Selective Trimmed Average: A Resilient Federated Learning Algorithm With Deterministic Guarantees on the Optimality Approximation

Mojtaba Kaheni, *Member, IEEE*, Martina Lippi, *Member, IEEE*, Andrea Gasparri, *Senior Member, IEEE*, and Mauro Franceschelli, *Senior Member, IEEE*

***Abstract*—The federated learning (FL) paradigm aims to distribute the computational burden of the training process among several computation units, usually called *agents* or *workers*, while preserving private local training datasets. This is generally achieved by resorting to a server–worker architecture where agents iteratively update local models and communicate local parameters to a server that aggregates and returns them to the agents. However, the presence of adversarial agents, which may intentionally exchange malicious parameters or may have corrupted local datasets, can jeopardize the FL process. Therefore, we propose selective trimmed average (SETA), which is a resilient algorithm to cope with the undesirable effects of a number of misbehaving agents in the global model. SETA is based on properly filtering and combining the exchanged parameters. We mathematically prove that the proposed algorithm is resilient against data and local model poisoning attacks. Most resilient methods presented so far in the literature assume that a trusted server is in hand. In contrast, our algorithm works both in server–worker and shared memory architectures, where the latter excludes the necessity of a trusted server. The theoretical findings are corroborated through numerical results on MNIST dataset and on multiclass weather dataset (MWD).**

***Index Terms*—Adversarial attacks, distributed optimization, multiagent systems, resilient federated learning (FL).**

I. Introduction

THE BASIC idea behind federated learning (FL) is to distribute the training process among several computation units, referred to as *agents* in the following, for example, computers, smartphones, possessing local training datasets that may be heterogeneous [1], [2]. Although agents generally do not want to disclose their local training datasets to preserve their privacy [3], they are interested in jointly learning a globally optimal model. The conventional architecture in FL is the server–worker structure, as depicted in Fig. 1. In this architecture, starting from a global model, each agent i moves toward the optimal local model for its training dataset, $\mathcal{D}_i$, through a stochastic gradient descent (SGD) algorithm and communicates the resulting parameters $\mathbf{w}_i$ to a server. In the next step, the server updates the global model by aggregating the received parameters $\mathbf{w}_i$, and sends the updated global parameters $\mathbf{w}$ back to agents. The average of received parameters is the most conventional aggregation rule in nonadversarial environments [4], [5], [6], [7].

However, all networked systems are threatened by cyber-attacks [8], [9] and FL is not an exemption. Adversarial agents can perform two categories of attacks: 1) data poisoning [10], [11], [12] and 2) local model poisoning [13] attacks. In data poisoning attacks, the adversaries inject malicious data into the local training set of compromised agents, while the latter run the learning process and honestly send the results to the server. In model poisoning attacks, the adversaries intentionally exchange malicious parameters to corrupt the global model. Recent results [14], [15], [16] show that even a single misbehaving agent can arbitrarily manipulate the global model if the average aggregation rule is implemented.

To overcome this issue, alongside methodologies that aim to detect and isolate misbehaving agents, such as [17] and [18], several aggregation rules, for example, trimmed average or median [14], Krum [15], Bulyan [19], Byrd-SAGA [16], Zeno [20] and RSA [21], have been proposed to make distributed learning resilient against adversaries in server–worker architectures. The common idea behind most existing resilient distributed learning algorithms is that *outlier* local parameters must be filtered out so as not to have any influence on the global model. For instance, the trimmed average aggregation rule [14] computes the coordinatewise average of the vectors of model parameters, discarding β percentage of the highest and lowest values, where β is a design parameter. A similar idea was recently proposed in [22], wherein a trimmed average is applied to the estimate of the global gradient vector by agents, discarding β percentage of the highest

Manuscript received 12 April 2023; revised 12 August 2023 and 5 December 2023; accepted 23 December 2023. Date of publication 23 January 2024; date of current version 23 July 2024. This work was supported in part by the Knowledge Foundation (KKS) through Safe and Secure Adaptive Collaborative Systems (SACSys) under Grant 20190021, and in part by the Swedish Agency for Innovation Systems (Vinnova) through GREENER: Intelligent Energy Management in Connected Construction Sites under Grant 2019-05877. This article was recommended by Associate Editor P. Shi. *(Mojtaba Kaheni and Martina Lippi are co-first authors.) (Corresponding author: Mauro Franceschelli.)*

Mojtaba Kaheni is with the Akademin för Innovation, Design och Teknik (IDT), Mälardalen University, 722 20 Västerås, Sweden (e-mail: mojtaba.kaheni@mdu.se).

Martina Lippi and Andrea Gasparri are with the Department of Civil, Computer Science and Aeronautical Technologies Engineering, Roma Tre University, 00154 Rome, Italy (e-mail: martina.lippi@uniroma3.it; andrea.gasparri@uniroma3.it).

Mauro Franceschelli is with the Department of Electrical and Electronic Engineering, University of Cagliari, 09122 Cagliari, Italy (e-mail: mauro.franceschelli@unica.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2024.3350387.

Digital Object Identifier 10.1109/TCYB.2024.3350387

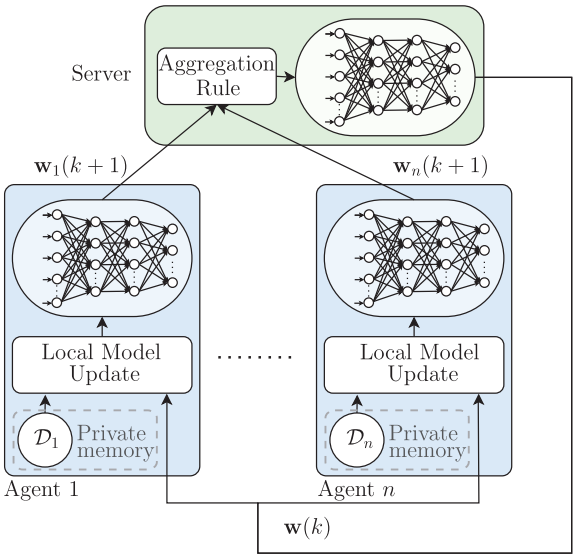

Fig. 1. Server–worker architecture where n agents locally update their models and send the computed parameters $\mathbf{w}_i$ to a server that aggregates them into a global parameter vector $\mathbf{w}$.

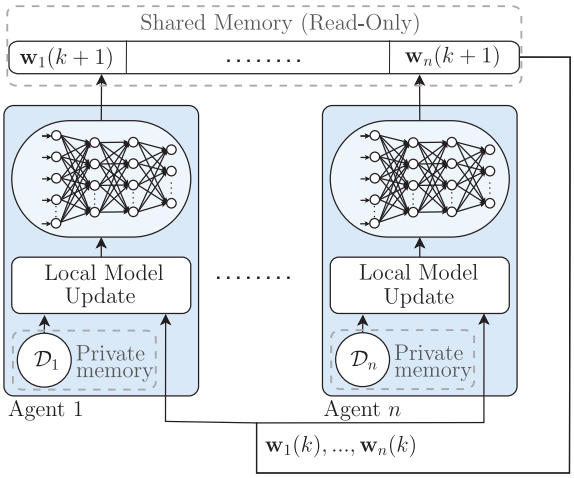

Fig. 2. Shared memory architecture where n agents locally update their models and share the computed parameters.

and lowest-coordinatewise values. The median aggregation rule [14] considers the coordinatewise median of the vectors of model parameters received from agents. A comparison between trimmed average and median aggregation rules can be found in [14]. In Krum [15], the local parameter vector having the lowest distance to others is selected as global model. An extension of Krum is provided by Bulyan [19] where parameters are updated according to a two-stage approach: 1) the set of local parameters with the lowest distance from others is recursively determined and 2) then, they are combined by discarding the farthest values from the coordinatewise median. A geometric median-based robust aggregation on corrected stochastic gradients is proposed in Byrd-SAGA [16] reducing the stochastic gradient-induced noise from regular agents, that is, not adversarial. A further approach based on computing and exchanging redundant gradients by the workers is proposed in [23] to overcome the computational complexity of median-based approaches. In addition, the algorithm Zeno presented in [20] exploits the knowledge of a training dataset by the server to score the gradients received by the workers. The approach is extended in [24] to handle asynchronous communication and an arbitrary number of adversaries. Finally, RSA in [21] is based on introducing a regularization term in the objective function to robustify the learning task and forcing the workers' local parameters to be close to the server's one.

A key assumption in the resilient aggregation rules presented so far is the availability of a trusted server, which represents a server unit capable of aggregating the parameters in a fault-free and attack-free manner. However, in case of attacks to this unit, the entire learning process is at severe risk since malicious global models can be transferred to all workers. Furthermore, many of them, for example, [14], [15], [19], [20], and [24], assume independent and identically distributed (IID) local datasets which are unrealistic in FL where each agent holds a private dataset.

In this article, we propose a resilient algorithm that is also applicable when all agents communicate according to a virtual shared memory architecture, as depicted in Fig. 2, in addition to the server–worker one. In the shared memory architecture, we consider that each agent i holds a local private memory, where the dataset $\mathcal{D}_i$ is stored, and shares the estimated local parameters $\mathbf{w}_i(k+1)$ in a memory shared with the other agents. In this shared memory, each agent can read the memory areas of all others and can write only on its own. This enables the omission of the trusted server unit and is equivalent to letting all the agents communicate with each other and exchange local parameters. Note that this architecture ensures privacy of local data in alignment with the FL paradigm. However, in this way each agent has complete knowledge of the network, the local parameters of all agents, as well as the update rules. This implies that adversaries can exploit this information to craft serious attacks, as detailed in [13], for which we demonstrate resilience. More specifically, we look at the resilient FL problem from the perspective of *resilient distributed optimization* [25], [26], [27], [28], aiming to find suboptimal solutions that are not affected by adversarial agents, without the need to explicitly detect and isolate them as done, for instance, in [29]. We present a resilient algorithm, called selective trimmed average (SETA), based on a coordinatewise trimmed average of local parameters to update the parameters of each regular agent. For each coordinate, trimmed values are selected based on whether the coordinate itself is evaluated as an outlier or not. We assume the general case where adversarial agents can collude with each other and can decide whether to perform data poisoning or local model poisoning attacks.

To the best of our knowledge, this is the first work that provides deterministic formal guarantees for resilient FL, also relaxing the server–worker architecture. Furthermore, SETA achieves better performance with non-IID datasets compared to most state-of-the-art baselines under different types of attacks. Note that a relaxation of the architecture can also be found in [30] and [31] where distributed protocols are designed but IID datasets are required.

The trimmed average is the closest existing approach to the one in this article. However, we identify the following fundamental differences.

1) In the trimmed average, agents have the same global model to evolve at each epoch. In contrast, in our algorithm, the starting model at each epoch may be different for the agents.
2) It is possible to execute our algorithm in an architecture with shared memory, which relaxes the need to have a trusted server unit.
3) We provide deterministic theoretical guarantees as opposed to statistical results published for trimmed average and many other aggregation rules.

The proposed algorithm is built on our previous work [32] in resilient distributed optimization of scalar functions. In particular, the results of our previous work are extended to resilient FL and multidimensional problems in this article as well as a more detailed mathematical analysis is provided.

In summary, the main contributions of this article are as follows.

1) We propose a resilient FL algorithm, called (SETA), aimed to cope with both data and local model poisoning attacks with non-IID local datasets.
2) We extend resilient FL to the case of shared memory architecture while guaranteeing that the additional information shared by regular agents, and accessible by adversaries, does not threaten the learning process.
3) We provide deterministic mathematical analysis, as well as simulations on the MINST dataset and the realistic multiclass weather dataset (MWD) [33], to confirm the effectiveness of our algorithm.

The remainder of this article is organized as follows. In Section II, preliminary notions in multiagent systems and distributed optimization are reviewed. FL problem in nonadversarial setting is introduced in Section III. Section IV is devoted to the problem statement and introducing the proposed resilient FL algorithm. Section V focuses on the mathematical analysis. Numerical results are provided in Section VI to validate the approach and Section VII concludes this article.

A. Notation

Table I reports the main notation of this article. Unless specified otherwise, we denote scalar values with small regular font, vectors with bold font, and matrices with capital letters. In addition, $\mathbf{1}_n$ ($\mathbf{0}_n$) is an n-element vector all equal to 1 (0), $|.|$ denotes the cardinality of a set, $\lfloor . \rfloor$ is the floor function.

TABLE I
MAIN NOTATIONS INTRODUCED IN THIS ARTICLE

Variable	Meaning
n (n_r,n_a)	Number of total (regular, adversarial) agents
F	Maximum number of adversarial agents
$\mathbf{w}_i$	State vector of agent i, i.e., parameters' vector
m	Dimension of the state vector
w_i^z	Component z of the state vector $\mathbf{w}_i$
W	Matrix collecting the agents' states
$\mathcal{D}_i$	Local dataset of agent i
f_i	Local objective function of agent i
$\mathcal{V}_r$	Set of regular agents
$\mathbf{d}_i$	Gradient vector of f_i
$\mathbf{l}_i$	Weighted sum of the state of agent i and its in-neighbors' states
c	Step size
$\mathcal{V}_h^z$ ($\mathcal{V}_l^z$)	Set of F agents with highest (lowest) w_j^z
$\mathcal{V}_n^z$	Set of $n-2F$ agents not in $\mathcal{V}_h^z \cup \mathcal{V}_l^z$
i_r^z	Selected agent in $\mathcal{V}_n^z$ to exclude for the update of w_i^z in case $i \notin \mathcal{V}_n^z$
q	Cardinality of the set $\mathcal{V}_n^z$ equal to $n-2F$

II. PRELIMINARIES

Consider a network composed of n computation units which can interact with each other. In the remainder of the manuscript, we refer to each computation unit as an agent or node. Such a network is modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ represents the set of agents in the network and $\mathcal{E} \subseteq \{\mathcal{V} \times \mathcal{V}\}$ the set of communication links (or edges) among the agents, that is, if agent i sends information to j, then $(i, j) \in \mathcal{E}$. We denote the set of the in-neighbors of agent i as $\mathcal{N}_i = \{j \in \mathcal{V} \backslash \{i\} | (j, i) \in \mathcal{E}\}$. A graph $\mathcal{G}$ is defined *undirected* if the communication links are bidirectional, that is, if $(i, j) \in \mathcal{E}$ implies that $(j, i) \in \mathcal{E}$, and is defined *directed* otherwise. In addition, a graph $\mathcal{G}$ is called complete if there exists an edge between all the pairs of agents, that is, $\mathcal{N}_i = \mathcal{V} \backslash \{i\} \ \ \forall i \in \mathcal{V}$. A path $\pi_{i,j}$ between nodes i and j is a sequence of consecutive edges, starting from node i and ending in node j, that is, it is composed of the edges $\{(i, v_1), (v_1, v_2), \ldots, (v_m, j)\} \subset \mathcal{E}$, where $\{i, v_1, v_2, \ldots, v_m, j\} \subset \mathcal{V}$. A directed graph $\mathcal{G}$ is defined as strongly connected, if there exists a directed path between each pair of nodes (i, j) in $\mathcal{V}$. If there exists $\pi_{i,j}$ between nodes i and j, node j is said to be *reachable* from node i. Furthermore, if there exists an agent $i \in \mathcal{V}$ such that all agents in $\mathcal{V}$ are reachable from i, the graph $\mathcal{G}$ is said to be *rooted*. In case each edge $(j, i) \in \mathcal{E}$ is associated with a positive weight, $a_{ij} > 0$, the graph $\mathcal{G}$ is called weighted. The matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ collecting the weights is defined as *adjacency matrix*, that is, $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. A square matrix $A \in \mathbb{R}^{n \times n}$ with non-negative entries and with each row (column) summing to 1 is called row (column) stochastic. Moreover, A is called doubly stochastic if it is jointly row and column stochastic. If the edge weights $a_{ij}(k)$ are time-varying, the weighted graph is time-varying as well and is denoted by $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$. Let $\mathcal{E}_B(k)$ be the aggregated set of edges $\mathcal{E}(k)$ in the time interval $[k_0, k_0 + B)$ with $k_0 \in \mathbb{N}$, that is,

$$\mathcal{E}_B(k) = \bigcup_{k=0}^{B-1} \mathcal{E}(k_0 + k).$$

A time-varying graph $\mathcal{G}(k)$ is defined as jointly strongly connected, if there exists a finite positive integer B such that the graph $(\mathcal{V}, \mathcal{E}_B(k))$ is strongly connected for all finite k_0.

A. Transition Matrix

Let $\mathbf{w}_i(k) \in \mathbb{R}^m$ be the state vector of agent i at time step k, that is, in our case $\mathbf{w}_i(k)$ is the vector collecting the model parameters to optimize. The most common update rule of an agent i in consensus-based multiagent systems [34] consists of

a weighted summation over its own and its in-neighbors state vectors, that is,

$$\mathbf{w}_i(k+1) = a_{ii}(k)\mathbf{w}_i(k) + \sum_{j\in\mathcal{N}_i(k)} a_{ij}(k)\mathbf{w}_j(k). \tag{1}$$

According to the definition of adjacency matrix, (1) can be rewritten in matrix form as

$$W(k+1) = A(k)W(k) \tag{2}$$

where $W(k) = [\mathbf{w}_1(k), \ldots, \mathbf{w}_n(k)]^T \in \mathbb{R}^{n\times m}$ is the matrix containing the agents' states at time step k. If the adjacency matrix is row stochastic, the weighted summation becomes a *weighted average* over each agent in-neighbors' values and its own state value at time step k. By virtue of (2), we can define the following equation:

$$W(k+1) = A(k)A(k-1)W(k-1). \tag{3}$$

If we repeat this procedure, for all $s < k$ it holds

$$W(k+1) = A(k)A(k-1)\ldots A(s+1)A(s)W(s). \tag{4}$$

To compact (4), the *transition matrix* is defined as

$$\Phi(k,s) = A(k)A(k-1)\ldots A(s+1)A(s) \tag{5}$$

for all s and k with $k \geq s$, and $\Phi(k,k) = A(k)$, leading to

$$W(k+1) = \Phi(k,s)W(s). \tag{6}$$

From (6), we observe that if all the rows of the transition matrix asymptotically converge to the same stochastic vector, then agreement among the agents is reached, that is, it holds

$$\lim_{k\to\infty} \mathbf{w}_i(k) = \lim_{k\to\infty} \mathbf{w}_j(k) \quad \forall i,j \in \mathcal{V}.$$

We recall a lemma providing a condition for convergence in terms of connectivity and adjacency matrix weights.

Lemma 1 [35]: Consider a communication graph $\mathcal{G}$. Assume that there exists a scalar $\tau \in (0,1)$, such that $\forall i \in \mathcal{V}$, it holds $a_{ii}(k) \geq \tau$, and for all $i \neq j$, it holds either $a_{ij}(k) = 0$ or $a_{ij}(k) \geq \tau$. If $\mathcal{G}$ is rooted and the adjacency matrix $A(k)$ is row stochastic $\forall k$, then there exist two positive scalars $B > 0$ and $\xi \in (0,1)$ and a stochastic vector $\phi(s) = [\phi_1(s), \phi_2(s), \ldots, \phi_n(s)]^T$ such that $\lim_{k\to\infty} \Phi(k,s) = \mathbf{1}_n\phi(s)^T$ and $|[\Phi(k,s)]_{i,j} - \phi_j(s)| \leq B\xi^{k-s}$.

B. Ancillary Definitions and Lemmas

We introduce the following definitions that will be used in the theoretical analysis.

Definition 1 [25]: A subset $\mathcal{S} \subset \mathcal{V}$ of agents is said to be *r-reachable*, with $r \in \mathbb{N}$, if there exists an agent $i \in \mathcal{S}$ such that $|\mathcal{N}_i \backslash \mathcal{S}| \geq r$.

Definition 2 [25]: For $r \in \mathbb{N}$, graph $\mathcal{G}$ is said to be *r-robust* if for all pairs of disjoint nonempty subsets, $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of $\mathcal{S}_1$ or $\mathcal{S}_2$ is r-reachable.

Definition 2 implies that a complete graph with n agents is $\lfloor (n+1)/2 \rfloor$-robust. We additionally consider the following lemmas.

Lemma 2 [36]: Suppose a graph $\mathcal{G}$ is r-robust. Let $\mathcal{G}'$ be a graph obtained by removing $r-1$ or fewer incoming edges from each node in $\mathcal{G}$. Then, $\mathcal{G}'$ is rooted.

Lemma 3 [37]: Let β be a positive scalar in $(0,1)$ and $\{\gamma_k\}$ be a positive scalar sequence. If $\lim_{k\to\infty} \gamma_k = 0$, then $\lim_{k\to\infty} \sum_{l=0}^{k} \beta^{k-l}\gamma_l = 0$.

III. Federated Learning Problem in Nonadversarial Settings

Traditional centralized learning algorithms require all training samples to be available to a central processing unit computing the optimal model. However, such algorithms may not be suitable for certain scenarios, where, for instance, 1) the owners of the training samples prefer not to disclose private information with a central processing unit or 2) the number of samples is too large, and it is impractical or even impossible to process them with a single processing unit. FL overcomes these limitations by distributing the learning process among multiple agents that hold private local datasets. This ensures privacy preservation and enables the processing of large datasets in a distributed fashion.

We now present the FL problem in a *nonadversarial* setting. Consider n agents that communicate with a server unit or share a memory and aim to collaboratively learn the parameters $\mathbf{w}$ of a global model. Each agent i has access to a local training dataset $\mathcal{D}_i$ and its local objective is to find the optimal model parameters $\mathbf{w} \in \mathbb{R}^m$, obtained by solving the following optimization problem:

$$\min_{\mathbf{w}} f(\mathbf{w}, \mathcal{D}_i) = \min_{\mathbf{w}} f_i(\mathbf{w}) \tag{7}$$

where f_i, generally referred to as loss function, depends on $\mathcal{D}_i$, for example, the mean square error (MSE) function can be chosen.

Assumption 1: The objective functions $f_i(\mathbf{w})$ are convex, and their gradients are continuous and bounded for bounded $|\mathbf{w}|$ $\forall i \in \mathcal{V}$, namely, $||\nabla f_i(\mathbf{w})|| \leq L$ if $|\mathbf{w}| < \infty$.

A sensible global model can be obtained as follows:

$$\min_{\mathbf{w}_i} \sum_{i=0}^{n} f_i(\mathbf{w}_i), \tag{8a}$$

$$\text{subject to } \mathbf{w}_i = \mathbf{w}_j \quad \forall i,j \in \mathcal{V} \tag{8b}$$

where each agent i holds a copy of the decision vector $\mathbf{w}_i$ and these copies are required to agree in (8b). The formulation in (8) can be viewed as a distributed optimization problem.

Remark 1: Assumption 1 is not too restrictive and several loss functions exist that satisfy it [38], [39]. Moreover, as discussed in [38], the main advantage of nonconvex loss functions is their ability to reduce the effects of outlier samples. Since our algorithm is resilient against manipulated samples, the importance of using these functions diminishes.

IV. Resilient Federated Learning Problem and Algorithm Design

As mentioned in the Introduction, it is possible to apportion all kinds of attacks in two categories, depicted in Fig. 3.

1) *Data Poisoning Attack (e.g., [40]):* The adversary injects deceptive samples in the local datasets $\mathcal{D}_i$ of some agents, while these agents run the learning process to solve (7) and honestly send the results to the server.

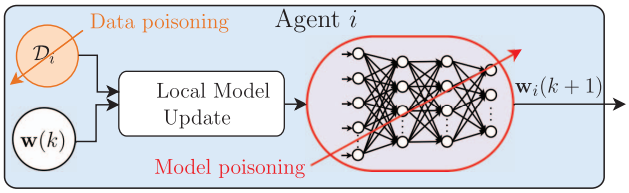

Fig. 3. Depiction of data (in orange) and model (in red) poisoning attacks.

Since the local objective $f_i(\mathbf{w})$ depends on the dataset, the effect of this attack is that a compromised agent i will try to locally optimize a different objective function $f_i^a(\mathbf{w})$. Note that it is usually difficult to detect this kind of attack [41] since agents execute the update algorithm correctly, while their dataset is corrupted. Next, we will refer to compromised agents as adversarial ones.

2) *Model Poisoning Attack (e.g., [13]):* The adversary manipulates the local model parameters $\mathbf{w}_i$ that are sent during the learning process. As a result, the adversaries do not solve (7). Instead, they generally optimize an adversarial objective aiming to mislead all the regular agents. This leads to the possibility of adversarial agents sending arbitrary model parameters $\mathbf{w}_i$ to other agents, with no constraints on their behavior being imposed.

Consider n agents among which $n_r \leq n$ are regular and aim to solve the FL problem in Section III and $n_a \leq F$ are adversarial, that is, they perform either data or model poisoning attacks, with F a positive constant and $n = n_r + n_a$. A sensible solution to (8) in an adversarial setting is to *ignore* the adversaries and find the optimizer among the regular agents

$$\begin{aligned} \min_{\mathbf{w}_i} \quad & \sum_{i\in\mathcal{V}_r} f_i(\mathbf{w}_i) \\ \text{subject to} \quad & \mathbf{w}_i = \mathbf{w}_j \quad \forall i,j \in \mathcal{V}_r \end{aligned} \tag{9}$$

where $\mathcal{V}_r$ represents the set of regular agents with nominal behavior. In the following, without loss of generality, we model shared memory architectures as complete graphs in which each agent receives the model parameters from all the other agents, that is, all the agents share the respective model parameters. Similar to (8), (9) can be viewed as a distributed optimization problem and, in particular, as a resilient distributed optimization problem.

A. Selective Trimmed Average Algorithm

We propose a resilient FL algorithm, referred to as (SETA) algorithm, aimed to solve (9). The basic idea behind SETA is that each regular agent filters out coordinatewise outlier values received from other agents and updates the parameter vector averaging the remaining values. Algorithm 1 summarizes the proposed SETA protocol, which is composed of three main phases.

For each time step k, in the first phase, each regular agent i gathers the local parameters $\mathbf{w}_j(k)$ from the other agents in the network and, for each coordinate $z \in \{1,\dots,m\}$, runs a clustering algorithm which builds the sets $\mathcal{V}_h^z(k)$ and $\mathcal{V}_l^z(k)$ comprising the highest and lowest-F values $w_j^z(k) \quad \forall j \in \mathcal{V}$, respectively, as well as the set $\mathcal{V}_n^z(k)$ comprising the remaining agents, that is, $\mathcal{V}_n^z(k) = \mathcal{V}\backslash(\mathcal{V}_h^z(k) \cup \mathcal{V}_l^z)(k)$.

Algorithm 1 Selective Trimmed Average (SETA) Protocol

Require: F, Cost function f_i related to dataset $\mathcal{D}_i \quad \forall i \in \mathcal{V}_r$
Each agent i runs indefinitely the following:
Phase 1 - Parameters clustering
Gather local parameters $\mathbf{w}_j(k)$, $j \in \mathcal{V}\backslash\{i\}$
for each $z \in \{1,\dots,m\}$ **do**
Cluster the values $w_j^z(k)$, $\forall j \in \mathcal{V}$ in 3 sets:
$\mathcal{V}_h^z(k) = \{F \text{ agents with highest } w_j^z(k)\}$
$\mathcal{V}_l^z(k) = \{F \text{ agents with lowest } w_j^z(k)\}$
$\mathcal{V}_n^z(k) = \mathcal{V}\backslash(\mathcal{V}_h^z \cup \mathcal{V}_l^z)$
end for
Phase 2 - Weights assignment
$q = n - 2F$
for each $z \in \{1,\dots,m\}$ **do**
if $i \in \mathcal{V}_n^z(k)$ **then**

$$a_{ij}^z(k) = \begin{cases} \frac{1}{q} & \text{if} \quad j \in \mathcal{V}_n^z(k) \\ 0 & \text{otherwise.} \end{cases}$$

else

$$i_r^z(k) = \begin{cases} \arg\min_{j\in\mathcal{V}_n^z(k)} w_j^z(k) & \text{if} i \in \mathcal{V}_l^z(k) \\ \arg\max_{j\in\mathcal{V}_n^z(k)} w_j^z(k) & \text{if} i \in \mathcal{V}_h^z(k) \end{cases}$$

$$a_{ij}^z(k) = \begin{cases} \frac{1}{q} & \text{if} \quad j \in \left\{\mathcal{V}_n^z(k)\backslash i_r^z(k)\right\} \\ \frac{1}{q} & \text{if} \quad j = i \\ 0 & \text{otherwise.} \end{cases}$$

end if
end for
Phase 3 - State update
for each $z \in \{1,\dots,m\}$ **do**
$l_i^z(k) = \sum_{j=1}^n a_{ij}^z(k) w_j^z(k)$
end for
$c(k) \leftarrow$ update step size [eq. (11)]
$\mathbf{d}_i(k) \leftarrow$ update gradient $f_i(\mathbf{w}_i)$
$\mathbf{w}_i(k+1) = \mathbf{l}_i(k) - c(k)\mathbf{d}_i(k)$

In the second phase, the weights $a_{ij}^z(k)$ for the in-neighbors are assigned. More in detail, if the agent i belongs to $\mathcal{V}_n^z(k)$, then the weights $a_{ij}^z(k)$ are set to $1/q \quad \forall j \in \mathcal{V}_n^z(k)$, and are set to 0 otherwise. In case the agent belongs to any of the outlier sets, that is, $i \in \mathcal{V}_h^z(k) \cup \mathcal{V}_l^z(k)$, the agent $i_r^z(k)$ with lowest- or highest-state value in $\mathcal{V}_n^z(k)$ is selected to be ignored if $i \in \mathcal{V}_l^z(k)$ or $i \in \mathcal{V}_h^z(k)$, respectively. Next, the weights a_{ij}^z are set to $1/q$ for $i = j$ and $\forall j \in \mathcal{V}_n^z(k)\backslash\{i_r^z(k)\}$, and are set to 0 otherwise.

An illustration of the effect of Phase 2 of SETA is provided in Fig. 4. Starting from the complete graph with n agents (in the circle on the left) representing the communication network, Phase 2 leads to m graphs, one for each component $z \in \{1,\dots,m\}$. Specifically, each graph (as depicted in the zoom on the right) can be viewed as a combination of a complete graph composed of $q = n - 2F$ agents (in the circle) and additional $2F$ agents (marked in gray) receiving information, but which do not have any influence on others, that is, the values of such $2F$ agents are filtered out by the other agents. This implies that the edges of the graph considered for each component are time-varying and depend on the identified clusters at each time step.

At this point, the third phase updates the states following the distributed subgradient optimization algorithm introduced in [37] and using the graphs obtained in Phase 2 for each component z, that is, the update rule for agent i is:

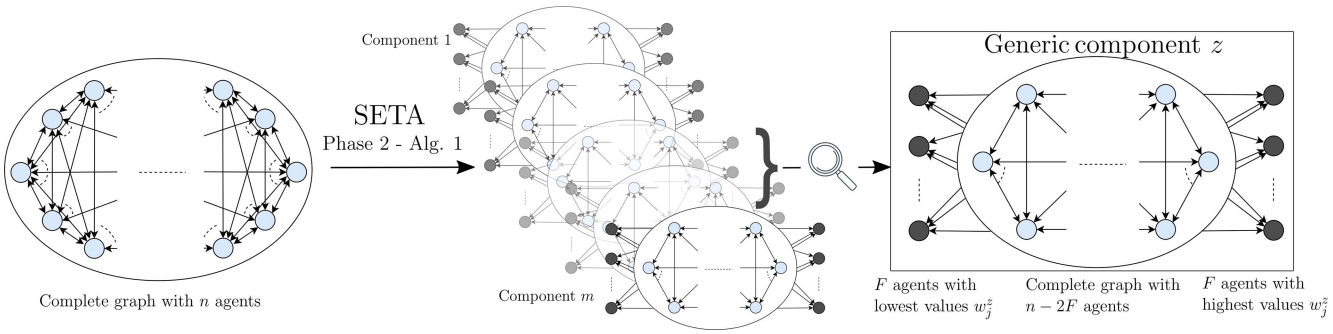

Fig. 4. Depiction of the graph reshape for each component $z \in \{1, \ldots, m\}$ given by the implementation of Phase 2 of SETA (Algorithm 1). Complete graphs are depicted by enclosing the agents in circles.

$$l_i^z(k) = \sum_{j=1}^{n} a_{ij}^z(k) w_j^z(k) \quad \forall z$$
$$\mathbf{w}_i(k+1) = \mathbf{l}_i(k) - c(k)\mathbf{d}_i(k) \tag{10}$$

with $c(k)$ the step size defined as

$$c(k) > 0,\ \sum_{k=0}^{\infty} c(k) = \infty,\ \sum_{k=0}^{\infty} c^2(k) < \infty. \tag{11}$$

The driving force behind SETA is the principle of self-trust. In Phase 2 of SETA, an agent may discover that its coordinate z significantly deviates from those of other agents, making it an outlier belonging to $\mathcal{V}_l^z(k)$ or $\mathcal{V}_h^z(k)$. Nevertheless, each agent maintains confidence in its own estimation and includes its own zth coordinates in the aggregation process, that is, $a_{ii}^z = (1/q)$, regardless of its perceived outlier status.

Remark 2: SETA is also suitable for a server–worker architecture in which, in Phase 1, all the local parameters $\mathbf{w}_i(k)$ are communicated to the central server, that builds the clusters $\mathcal{V}_h^z(k)$, $\mathcal{V}_l^z(k)$, $\mathcal{V}_n^z(k)$. Based on these, the server computes the variables $a_{ij}^z(k)$ defined in Phase 2 and updates the states $\mathbf{w}_i(k+1)$ as in Phase 3. Finally, the server sends back the updated states to each agent i according to the architecture in Fig. 1. Without loss of generality, we consider the shared memory architecture in our mathematical treatments.

The main difference compared to previous aggregation rules for server–worker architecture is that with SETA each agent i receives a different parameter update $\mathbf{w}_i(k+1)$, and these parameters converge to the same vector for all regular agents.

V. Mathematical Analysis

We now focus on proving that SETA is resilient to all types of attacks described in Section IV. To this aim, we first prove that if n and F satisfy the inequality for the complete graph to be $2F+1$ robust, that is,

$$\lfloor (n+1)/2 \rfloor \geq 2F+1 \tag{12}$$

then the agents reach consensus with SETA protocol and the consensus value is not dependent on adversarial agents. Next, we provide a deterministic bound on the convergence to the optimal solution of the resilient FL problem.

A. Agents Consensus

As discussed above, SETA produces m different adjacency matrices corresponding to m different graph topologies. We notice that each agent i filters out at most $2F$ of its in-neighbors' states to update each coordinate of the parameter vector $\mathbf{w}_i$. Considering Lemma 2, the $(2F+1)$-robustness property in (12) ensures that the resulting m graphs after implementing SETA are rooted.

Lemma 4: Consider n agents with $(2F+1)$-robust network in a shared memory architecture. Then, under Assumption 1, all agents executing SETA in Algorithm 1 converge to a constant consensus vector, $\overline{\mathbf{w}} \in \mathbb{R}^m$, that is,

$$\lim_{k\to\infty} \|\mathbf{w}_i(k) - \overline{\mathbf{w}}\| = 0 \quad \forall i \in \mathcal{V}_r$$

and $\overline{\mathbf{w}}$ is not influenced by adversarial agents.

Proof: To prove this result we consider two steps: 1) by assuming that regular agents are at consensus, we demonstrate that this is not affected by adversaries and 2) then we prove that regular agents actually reach a constant consensus vector $\overline{\mathbf{w}}$.

Regarding the first step, in case all the adversaries belong to $\mathcal{V}_l^z(k) \cup \mathcal{V}_h^z(k)$, they are filtered out by all regular agents according to SETA and cannot deviate the consensus value. We thus consider the case where K^z adversarial agents a_i belong to $\mathcal{V}_n^z(k)$, $i = 1, \ldots, K^z$. This implies that there must also exist K^z regular agents belonging to $\mathcal{V}_l^z(k)$ and K^z regular agents belonging to $\mathcal{V}_h^z(k)$. The component z of the state of adversarial agents a_i can be written as a linear combination of two filtered regular agents as follows:

$$w_{a_i}^z(k) = \sigma_i^z(k) w_{h_i}^z(k) + \big(1 - \sigma_i^z(k)\big) w_{l_i}^z(k) \tag{13}$$

for $i = 1, \ldots, K^z$ and $0 < \sigma_i^z(k) < 1$ where $w_{h_i}^z(k)$ and $w_{l_i}^z(k)$ represent the components z of the regular agents that belong to $\mathcal{V}_h^z(k)$ and $\mathcal{V}_l^z(k)$, respectively. Therefore, $A^z(k)$ is equivalent to an additional row stochastic adjacency matrix, $A^{z'}(k)$, where two filtered regular agents are considered in place of the remaining adversarial agents in $\mathcal{V}_n^z(k)$. It follows that the case of K^z adversarial agents in $\mathcal{V}_n^z(k)$ is mathematically equivalent to the situation in which the adversarial agents do not communicate their state, but rather send the states of the respective two regular agents according to (13). In this equivalent network graph, the regular agents filter out at most $2F$ of their incoming edges as well. Thus, since the graph representing the network is $2F+1$-robust, we can conclude, by virtue of Lemma 2, that the equivalent adjacency matrix $A^{z'}(k)$ is rooted and stochastic. Furthermore, by recalling that the adversarial agents are filtered out in $A^{z'}(k)$, we obtain that,

if the regular agents reach a consensus, this value cannot be influenced by the adversaries.

At this point, we focus on proving that the regular agents reach a consensus vector $\overline{\mathbf{w}}$. From (10), we have

$$\mathbf{w}_i(k+1) = \mathbf{l}_i(k) - c(k)\mathbf{d}_i(k). \tag{14}$$

We can rewrite the z^{th} value of $\mathbf{w}_i(k+1)$ in (14) as follows:

$$w_i^z(k+1) = l_i^z(k) - c(k)d_i^z(k). \tag{15}$$

According to the definition of $l_i^z(k)$ in (10), (15) is equal to

$$w_i^z(k+1) = \sum_{j=1}^{n} a_{ij}^z(k) w_j^z(k) - c(k)d_i^z(k). \tag{16}$$

Then, using the transition matrix defined in (5), we obtain $\forall i \in \mathcal{V}$ and $k > s$

$$\begin{aligned} w_i^z(k+1) = &\sum_{j=1}^{n} \left[\Phi^z(k,s)\right]_{i,j} w_j^z(s) \\ &- \sum_{r=s}^{k-1}\sum_{j=1}^{n} \left[\Phi^z(k,r+1)\right]_{i,j} \Big(c(r)d_j^z(r)\Big) \\ &- c(k)d_i^z(k) \end{aligned}$$

where Φ^z is the transition matrix associated with $[a_{ij}^z]$. Since by Assumption 1 gradients are bounded, it holds $\lim_{k\to\infty} c(k)d_i^z(k) = 0$ according to (11). Therefore, $\forall i_1, i_2 \in \mathcal{V}_r$, we have

$$\begin{aligned} &\lim_{k\to\infty}\Big(w_{i_1}^z(k+1) - w_{i_2}^z(k+1)\Big) \\ &= \lim_{k\to\infty} \underbrace{\sum_{j=1}^{n}\Big(\left[\Phi^z(k,s)\right]_{i_1,j} - \left[\Phi^z(k,s)\right]_{i_2,j}\Big) w_j^z(s)}_{i)} \\ &\quad - \lim_{k\to\infty} \underbrace{\sum_{r=s}^{k-1}\sum_{j=1}^{n}\Big(\left[\Phi^z(k,r+1)\right]_{i_1,j} - \left[\Phi^z(k,r+1)\right]_{i_2,j}\Big)}_{ii)} \\ &\quad \times \underbrace{\Big(c(r)d_j^z(r)\Big)}_{iii)}. \end{aligned} \tag{17}$$

Considering that the resulting graph after implementing SETA is equivalent to a row stochastic and rooted graph, in which the adversaries are filtered out, from Lemma 1 it follows $\forall i \in \mathcal{V}_r$:

$$\lim_{k\to\infty} \left[\Phi^z(k,s)\right]_{i,j} = \varphi_j^z(s) \tag{18}$$

which yields to

$$\lim_{k\to\infty}\Big(\left[\Phi^z(k,s)\right]_{i_1,j} - \left[\Phi^z(k,s)\right]_{i_2,j}\Big) = 0. \tag{19}$$

Since any $s < k$ can be selected for which $w_j^z(s)$ is bounded, we observe that the term 1) of (17) tends to zero. Regarding term 2), Lemma 1 leads to

$$\begin{aligned} -B\xi^{k-(r+1)} &\le \sum_{j=1}^{n}\Big(\left[\Phi^z(k,r+1)\right]_{i_1,j} - \left[\Phi^z(k,r+1)\right]_{i_2,j}\Big) \\ &\le B\xi^{k-(r+1)} \end{aligned}$$

with $B > 0$ and $0 < \xi < 1$. According to Assumption 1, we can write the following inequality for term iii):

$$-nLc(r) \le c(r)\sum_{j=1}^{n} d_j^z(r) \le nLc(r). \tag{20}$$

Thus, the overall term given by multiplying ii) and iii) is bounded in the interval

$$\left[-nLB\sum_{r=s}^{k-1}\xi^{k-(r+1)}c(r),\, nLB\sum_{r=s}^{k-1}\xi^{k-(r+1)}c(r)\right]. \tag{21}$$

In view of (11) and Lemma 3, both the extremes of the interval tend to zero. Therefore, it holds

$$\lim_{k\to\infty}\left\|\mathbf{w}_{i_1}(k) - \mathbf{w}_{i_2}(k)\right\| = 0 \quad \forall i_1, i_2 \in \mathcal{V}_r \tag{22}$$

proving that regular agents reach consensus. At this point, we prove that the consensus value is a constant vector, $\overline{\mathbf{w}}$. Recall that the adjacency matrix of the mathematically equivalent graph achieved by (13), where adversarial agents are filtered out, is row stochastic. Therefore, from (22) and the definition of $l_i^z(k)$ in (10), $\forall i \in \mathcal{V}_r$, it follows:

$$\begin{aligned} \lim_{k\to\infty} l_i^z(k) &= \lim_{k\to\infty}\sum_{j=1}^{n} a_{ij}^z(k) w_j^z(k) \\ &= \lim_{k\to\infty}\sum_{j\in\mathcal{V}_r} a_{ij}^z(k) w_j^z(k) = \lim_{k\to\infty} w_i^z(k). \end{aligned} \tag{23}$$

On the other hand, the update rule in (10) yields to

$$\lim_{k\to\infty} w_i^z(k+1) = \lim_{k\to\infty} l_i^z(k) - \lim_{k\to\infty} c(k)d_i^z(k). \tag{24}$$

Since in view of (11) it holds $\lim_{k\to\infty} c(k)d_i^z(k) = 0$, by combining (23) and (24), one obtains

$$\lim_{k\to\infty} w_i^z(k+1) = \lim_{k\to\infty} w_i^z(k) \tag{25}$$

which holds $\forall z$, $z \in \{1, \ldots, m\}$ and shows that consensus vector of regular agents is constant, proving the desired result. ■

B. Resiliency Analysis

In the previous section, we demonstrated that the regular agents reach a consensus that is not influenced by adversaries. We now need to prove that the agents converge to a sensible solution of the optimization problem in (9). To demonstrate this result, we first consider the following auxiliary lemma.

Lemma 5: If Assumption 1 holds, by implementing SETA in Algorithm 1 $\forall i_l \in \mathcal{V}_l^z(k)$ $\forall i_n \in \mathcal{V}_n^z(k)$ and $\forall i_h \in \mathcal{V}_h^z(k)$, $z \in \{1, \ldots, m\}$, there exists a finite time step $T < \infty$ such that for all $k > T$, it holds $d_{i_l}^z(k) > d_{i_n}^z(k) > d_{i_h}^z(k)$.

Proof: To prove the result, first note that according to (10), the difference $l_{i_l}^z(k) - l_{i_n}^z(k)$ can be written as

$$l_{i_l}^z(k) - l_{i_n}^z(k) = \frac{1}{q}\Big(w_{i_l}^z(k) - w_{i_{\tilde{r}}^z(k)}^z(k)\Big). \tag{26}$$

Given the definition of $\mathcal{V}_l^z(k)$ and $\mathcal{V}_n^z(k)$ (Phase 1 of SETA), it holds by construction $w_{i_l}^z(k) - w_{i_{\tilde{r}}^z(k)}^z(k) < 0$.

Therefore, $l^z_{i_l}(k) - l^z_{i_n}(k) < 0$. Similarly, we can prove that $l^z_{i_n}(k) - l^z_{i_h}(k) < 0$. Therefore $\forall k$ it holds

$$l^z_{i_l}(k) < l^z_{i_n}(k) < l^z_{i_h}(k). \tag{27}$$

Next, recall that according to (10), the update rules for agents i_l and i_n are

$$\begin{aligned} w^z_{i_l}(k+1) &= l^z_{i_l}(k) - c(k)d^z_{i_l}(k) \\ w^z_{i_n}(k+1) &= l^z_{i_n}(k) - c(k)d^z_{i_n}(k). \end{aligned} \tag{28}$$

Then, if $d^z_{i_l}(k) \geq d^z_{i_n}(k)$, from (27) and (28) it follows:

$$l^z_{i_l}(k) - c(k)d^z_{i_l}(k) < l^z_{i_n}(k) - c(k)d^z_{i_n}(k) \tag{29}$$

leading to

$$w^z_{i_l}(k+1) \ < w^z_{i_n}(k+1), \text{ if } \quad d^z_{i_l}(k) \geq d^z_{i_n}(k). \tag{30}$$

With the same argumentation, one obtains

$$w^z_{i_n}(k+1) \ < w^z_{i_h}(k+1), \text{ if } \quad d^z_{i_n}(k) \geq d^z_{i_h}(k). \tag{31}$$

At this point, since in view of Assumption 1 the functions $f_i(.)$ are differentiable $\forall i$, and their gradients $\mathbf{d}_i(k)$ are continuous, we observe that, by virtue of Lemma 4, the gradients converge to different constant vectors according to the consensus value of states. This implies that there exists a large enough time step index s such that the ordering of gradients does not change for $\forall k \geq s$, that is, it holds $\forall i, j$, that if $d^z_i(s) \geq d^z_j(s)$ then $d^z_i(k) \geq d_j(k)^z \ \ \forall k \geq s$ and $z \in \{1, \ldots, m\}$.

In view of (26), the difference $l^z_{i_n}(k) - l^z_{i_l}(k)$ is equal to

$$l^z_{i_n}(k-1) - l^z_{i_l}(k-1) = \frac{1}{q}\Big(w^z_{i^z_r(k-1)}(k-1) - w^z_{i_l}(k-1)\Big). \tag{32}$$

Thus, the difference $w^z_{i_n}(k) - w^z_{i_l}(k)$ can be written as

$$\begin{aligned} w^z_{i_n}(k) - w^z_{i_l}(k) = \frac{1}{q}&\Big(w^z_{i^z_r(k-1)}(k-1) - w^z_{i_l}(k-1)\Big) \\ &- c(k-1)\Big(d^z_{i_n}(k-1) - d^z_{i_l}(k-1)\Big). \end{aligned} \tag{33}$$

Considering now the difference $w^z_{i^z_r(k-1)}(k-1) - w^z_{i_l}(k-1)$ in (33), it holds

$$\begin{aligned} &w^z_{i^z_r(k-1)}(k-1) - w^z_{i_l}(k-1) \\ &= \frac{1}{q}\Big(w^z_{i^z_r(k-2)}(k-2) - w^z_{i_l}(k-2)\Big) \\ &\quad - c(k-2)\Big(d^z_{i^z_r(k-1)}(k-2) - d^z_{i_l}(k-2)\Big) \end{aligned} \tag{34}$$

where $i^z_r(k-1) \in \mathcal{V}^z_n(k-1)$ by construction. Let us define

$$D = \inf_{k \geq s}\Big\{d^z_{i^z_r(k+1)}(k) - d^z_{i_l}(k)\Big\} > 0 \tag{35}$$

$$C = \sup_{k \geq s}\Big\{w^z_{i^z_r(k)}(k) - w^z_{i_l}(k)\Big\} > 0 \tag{36}$$

for which it holds $0 < C < \infty$, since regular agents reach consensus (Lemma 4).

We first analyze the case in which $i^z_r(k) \in \mathcal{V}^z_n(k-1)$ or $i^z_r(k) \in \mathcal{V}^z_h(k-1)$, that is, the agent removed at time step k was contained in $\mathcal{V}^z_n(k-1) \cup \mathcal{V}^z_h(k-1)$ at the previous time step. From the definition of $i^z_r(k)$ in Phase 2 of Algorithm 1, we observe that in this case it holds $d^z_{i^z_r(k)}(k-1) \geq d^z_{i_n}(k-1)$. Otherwise, with a similar argumentation to (30), we would obtain $w^z_{i_n}(k) < w^z_{i^z_r(k)}(k)$, which is a contradiction given the definition of $i^z_r(k)$ for $i_l \in \mathcal{V}^z_l(k)$. We want to prove that if $d^z_{i_l}(s) < d^z_{i_n}(s)$, then the agents i_l and i_n switch at a finite time step T, that is, there exists $s < T < \infty$ such that $w^z_{i_n}(T) < w^z_{i_l}(T)$. From (34) we can recursively find out that, the following inequality is verified for $k > s$:

$$w^z_{i_n}(k) - w^z_{i_l}(k) \leq \frac{C}{q^{k-s}} - D\sum_{r=1}^{k-s}\frac{c(k-r)}{q^{r-1}}. \tag{37}$$

To prove that there exists a time step $T < \infty$ such that $w^z_{i_n}(T) - w^z_{i_l}(T) < 0$, we show that the first term in the right-hand side of (37) tends to zero faster than the second one. In view of (37), we obtain

$$\begin{aligned} \lim_{k\to\infty}\frac{D\sum_{r=1}^{k-s}\frac{c(k-r)}{q^{r-1}}}{\frac{C}{q^{(k-s)}}} &= \frac{D}{C}\lim_{k\to\infty}\sum_{r=s}^{k-1} q^{(r+1-s)}c(r) \\ &\geq \frac{D}{C}\lim_{k\to\infty}\sum_{r=s}^{k-1} c(r) = \infty. \end{aligned} \tag{38}$$

We now consider the case in which $i^z_r(k) \in \mathcal{V}^z_l(k-1)$, that is, the agent $i^z_r(k) \in \mathcal{V}^z_n(k)$ was previously contained in $\mathcal{V}^z_l(k-1)$ at time step $k-1$. In this case, according to (30), there must exist an agent i^z_s with higher gradient, that is, $d^z_{i^z_r(k)}(k-1) \leq d^z_{i^z_s}(k-1)$, which switched its position with $i^z_r(k)$, that is, such that $i^z_s \in \mathcal{V}^z_n(k-1) \cup \mathcal{V}^z_h(k-1)$ and $i^z_s \in \mathcal{V}^z_l(k)$. This implies that the switching is a step toward sorting the agents with respect to their gradients. We can then reinitialize $s = k$ and go forward again. Since the number of agents is bounded and all the replacements are toward sorting the agents, there exist finite time steps in which it holds $i^z_r(k) \in \mathcal{V}^z_l(k-1)$. Considering a time step s in which $i^z_r(k) \in \mathcal{V}^z_n(k-1) \cup \mathcal{V}^z_h(k-1)$ for all $k > s$, the same case as in the above is obtained and it is proven that there exists a time step $T < \infty$ such that $w^z_{i_n}(T) - w^z_{i_l}(T) < 0$. We can write a similar argument for $d^z_{i_n}(s) < d^z_{i_h}(s)$ and $w^z_{i_n}(s) < w^z_{i_h}(s)$. Moreover, from (30) and (31) we observe that if the regular agents are ordered they will not change their sets. This concludes the proof. ∎

At this point, we can formally state our main theorem regarding the resiliency of SETA in case of attacks. We define the powerset $\mathcal{P}(\mathcal{V}_r)$ of the set of regular agents $\mathcal{V}_r$ and its subset $\mathcal{S}$ equal to $\mathcal{S} = \{x \in \mathcal{P}(\mathcal{V}_r) \,|\, |x| = n - 2F\}$.

Theorem 1: Consider n agents among which up to F can be adversarial. Assume F satisfies (12) and Assumption 1 holds. Then, by implementing SETA in Algorithm 1 in a shared memory architecture, the states $\mathbf{w}_i$ of regular agents $i \in \mathcal{V}_r$ converge in the smallest hypercube containing all $\mathbf{w}^*_j$ defined as

$$\mathbf{w}^*_j = \arg\, min_{\mathbf{w}} \sum_{i\in\Omega_j} f_i(\mathbf{w})$$

for $\Omega_j \in \mathcal{S}$, $j = 1, \ldots, |\mathcal{S}|$.

Proof: By Lemma 5, it follows that there exists a time step T in which the regular agents do not swap their position

between sets $\mathcal{V}_h^z(k)$, $\mathcal{V}_l^z(k)$ and $\mathcal{V}_n^z(k)$ for $k > T$. Let us consider $k > T$ and define the sets Ω_h^z and Ω_l^z containing q regular agents with highest and lowest-state values $w_i^z \ \forall i$, respectively. Let us introduce the following auxiliary vectors given the component z:

$$\overline{\mathbf{M}} = \arg\min_{\mathbf{w}} \sum\nolimits_{i\in\Omega_h^z} f_i(\mathbf{w})$$
$$\underline{\mathbf{m}} = \arg\min_{\mathbf{w}} \sum\nolimits_{i\in\Omega_l^z} f_i(\mathbf{w}). \tag{39}$$

From Lemma 4, it follows that the regular agents reach consensus despite the presence of adversaries in the network. Let $\overline{\mathbf{w}}$ be the consensus vector, that is, $\overline{\mathbf{w}} = \mathbf{w}_i(k) \ \ \forall i \in \mathcal{V}_r$ as $k \to \infty$. To prove the theorem, we show that the equivalent following inequality is satisfied:

$$\underline{m}^z \le \overline{w}^z \le \overline{M}^z \qquad \forall z \in \{1, \dots, m\}. \tag{40}$$

Assume by contradiction that $\overline{w}^z = \overline{M}^z + \epsilon$, with $\epsilon > 0$. We introduce the following variables:

$$M^z(k) = \sum_{i\in\Omega_h^z} w_i^z(k), \ m^z(k) = \sum\nolimits_{i\in\Omega_l^z} w_i^z(k). \tag{41}$$

By considering that there are maximum F adversarial agents and that it holds $|\Omega_h^z| = q = n - 2F$, we deduce that there must exist at least F regular agents with lower states $w_i^z(k)$ which are not included in Ω_h^z. This implies that $\mathcal{V}_l^z(k) \cap \Omega_h^z = \emptyset$ for all $k > T$. By applying a similar reasoning, we obtain $\mathcal{V}_h^z(k) \cap \Omega_l^z = \emptyset$ for all $k > T$. Therefore, by virtue of the update procedure in (10), the following inequality holds true:

$$\frac{1}{q} m^z(k) \le l_i^z(k) \le \frac{1}{q} M^z(k)$$

which leads to

$$w_i^z(k+1) \ \le \frac{1}{q} M^z(k) - c(k) d_i^z(k). \tag{42}$$

Considering (42) and (41) for $M^z(k+1)$, it follows:

$$M^z(k+1) \ \le M^z(k) - c(k) \sum_{i\in\Omega_h^z} d_i^z(k) \tag{43}$$

which can be generalized for $M^z(k+Z)$ as

$$M^z(k+Z) \ \le M^z(k) - \sum_{k=k_0}^{Z-1} \left(c(k) \sum_{i\in\Omega_h^z} d_i^z(k) \right). \tag{44}$$

Since we assumed $\overline{w}^z = \overline{M}^z + \epsilon$, there exists a time step $k_0 > T$ such that the following inequality is verified for $k \ge k_0$ and $i \in \mathcal{V}_r$:

$$\overline{M}^z + \frac{1}{2}\epsilon \le w_i^z(k) \le \overline{M}^z + 2\epsilon \tag{45}$$

which, summing for all $j \in \Omega_h$ and considering the definition in (41), leads to

$$q\left(\overline{M}^z + \frac{1}{2}\epsilon\right) \le M^z(k) \le q\left(\overline{M}^z + 2\epsilon\right). \tag{46}$$

In view of (39) and the convexity of the loss functions, we observe that if it holds $w_i^z(k) = \overline{M}^z$ for all $i \in \Omega_h^z$, then the sum of respective gradients are null, that is, it holds $\sum_{i\in\Omega_h^z} d_i^z(k) = 0$. Therefore, in order to fulfill (45) it must hold $\sum_{i\in\Omega_h^z} d_i^z(k) > 0$ for $k > k_0$. At this point, recalling that $\sum_{k=k_0}^{\infty} c(k) = \infty$, we obtain from (44) that, for a large enough Z, it holds $M^z(k_0 + Z) < q(\overline{M}^z + (1/2)\epsilon)$, which is in contradiction with (46). By applying a similar reasoning, one can reach $\underline{m}^z \le \overline{w}^z$. This concludes the proof. ■

Remark 3: The adversarial agent's behavior is not constrained in the above proof. Therefore, SETA is resilient against both data and model poisoning attacks described in Section IV.

Remark 4: The challenges of training FL algorithms are significantly increased with non-IID datasets, as extensively discussed in [42]. In some extreme cases, such as when each agent possesses data samples, containing only one class in a multiclass classification problem, the non-IID distribution can lead the global model to fail in achieving satisfactory performance. In the mathematical analysis presented in this article, no assumptions about the properties of the datasets are made. This means that Algorithm 1 leads the states of all regular agents to a consensus vector, which is not influenced by adversaries and belongs to the hypercube defined in Theorem 1, regardless of whether the dataset is IID or non-IID. However, as noted in [43], the local decision vectors are closer in the IID case than in the non-IID case. Therefore, the convergence hypercube is a smaller neighborhood around the global optimal solution in the IID case compared to the non-IID case. Consequently, we can expect more accurate outcomes with IID datasets than with non-IID datasets.

The above theorem provides a region of convergence of the regular agents. A condition for reaching the optimal solution to (9) can be defined using the concept of *redundancy* in distributed optimization. In particular, the work in [44] proves that a necessary condition for finding a solution to (9) with up to F adversarial agents is that the cost functions $f_i(\mathbf{w})$ fulfill the *2F-redundancy* property, that is, that for any subset of agents, Ω, where $|\Omega| = n - 2F$ the following holds true

$$\arg\min_{\mathbf{w}} \sum_{i\in\Omega} f_i(\mathbf{w}) \in \mathcal{W}^* \tag{47}$$

where $\mathcal{W}^*$ is the convex set of optimal solutions of (9). Based on (47), we identify the condition for SETA to converge to an optimal solution of (9).

Corollary 1: Assume that the conditions of Theorem 1 hold and the $2F$-redundancy property is fulfilled. Then, by implementing SETA in Algorithm 1, the states $\mathbf{w}_i$ of regular agents $i \in \mathcal{V}_r$ converge to an optimal solution of (9).

Proof: The proof easily follows considering that, if the 2F-redundancy property is verified, then it holds $\underline{\mathbf{m}}, \overline{\mathbf{M}} \in \mathcal{W}^*$. According to Assumption 1, the loss functions are convex and from Theorem 1 $\underline{m}^z \le \overline{w}^z \le \overline{M}^z$, therefore, it holds $\overline{\mathbf{w}} \in \mathcal{W}^*$. ■

Note that reaching the exact optimal solution in $2F$-redundant problems can be viewed as a metric to evaluate resilient optimization algorithms. In particular, since $2F$-redundancy is a necessary condition to find the exact optimal solution of (9), a well-designed algorithm should converge to this solution when the $2F$-redundancy property is met.

VI. Simulation Results

In this section, we validate the resiliency of SETA against several attack types using two datasets with different levels of complexity and compare it against different baselines.

A. Setting

Datasets: We validate the proposed algorithm with two datasets: 1) the MNIST dataset [45], [46], collecting gray scale images of digits with resolution 28×28 and 2) the MWD in [33], collecting colored images of different weather conditions, that is, cloudy, sunny, rainy, and sunrise, with variable resolution. In both cases, our FL objective is to perform classification [of digits for 1) and weather condition in 2)] considering that each agent has access to a private local dataset D_i. The choice of these two datasets is motivated by the fact that the former a widely used standard dataset in literature, allowing us to conduct a relatively simple initial validation of our method; the latter presents a more challenging case study, which is valuable for validating the algorithm in realistic contexts that have practical applications. Specifically, as we operate within the context of precision agriculture robotics as for the European project CANOPIES,[1] the classification of weather conditions can help robots operate more safely in their environment. For instance, if the robots detect cloudy or rainy weather, they may decide to move to a sheltered location to avoid damage or to speed up their activities. The MNIST dataset is composed of 60000 training samples and 10000 testing samples, while MWD includes 1125 samples in total which we resize to a resolution of 50×50 and randomly split into 80% for training and 20% for validation.

In the following, we consider both IID and non-IID distribution of the training datasets among the agents. In the IID case, the training samples are uniformly distributed among the agents. In the non-IID case, the local datasets $\mathcal{D}_i$ are composed of random samples associated with k classes of the dataset. In the MNIST dataset, we select $k \sim \mathcal{U}(3, 4)$, that is, each agent has samples of either three or four digits in the local dataset, while for MWD case, we consider $k \sim \mathcal{U}(1, 2)$, that is, each agent has samples of one or two weather conditions (out of four). The testing samples are used to evaluate the performance.

Agents: In the MNIST case study, we analyze a system comprising $n = 100$ agents, out of which $n_a = F = 20$ agents are randomly designated as adversarial agents. In contrast, for the MWD case study, we consider a system consisting of $n = 50$ agents, out of which $n_a = F = 10$ agents are randomly selected as adversarial agents. Each agent has a two-layer fully connected neural network with 100 hidden units in the MNIST case and 256 units in the MWD case. Leaky ReLU and Softmax activation functions are used for the hidden and the output layers, respectively. Note that, although the leaky ReLU is not differentiable at the origin, continuous pseudo derivatives of leaky ReLU are proposed in the literature, for example, [47], which can be used to satisfy Assumption 1 and the mathematical soundness of the backpropagation learning procedure. Weights are initialized by each agent according to a uniform distribution $\mathcal{U}(\mathbf{0}_m, 0.01 \cdot \mathbf{1}_m)$ and models are trained for 1000 steps with a step size $c(k)$ evolving as

$$c(k) = \begin{cases} c_0, & \text{if } k \leq s \\ \gamma\, c_0/((k-s)c_0 + \gamma), & \text{if } k > s \end{cases}$$

which fulfills conditions in (11) and where s are warm-up steps set to $s = 300$, γ is a positive constant $\gamma = 500$ and $c(0) = 1$ for the MNIST case and $c(0) = 0.01$ with MWD case.

Attacks and Baselines: To validate the resiliency of SETA, we implement three local model poisoning attacks and one data poisoning attack. Regarding the former, we consider: 1) *Gaussian* attack, as reported for example in [21] and [30], where each adversarial agent i sends a parameter vector $\mathbf{w}_i$ obtained according to a Gaussian distribution, that is, $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}_m, \mathbf{1}_m)$; 2) *model flipping* attack, as used for example in [30], where each adversary flips the sign of weights computed according to SETA, and communicates the flipped parameters; and 3) *optimization*-based attack, introduced in [13], where each adversary determines the local parameter vector by solving an optimization problem. In particular, given the direction along which the global parameter vector would be updated in the absence of attacks, the optimization objective is to deviate the global parameter vector as much as possible toward the inverse of this direction. Regarding the data poisoning attack, a *label flipping* attack [40] is considered. For the MNIST case, each label is exchanged with the previous digit, that is, 1 is set instead of 2, 2 instead of 3 and so on, while 9 is used in place of label 0. Similarly, for the MWD, we exchange each label with the previous one in the ordered list consisting of cloudy, rainy, sunny, and sunrise. Hence, samples originally belonging to the class rainy are assigned the label cloudy, and so on. For all the attacks, we consider that the adversarial agents begin the attack from the start of our simulations.

To compare results, we consider the following baselines.

1) Centralized SGD, representing the *ideal* case where data is not distributed among agents and a single server computes the parameters without any attack, that is, no aggregation rule is used and no adversaries are present. Therefore, this baseline provides an upper bound for the accuracy of SETA and the other FL baselines.
2) Average aggregation, which is the typical aggregation rule where the parameters are updated by performing a simple coordinatewise average of all agents' local parameters.
3) FedProx [48], that introduces a penalty term in the optimization objective to mitigate the impact of stragglers and non-IID data. It encourages local models at each agent to be close to a global model while considering the differences in local datasets.
4) Median aggregation, where the coordinatewise median is computed to update the parameters.
5) Trimmed average, where parameters are updated by filtering out the F highest and smallest values in a coordinatewise fashion and computing the remaining average values.

[1] https://canopies.inf.uniroma3.it/

TABLE II
PERCENTAGE ACCURACY ON THE MNIST TEST SET ACHIEVED BY ALL THE CONSIDERED AGGREGATION RULES USING IID (LEFT) AND NON-IID (RIGHT) DATA DISTRIBUTIONS. GAUSSIAN, MODEL FLIPPING, OPTIMIZATION-BASED, AND LABEL FLIPPING ATTACKS ARE CONSIDERED. BEST RESULTS IN BOLD

		Results with iid distribution				Results with non-iid distribution			
	Algorithm	Gaussian	Model flip.	Opt.-based	Label flip.	Gaussian	Model flip.	Opt.-based	Label flip.
MNIST	Average	13.700	11.350	11.350	93.860	13.290	9.740	8.920	91.950
	FedProx	83.340	84.330	9.820	92.230	44.480	53.550	8.920	89.270
	Median	93.680	91.290	95.420	94.140	59.375	24.107	82.143	62.500
	Trim. avg.	93.760	89.540	**95.640**	**94.500**	93.430	82.370	95.350	**93.020**
	Krum	87.810	89.850	15.920	89.970	50.893	56.696	29.018	66.518
	Bulyan	91.720	**93.540**	10.520	93.870	53.270	66.870	19.600	78.830
	SETA	**93.800**	90.620	95.570	94.480	**93.760**	**82.390**	**95.480**	92.970

6) Krum [15], where the local parameter vector with lowest distance to its $n-F-2$ closest local parameter vectors is used.
7) Bulyan [19], where $n-2F$ local parameter vectors are recursively selected by resorting to an aggregation rule, and then they are combined by discarding, for each coordinate, the $2F$ values that are furthest from the median and by averaging the remaining ones. As aggregation rule, we resort to Krum as done in [19].

B. Results

MNIST Case Study: The performance reached with different aggregation rules and attacks is reported in Table II for the MNIST dataset. Percentage accuracy on the test set using the weights at the last training step is shown. In particular, on the left IID distribution for the local datasets $\mathcal{D}_i$ is considered, while on the right non-IID distribution is used. Results in case of Gaussian (first column of each block), model flipping (second column), optimization-based (third column) and label flipping (forth column) attacks are provided. Accuracy equal to 97.1% in case of no attack (not reported in the table) is achieved by the centralized SGD, representing the performance to aim with FL approaches. Starting from the case of IID distribution (in the left part of the table), we can observe that a maximum decrease in performance equal to ≈7% is achieved by SETA compared to the centralized in case of model flipping, while a decrease lower than 3% is reached with the other attacks. Similar performance is obtained by median and trimmed average aggregation rules. Significantly lower performance is reached instead using average aggregation rule achieving $< 14\%$ with most attacks. Poor performance (lower than 16%) in case of optimization-based attack is also obtained with FedProx, Krum and Bulyan methods, although Bulyan achieves the highest accuracy in case of model flipping attack, reaching 93.54% compared to 90.62% of SETA. In the case of non-IID distribution (right part of the table), we can notice that a significant drop in performance is recorded with most attacks when using average and FedProx aggregation rules, both achieving, for instance, only $\approx 10\%$ with optimization-based attack. An overall performance decrease is also recorded with median, Krum, and Bulyan, achieving performance lower than 30% in the respective worst cases. Best accuracy is achieved instead with most of the attacks using the proposed SETA. In particular, a maximum decrease in performance, with respect to the ideal centralized case, equal to $\approx 15\%$ is achieved in the case of model flipping, while a decrease lower than 5% is reached with the other attacks. Slightly lower performance is obtained in general by trimmed average compared to SETA.

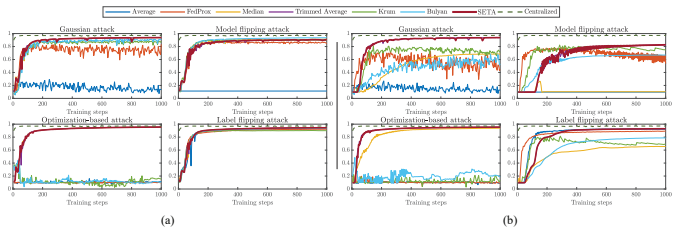

Fig. 5. Accuracy on the MNIST test set during the learning process using IID (left) and non-IID (right) data distributions. Results achieved by average (dark blue), Fedprox (light red), median (yellow), trimmed average (purple), Krum (light green), Bulyan (light blue), and SETA (thick red lines) as well as by the centralized architecture (dashed dark green lines) are reported. Three model poisoning attacks (top part and bottom left of each figure) and one data poisoning attack (bottom right of each figure) are considered. (a) MNIST case study: IID data distribution. (b) MNIST case study: Non-IID data distribution.

Fig. 5(a) and (b) show the accuracy on the test set during the learning process with IID and non-IID distributions, respectively. The four different attacks are reported, that is, Gaussian (in the top left), model flipping (in the top right), optimization-based (in the bottom left), and label flipping (in the bottom right) attacks. Centralized SGD results are shown with dotted dark green lines, while average, FedProx, median, trimmed average, Krum, Bulyan, and the proposed SETA algorithm are reported with dark blue, light red, yellow, purple, light green, light blue, and dark red solid lines, respectively. First, the plots confirm that the results in Table II are also observed during the entire learning process. More specifically, we can observe that in all the cases SETA outperforms the others approaching more closely the centralized results and, as expected, shows comparable performance with respect to the trimmed average. Second, the plots show the robustness of the proposed algorithm compared to the other baselines toward chattering phenomena that are induced by the model flipping and the optimization-based attacks.

MWD Case Study: Table III summarizes the results obtained with MWD using the different aggregation rules and attacks. Results with IID (left) and non-IID (right) distributions of the dataset are shown. Similar observations to the MNIST

TABLE III
PERCENTAGE ACCURACY ON THE MWD TEST SET ACHIEVED BY ALL THE CONSIDERED AGGREGATION RULES USING IID (LEFT) AND NON-IID (RIGHT) DATA DISTRIBUTIONS. GAUSSIAN, MODEL FLIPPING, OPTIMIZATION-BASED, AND LABEL FLIPPING ATTACKS ARE CONSIDERED. BEST RESULTS IN BOLD

		Results with iid distribution				Results with non-iid distribution			
	Algorithm	Gaussian	Model flip.	Opt.-based	Label flip.	Gaussian	Model flip.	Opt.-based	Label flip.
MWD	Average	7.589	31.696	29.018	80.357	9.375	29.018	29.018	**81.696**
	FedProx	50.893	77.232	16.518	73.661	36.161	58.036	29.018	51.339
	Median	**81.250**	80.804	82.143	80.804	59.375	24.107	82.143	62.500
	Trim. avg.	**81.250**	79.464	**83.929**	79.911	**79.911**	67.857	81.250	78.571
	Krum	49.107	72.768	29.018	75.000	50.893	56.696	29.018	66.518
	Bulyan	42.411	**81.250**	21.875	**81.696**	51.339	**80.804**	42.857	76.786
	SETA	**81.250**	79.464	82.589	79.911	**79.911**	66.964	**83.482**	78.571

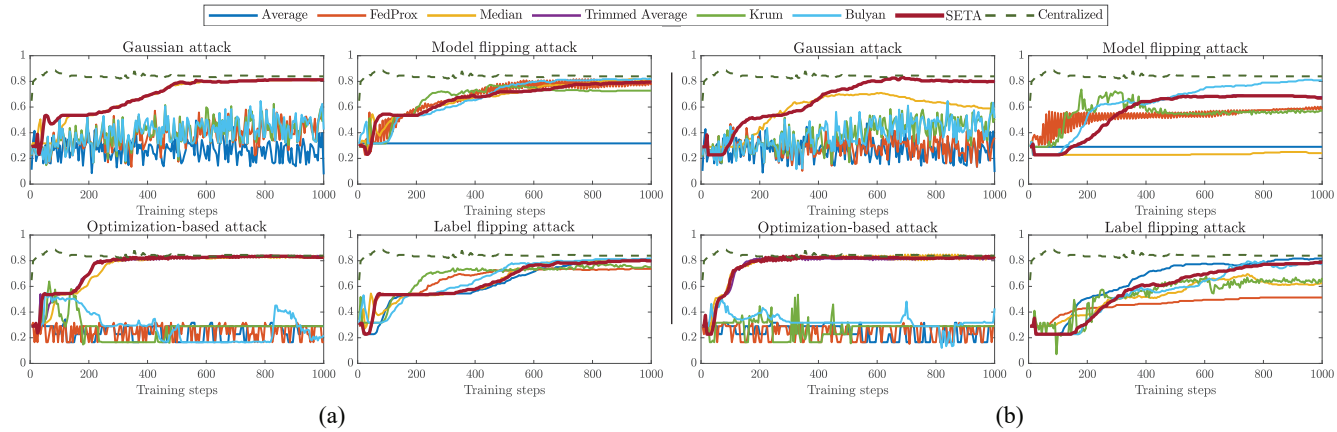

Fig. 6. Accuracy on the MWD test set during the learning process using IID (left) and non-IID (right) data distributions achieved by the average (in dark blue), Fedprox (light red), median (yellow), trimmed average (purple), Krum (light green), Bulyan (light blue), and SETA (thick red lines) as well as by the centralized architecture (dashed dark green lines) are reported. Three model poisoning attacks (top part and bottom left of each figure) and one data poisoning attack (bottom right of each figure) are shown. (a) MWD case study: IID data distribution. (b) MWD case study: Non-IID data distribution.

case apply to the MWD case study. In this case, classification accuracy equal to 83.93% is achieved in the ideal scenario of no attack and centralized SGD (not reported in the table). For the IID distribution setting, we can notice that SETA achieves a maximum performance decrease of approximately 4.5%, compared to the centralized approach, with the model flipping attack. However, both Gaussian and optimization-based attacks result in a performance decrease of less than 3% using SETA. Similar performance is obtained when using the trimmed average and median aggregation rules. As for MNIST, lower performance is obtained in general with average, FedProx, Krum, and Bulyan aggregation rules. Specifically, average, FedProx, and Krum rules achieve performance lower than 51% for Gaussian and optimization-based attacks, while Bulyan obtains the best performance equal to 81.25% for the model flipping attack, but performs poorly on Gaussian and optimization-based attacks, achieving accuracy lower than 42%. For the non-IID distribution setting (on the right), an overall performance decrease is obtained with average, FedProx, median, Krum, and Bulyan aggregation rules, except for the label flipping attack with average method, reaching best performance $\approx$ 82% and the model flipping attack with Bulyan method, reaching best performance $\approx$ 81%. In the remaining attacks, the best accuracy, similar to the trimmed average, is achieved by SETA. Specifically, a maximum performance decrease, compared to the centralized case, equal to $\approx$ 17% is achieved in the case of model flipping, while a decrease lower than 5.4% is reached with the other attacks.

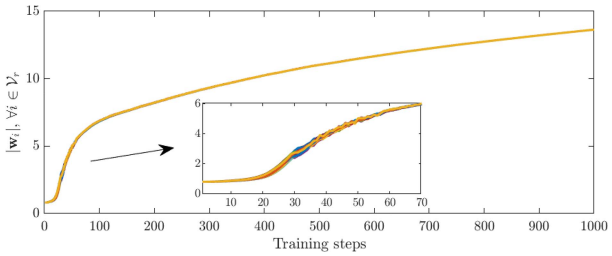

Fig. 7. Evolution of the norm of the state vectors $\mathbf{w}_i \quad \forall i \in \mathcal{V}_r$ of regular agents using SETA.

Similar to Fig. 5, we show the accuracy on the MWD test set during the learning process in Fig. 6(a) and (b), for the IID and non-IID distributions, respectively. As for the MNIST case study, the figure shows that the results in Table III are also recorded during the entire learning process. Furthermore, the figure makes evident that the learning process is significantly more challenging in this case study compared to the MNIST one. However, in all tests, SETA is able to closely approach the centralized results without any chattering phenomena.

Consensus of the Agents: Fig. 7 reports the evolution of the norm of the state vectors $\mathbf{w}_i \quad \forall i \in \mathcal{V}_r$ associated with regular

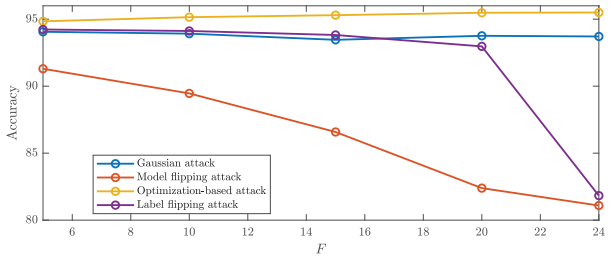

Fig. 8. Accuracy of SETA with varying number of adversaries using MNIST dataset and non-IID distribution.

agents that is achieved using SETA. For instance, the case of non-IID distribution and optimization-based attack with MNIST dataset is considered, but similar trends are observed in the other cases. The figure shows that, coherently with Lemma 4, during the initial training steps (reported in the zoom figure), different norm values are recorded, while as the training advances, the agents reach consensus on the weights.

Impact of Varying Number of Adversaries: We analyzed the performance of SETA when varying the number of adversaries within the set $\{5, 10, 15, 20, 24\}$, where each value satisfies the $2F + 1$ robustness condition with $n = 100$. Fig. 8 depicts the accuracy achieved by SETA under various attacks, that is, Gaussian, model flipping, optimization-based, and label flipping attacks, using the MNIST dataset and non-IID distribution. The figure shows that stable results are obtained with Gaussian and optimization-based attacks as the number of adversaries increases. In contrast, a progressive decrease in performance is observed in the case of the model flipping attack, making it the most severe attack for SETA in our tests. Finally, a stable behavior is observed with the label flipping attack until $F = 20$, while a noticeable drop in performance is evident at $F = 24$. This can be motivated by the fact that, with non-IID data distribution, increasing the number of adversarial agents might cause the number of adversarial flipped samples of a specific digit to exceed the number of benign samples of it. As a result, correct classification of that digit becomes challenging, leading to a sudden 10% drop in accuracy.

VII. Conclusion

In this article we proposed a resilient FL algorithm, namely, SETA, to tackle the presence of adversarial agents that can compromise the distributed learning process. Given local models of the agents, SETA first performs a coordinatewise clustering of the local parameters. Then, it applies a coordinatewise trimmed average, in which the trimmed values are selected according to the respective cluster. SETA enables FL both in standard server–worker architecture and in shared memory settings, where a trusted server is not needed. We formally proved the convergence bounds of the algorithm against model and data poisoning attacks and validated the approach using MNIST and MWD datasets. We compared the performance with respect to average, median, trimmed average, FedProx, Krum, and Bulyan aggregation rules in case of different attack types. Simulation results confirmed the effectiveness of SETA in adversarial settings, providing generally better performance than average, median, FedProx, Krum, and Bulyan aggregation rules as well as comparable results to trimmed average.

As part of future work, we aim to evaluate SETA's performance on additional real-world and heterogeneous datasets and extend it to fully distributed settings, eliminating both the shared memory and the need for a trusted server.

References

[1] J. Le, X. Lei, N. Mu, H. Zhang, K. Zeng, and X. Liao, "Federated continuous learning with broad network architecture," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3874–3888, Aug. 2021.

[2] X. Li, Z. Qu, B. Tang, and Z. Lu, "FedLGA: Toward system-heterogeneity of federated learning via local gradient approximation," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 401–414, Jan. 2024.

[3] L. Zhang, W. Cui, B. Li, Z. Chen, M. Wu, and T. S. Gee, "Privacy-preserving cross-environment human activity recognition," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1765–1775, Mar. 2023.

[4] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[5] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in Industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[6] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[7] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Oct. 2021.

[8] R. Moghadam and H. Modares, "Resilient autonomous control of distributed multiagent systems in contested environments," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3957–3967, Nov. 2019.

[9] C. Deng and C. Wen, "MAS-based distributed resilient control for a class of cyber-physical systems with communication delays under DoS attacks," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2347–2358, May 2021.

[10] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2018, pp. 19–35.

[11] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. 32nd Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8011–8021.

[12] H. Li, G. Li, and Y. Yu, "ROSA: Robust salient object detection against adversarial attacks," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4835–4847, Nov. 2020.

[13] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Security Symp.*, 2020, pp. 1605–1622.

[14] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[15] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 118–128.

[16] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks," *IEEE Trans. Signal Process.*, vol. 68, pp. 4583–4596, Jul. 2020, doi: 10.1109/TSP.2020.3012952.

[17] A. Gouissem, K. Abualsaud, E. Yaacoub, T. Khattab, and M. Guizani, "Collaborative Byzantine resilient federated learning," *IEEE Internet Things J.*, vol. 10, no. 18, pp. 15887–15899, Sep. 2023.

[18] H. Masuda, K. Kita, Y. Koizumi, J. Takemasa, and T. Hasegawa, "Byzantine-resilient secure federated learning on low-bandwidth networks," *IEEE Access*, vol. 11, pp. 51754–51766, 2023.

[19] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.

[20] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6893–6901.

[21] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1544–1551.
[22] Y. Tao et al., "Byzantine-resilient federated learning at edge," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2600–2614, Sep. 2023.
[23] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 903–912.
[24] C. Xie, S. Koyejo, and I. Gupta, "Zeno++: Robust fully asynchronous SGD," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10495–10503.
[25] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2018.
[26] Z. Wu and Z. Li, "Distributed robust optimization algorithms over uncertain network graphs," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4451–4458, Jun. 2022.
[27] L. Su and N. H. Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2227–2233, May 2020.
[28] Y. Lou, L. Yu, S. Wang, and P. Yi, "Privacy preservation in distributed subgradient optimization algorithms," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2154–2165, Jul. 2018.
[29] W. Zeng and M.-Y. Chow, "Resilient distributed control in the presence of misbehaving agents in networked control systems," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2038–2049, Nov. 2014.
[30] S. Guo et al., "Byzantine-resilient Decentralized stochastic gradient descent," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 4096–4106, Jun. 2022.
[31] Z. Yang and W. U. Bajwa, "ByRDiE: Byzantine-resilient distributed coordinate descent for Decentralized learning," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 4, pp. 611–627, Dec. 2019.
[32] M. Kaheni, E. Usai, and M. Franceschelli, "Resilient constrained optimization in multi-agent systems with improved guarantee on approximation bounds," *IEEE Control Syst. Lett.*, vol. 6, pp. 2659–2664, 2022.
[33] G. Ajayi, 2018, "Multi-class weather Dataset for image classification," Dataset, Mendeley. [Online]. Available: https://data.mendeley.com/datasets/4drtyfjtfy/1.
[34] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. Am. Control Conf.*, 2005, pp. 1859–1864.
[35] M. Cao, A. S. Morse, and B. D. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 575–600, 2008.
[36] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Select. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.
[37] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
[38] L. Zhao, M. Mammadov, and J. Yearwood, "From convex to Nonconvex: A loss function analysis for binary classification," in *Proc. IEEE Int. Conf. Data Min. Workshops*, 2010, pp. 1281–1288.
[39] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, 2004.
[40] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. Eur. Symp. Res. Comput. Security*, 2020, pp. 480–501.
[41] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5252–5256.
[42] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," 2021, *arXiv:2106.06843*.
[43] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.
[44] N. Gupta and N. H. Vaidya, "Fault-tolerance in distributed optimization: The case of redundancy," in *Proc. Symp. Princ. Distrib. Comput.*, 2020, pp. 365–374.
[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[46] Y. Zhang, M. Cui, L. Shen, and Z. Zeng, "Memristive quantized neural networks: A novel approach to accelerate deep learning on-chip," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 1875–1887, Apr. 2021.
[47] Z. Hu, Y. Li, and Z. Yang, "Improving convolutional neural network using pseudo derivative ReLU," in *Proc. Int. Conf. Syst. Informat.*, 2018, pp. 283–287.
[48] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

Mojtaba Kaheni (Member, IEEE) received the M.Sc. and Ph.D. degrees in control engineering from the Shahrood University of Technology, Shahrood, Iran, in 2011 and 2019, respectively.

He is a Postdoctoral Researcher with the Akademin för Innovation, Design och Teknik, Mälardalen University, Västerås, Sweden. He was a Visiting Scholar with the University of Florence, Florence, Italy, from May 2017 to October 2017, and he has also served as a Postdoctoral Researcher with the University of Cagliari, Cagliari, Italy, from August 2020 till December 2022. His research interests include distributed optimization, multiagent systems, and nonlinear control.

Martina Lippi (Member, IEEE) received the M.Sc. (cum laude) and Ph.D. degrees in information engineering from the University of Salerno, Fisciano, Italy, in 2017 and 2020, respectively.

In 2019, she was a Visiting Scholar with the KTH Royal Institute of Technology, Stockholm, Sweden. From November 2020 to June 2022, she was a Postdoctoral Researcher with Roma Tre University, Rome, Italy, where she has been an Assistant Professor since June 2022. Her research interests include multimanipulator systems, distributed control, and human–robot interaction.

Andrea Gasparri (Senior Member, IEEE) received the Laurea (cum laude) degree in computer science and the Ph.D. degree in computer science and automation from Roma Tre University, Rome, Italy, in 2004 and 2008, respectively.

He is currently a Professor with the Department of Civil, Computer Science and Aeronautical Technologies Engineering, Roma Tre University. His research interests include robotics, sensor networks, networked multiagent systems, and precision agriculture.

Dr. Gasparri was an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS from 2017 to 2021. Since 2021, he has been an Associate Editor for the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. He has been a member of the Steering Committee for the IEEE RAS Technical Committee on Multirobot Systems since 2014, the IEEE CSS Technical Committee on Networks and Communications since 2015, and the IEEE Technical Committee on Agricultural Robotics since 2021.

Mauro Franceschelli (Senior Member, IEEE) received the Laurea degree (cum laude) in electronic engineering and the Ph.D. degree from the University of Cagliari, Cagliari, Italy, in 2007 and 2011, respectively.

He was a Visiting Professor with the Georgia Institute of Technology, Atlanta, GA, USA, and the University of California at Santa Barbara, Santa Barbara, CA, USA. He is an Associate Professor with the Department of Electrical and Electronic Engineering, University of Cagliari. His research interests include consensus problems, gossip algorithms, multiagent systems, multirobot systems, nonsmooth analysis, distributed optimization, and electric demand side management.

Dr. Franceschelli received a Fellowship from the National Natural Science Foundation of China at Xidian University, Xi'an, China, in 2013. In 2015, he was awarded a position of an Assistant Professor (RTD-A) funded by the Italian Ministry of Education, University and Research. He serves as an Associate Editor for several conferences since 2015, for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING since 2021, and for *IFAC Nonlinear Analysis: Hybrid Sytems* since 2023. He is an Officer (Secretary) for the IEEE Italy Section Chapter of the IEEE Control Systems Society. He is a member of the Conference Editorial Board for the IEEE Control Systems Society since 2019.

Open Access funding provided by "Università degli Studi di Cagliari" within the CRUI CARE Agreement