarial Training [13], which involves incorporating adversarial samples to the training set to train the model or learner. Another strategy is Defensive Distillation [14], focuses on enhancing robustness through the use of model distillation. However, a recent study [15] demonstrated that conventional adversarial training methods frequently suffer from overfitting. In contrast, distillation techniques are computationally demanding and pose challenges in optimization. Furthermore, widely adopted defense approaches for deep neural networks (DNNs), such as semi-supervised robust training [16] do not easily translate to the realm of meta-learning due to the inherent bi-level optimization nature of meta-learners.

To the best of our knowledge, FML in the context of first-order adversarial samples remained unexplored. We demonstrate that a conventional approach is inadequate against first-order adversary attacks such as PGD, prompting us to introduce a new and successful technique called FLAMINGO.

C. Contributions

- Introduce a cutting-edge adversarial meta-training technique designed to enhance the robustness of FML models, integrating consistency regularization with varied data augmentation tactics for improved resilience.
- Employ a strategic mix of both weak and strong data augmentations during the meta-training phase to enhance the model’s defenses against adversarial attacks and to prevent overfitting.
- Demonstrate how FLAMINGO excels beyond traditional FL approaches in safeguarding against adversarial attacks, while minimizing resource consumption.

II. Adversarial Federated-Meta Learning

The high-level idea of adversarial federated meta-learning is to train distributed edge devices by incorporating adversarial examples directly into their training samples. Exposing meta-learners to adversarially perturbed examples during their training enhances their resilience against the subtle yet potent adversarial attacks that have the potential to compromise their performance. Our objective is to develop a robust meta-learner for FL that takes local data samples from their respective clients as input $\mathcal{D} = \{D_1, D_2..., D_n\}$ and returns a learner with parameter θ that optimizes the average classification accuracy on the corresponding test sets $\mathcal{D}^t = \{D_1^t, D_2^t..., D_n^t\}$. During meta-testing phase, we assess the meta-learner’s ability to generalize to new tasks, even when these tasks may involve adversarial samples in both the training and testing datasets. An ideal meta-learner should yield a learning model proficient in handling new tasks exclusively with clean samples (without perturbation) while exhibiting only marginal performance decline with tasks including adversarial samples.

A. Federated Meta Learning

In this paper, we select Meta-SGD [10] as the baseline meta-learner, attributing to its enhanced generalization and optimization capabilities over MAML [3]. Meta-training follows an iterative approach, where each iteration involves sampling a batch of tasks from a task distribution $\mathcal{T}$ across a meta-training set. Within each iteration, a task T is randomly sampled with a specified number of classes, referred to as ‘ways’, and their associated samples, known as ‘shots’. For instance, in a scenario termed ‘5 way 1 shot learning’, each task includes five randomly selected classes, with one data sample available for each class. A sampled task T consists of a support set, $\mathcal{S}_T = \{(x_m, y_m)\}_{m=1}^{|\mathcal{S}_T|}$ and a query set, $Q_T = \{(x_m^{'}, y_m^{'})\}_{m=1}^{Q_T}$. Support and query sets are kept disjoint to maximize the ability of generalization for the meta-learner. During Meta-training, Meta-SGD maintains parameter θ that acts as the initial value of the parameter model f for each task T. Note that here θ embodies the state of a learner, serving as a basis for initializing the learner for any new task. Initially, the model f_θ gets trained on the support set $\mathcal{S}_T$ and generates an updated model f_{θ_T}, using one or multiple gradient descent steps with training loss $\mathcal{L}_{\mathcal{S}_T}(\theta) = \frac{1}{|\mathcal{S}_T|}\sum_{(x,y)\in\mathcal{S}_T} \ell(f_\theta(x), y)$. This process is also referred to as the *inner update*. The updated model f_{θ_T} is then evaluated on the query set $\mathcal{Q}_T$ and generates meta test loss $\mathcal{L}_{Q_T}(\theta_T) = \frac{1}{|Q_T|}\sum_{(x',y')\in Q_T} \ell(f_{\theta_T}(x'), y')$. Finally, to minimize the test loss, the meta-learners’ parameters go through an update process, referred to as the *outer update*. Unlike MAML [3], Meta-SGD incorporates α, an inner learning rate specific to each task. This enables the meta-learner to simultaneously learn the initialization parameter θ and inner learning rate α from the given tasks, enhancing the optimization and speed of the Meta-SGD algorithm. The optimization function of Meta-SGD can be expressed as follows:

$$\min_{\theta,\alpha} \mathcal{E}_{T\sim\mathcal{T}}\left[\mathcal{L}_{Q_T}\left(\theta - \alpha \circ \nabla\mathcal{L}_{\mathcal{S}_T}(\theta)\right)\right] \qquad (1)$$

Here, α is a vector of the same size as θ and $\circ$ denotes element-wise product. To incorporate meta-training with FL, an FL server initializes parameter θ and transmits it to the selected clients. Each client c independently trains a meta-learner using their own local support set, denoted as $\mathcal{D}_S^T$, and then evaluates the model’s performance on a query set, represented as $\mathcal{D}_Q^T$. The resulting test loss $\mathcal{L}_{D_Q^T}(\theta_T)$ is subsequently utilized to adjust the meta-learner’s parameters, and these updated parameters, denoted as q_c, are communicated back to the server. The server retains the initialized parameters and updates them by aggregating weights received from the subset of clients.

B. Adversarial Meta Training

We introduce a novel adversarial meta-training technique for strengthening the meta-learner against adversarial attacks. This approach considers the inclusion of adversarial samples during meta-training. Central to our approach is the optimization of a Consistency Regularization (CR) loss, which encourages the model to maintain consistent predictions across perturbed versions of the same input, as illustrated in Fig. 1.

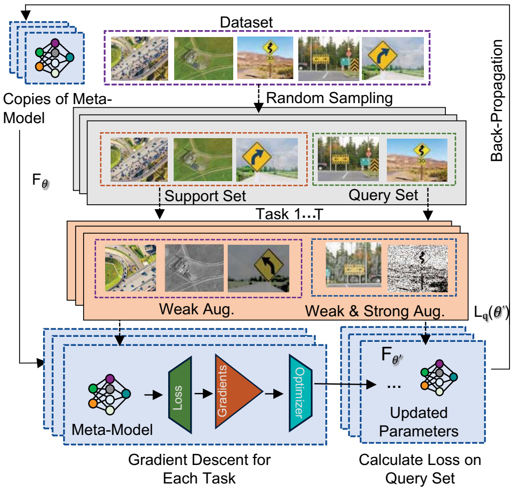

Fig. 1: Proposed Adversarial Meta-Learning approach.

Our methodology involves two key augmentation strategies for meta-training: weak augmentation and strong augmentation, aiming to minimize the discrepancy in the model's predictions between these augmentations. The process begins with the random selection of a task T from a task distribution $\mathcal{T}$ for baseline meta-learner, which itself is derived from the local data samples $\mathcal{D}$ of the respective clients. For each task within the support set D_S^T, we apply a weak augmentation strategy on the data samples. Formally, for task T, we denote the weakly augmented sample as $x_w = x + p_w$, where p_w represents the weak augmentation. Consider $X = \{x_1, x_2, ..., x_n\}$ as the original, unmodified data samples from the support set. Upon applying a weak augmentation, these samples are transformed into a set of weakly augmented samples, denoted by $X_w = \{x_{w_1}, x_{w_2}, ..., x_{w_n}\}$. The meta-learner, represented as f_θ, is then trained using both the original (clean) and the weakly augmented samples, resulting in the updated parameters θ_T for the clean samples and θ_{T_w} for the weakly augmented samples, respectively. In this process, f_θ also generates the support set loss $\mathcal{L}_{D_S^T}$, which is defined as follows:

$$\mathcal{L}_{D_S^T}(\theta) = \frac{1}{\left|D_S^T\right|} \sum_{(x,y) \in D_S^T} \ell\left(f_\theta\left(x\right), y\right) + \frac{1}{\left|D_S^T\right|} \sum_{(x_w,y) \in} D_{S_w}^T \ell\left(f_\theta\left(x_w\right), y\right) \tag{2}$$

In this equation, $D_{S_w}^T$ represents the weakly augmented support set. During the *inner update* phase, our AMT strategy calculates its initial CR training loss, $\mathcal{L}_{CR_s}$, by calculating the Kullback-Leibler (KL) divergence between predictions from the original model and its weakly augmented counterpart. The primary objective is to minimize this KL loss, fostering a meta-learner that is both more robust and generalized. The CR training loss for the *inner update* is expressed as follows:

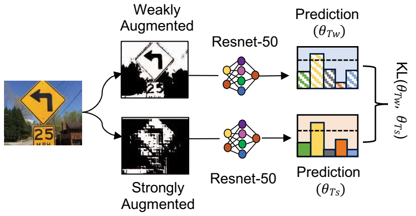
Weakly Augmented
Resnet-50
Prediction (θ_{Tw})
KL(θ_{Tw}, θ_{Ts})
Strongly Augmented
Resnet-50
Prediction (θ_{Ts})

Fig. 2: Consistency Regularization for Query Set.

$$\mathcal{L}_{CR_s} = \mathrm{KL}\left(f(\cdot;\theta_T), f\left(\cdot;\theta_{T_w}\right)\right) \tag{3}$$

We aggregate the *CR* training loss $\mathcal{L}_{CR_s}$ with support set loss $\mathcal{L}_{D_S^T}$ to calculate combined loss,

$$\mathcal{L}_s(\theta) = \frac{1}{|D_S^T|} \sum_{(x,y) \in D_S^T} \ell\left(f_\theta\left(x\right), y\right) + \frac{1}{|D_S^T|} \sum_{(x_w,y) \in D_{S_w}^T} \ell(f_\theta\left(x_w\right), y) + \mathcal{L}_{CR_s} \tag{4}$$

Here, λ is a hyperparameter that controls the strength of the CR term. During the *outer update* phase, both weak and strong augmentations are applied to the samples in the query set D_Q^T. For strong augmentation, we opt for RandAugment [17] strategy, which applies a series of randomly selected augmentations to the input images. These augmentations include operations like rotation, scaling, shearing, translation, and color adjustments. Formally, we can also represent strongly augmented sample by $x_s^{'}$, where $x_s^{'} = x^{'} + p_s$ and p_s denotes strong augmentation. Additionally, clean data samples can be denoted as $X^{'} = \{x^{'}, x^{'}, ..., x^{'}\}$, with $X's = \{x_{s_1}^{'}, x_{s_2}^{'}, ..., x_{s_n}^{'}\}$ denoting strongly augmented samples and $X_w^{'} = \{x_{w_1}^{'}, x_{w_2}^{'}, ..., x_{w_n}^{'}\}$ representing weakly augmented samples.

Upon generating strongly augmented $D_{Q_s}^T$ and weakly augmented $D_{Q_w}^T$ versions of the query set D_Q^T, the updated model f_{θ_s} from inner update undergoes one or more gradient updates using these augmented samples. This process leads to the generation of updated parameters $\theta_{T_w}^{'}$ for weakly augmented samples and $\theta_{T_s}^{'}$ for strongly augmented samples. Moreover, the function f_{θ_s} produces an updated query set loss, denoted as $\mathcal{L}_{D_Q^T}$. To reinforce CR, a comparison of the dissimilarity between the newly updated parameters is made by calculating the KL divergence. This CR process, aiming to maintain consistent predictions by the model across different levels of data augmentation, is presented in Fig. 2. We calculate the CR loss, $\mathcal{L}_{CR_q}$ for the query set as follows:

$$\mathcal{L}_{CR_q} = \mathrm{KL}\left(f(\cdot;\theta_{T_s}^{'}), f\left(\cdot;\theta_{T_w}^{'}\right)\right) \tag{5}$$

After that, we aggregate the CR loss $\mathcal{L}_{CR_q}$ with the query set loss $\mathcal{L}_{D_Q^T}$ that produces the combined loss,

$$\mathcal{L}_q(\theta_s) = \frac{1}{\left|D_Q^T\right|} \sum_{(x',y') \in D_{Q_s}^T} \ell\left(f_{\theta_s}\left(x_s'\right), y'\right) + \frac{1}{\left|D_Q^T\right|} \sum_{(x_w',y') \in D_{Q_w}^T} \ell\left(f_{\theta_s}\left(x_w'\right), y'\right) + \mathcal{L}_{CR_q} \tag{6}$$

Here, λ' serves as a hyperparameter that controls the influence of the CR loss ($\mathcal{L}_{CR_q}$) on the query set. As a final step for our AMT strategy, we combine support set loss ($\mathcal{L}_s$) and query set loss ($\mathcal{L}_q$). The final optimization objective, i.e., minimizing the combined loss using an inner learning rate parameterized by α, is represented as follows:

$$\min_{\theta,\alpha} \mathcal{E}_{T\sim\mathcal{T}} \left[\mathcal{L}_q \left(\theta - \alpha \circ \nabla \mathcal{L}_s(\theta)\right)\right] \tag{7}$$

In our proposed AMT mechanism, the incorporation of both strong and weak augmentation strategies plays a pivotal role in fine-tuning the meta-learner to enhance its resilience against unforeseen adversarial attacks and prevent overfitting. Our experiments reveal that using strong augmentation on the support set leads to early divergence of the model during training, and consequently, strong augmentation is deliberately applied only to the query set. We delve further into this observation in later sections, providing evidence through empirical results. The procedure for our adversarial meta-training approach, in the context of FL, is outlined in Algorithm 1.

Algorithm 1: FLAMINGO: FL and Adversarial Meta-Training with Consistency Regularization

```
Executes on Server
$\theta$ and $\alpha$ gets initialized
for each iteration $i = 0, 1, 2, 3 \ldots$ do
    Sample random set of $U_c$ of $n$ clients and distribute $\theta$, $\alpha$
    for each client $c \in U_c$ do
        Query set loss $q_c \leftarrow$ AdMetaTraining$(\theta, \alpha)$
    Update parameters $(\theta, \alpha) \leftarrow (\theta, \alpha) - \frac{\beta}{n} \sum_{c \in U_c} q_c$
Executes on Client $c$
AdMetaTraining $(\theta, \alpha)$
    Sample support set $D_S^T$ and query set $D_Q^T$ from task distribution $\mathcal{T}$
    $D_{S_w}^T, D_{Q_w}^T \leftarrow$ WeakAugmentation($D_S^T$, $D_Q^T$)
    $D_{Q_s}^T \leftarrow$ StrongAugmentation($D_Q^T$)
    $\mathcal{L}_{D_S^T}(\theta) \leftarrow \frac{1}{|D_S^T|} \sum_{(x,y) \in D_S^T} \ell\left(f_\theta\left(x\right), y\right) + \frac{1}{|D_S^T|} \sum_{(x_w,y) \in D_{S_w}^T} \ell\left(f_\theta\left(x_w\right), y\right)$
    $\mathcal{L}_{CR_s} \leftarrow \mathrm{KL}\left(f(\cdot;\theta_T), f\left(\cdot;\theta_{T_w}\right)\right)$
    $\mathcal{L}_s(\theta) \leftarrow \mathcal{L}_{D_S^T} + \lambda \mathcal{L}_{CR_s}$
    $\theta_s \leftarrow \theta - \alpha \circ \nabla \mathcal{L}_s(\theta)$
    $\mathcal{L}_{D_Q^T}(\theta_s) \leftarrow \frac{1}{|D_Q^T|} \sum_{(x'_s,y') \in D_{Q_s}^T} \ell\left(f_{\theta_s}\left(x'_s\right), y'\right) + \frac{1}{|D_Q^T|} \sum_{(x'_w,y') \in D_{Q_w}^T} \ell\left(f_{\theta_s}\left(x'_w\right), y'\right)$
    $\mathcal{L}_{CR_q} = \mathrm{KL}\left(f(\cdot;\theta'_{T_s}), f\left(\cdot;\theta'_{T_w}\right)\right)$
    $\mathcal{L}_q(\theta_s) \leftarrow \mathcal{L}_{D_Q^T} + \lambda' \mathcal{L}_{CR_q}$
    $q_c \leftarrow \nabla_{(\theta,\alpha)} \mathcal{L}_q(\theta_s)$
    return $q_c$ to server
```

C. Launching Intrusion Attack in an FL Environment

To assess the resilience and effectiveness of FLAMINGO, we employ an enhanced form of the "first-order adversary attack", PGD [11]. Perturbations generated by PGD are generally imperceptible to human vision as shown in Fig. 3, and have the capability to mislead deep learning models or meta-learners towards misclassification, often with a high level of confidence in the resulting prediction. PGD achieves adversarial perturbation by iteratively adjusting the pixel values of an input image based on the gradient of the loss function. As it is challenging to establish a metric quantifying the capacity of human vision, p-norms are commonly employed to regulate the magnitude and quantity of perturbations introduced into an image. The p-norm, denoted as ℓ_p, calculates the distance $\|x - x'\|_p$ in the input space between an original input sample x and adversarial example x'. The parameter p ranges from 0 to ∞, where ℓ_∞ quantifies the maximum disparity across all corresponding pixels between the perturbed image and its original counterpart. PGD operates in a sequential manner, calculating the gradient of the loss function with respect to the image's pixel values at each step. This calculated gradient is multiplied by a predetermined step size, ϵ, and subsequently added to the pixel values to produce a perturbed version of the image. The resulting perturbed image is then clipped to ensure pixel values remain within a valid range (e.g., $[0, 255]$).

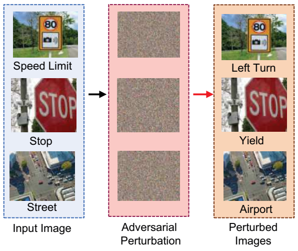

Fig. 3: Adversarial attack with PGD.

III. Experimental Results

A. Datasets

The datasets chosen for this research are designed to accommodate devices with limited resources, allowing the FL-meta models to efficiently address challenges posed by limited computational power. Experimental evaluations are conducted the EuroSAT [18] and LISA [19] datasets to evaluate the performance and robustness of the proposed FLAMINGO.

B. Performance Evaluation

We conducted all of our experiments with 5 FL clients and executed up to 40 communication rounds of FL training. For our proposed method FLAMINGO and state-of-the-art vanilla Meta-SGD, we additionally performed 50 meta-training rounds in order to facilitate few-shot learning. In addition to that, we conduct 5 way 1 shot and 5 way 5 shot task classifications for proposed FLAMINGO and the vanilla Meta-SGD. We used ResNet-50 as the baseline model and to regulate the effect of CR, We set both of the CR terms, λ and λ' to 1. Subsequently, we assess their performance in comparison to conventional FL algorithms like FedProx [20] and FedNova [21], both equipped with robust adversarial learning method called FGSM [22] adversarial training. For our attack model, we perform PGD attack on selected clients with 10 iterations under $\epsilon = 2/255$ with ℓ_2 norm constraint.

Fig. 4 illustrates the performance of conventional FL models such as FedProx [20], FedNova [21], and Vanilla Meta-SGD under the influence of a PGD attack. In the federated setting, clients are randomly chosen for the PGD attack, while the remaining clients are trained using clean samples. Fig. 4 (left) demonstrates the instability observed in the server accuracy of these FL models on the EuroSAT dataset when subjected to the PGD attack. As anticipated, the same pattern of instability is observed in our experiment with Lisa dataset in Fig. 4 (right), with a majority of the models exhibiting significant divergence in the early stages of training due to the adversarial attack. In both scenarios, Meta-SGD (5 way 5 shot) demonstrates a degree of resilience against the PGD attack, although it ultimately falls short of achieving the desired accuracy.

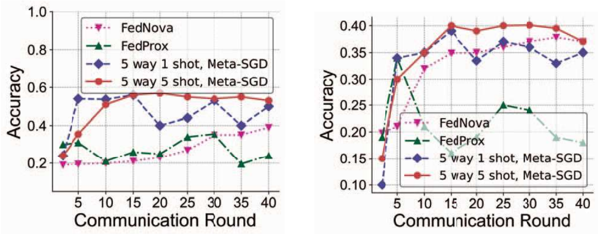

Fig. 4: Performance of conventional FL methods in the presence of PGD attack under l_2 norm: Server accuracy on EuroSAT dataset (left) and on LISA dataset (right).

From the above discussion, a natural question may arise: how is it feasible to attain a server test accuracy of 50% despite the impact of the PGD attack? To address this query, we present our comprehensive experiment with Vanilla Meta-SGD on the EuroSAT dataset in Fig. 5. It is evident from Fig. 5 (left) that even though 3 of the attacked clients are yielding poor performance, with accuracy below 20%, the 2 clients trained with clean samples are achieving over 90% accuracy. In addition, Fig. 5 (right) depicts how attacked clients diverge from loss minimization at an early stage. Given the federated setting, all client updates are merged on the server. This implies that the server model is created via contributions from models trained on both clean and adversarial samples. The server's incorporation of updates from models trained on clean samples generates the performance outcomes shown in Fig. 4.

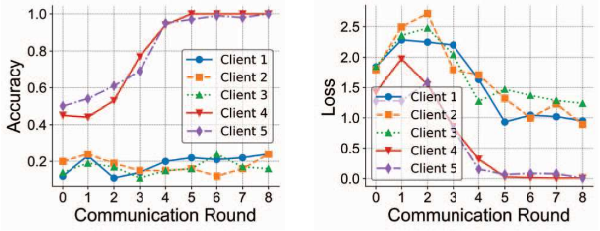

Fig. 5: Accuracy (left) and loss (right) for individual training client on the EuroSAT dataset utilizing Meta-SGD.

Next, we explore the effectiveness of FLAMINGO, under the PGD attack. We conduct an empirical analysis on FLAMINGO using both the EuroSAT dataset (Fig. 6) and the Lisa dataset (Fig. 7). In terms of server accuracy illustrated in Fig. 6 (left) and Fig. 7 (left), FLAMINGO with a 5 way 5 shot task classification surpasses both FedProx and FedNova employing the FGSM adversarial training method. Furthermore, it exhibits enhanced stability across both datasets. Additionally, We can also observe that FLAMINGO with 5 way 5 shot classification task, exhibits much more stable performance due to having more data samples per class during training, in contrast to the 5 way 1 shot classification task. This stability is reflected in the loss minimization performance, as depicted in Fig. 6 (right) and Fig. 7 (right), where FLAMINGO exhibits superior and more stable results compared to FedProx and FedNova [21] with FGSM. Furthermore, the difference between the best accuracy and final accuracy achieved using FLAMINGO on both datasets is marginally low (below 3%), which indicates FLAMINGO effectively mitigates overfitting.

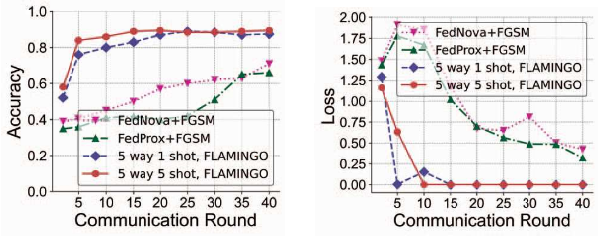

Fig. 6: Performance comparison of different adversarial training methods on EuroSAT dataset in the presence of PGD attack under l_2 norm: Server accuracy across multiple FL communication rounds (left) and loss minimization (right).

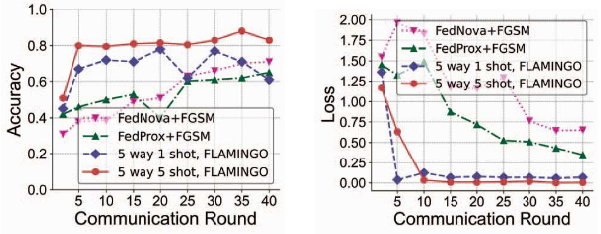

Fig. 7: Server accuracy (left) and loss of various adversarial training methods (right) on LISA dataset in the presence of PGD attack under l_2 norm.

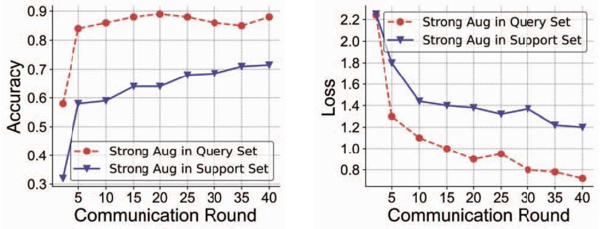

Fig. 8: Server accuracy (left) and loss (right) of FLAMINGO on EuroSAT with PGD adversarial images.

Now, we demonstrate the reasoning behind exclusively applying strong augmentation in the query set rather than the support set. Our empirical analysis, illustrated in Fig. 8 for EuroSAT, involving 5 way 5 shot classification tasks and step size $\epsilon = 2/255$, provides additional validation for our initial hypothesis. Fig. 8 (left) reveals that FLAMINGO's server accuracy decreases when strong augmentation is applied to the support set, as opposed to its application in the query set. This trend is consistently reflected in loss minimization, as shown in Fig. 8 (right). The incorporation of strong augmentation leads to premature divergence during the training process in FLAMINGO, resulting in notable decline in accuracy performance, as depicted in Fig. 8 (left). Moreover, the findings presented in Fig. 9 further corroborate our initial claim that FLAMINGO is more resource-efficient compared to conventional FL with FGSM adversarial training [22].

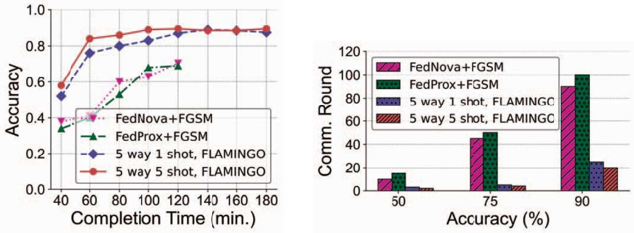

Fig. 9: Analyzing Resource Consumption of Adversarial Training Methods on EuroSAT Dataset with ℓ_2 PGD Attacks: Server Accuracy vs. Completion Time (left) and Communication Rounds for Target Accuracy (right).

FLAMINGO achieves the desired accuracy swiftly compared to other methods, surpassing 80% within just 60 minutes (presented in Fig. 9 (left)). Although it takes a slightly higher duration to complete 40 communication rounds, FLAMINGO shows potential for early termination once the desired accuracy is reached. Furthermore, Fig. 9 (right) underscores FLAMINGO's effectiveness in requiring fewer communication rounds to achieve the target accuracy. It surpasses 75% accuracy within just 5 communication rounds, while other FL methods with FGSM adversarial training require up to 50 rounds to achieve the same 75% accuracy threshold.

IV. Conclusion

In this paper, we introduced FLAMINGO, a novel AMT method, designed to enhance the robustness of FML achieved through the integration of adversarial training and consistency regularization. Within FLAMINGO, we employ diverse augmentation strategies during the meta-training process to enhance the model's resilience against adversarial attacks as well as prevent overfitting. An extensive series of ablation studies validate our design choices, particularly the utilization of strong augmentation exclusively on query sets to prevent premature divergence. Through comprehensive experiments conducted on image classification datasets, including EuroSAT and LISA, underscore the effectiveness of FLAMINGO. In the face of PGD attacks employing 10 iterations, FLAMINGO demonstrates remarkable performance, reaching faster convergence in such demanding setting. This achievement is in stark contrast to vanilla Meta-SGD (approximately 50%) and FedNova with FGSM (about 65%) for EuroSAT and achieves over 20% increase in accuracy on LISA dataset compared to other models. Additionally, it delivers a notable 50% increase in accuracy on the LISA dataset when compared to other models. Furthermore, FLAMINGO exhibits accelerated convergence, achieving 80% accuracy on EuroSAT in less than half the time compared to alternative methods.

V. Acknowledgement

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. CRII-IIS-RI-2348145. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, and W. Zhuang, "Efficient federated meta-learning over multi-access wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1556–1570, 2022.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[3] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[4] M. Z. Hossain and A. Imteaj, "Fedavo: Improving communication efficiency in federated learning with african vultures optimizer," *arXiv preprint arXiv:2305.01154*, 2023.

[5] A. R. Shahid, A. Imteaj, S. Badsha, and M. Z. Hossain, "Assessing wearable human activity recognition systems against data poisoning attacks in differentially-private federated learning," in *2023 IEEE International Conference on Smart Computing*. IEEE, 2023, pp. 355–360.

[6] M. Z. Hossain, A. Imteaj, S. Zaman, A. R. Shahid, S. Talukder, and M. H. Amini, "Flid: Intrusion attack and defense mechanism for federated learning empowered connected autonomous vehicles (cavs) application," in *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, 2023, pp. 1–8.

[7] T. Liu, X. Hu, and T. Shu, "Facilitating early-stage backdoor attacks in federated learning with whole population distribution inference," *IEEE Internet of Things Journal*, 2023.

[8] T. Kim, S. Singh, N. Madaan, and C. Joe-Wong, "Characterizing internal evasion attacks in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 907–921.

[9] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3310–3322, 2020.

[10] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[12] M. Goldblum, L. Fowl, and T. Goldstein, "Adversarially robust few-shot learning: A meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 886–17 895, 2020.

[13] C. Yin, J. Tang, Z. Xu, and Y. Wang, "Adversarial meta-learning," *arXiv preprint arXiv:1806.03316*, 2018.

[14] N. Papernot and P. McDaniel, "Extending defensive distillation," *arXiv preprint arXiv:1705.05264*, 2017.

[15] R. Wang, K. Xu, S. Liu, P.-Y. Chen, T.-W. Weng, C. Gan, and M. Wang, "On fast adversarial robustness adaptation in model-agnostic meta-learning," *arXiv preprint arXiv:2102.10454*, 2021.

[16] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," *Advances in neural information processing systems*, vol. 32, 2019.

[17] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[18] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[19] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 4, pp. 1484–1497, 2012.

[20] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[21] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.