

AdaFed: Performance-based Adaptive Federated Learning

ALESSANDRO GIUSEPPI, University of Rome “La Sapienza”, Italy

LUCREZIA DELLA TORRE, University of Rome “La Sapienza”, Italy

DANILO MENEGATTI, University of Rome “La Sapienza”, Italy

ANTONIO PIETRABISSA, University of Rome “La Sapienza”, Italy

Federated Learning is a distributed and privacy-preserving machine learning technique that allows local clients to learn a model without sharing their own data by coordinating with a global server. In this work, we present the Adaptive Federated Learning (AdaFed) algorithm, which aims at improving the training performance of deep neural networks in Federated Learning settings by: (i) dynamically weighting the local models in the model averaging procedure; (ii) by adapting the loss function used by the federation at every communication round. We discuss the specialisation of AdaFed for both classification and regression tasks, providing several validation examples. Due to its adaptive design, the AdaFed algorithm showed a robust behaviour against unbalanced data distributions and adversarial clients.

CCS Concepts: • **Computer systems organization** → **Distributed architectures**; • **Computing methodologies** → *Distributed algorithms*; **Distributed artificial intelligence**; *Multi-agent systems*; **Intelligent agents**; **Cooperation and coordination**.

Additional Key Words and Phrases: Federated Learning, Deep Neural Networks, Distributed Learning Systems

ACM Reference Format:

Alessandro Giuseppe, Lucrezia Della Torre, Danilo Menegatti, and Antonio Pietrabissa. 2022. AdaFed: Performance-based Adaptive Federated Learning. *ACM Trans. Graph.* 37, 4, Article 111 (August 2022), 6 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Federated Learning (FL) is a distributed learning solution to address Machine Learning (ML) problems without the need of collecting the available data in a single data center. FL finds application in scenarios in which the data are distributed over a multitude sources that, for privacy or communication constraints, cannot share it among them or with a centralised entity. The need of analysing data *locally* becomes of crucial importance when dealing with personal data (e.g., collected by smartphones) or sensitive information (e.g., regarding the customers of a company). FL is then a technology enabler to analyse data in heavily regulated fields such as healthcare [Li et al. 2020a].

Authors' addresses: Alessandro Giuseppe, giuseppi@diag.uniroma1.it, University of Rome “La Sapienza”, Via Ariosto 25, Rome, Italy, 00187; Lucrezia Della Torre, dellatorre@diag.uniroma1.it, University of Rome “La Sapienza”, Via Ariosto 25, Rome, Italy, 00187; Danilo Menegatti, menegatti@diag.uniroma1.it, University of Rome “La Sapienza”, Via Ariosto 25, Rome, Italy, 00187; Antonio Pietrabissa, pietrabissa@diag.uniroma1.it, University of Rome “La Sapienza”, Via Ariosto 25, Rome, Italy, 00187.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

0730-0301/2022/8-ART111 \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

In FL, the server's model, is updated at every *communication round* by averaging the models of the federated clients, trained on their locally available data. In this work, we propose the Adaptive Federated Learning algorithm, AdaFed, a two-step procedure to improve both the model averaging and the local training processes by i) dynamically weighting the client models contributions to the federation based on their performance, and ii) adapting the federated loss function depending the global model performance at each communication round.

For the sake of presentation clarity, the proposed solution is designed to extend the formulation and results of original FL algorithm Federated Averaging (FedAvg) [McMahan et al. 2016a], but the concepts behind the proposed innovations are independent of the specific implementation and may be seamlessly translated to other algorithms such as FedProx [Li et al. 2020b]. The evaluation of AdaFed in terms of versatility, performance improvements and capability of addressing new challenging scenarios is illustrated by comparison with FedAvg on different tasks involving several different data-sets.

The remainder of this paper is organized as follows: In Section 2 we present an overview of related work. In Section 3 we present our framework, AdaFed, specialising it for classification and regression tasks in Section 3.1 and 3.2. Some validation experiments are presented in Section 4, while Section 5 draws the conclusions and highlights future work.

2 RELATED WORKS AND MAIN CONTRIBUTIONS

FL was introduced in 2016 by the authors of [McMahan et al. 2016b] and, in its original formulation [McMahan et al. 2016a,b], FL was proposed specifically to address the collaboration among a group of smartphones and consisted in an iterative model-averaging procedure. According to such procedure, local model updates, independently computed by the smartphones, were gathered and averaged by a centralised server that then propagated the updated *global* model in the network.

As discussed in the recent surveys [Li et al. 2019; Lim et al. 2020; Yang et al. 2019], since its introduction FL was extended to different architectures and uses cases, with several works focusing on enhancing its privacy preserving and security related characteristics [Bonawitz et al. 2017; Geyer et al. 2017; Nasr et al. 2018; Truex et al. 2019] to prevent direct or indirect data leakage [Wang et al. 2019], and on reducing the communication cost associated to the distributed training [Konečný et al. 2016; Lin et al. 2017b; Sattler et al. 2019]. FL found application in several domains, spacing from smartphones/Internet of Things tasks such as Natural Language Processing (NLP) [Hard et al. 2018; Ramaswamy et al. 2019; Yang et al. 2018], image analysis [Liu et al. 2020; Luo et al. 2019] and

distributed sensing and computing [Imteaj and Amini 2019; Lim et al. 2020], to scenarios in which organisations and institutions cooperate to obtain better models to analyse complex and highly confidential data, as typical in the healthcare domain [Brisimi et al. 2018; Chen et al. 2020a; Sheller et al. 2019]. Contrary to federated database systems, in which data can be distributed freely by a central entity, the typical scenario for a FL application is characterised by the need of analysing data partitioned *as given*, implying that FL algorithms need to be able to cope with data that are:

- non-IID and imbalanced, as the geographical dislocation of the federated organisations or data sources may significantly affect the data distribution and collecting procedures;
- extremely distributed, as in scenarios in which the federated entities are smart connected devices their number is likely to be orders of magnitude more than the quantity of their individual data samples.

This work explores the concept of associating to each client a different weight in the model averaging procedure depending on their contribution to the federation. Several other works explore this research direction, as the FOCUS algorithm [Chen et al. 2020b] where the authors design a procedure to determine the weighting factors based on a credibility score assigned to each client. Such credibility score aims at minimizing the sensitivity of the federation model to a possible disparity in the labeling quality of the clients' datasets, hence addressing two of the main implications of considering distributed data sources: the different collecting procedures and human factors. The FOCUS algorithm relies on the idea that the combination of the performance of the global model on the clients' datasets and the performance of the clients' models on the server dataset (which is assumed to be correctly labeled) provides an indicator of the quality of the clients' labelling. The framework proposed by ADAFED is not specifically designed for providing robustness to labelling quality disparity, and focuses more on the evaluation of the clients' model instead of inferring the quality of their data. Furthermore, FOCUS determines its credibility scores under the assumption that a correctly labeled client dataset is IID with the one available to the server (Theorem 1 in [Chen et al. 2020b]), whereas, as we will show in the simulations, ADAFED does not make any assumption on the clients' data. In fact, ADAFED will be shown to be an enabler approach to allow FL algorithms to cope with scenarios in which data are distributed in extremely unbalanced ways and even in the presence of malicious/compromised clients. A similar approach is followed by [Wang et al. 2020], where a federated way to estimate the Shapey Value (SV) [Roth 1988], that captures the value of the clients' data for the federation, is proposed. The federated estimation of the SV allows several useful properties, as the detection of poor/noisy labelling, malicious clients and communication minimisation. The main idea behind the algorithm in [Wang et al. 2020] is to sort the clients according to their SV (i.e., their contribution to the federation) and then let only the most contributing ones participate to the model averaging. ADAFED, on the other hand, performs a weighted model averaging, where the weights given to all the clients are dynamically adapted based on their model performance, reducing the impact of - or even excluding - the low performing/malicious ones. We mention that a federated

version of SV was also explored in [Song et al. 2019], where a similar metric is developed to allow the distribution of revenue/profits to the clients depending on their contribution to the federation. In this sense, a future work may explore the combination of federated SV estimation with the adaptive model averaging included in ADAFED by utilising the SV values as clients' weights. One of the most promising contribution in the FL field was made by the authors of [Li et al. 2020b] with the so-called FEDPROX algorithm. FEDPROX, similarly to ADAFED, builds on top of FEDAVG formulation to cope with some of its limitations, and in particular proposes two improvements to deal with systems' heterogeneity (i.e., significant differences in the optimisation capabilities of the federation clients) and statistical heterogeneity (i.e., significant non-IID nature of the data distribution). FEDPROX archives its properties by proposing an extended loss function that includes a so-called proximal term that limits the impact of different local training settings (e.g., epochs number, optimiser settings, ...) and non-IID local data distribution on the clients' models. In principle, ADAFED may be deployed on a federation that utilises a loss function with the structure proposed by FEDPROX, but the impact of the adaptive features of ADAFED on the convergence properties of FEDPROX should be explored in detail in a future work.

Another contribution of this work is in the direction of combining FL with the recent research trend of *adaptive loss functions* [Barron 2019; Heydari et al. 2019; Miranda and Zuben 2015; Teixeira et al. 2019], that showed promising results in problems characterised by complex loss functions. An example is multi-objective learning, where the models are required to optimize multiple loss functions at the same time seeking a Pareto-like optimality [Heydari et al. 2019]. Complex loss functions are also found in computer vision [Barron 2019; Redmon et al. 2016; Teixeira et al. 2019] where they are commonly used in solutions such as You-Only-Look-Once (YOLO) [Redmon et al. 2016] or Single-Image-Super-Resolution (SISR) [Ayyoubzadeh and Wu 2020]. The fundamental idea at the basis of adaptive loss approaches is that the loss function itself evolves during training depending on the recent intra-training performances of the model. FL, due to its iterative nature, provides an ideal setting for such an updating solution, as the models' loss function can be adapted at every communication round.

A summary of the main contributions of the paper follows:

- (1) This work introduces a new adaptive federated learning algorithm, AdaFed, in which two mechanisms drive and speed up the learning process
 - (a) by dynamically weighting the contributions of the clients' models in the server's model based on their performance (over a representative test set), and
 - (b) by modifying the local training with an adaptive update of the loss functions of the clients' models at each communication round, depending on the performance achieved by the server's model on its test set.
- (2) It also illustrates the versatility of ADAFED in learning different models, including convolutional neural networks for both classification and regressions tasks, also in the transfer learning setting.

Algorithm 1 AdaFed: Performance-Based Adaptive Federated Learning algorithm

```

1: SERVER'S UPDATE:
2: for each round  $t = 1, 2, \dots, R$  do
3:   ClientsUpdate
4:   for each client  $i = 1, 2, \dots, K$  do
5:     receive the client's model  $w_i$ 
6:     evaluate  $w_i$  on the server test set
7:     use the evaluation to determine the weight  $p_i$ 
8:   end for
9:   update the server's model  $w_S \leftarrow \frac{\sum_{i=1}^{n_c} w_i p_i}{\sum_{i=1}^{n_c} p_i}$ 
10:  evaluate its performance  $p_S$  on the server test set
11:  adapt the loss function  $l$  depending on  $p_S$ 
12:  propagate  $w_S$  and  $l$  to the clients
13: end for

14: ClientsUpdate:
15: for each client  $i = 1, 2, \dots, K$  do
16:    $w_i \leftarrow w_S$ 
17:   for each local epoch  $j = 1, 2, \dots, E$  do
18:     for each mini-batch  $b$  of size  $B$  do
19:        $w_i \leftarrow w_i - \eta \nabla l(w, b)$ 
20:     end for
21:   end for
22:  return  $w_i$  to server
23: end for

```

- (3) It discusses through simulations the performance improvement that the proposed method is able to achieve in various scenarios with respect to one of the most common baseline FL algorithm, FEDAVG [McMahan et al. 2016a].

3 ADAFED: ADAPTIVE FEDERATED LEARNING

The proposed solution for FL draws inspiration from the well known ensemble learning algorithm AdaBoost (AdaBoost) [Freund and Schapire 1995]. In AdaBoost, a weak learner is trained at each epoch (commonly in the form of a *decision stump* for classification tasks) and the data samples are associated with a weighting factor proportional to their contribution to the loss attained by the learner. Consequently, at the next epoch, the new learner will focus more on the miss-evaluated data.

The AdaFed algorithm pseudo-code is written in the Algorithm 1 table. Differently from AdaBoost solutions, AdaFed does not involve ensemble learners and, instead, aims at making the model averaging procedure at the backbone of FL an adaptive process to improve the performance of the global model learned by the federation. This result is pursued by the following two step-procedure, that will be specialised for classification problems in section 3.1 and for regression problems in section 3.2:

- (1) **Weighted Model Average:** during the server update (lines 4-9 of Algorithm 1), the collected models are evaluated on a common test set and are weighted according to their performance for averaging. The performance metric can in principle

be any quantity that captures how well the model performs for the given task (e.g., accuracy for classification tasks), or may even be set to a quantity such as the SV to evaluate the client's contribution to the federation.

- (2) **Adaptive Loss:** the server propagates to the federation both the updated model and a new loss function, which was adapted to the performance of the server own model on a dedicated test set according to a use-case dependant metric (e.g., update class weights for classification tasks depending on recall) (lines 10-12 of Algorithm 1).

We formulated the Algorithm 1 similarly to FEDAVG, as this allows for a clearer presentation of its innovations. We mention that privacy preserving features, communication minimization policies (e.g., random selection of clients for each averaging) and other enhancements (e.g., the heterogeneity reduction approaches of FEDPROX) may be included in the ADAFED formulation as long as they are compatible with the standard FL algorithm structure of iterative local updates and centralised model averaging.

As in [Chen et al. 2020b], ADAFED aims at evaluating the contribution level of the various clients to determine their weights in the model averaging, and to do so ADAFED assumes the availability of a representative server test set on which it is possible to evaluate the performance of the various clients' models for the given task, without any significant increase in the training complexity.

The combination of a dynamic weighted model averaging procedure with an adaptive loss function allows the federation to: i) give more weight to better performing clients, while reducing or preventing the negative influence of malicious/noisy ones (by lowering or setting to 0 their weights) and ii) give more attention to data samples needed to improve the model performance, as in the case of the most rare/harder to discern labels in classification tasks.

As mentioned, in the remainder of the paper, we will neglect aspects linked to communication efficiency and privacy preserving solution, as they are already well documented in the works [Li et al. 2019; McMahan et al. 2016a; Sattler et al. 2019; Yang et al. 2019] and are beyond the scope of the present work, which instead focuses on the model averaging and adaptive loss aspects.

3.1 Application: ADAFED for Classification Tasks

Multi-class classification is one of the most common ML task examples. A typical choice for the loss function utilised in this setting is the categorical cross-entropy:

$$l(X, Y) = -\frac{1}{M} \sum_{c=1}^C \sum_{m=1}^M y_m^c \log(\hat{y}_m^c), \quad (1)$$

where $x_m \in X$ and $y_m \in Y$ are the m -th data sample and label, respectively, in the dataset (X, Y) , M and C are respectively the number of data samples and classes, y_m^c and \hat{y}_m^c denote the c -th component of the vectors y_m and \hat{y}_m and are respectively the true and predicted labels for the sample m regarding class c . Note that \hat{y}_m^c is typically produced, for single-label problems, by a deep neural network with a *softmax* output activation function and can be interpreted as the probability of correctness for the given label.

As it is, the categorical cross-entropy does not compensate for imbalanced class distributions, that in our reference scenarios are

likely to characterise the datasets available to the various clients. A typical solution to improve the learning process in this kind of scenarios is to utilise a weighted categorical cross-entropy of the form

$$l(X, Y) = -\frac{1}{M} \sum_{c=1}^C \sum_{m=1}^M \kappa^c y_m^c \log(\hat{y}_m^c), \quad (2)$$

where κ^c is a class-dependent weight. In [Cui et al. 2019] the κ^c were set as proportional to the inverse of the number of data samples available for each class, while in Focal Loss [Lin et al. 2017a] a complex weight is associated to each class based on its classification difficulty. To the best of the authors knowledge, the weighting approaches available in literature either introduce static, rule based, coefficients or consider the weights as hyper parameters, which is in contrast to what was recently proposed for regression in [Barron 2019], where the loss is dynamically modified during training.

Inspired by [Barron 2019; Cui et al. 2019; Lin et al. 2017a], we choose the class dependent weight κ^c to be inversely proportional to the performance p_S obtained by the server model on the server test set with respect to its corresponding class, expressed by means of its F_1 -score, e.g. $\kappa^c = 1/(F_1^c + \epsilon)$, where $\epsilon < 1$ is a design parameter to limit κ^c . Note that this choice is however arbitrary as other metrics to evaluate p_S can be chosen, as shown in Section 4. The rationale behind this *adaptive loss logic* is to encourage the clients to focus, during their training phase, on classes which are misclassified by the global model.

Referring to Algorithm 1, this choice translates in having $p_S = \{F_1^c, \text{ for } c = 1, \dots, C\}$ (line 10) and updating the loss function l with the new weights κ^c derived from the F_1^c -scores (line 11).

The same idea is used in the Weighted Model Average step. Note that, in principle, the weight p_i of the i -th client can be computed via the performance of its model over the server test set with a different metric than the one used for the Adaptive Loss step, so instead of the F1 score one may utilise for example the model accuracy or the diagnostic odds ratio, depending on the specific use case

3.2 Application 2: AdaFed for Regression Tasks

The same principles discussed in section 3.1 can be directly adapted to regression tasks. For example, as it will be discussed in the simulation section, for model averaging in regression problems it may be suitable to select an application-dependent metric not directly related to the loss. In our example (simulation 3), where the objective is to estimate the number of cells in a given microscope photo, we will consider the mean absolute percentage error on the counting of cells, whereas the models' loss will be related to the mean squared error between a target image and the image generated by the deep neural network. Furthermore, it is in principle possible to consider a loss function as the one presented in [Barron 2019] and adapt or tune its hyper parameters at each communication round, depending on the server's model performance on its dataset.

4 EXPERIMENTS

In this section, ADAFED is tested and compared to FEDAVG in different scenarios. The focus of the experiments is on the performance of the server's model, neglecting communication efficiency arguments

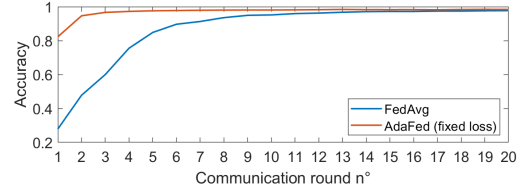


Fig. 1. Simulation 1 (MNIST): Evolution of the server model accuracy over communication rounds, evaluated on the separated test set of the server.

for the sake of the results clarity. Nevertheless, in principle ADAFED can be seamlessly modified to accommodate for the differential privacy and communication efficiency improvements proposed in literature to extend FEDAVG. For these reasons, in the following, we consider that all the clients send their model to the server at each communication round.

In the simulations we train our models on the MNIST [LeCun et al. 2010] dataset to better evaluate the effect of each of the proposed enhancements. The MNIST dataset consists in a set of 60k+10k labeled images of handwritten digits (from 0 to 9), and represents one of the most common baselines for classification tasks. Simulation 1 will discuss the benefit of the Weighted Model Average step, Simulation 2 will detail the effects of the Adaptive Loss step, and Simulation 3 will validate the robustness of the federation against adversarial clients. For the MNIST simulations, we set the parameters of the Algorithm 1 (Table 1) as $E = 5$ (number of epochs in the clients' update), $B = 100$ (batch size) and $R = 20$ (number of communication rounds). The server test set is composed by the whole MNIST test set, whereas clients' data are distributed as described in each of the following sections.

4.0.1 Simulation 1 (MNIST) - Validation of Weighted Model Averaging. In this experiment, we consider a federation constituted by $K = 6$ clients. The first five clients have access to 5500 samples from two classes only, with no overlapping, except for classes 0 and 5 which have 100 and 200 samples only, while the sixth one has 500 samples from each classes. The model of each client is a simple deep convolutional neural network from the official documentation of Keras [Official Keras.io documentation [n.d.]], composed by two 2-D convolutional layers followed by two dense layers. The performance weight p_i of the i -th client for ADAFED is computed as its accuracy times the number of its available data samples, i.e., $p_i = \text{accuracy}_i \times \text{\#training-data}_i$.

Figure 1 shows that ADAFED starts with a higher accuracy from the very first communication round, thus attaining a faster convergence rate compared to FEDAVG. The reason behind this behaviour is the presence of a client whose dataset is IID with respect to the server test set; despite its limited amount of data, the sixth agent exhibits from the start a better accuracy compared to the other clients which have visibility only on two different classes, meaning that its performance weight is significantly higher than the others in the model averaging procedure. Starting the next round with a better baseline model, ADAFED achieves, from the second round, an accuracy level achieved by FEDAVG only after twelve rounds.

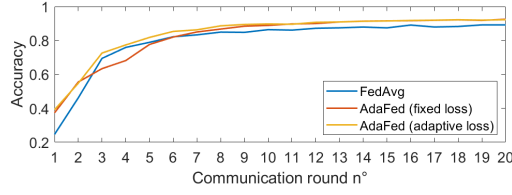


Fig. 2. Simulation 2 (MNIST): Evolution of the server model accuracy over communication rounds, evaluated on the separated test set of the server.

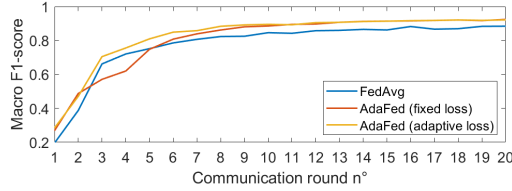


Fig. 3. Simulation 2 (MNIST): Evolution of the server model macro F1-score over communication rounds, evaluated on the separated test set of the server.

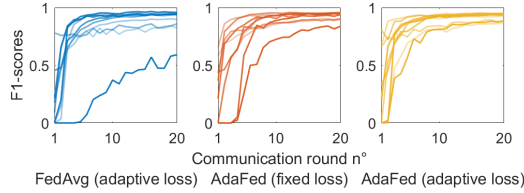


Fig. 4. Simulation 2 (MNIST): Per-class evolution of the server model F1-scores over communication rounds, evaluated on the separated test set of the server.

4.0.2 Simulation 2 (MNIST) - Validation of adaptive loss function. In this experiment, we want to evaluate the impact of adapting the clients loss function after each communication round. The scenario is similar to the one of Simulation 1, with data now randomly distributed in a less uniform way, as shown in Table 1.

The weighted categorical cross-entropy loss \mathcal{L}_2 of the clients is updated after each communication round according to the server model F1 scores for the various classes. In particular, the class weights κ^c are set to be equal to $1/(F_1^c + 0.1)$. The logic behind this choice is to increase the contribution to the loss for the classes that are more commonly miss-classified (i.e., $F_1^c \rightarrow 0$) by a factor of 10, whereas the weights of the better recognised classes (i.e., $F_1^c \rightarrow 1$) are slightly lowered.

Figure 2 reports the accuracy evolution of FedAvg compared with two different AdaFed federations, one implementing both the weighted averaging and the adaptive loss procedure described above, and another that only implements the weighted model averaging. Note that, in this simulation and in the following ones, for simplicity we set for AdaFed $p_i = \text{accuracy}_i$.

Both the AdaFed implementations are able to offset the FedAvg one by about 3%; moreover the adaptive loss one attains convergence faster. The reason for this different behaviour can be seen in Figure 3, which reports the Macro F1-scores (i.e. the arithmetic mean of the F_1^c -scores of all the classes). The adaptive loss implementation

of ADAFED outperforms the others, meaning that it is able to better discern classes for which limited samples are available. Figure 4 details the F_1^c -scores for all the ten classes of the datasets, and it is clear that ADAFED performs better in this sense, with the adaptive loss significantly contributing in having the scores of the classes evolve more uniformly and converge more rapidly.

4.0.3 Simulation 3 (MNIST) - Robustness to adversarial actors. Considering the same data distribution of Simulation 2, we show in this experiment the resiliency of ADA-FED to adversarial clients that try to lower the federation performances. To this end, we now add two additional clients 7 and 8 (i.e., $K = 8$) with the same data distribution of clients 3 and 4 but with incorrect labeling for respectively 50% and 100% of their samples. Furthermore, the malicious clients also discard the federated model they receive from the server and update their model only on the basis of their own data.

Figures 5 and 6 show that ADAFED (implemented with the adaptive loss and weighted averaging as in the previous simulation) is practically unaffected by the presence of the new malicious clients (the difference with respect to the previous simulation is about 0.01% in accuracy and 0.015 in the macro F1 score), while on the contrary FedAvg shows a significant drop in both accuracy (about 2%) and in the macro F1-score (about 0.1). We note that, in more complex cases, a different strategy may be followed for ADAFED to select the weights p_i (e.g. a quadratic or cubic function of the accuracy) to further decrease or annihilate the weights of bad performing (and hence potentially adversarial) clients.

5 CONCLUSIONS

In a federated learning setting, ADAFED envisages the dynamic update at every communication round of the clients' models loss functions, depending on the performance achieved by the server's model, combined with a weighted model averaging procedure that depends on the individual clients' model evaluation. This allows for a more efficient and robust training process, as the clients that contribute more to the federation are given more weight.

Future research directions involve the explicit inclusion of communication efficiency and privacy-related features in the framework. More extensive tests are also required to evaluate the algorithm in heavily distributed settings with hundreds of clients.

6 ACKNOWLEDGMENTS

This work has been partially funded by the Lazio region, in the scope of the project FedMedAI, POR FESR Lazio 2014 – 2020 (Azione 1.2.1), Prot. n. A0375-2020-36491 - 23/10/2020

Table 1. Simulation 2 - Data distribution among clients

Client	Classes									
1	10	0	30	10	30	50	20	20	10	10
2	10	0	0	500	100	0	0	500	100	500
3	0	0	30	500	100	150	500	0	0	500
4	0	0	30	0	100	0	500	500	100	0
5	0	10	30	500	0	0	500	500	0	500
6	0	10	10	10	10	100	10	0	10	3000

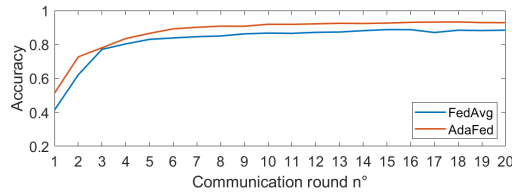


Fig. 5. Simulation 3 (MNIST): Evolution of the server model accuracy over communication rounds, evaluated on the separated test set of the server.

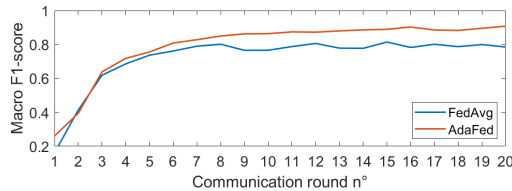


Fig. 6. Simulation 3 (MNIST): Evolution of the server model Macro F1-Score over communication rounds, evaluated on the separated test set of the server.

REFERENCES

- Seyed Mehdi Ayyoubzadeh and Xiaolin Wu. 2020. Adaptive Loss Function for Super Resolution Neural Networks Using Convex Optimization Techniques. arXiv:2001.07766
- Jonathan T. Barron. 2019. A General and Adaptive Robust Loss Function. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. <https://doi.org/10.1109/cvpr.2019.00446>
- Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM. <https://doi.org/10.1145/3133956.3133982>
- Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch. Paschalidis, and Wei Shi. 2018. Federated learning of predictive models from federated Electronic Health Records. *International Journal of Medical Informatics* 112 (April 2018), 59–67. <https://doi.org/10.1016/j.ijmedinf.2018.01.007>
- Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020a. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *IEEE Intelligent Systems* (2020), 1–1. <https://doi.org/10.1109/mis.2020.2988604>
- Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Biao Chen, and Zhiqi Shen. 2020b. Focus: Dealing with label quality disparity in federated learning. arXiv:2001.11359
- Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. <https://doi.org/10.1109/cvpr.2019.00949>
- Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 23–37. https://doi.org/10.1007/3-540-59119-2_166
- Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. arXiv:1712.07557
- Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604
- A. Ali Heydari, Craig A. Thompson, and Asif Mehmood. 2019. SoftAdapt: Techniques for Adaptive Loss Weighting of Neural Networks with Multi-Part Loss Functions. arXiv:1912.12355
- Ahmed Imteaj and M. Hadi Amini. 2019. Distributed Sensing Using Smart End-User Devices: Pathway to Federated Learning for Autonomous IoT. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. <https://doi.org/10.1109/csci49370.2019.00218>
- Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. arXiv:1610.05492
- Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist> 2 (2010).
- Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. 2019. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. arXiv:1907.09693
- Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020a. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (May 2020), 50–60. <https://doi.org/10.1109/msp.2020.2975749>
- Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020b. Federated Optimization in Heterogeneous Networks. arXiv:1812.06127 [cs.LG]
- Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* (2020), 1–1. <https://doi.org/10.1109/comst.2020.2986024>
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017a. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. <https://doi.org/10.1109/iccv.2017.324>
- Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2017b. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. arXiv:1712.01887
- Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. 2020. FedVision: An Online Visual Object Detection Platform Powered by Federated Learning. arXiv:2001.06202
- Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. 2019. Real-World Image Datasets for Federated Learning. arXiv:1910.11089
- H. Brendan McMahan, Eider Moore, Daniel Ramage, and Seth Hampson. 2016a. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP volume 54*. arXiv:1602.05629
- H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016b. Federated Learning of Deep Networks using Model Averaging. arXiv:1602.05629
- Conrado Silva Miranda and Fernando José Von Zuben. 2015. Multi-Objective Optimization for Self-Adjusting Weighted Gradient in Machine Learning Tasks. arXiv:1506.01113
- Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks. arXiv:1812.00910
- Official Keras.io documentation. [n.d.]. Example of Convolutional Neural Network for MNIST. https://keras.io/examples/mnist_cnn/
- Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated Learning for Emoji Prediction in a Mobile Keyboard. arXiv:1906.04329
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. <https://doi.org/10.1109/cvpr.2016.91>
- Alvin E Roth. 1988. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.
- Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems* (2019), 1–14. <https://doi.org/10.1109/tnnls.2019.2944481>
- Micah J. Sheller, G. Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2019. Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer International Publishing, 92–104. https://doi.org/10.1007/978-3-030-11723-8_9
- Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit Allocation for Federated Learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. <https://doi.org/10.1109/bigdata47090.2019.9006327>
- Brian Teixeira, Birgi Tamersoy, Vivek Singh, and Ankur Kapoor. 2019. Adaloss: Adaptive Loss Function for Landmark Localization. arXiv:1908.01070
- Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A Hybrid Approach to Privacy-Preserving Federated Learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security - AISec19*. ACM Press. <https://doi.org/10.1145/3338501.3357370>
- Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. A Principled Approach to Data Valuation for Federated Learning. arXiv:2009.06192 [cs.LG]
- Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. IEEE. <https://doi.org/10.1109/infocom.2019.8737416>
- Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (Feb. 2019), 1–19. <https://doi.org/10.1145/3298981>
- Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied Federated Learning: Improving Google Keyboard Query Suggestions. arXiv:1812.02903