Towards Sybil Resilience in Decentralized Learning

Thomas Werthenbach
Delft University of Technology
Delft, The Netherlands
T.A.K.Werthenbach@student.tudelft.nl

Johan Pouwelse
Delft University of Technology
Delft, The Netherlands
J.A.Pouwelse@tudelft.nl

***Abstract*—**

***Index Terms*—decentralized learning, sybil attack, sybil resilience**

I. Introduction

The rise of machine learning has resulted in an increasing number of everyday-life intelligent applications. As such, machine learning has been used in personal assistants [1], recommendation in social media [2] and music [3], and cybersecurity [4]. However, accurate machine learning models require large training datasets [5], [6], which can often be hard to obtain and store due to recent privacy legislation [7]. Federated learning [8] has become a promising alternative and widely adopted tool for crowd sourcing computationally expensive machine learning operations, reportedly having been used for training numerous industrial machine learning models [9]–[13]. Federated learning ensures the protection of privacy, as the user's data will not leave their device during training.

With federated learning, in contrast to centralized machine learning, training takes place on the end-users' personal devices, which are often referred to *edge devices* or *nodes*. The resulting trained models are communicated to a central server, commonly referred to as the *parameter server*, which aggregates these models using some predefined methodology. By only sharing the end user-trained models with the parameter server, the user's privacy is preserved, while obtaining comparable performance compared to centralized machine learning [14]. While there exist attacks in which training data can be reconstructed based on the gradient of the trained models [15], [16], defense mechanisms against this attack have been proposed [17], [18].

However, federated learning suffers from some disadvantages. For instance, the parameter server aggregates the models of all participating nodes, inducing heavy communication costs and a potential bottleneck in the learning process affecting the overall convergence time [19]. Secondly, the scalability in terms of the amount of nodes heavily varies depending on the aggregation method. In secure and robust federated learning aggregation methods, the incorporation of additional nodes during aggregation may result in significantly increased computational effort for the parameter server [20]. Thirdly, the parameter server performing the aggregation poses a single-point of failure [21]. Disruptions to the parameter server can cause downtime and hinder the overall model training process, particularly in architectures where edge devices require the globally aggregated model before continuing their training. An upcoming alternative aiming to resolve these issues is *decentralized learning*, also commonly referred to as *decentralized federated learning*. In decentralized learning, there exists no dedicated parameter server performing the aggregation and the edge devices form a distributed network, e.g. a peer-to-peer network, in which each node individually performs aggregation on their neighbours' models (see Figure 1). While the information available during aggregation is more limited relative to federated learning, it has been shown that decentralized learning has the potential to obtain similar results compared to federated learning [22]. Models are exchanged between individual devices and aggregated on individual scale using some predefined aggregation method, alleviating the communicative bottleneck and single point of failure issues imposed on federated learning, and paving the path for boundless scalability.

While decentralized learning solves the scalability challenges faced in federated learning, it is still vulnerable to byzantine environments [23]. Since the predefined aggregation method in decentralized learning does not have access to all models in the network, aggregation is performed with less information compared to federated learning, resulting in relatively less resistance against possible poisoning attacks [24]. Poisoning attacks are can generally be categorized in two categories, namely those of *targeted poisoning attacks* and *untargeted poisoning attacks*. Targeted poisoning attacks focus on achieving a specific goal an adversary aims to achieve such as the label-flipping attack [25], [26] and the backdoor attack [27]–[29]. On the other hand, untargeted poisoning attacks aim to hinder the result of the training process in some way without any particular goal in mind. The effect of these attacks can often be amplified through combining them with the Sybil attack [30], in which an adversary controls a substantial amount of nodes to increase its influence. As such, an adversary may deploy the Sybil attack to rapidly spread their poisoned model through the network. In this work, we focus exclusively on targeted poisoning attacks amplified by Sybil attacks in decentralized learning.

Prior work on resilience against poisoning attacks combined with Sybil attacks in distributed machine learning has mainly been done in federated learning settings. One popular example of such work is *FoolsGold* [31], which aims to increase Sybil resilience under the assumption that all Sybils will

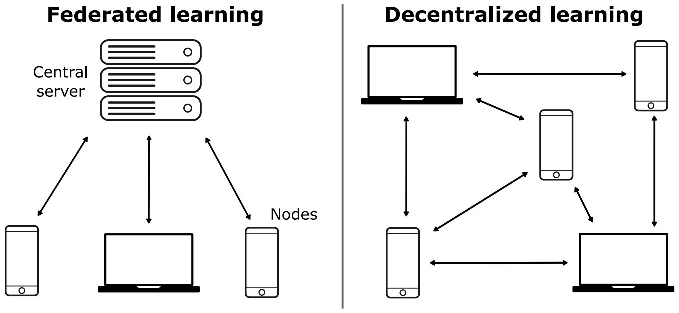

Fig. 1. Federated learning compared to decentralized learning. Arrows represent a connection between two nodes and indicates the two connecting nodes share model updates with eachother.

broadcast similar gradients during each round of training. By dynamically adapting the aggregation weights of peers' models based on their similarity with others, experimental results suggest that FoolsGold has the potential to provide effective protection against Sybil attacks in small-scale and simple federated learning settings.

In this work, we experimentally demonstrate *FoolsGold*'s inability to scale to an unbounded number of edge devices in federated learning and inept defensive capabilities against targeted poisoning attacks in decentralized learning. We suggest an improved version of FoolsGold, named NAME-OF-ALGORITHM[1], which shows significant resilience towards defending against targeted poisoning attacks whilst enjoying the boundless scalability offered by decentralized learning. More specifically, we achieve this by introducing a probabilistic gossiping mechanism for knowledge spreading. Finally, we empirically evaluate this algorithm on numerous types of Sybil attacks and show its ability to obtain increased Sybil resilience.

To the best of our knowledge, there exists only a single other work on defensive algorithms against poisoning attack in decentralized learning [32]. Moreover, this paper is the first to study Sybil attacks in decentralized learning. In short, our contributions are the following:

- We evaluate FoolsGold, a popular Sybil resilience algorithm in federated learning, and assess its compatibility with decentralized learning in Section III.
- We present NAME-OF-ALGORITHM, a pioneering algorithm for Sybil resilience with boundless scalability in decentralized learning, in Section VI.
- We perform an empirical evaluation of NAME-OF-ALGORITHM's performance in VII
- Maybe: We provide a convergence analysis on NAME-OF-ALGORITHM in section VIII.

II. Background

A. Federated learning

Federated learning was first proposed by Google [8] as an alternative for training machine learning models on anonymized user data

- Explain more in-depth how federated learning works → formal definitions
- Refer to Figure 1
- Explore some implementations of popular (simple) FL algorithms.
- FedAVG
- FedSGD

B. Decentralized learning

- Explain more in-depth how decentralized learning works → formal definitions?
- Refer to Figure 1
- Explore some implementation of popular (simple) DL algorithms.

C. Targeted poisoning attacks

- Briefly revisit targeted and untargeted poisoning attacks. We focus on targeted.
- Provide formal definitions of the label-flipping attack and the backdoor attack.

D. The Sybil attack

- Formal definition of Sybil attack
- In our context, most Sybil attacks may use botnets to increase their reachability and network throughput.
- Seuken and Parks on strongly and weakly benificial Sybil attacks.

III. Related work

A. FoolsGold

Explain FoolsGold [31] and show two graphs in which FoolsGold is used in both federated and decentralized settings (and show that it does not work as well in decentralized learning if there is no more than a single attack edge to every honest node).

How our work is different:

- It can be deployed in decentralized learning.
- It suffers less from the computationally expensive aggregation method. According to Foolsgold's authors, the cosine similarity function was the most expensive operation.

Furthermore, we performed an extensive evaluation of FoolsGold in both federated learning and decentralized learning. These are our results...

B. Resilient Averaging Gradient Descent

Resilient Averaging Gradient Descent (RAGD) [32] is a novel algorithm for mitigating poisoning attacks in decentralized learning.
How our work is different:

- RAGD naively assumes that malicious model updates will be quite different compared to honest model, but this may not necessarily be the case for label-flipping attacks or backdoor attacks.

[1] NAME-OF-ALGORITHM stands for

- RAGD assumes the existence of a static adjacency matrix, defining the edge weights between any two nodes. It also assumes that any attack edge has a weight of $0 < \epsilon < \frac{1}{2}$.
- We assume that nodes will not be fully connected.

C. Krum

Distance based

IV. Preliminaries

1) We assume that there exists some incentive for utilizing Sybils. This may be an upper bound on the maximum amount of connections any node can have with other nodes. An alternative may be a communication bottleneck, such as network speed, which incentivizes the use of a botnet as sybils to help distribute the poisoned model more rapidly.
2) We assume that adversaries perform a Sybil attack through hijacking other nodes such that they can play as a man-in-the-middle, thereby
3) maybe we don't need to send entire models, but just the cosine similarity between their own model and the other model?
4) WEAKNESS: as nodes have access to much less information in decentralized learning, it may incorrectly classify honest nodes as sybils if they are remotely similar.

V. Threat model

VI. Design

- Explain FoolsGold (cannot assume everyone knows it)
- Pseudocode?
- Explain gossiping models → the probabilistic property occurs two-fold, ① when selecting a peer to request a model from and ② when selecting what model to send to the requesting peer.
- Add figure
- Include somewhere that our max degree dampens single-attackers and that gossip mechanism prevents multi-attackers

VII. Evaluation

A. Experimental setup

- DAS6 → IPv8 → Gumby
- Attacks:
  - Label-flipping attack. from [31], [33]
  - backdoor attack. from [31]
  - a little is enough? from [33]
  - fall of empires? from [33]
  - sign-flipping? from [33]

B. Results

VIII. Analysis

IX. Discussion

X. Conclusion

References

[1] E. V. Polyakov, M. S. Mazhanov, A. Y. Rolich, L. S. Voskov, M. V. Kachalova, and S. V. Polyakov, "Investigation and development of the intelligent voice assistant for the internet of things using machine learning," in *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, 2018, pp. 1–5.

[2] B. T.K., C. S. R. Annavarapu, and A. Bablani, "Machine learning algorithms for social media analysis: A survey," *Computer Science Review*, vol. 40, p. 100395, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013721000356

[3] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 627–636. [Online]. Available: https://doi.org/10.1145/2647868.2654940

[4] S. A. Salloum, M. Alshurideh, A. Elnagar, and K. Shaalan, "Machine learning and deep learning techniques for cybersecurity: A review," in *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*, A.-E. Hassanien, A. T. Azar, T. Gaber, D. Oliva, and F. M. Tolba, Eds. Cham: Springer International Publishing, 2020, pp. 50–57.

[5] J. Prusa, T. M. Khoshgoftaar, and N. Seliya, "The effect of dataset size on training tweet sentiment classifiers," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 96–102.

[6] J. Hestness, S. Narang, N. Ardalani, G. F. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," *CoRR*, vol. abs/1712.00409, 2017. [Online]. Available: http://arxiv.org/abs/1712.00409

[7] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash, "Data minimization for gdpr compliance in machine learning models," *AI and Ethics*, pp. 1–15, 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[9] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[10] P. Navarro, C. Fernández, R. Borraz, and D. Alonso, "A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3d range data," *Sensors*, vol. 17, no. 12, p. 18, Dec 2016. [Online]. Available: http://dx.doi.org/10.3390/s17010018

[11] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *CoRR*, vol. abs/1811.03604, 2018. [Online]. Available: http://arxiv.org/abs/1811.03604

[12] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *CoRR*, vol. abs/1812.02903, 2018. [Online]. Available: http://arxiv.org/abs/1812.02903

[13] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," *CoRR*, vol. abs/1903.10635, 2019. [Online]. Available: http://arxiv.org/abs/1903.10635

[14] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving ai," *Communications of the ACM*, vol. 63, no. 12, pp. 33–36, 2020.

[15] L. Lyu and C. Chen, "A novel attribute reconstruction attack in federated learning," *CoRR*, vol. abs/2108.06910, 2021. [Online]. Available: https://arxiv.org/abs/2108.06910

[16] H. Yang, M. Ge, K. Xiang, and J. Li, "Using highly compressed gradients in federated learning for data reconstruction attacks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 818–830, 2023.

[17] H. S. Sikandar, H. Waheed, S. Tahir, S. U. R. Malik, and W. Rafique, "A detailed survey on federated learning attacks and defenses," *Electronics*, vol. 12, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/2/260

[18] P. Qiu, X. Zhang, S. Ji, Y. Pu, and T. Wang, "All you need is hashing: Defending against data reconstruction attack in vertical federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2212.00325

[19] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A communication-efficient algorithm for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3973–3983. [Online]. Available: https://proceedings.mlr.press/v119/hamer20a.html

[20] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning," *CoRR*, vol. abs/2009.11248, 2020. [Online]. Available: https://arxiv.org/abs/2009.11248

[21] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X2033065X

[22] I. Hegedűs, G. Danner, and M. Jelasity, "Decentralized learning works: An empirical comparison of gossip learning and federated learning," *Journal of Parallel and Distributed Computing*, vol. 148, pp. 109–124, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731520303890

[23] J. Hou, F. Wang, C. Wei, H. Huang, Y. Hu, and N. Gui, "Credibility assessment based byzantine-resilient decentralized learning," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–12, 2022.

[24] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Computer Security – ESORICS 2020*, L. Chen, N. Li, K. Liang, and S. Schneider, Eds. Cham: Springer International Publishing, 2020, pp. 480–501.

[25] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Defending against the label-flipping attack in federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2207.01982

[26] D. Li, W. E. Wong, W. Wang, Y. Yao, and M. Chau, "Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means," in *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, 2021, pp. 551–559.

[27] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[28] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *CoRR*, vol. abs/1911.07963, 2019. [Online]. Available: http://arxiv.org/abs/1911.07963

[29] C. Wu, X. Yang, S. Zhu, and P. Mitra, "Mitigating backdoor attacks in federated learning," *CoRR*, vol. abs/2011.01767, 2020. [Online]. Available: https://arxiv.org/abs/2011.01767

[30] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 251–260.

[31] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *CoRR*, vol. abs/1808.04866, 2018. [Online]. Available: http://arxiv.org/abs/1808.04866

[32] Y. Mao, D. Data, S. Diggavi, and P. Tabuada, "Decentralized learning robust to data poisoning attacks," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 6788–6793.

[33] S. Farhadkhani, R. Guerraoui, N. Gupta, L. N. Hoang, R. Pinot, and J. Stephan, "Making byzantine decentralized learning efficient," 2022. [Online]. Available: https://arxiv.org/abs/2209.10931