Defending Byzantine Attacks in Ensemble Federated Learning: A Reputation-based Phishing Approach

Beibei Li, *Member, IEEE,* Peiran Wang, *Student Member, IEEE,* Qinglei Kong, *Member, IEEE,* Yuan Zhang, *Member, IEEE,* and Rongxing Lu, *Fellow, IEEE*

***Abstract*—Emerging as a promising distributed learning paradigm, federated learning (FL) has been widely adopted in many fields. Nonetheless, a big challenge for FL in real-world implementation is Byzantine attacks, where compromised clients can mislead or poison the training model by falsifying or manipulating the local model parameters. To solve this problem, in this paper, we present a reputation-based Byzantine robust-FL scheme (called FLPhish) for defending Byzantine attacks under the Ensemble Federated Learning architecture (called EFL). Specifically, we first develop a novel ensemble FL architecture, EFL, which allows FL compatible with different deep learning models in different clients. Second, we craft a phishing algorithm for the EFL architecture to identify possible Byzantine behaviors. Third, a Bayesian inference based reputation mechanism is devised to measure each client's level of confidence and to further identify Byzantine clients. Last, we strictly analyze how the FLPhish scheme defend against backdoor attacks. Extensive experiments under different settings demonstrate that the proposed FLPhish achieves great efficacy in defending Byzantine attacks in EFL. FLPhish is tested with different fractions of Byzantine clients and different degrees of distribution imbalance. [1]**

***Index Terms*—Federated learning, ensemble learning, Bayesian inference-based reputation, phishing.**

I. Introduction

MANY elements of our daily lives and society have benefited from deep learning tasks in natural language processing, computer vision, and anomaly detection. To learn complex rules, such activities necessitate a large dataset. In most cases, these huge datasets are acquired by the application developers from users, such as the shopping app users' purchase record data, patients' clinical data and etc. Nonetheless, in recent years, there has been an explosion in social concerns about personal privacy, making it difficult to get data directly from users anymore. Under these circumstances, each individual's data is referred to as an 'Isolated Data Island'. The existence of each 'Isolated Data Island' drives the development of privacy-preserving solutions like Federated Learning (FL) [1]–[3]. Bonawitz *et al.* built the first FL system which is operated on Google's mobile phone to train a global model based on TensorFlow[1]. Its FL system could be operated on thousands of mobile phones. Moreover, a team of WeBank developed an FL scheme called FATE[2] for credit risk prediction. Furthermore, some former researchers have also applied FL in some industrial cyber–physical Systems [4]–[6].

FL is a distributed machine learning paradigm, which allows

TABLE I
Summary of Notations

Term	Description
s	central server in FL
c_i	the ith client in FL, $i = 1, 2, 3, ..., u$
d_i	the local dataset preserved by the ith client
C	the ensemble of all the clients
u	the number of clients
D_t	the unlabeled dataset chosen by s in each procedure
D	the unlabeled dataset preserved by s
n	the number of samples in D_t
B_t	the labeled dataset (*'bait'*) chosen by s in each procedure
B	the labeled dataset preserved by s
m	the number of samples in B_t
a_i^t	the accuracy of predictions of B_t made by c_i in tth procedure
q_i	the label of c_i to judge it is a malicious client or not
r_q	the threhold of malicious clients
x_l^t	the lth data point in D_t
b_i	the Byzantine attacker
σ	the 'trigger' in the backdoor attack
ι	the backdoor label in the backdoor attack
$\mathbf{M}$	global model preserved by s
$\mathbf{m_i}$	local models trained by the ith client
$\mathbf{k_i^t}$	the predictions (*'knowledge'*) made by the ith client in the tth procedure
$\mathbf{\hat{y}_l^t}$	the ensembled prediction of data point x_l^t
$\mathbf{\hat{y}_l^i}$	the prediction of lth data point made by ith client
$\mathbf{K_t}$	the aggregated labels (predictions) of the tth iteration's unlabeled dataset

This paper is an extended version of the paper titled 'FLPhish: Reputation-Based Phishing Byzantine Defense in Ensemble Federated Learning', which was published in IEEE ISCC 2021, and awarded 'Best Paper'.

B. Li and P. Wang are with the School of Cyber Science and Engineering, Sichuan Universsity, Chengdu, Sichuan, China 610065. Email: libeibei@scu.edu.cn; wangpeiran@stu.scu.edu.cn.

Q. Kong is with the Future Network of Intelligence Institute, The Chinese University of Hong Kong, Shenzhen, China 518172, and also with The University of Science and Technology of China, Hefei, China 230052. Email: kql8904@163.com.

Y. Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China 610054. Email: zy_loye@126.com.

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada E3B 5A3. Email: rlu1@unb.ca.

[1] https://federated.withgoogle.com/
[2] https://github.com/FederatedAI/FATE

the central server in the paradigm to produce a global model without getting each individual's private data. Instead of gathering private data from each user, the central server in FL aggregates all the model gradient updates from distributed clients to its global model. In each iteration of FL, the central server sends a model to each client. Each client updates the model using its private data and sends the model gradient update back to the central server. In the central server, all the clients' updates are aggregated to a global model gradient update, and the global model gradient update is utilized to update the global model. Thus, FL not only protects each participating individual's privacy, but also leverages the capabilities of the end users' computation and storage.

Since thousands of clients from different sources may participate in the training process, security issues also exist in the distributed FL system. On one hand, former researchers have already studied the privacy problems of FL and have proposed the corresponding schemes to enhance privacy protection in FL [7]–[9]. On the other hand, FL faces threats from the poisoning attacks launched by malicious attackers among the FL clients [10], [11]. And such attacks are referred to as Byzantine attacks in wireless communication network [12]–[15]. By poisoning the clients' datasets or directly manipulating the gradient updates, the incorrect gradient updates are sent by the malicious clients to the central server, which causes the global model to learn incorrect knowledge. As a result, this process renders the central server's global model obsolete. Furthermore, Byzantine attacks can be separated into two types according to the attack consequences. In the first type of Byzantine attacks, called denial-of-service attack (including untargeted attacks, targeted attacks, e.g), the Byzantine attackers intend to disturb the global model thus making it produce wrong predictions of the normal dataset [16]–[20]. In another type of Byzantine attacks, called backdoor attacks, the disturbed global model will make wrong predictions of the data samples which have 'backdoor' in them [21]–[25].

Former researchers have offered certain Byzantine-robust techniques to deal with malicious Byzantine clients under the FL application settings [26]–[35]. Byzantine-robust techniques try to construct a global model with high accuracy in the presence of a finite number of malicious clients. According to their different mechanisms, we divide Byzantine-robust approaches into two major types. The first (named Byzantine-Detection) is based on the development of a Byzantine-robust aggregation algorithm that distinguishes suspected clients from benign clients. The suspected clients' gradient updates are subsequently removed from the aggregation process by the server. For instance, in the DRACO scheme proposed by Chen *et al.*, each node analyzes duplicate gradients that the parameter server uses to mitigate the effects of adversarial updates [27]. Another Byzantine-robust technique (named Byzantine-Tolerance) seeks to ensure that the aggregation process is tolerant to poisoned updates from Byzantine clients without excluding Byzantine clients like Median [29]. In Median, The FL server sorts the values of each parameter and picks the median value of each parameter as the value to be utilized in global model updates. In this study, we provide a unique reputation-based phishing method (named FLPhish) to protect against Byzantine attacks in EFL, based on the preceding research. Our contributions are four-fold:

- We design a new FL architecture, Ensemble Federated Learning (called EFL), which utilizes an unlabeled dataset to replace the gradient updates in typical FL. This architecture is flexible for it is compatible with different deep learning models in different clients.
- We craft a 'phishing' method based on EFL to detect Byzantine attacks. The 'phishing' method employs the labeled dataset to detect the potential Byzantine clients in the EFL system, which preserves the security of EFL.
- We present a Bayesian inference-based reputation mechanism to promote FLPhish's aggregation. The reputation mechanism gives each client a reputation to measure its confidence value and identifies the clients with low reputation values as Byzantine clients, which helps FLPhish identify the Byzantine clients with higher accuracy.

II. Related Work

In this section, we discuss about the related research work about the proposed Byzantine defense methods in FL and the proposed reputation mechanism in cybersecurity research.

A. Byzantine Defense Methods in Federated Learning

Byzantine-robust schemes are very important for FL to enhance its security as Byzantine attacks can cause great damages to the FL system. Recent years have witnessed the increasing interest in the research of Byzantine-robust schemes in the context of FL. Most of the current Byzantine-robust FL methods tend to make a more robust aggregation rule which aims to tolerate the presence of Byzantine clients. For example, in 2017, Chen *et al.* developed an approach called Krum, which selects one client's update as a global model based on a square-distance score in each iteration [26]. In the same year, Blanchard *et al.* proposed two Byzantine-tolerant FL aggregation rules called Trimmed mean and Median [29]. Trimmed Mean considers each parameter of the model update individually. Trimmed Mean sorts the parameter of the model updates collected, and cuts off the largest ones and the smallest ones. Median sorts the values of each parameter of all local model updates as well. Besides it considers the median value of each parameter as the value of the parameter in the global model update. In 2018, Chen *et al.* designed an approach called DRACO to evaluate redundant gradients that are used by the parameter server to eliminate the effects of adversarial updates. In 2019, Xie *et al.* proposed Zeno, which uses a ranking-based preference mechanism [28]. The server computes a score for each client by using the stochastic zero-order oracle. Then Zeno presents a ranking list of clients based on the estimated descent of the loss function and the magnitudes. At last, Zeno computes the global model update by aggregating the clients with the highest scores. In 2020, SLSGD developed by Xie *et al.* also uses trimmed mean as the robust aggregation rules for Byzantine-robust FL [36]. In the same year, Cao *et al.* proposed a Byzantine-tolerant scheme: FLTrust to introduce the use of trust [30]. In each iteration, the server calculates a trust score for each client at first and lowers the trust score if the client's local model update's direction deviates more from the direction of the global model update. The client with a trust score lower than the threshold is considered a malicious client. In 2021, a privacy-enhanced FL (PEFL) framework is presented by Liu *et al.* [37]. PEFL takes advantage of homomorphic encryption to protect the privacy of the clients. Furthermore, a channel using the effective gradient data extraction is provided for the server to punish poisoners.

B. Reputation Mechanism in Cybersecurity

The reputation mechanism is valued as a way to measure an entity's performance in a long term, such as in an online social network [38], and in a smart grid system, [39], [40]. In 2012, Das *et al.* first presented a dynamic trust computation model called SecuredTrust. This framework is used to distribute the workload and deal with the altering behavior of malicious clients [41]. To calculate and manage trust and reputation of CSP and SNP services, Zhu *et al.* proposed an authenticated trust and reputation calculation and management system in wireless sensor networks and cloud computing in 2015 [42]. Lei *et al.* presented a reputation-based Byzantine Fault Tolerance rule in 2018, which uses a reputation model to assess the performance of each node in the blockchain system [43]. If the system detects any malicious behavior, the nodes' discourse rights and reputation in the voting process are reduced. They also provided a primary change method based on reputation. The node with a higher reputation would have more chances to generate fresh valid blocks, lowering the system's security risk. In 2020, Chouikhi *et al.* developed a reputation computing and credibility model to improve network efficiency [44]. They measured a vehicle's behavior toward other vehicles and network services using its reputation score or worth. And a vehicle's credibility is utilized to determine the correctness of a reputation score it offers. In the same year, Wen *et al.* designed a Dirichlet reputation-based approach, and used the reputation score to choose a trustworthy Helper as a friendly jammer in a wireless cooperative system (WCS) [45]. Furthermore, they devised a multi-threshold fake noise detection approach. They gave ratings on a scale of one to ten. The graded ratings were directly represented and reflected in the generated reputation scores in the Dirichlet reputation-based method. In 2021, Liang *et al.* introduced an intrusion detection system with a Markov-based reputation algorithm [46]. The RS-HgMTD model of the Hidden Generalized Mixture Transition Distribution (HgMTD) is designed to help each vehicle in the VANET assess the creditworthiness of its neighbors.

III. Models and Design Goals

In this section, we discuss the system model, show the threat model and identify our design goals.

A. System Model

We first show the system architecture of a typical FL with two entities, FL server, and a group of FL clients.

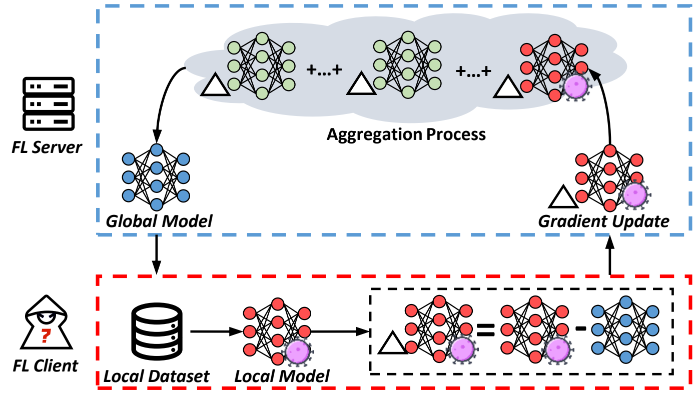

Fig. 1. System Model&Threat Model.

1) FL Server: In each iteration, the FL server s provides a global model to each client. The FL server aggregates all of the gradient updates to a global update based on FedAvg after receiving them from all of the clients. The FL server updates the global model after the aggregation process.

2) FL Client: The local dataset d_i gathered by each FL client c_i (c_i denotes the ith client in FL) is preserved by each FL client c_i. To update the model obtained from the FL server, the FL client c_i uses its local dataset d_i. The model gradient updates are then sent back to the FL server. Meanwhile, it repeats the preceding steps throughout the FL process until the FL server s stops transmitting new models.

B. Threat Model

Byzantine attacks are a problem in the current system. We separate Byzantine attacks into two types according to the attack consequences:

1) Denial-of-Service Byzantine Attack: In denial-of-service attack, the Byzantine attackers intend to disturb the global model thus making it produce wrong predictions of the normal dataset [16]–[20]. The label flipping attack in the current system model can be used by a malicious Byzantine client b_i to launch Byzantine attacks against the global model. Label flipping attacks require b_i to change the labels of training data while ensuring that the data's features remain unchanged [47]. The local model of the Byzantine client b_i is trained with incorrect labels, resulting in a 'poisoned' model with low accuracy. Then Byzantine client b_i dispatches the false model gradient updates to the central server. Therefore the false model gradient updates cause the central server to learn the falsely distilled knowledge from clients. The server s's aggregation process is performed on FedAvg which takes each client c_i's dataset d_i's size as the aggregation weight for c_i. This means that a client c_i with a larger size of d_i gets a larger aggregation weight. Meanwhile FedAvg takes the size of d_i declared by c_i as d_i's real size which means b_i can declare a fake size value larger than d_i's real size value to enlarge the impact of attack. If the weight of the malicious clients reaches a threshold, the central server is misguided to produce false predictions.

In section V, we first divided denial-of-service attacks into 2 types of attacks: The first one is untargeted attack which makes the Byzantine attackers change the label of data samples from a type into another type (for instance, all the samples with label '5' are changed to label '0'). The another one is random attack which makes the Byzantine attackers randomly change the label of data samples. Then we evaluated our framework against the 2 types of denial-of-service attacks.

2) Backdoor Byzantine Attack: In backdoor attacks, called backdoor attacks, the disturbed global model will make wrong predictions of the data samples which have 'backdoor' in them [21]–[25]. Backdoor attackers in FL need to embed some 'triggers' (noted as σ) in their local dataset. Then they relabel the data samples with σ as the target label ι. Each backdoor attacker adopts the preprocessed local dataset with σ to update the global model it received from the FL server. Then it transfers the poisoned model updates which contain the information that the data sample with σ is predicted as the target label ι to the FL server. After receiving the poisoned model updates, the FL server updates the global model using the poisoned model updates of the backdoor attackers. After the update, the FL server's model misclassifies the data with the σ to the target label ι of backdoor attackers. Take a backdoor attack process towards the construction of an FL on CIFAR-10 as an example. The backdoor attacker adds a grey square as σ in each data sample. Each data sample with a σ is labeled as 'cat'. Then the backdoor attacker uses these data samples to update the global model from the central server and transfers the model gradient updates containing the σ information to the central server. After that, the central server updates the global model via the model gradient updates. Thus, the global model learns the σ information from the backdoor attacker. It misclassifies the data sample with the σ as 'cat' as well.

In section IV.D, we did theoretical analysis about how our FLPhish scheme can defend against backdoor Byzantine attacks.

C. Design Goals

The proposed FLPhish scheme's main goal is to provide a reliable method for accurately resisting opportunistic Byzantine attacks in our EFL system. The following are our design goals:

1) The typical FL design has several flaws, including incompatibility with various deep learning models in different clients and significant communication costs. As a result, we created EFL, a novel FL architecture inspired by the idea of ensemble learning. It lowers the cost of network transfers and expands our ability to defend against Byzantine attacks in FL.

2) The proposed EFL architecture currently needs a robust Byzantine attack protection mechanism. We urgently seek an efficient solution to deal with malicious Byzantine clients in FL because they cannot be trusted. In our proposed EFL system, we describe a phishing-based approach to guard against Byzantine attacks.

3) As the performance of each client remains not stable in each iteration, it is important for our scheme to accurately measure each client's level of confidence in the long term. Therefore, we further propose an effective Bayesian inference-based reputation scheme based on our phishing-based model to spot Byzantine attacks compromised by malicious users.

IV. Conclusion

The conclusion goes here.

Acknowledgments

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

Appendix
Proof of the Zonklar Equations

Use `\appendix` if you have a single appendix: Do not use `\section` anymore after `\appendix`, only `\section*`. If you have multiple appendixes use `\appendices` then use `\section` to start each appendix. You must declare a `\section` before using any `\subsection` or using `\label` (`\appendices` by itself starts a section numbered zero.)

References Section

You can use a bibliography generated by BibTeX as a .bbl file. BibTeX documentation can be easily obtained at: http://mirror.ctan.org/biblio/bibtex/contrib/doc/ The IEEEtran BibTeX style support page is: http://www.michaelshell.org/tex/ieeetran/bibtex/

Simple References

You can manually copy in the resultant .bbl file and set second argument of `\begin` to the number of references (used to reserve space for the reference number labels box).

References

[1] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *2019 the 2nd Systems and Machine Learning Conference (SysML)*, Stanford, CA, USA, Mar. 31-Apr. 2 2019.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Tech.*, vol. 10, no. 2, Jan. 2019.

[3] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021.

[4] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Sep. 2021.

[5] Q. Kong, F. Yin, R. Lu, B. Li, X. Wang, S. Cui, and P. Zhang, "Privacy-preserving aggregation for federated learning-based navigation in vehicular fog," *IEEE Trans. Ind. Informat.*, Apr. 2021.

[6] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, 2020.

[7] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, Apr. 2020.

[8] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, Jul. 2020.

[9] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, “Beyond inferring class representatives: User-level privacy leakage from federated learning,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 2512–2520.

[10] C. Miao, Q. Li, L. Su, M. Huai, W. Jiang, and J. Gao, “Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing,” in *Proceedings of the 2018 World Wide Web Conference*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee (WWW), 2018, p. 13–22.

[11] R. Laishram and V. V. Phoha, “Curie: A method for protecting svm classifier from poisoning attack,” *arXiv preprint arXiv:1606.01584*, 2016.

[12] C.-Y. Wei, P.-N. Chen, Y. S. Han, and P. K. Varshney, “Local threshold design for target localization using error correcting codes in wireless sensor networks in the presence of byzantine attacks,” *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1571–1584, Feb. 2017.

[13] X. Liu, T. J. Lim, and J. Huang, “Optimal byzantine attacker identification based on game theory in network coding enabled wireless ad hoc networks,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2570–2583, Feb. 2020.

[14] R. Cao, T. F. Wong, T. Lv, H. Gao, and S. Yang, “Detecting byzantine attacks without clean reference,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2717–2731, Jul. 2016.

[15] Y. Amir, C. Danilov, D. Dolev, J. Kirsch, J. Lane, C. Nita-Rotaru, J. Olsen, and D. Zage, “Steward: Scaling byzantine fault-tolerant replication to wide area networks,” *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 1, pp. 80–93, Sep. 2010.

[16] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Model poisoning attacks in federated learning,” in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*, Palais des Congrès de Montréal, Montréal CANADA, Dec. 2-8 2018.

[17] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to byzantine-robust federated learning,” in *29th USENIX Security Symposium (USENIX Security)*, Boston, MA, USA, Aug. 12-14 2020.

[18] B. Biggio, B. Nelson, and P. Laskov, “Poisoning attacks against support vector machines,” in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, New York, NY, USA, Jun. 26-Jul. 1 2012.

[19] C. Yang, Q. Wu, H. Li, and Y. Chen, “Generative poisoning attack method against neural networks,” *arXiv preprint arXiv:1703.01340*, 2017.

[20] M. Sun, J. Tang, H. Li, B. Li, C. Xiao, Y. Chen, and D. Song, “Data poisoning attack against unsupervised node embedding methods,” *arXiv preprint arXiv:1810.12881*, 2018.

[21] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, “Exploiting machine learning to subvert your spam filter,” in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Francisco, California, USA, Apr. 14-15 2008.

[22] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual, Aug. 26-28 2020.

[23] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?” *arXiv preprint arXiv:1911.07963*, Nov 2019.

[24] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, “Attack of the tails: Yes, you really can backdoor federated learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*. Virtual: Curran Associates, Inc., Dec. 6-12 2020.

[25] C. Xie, K. Huang, P.-Y. Chen, and B. Li, “Dba: Distributed backdoor attacks against federated learning,” in *7th International Conference on Learning Representations (ICLR)*, New Orleans, USA, May 6-9 2019.

[26] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, Dec. 4-9 2017.

[27] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, “DRACO: Byzantine-resilient distributed training via redundant gradients,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Stockholmsmässan, Stockholm SWEDEN, Jul. 10-15 2018.

[28] C. Xie, S. Koyejo, and I. Gupta, “Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance,” in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, Jun. 9-15 2019.

[29] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Stockholmsmässan, Stockholm SWEDEN, Jul. 10-15 2018.

[30] X. Cao, M. Fang, J. Liu, and N. Gong, “FLTrust: Byzantine-robust federated learning via trust bootstrapping,” in *2021 Network and Distributed System Security Symposium (NDSS)*, Feb. 21-25 2021.

[31] J. So, B. Güler, and A. S. Avestimehr, “Byzantine-resilient secure federated learning,” *IEEE J. Sel. Areas Commun*, pp. 1–1, Jul. 2020.

[32] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, “Robust federated learning in a heterogeneous environment,” *arXiv preprint arXiv:1906.06629*, Jun. 2019.

[33] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, “On the byzantine robustness of clustered federated learning,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Catalonia, Spain, May. 4-8 2020, pp. 8861–8865.

[34] A. Portnoy and D. Hendler, “Towards realistic byzantine-robust federated learning,” *arXiv preprint arXiv:2004.04986*, Apr. 2020.

[35] L. Muñoz-González, K. T. Co, and E. C. Lupu, “Byzantine-robust federated machine learning through adaptive model averaging,” *arXiv preprint arXiv:1909.05125*, Sep. 2019.

[36] C. Xie, O. Koyejo, and I. Gupta, “SLSGD: Secure and efficient distributed on-device machine learning,” in *Machine Learning and Knowledge Discovery in Databases (PKDD)*, Ghent, Belgium, Sep. 14-18 2020.

[37] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, “Privacy-enhanced federated learning against poisoning adversaries,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4574–4588, 2021.

[38] G. Liu, Q. Yang, H. Wang, and A. X. Liu, “Trust assessment in online social networks,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 994–1007, May 2021.

[39] B. Li, R. Lu, W. Wang, and K.-K. R. Choo, “DDOA: A dirichlet-based detection scheme for opportunistic attacks in smart grid cyber-physical system,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2415–2425, Jun. 2016.

[40] B. Li, R. Lu, and G. Xiao, *Detection of False Data Injection Attacks in Smart Grid Cyber-Physical Systems*. Springer, 2020.

[41] A. Das and M. M. Islam, “SecuredTrust: A dynamic trust computation model for secured communication in multiagent systems,” *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 2, pp. 261–274, 2012.

[42] C. Zhu, H. Nicanfar, V. C. M. Leung, and L. T. Yang, “An authenticated trust and reputation calculation and management system for cloud and sensor networks integration,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 118–131, 2015.

[43] K. Lei, Q. Zhang, L. Xu, and Z. Qi, “Reputation-based byzantine fault-tolerance for consortium blockchain,” in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, Sentosa, Singapore, Dec. 11-13 2018, pp. 604–611.

[44] S. Chouikhi, L. Khoukhi, S. Ayed, and M. Lemercier, “An efficient reputation management model based on game theory for vehicular networks,” in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, Sydney, Australia, Nov. 16-19 2020, pp. 413–416.

[45] Y. Wen, Y. Huo, T. Jing, and Q. Gao, “A reputation framework with multiple-threshold energy detection in wireless cooperative systems,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Virtual, Jun. 7-11 2020, pp. 1–6.

[46] J. Liang and M. Ma, “ECF-MRS: An efficient and collaborative framework with markov-based reputation scheme for idss in vehicular networks,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 278–290, 2021.

[47] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, “Understanding distributed poisoning attack in federated learning,” in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, Tianjin, China, Dec. 4-6 2019.

REFERENCES

[1] *Mathematics Into Type*. American Mathematical Society. [Online]. Available: https://www.ams.org/arc/styleguide/mit-2.pdf

[2] T. W. Chaundy, P. R. Barrett and C. Batey, *The Printing of Mathematics*. London, U.K., Oxford Univ. Press, 1954.

[3] F. Mittelbach and M. Goossens, *The LaTeXCompanion*, 2nd ed. Boston, MA, USA: Pearson, 2004.

[4] G. Grätzer, *More Math Into LaTeX*, New York, NY, USA: Springer, 2007.

[5] M. Letourneau and J. W. Sharp, *AMS-StyleGuide-online.pdf,* American Mathematical Society, Providence, RI, USA, [Online]. Available: http://www.ams.org/arc/styleguide/index.html

[6] H. Sira-Ramirez, “On the sliding mode control of nonlinear systems,” *Syst. Control Lett.*, vol. 19, pp. 303–312, 1992.

[7] A. Levant, "Exact differentiation of signals with unbounded higher derivatives," in *Proc. 45th IEEE Conf. Decis. Control*, San Diego, CA, USA, 2006, pp. 5585–5590. DOI: 10.1109/CDC.2006.377165.
[8] M. Fliess, C. Join, and H. Sira-Ramirez, "Non-linear estimation is easy," *Int. J. Model., Ident. Control*, vol. 4, no. 1, pp. 12–27, 2008.
[9] R. Ortega, A. Astolfi, G. Bastin, and H. Rodriguez, "Stabilization of food-chain systems using a port-controlled Hamiltonian description," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2000, pp. 2245–2249.

BIOGRAPHY SECTION

If you have an EPS/PDF photo (graphicx package needed), extra braces are needed around the contents of the optional argument to biography to prevent the LaTeX parser from getting confused when it sees the complicated `\includegraphics` command within an optional argument. (You can create your own custom macro containing the `\includegraphics` command to make things simpler here.)

If you include a photo:

Michael Shell Use `\begin{IEEEbiography}` and then for the 1st argument use `\includegraphics` to declare and link the author photo. Use the author name as the 3rd argument followed by the biography text.

If you will not include a photo:

John Doe Use `\begin{IEEEbiographynophoto}` and the author name as the argument followed by the biography text.