Enhancing Byzantine-Resistant Aggregations with Client Embedding

Anonymous ACL submission

Abstract

Byzantine-resistant aggregations detect poisonous clients and discard them to ensure that the global model is not poisoned or attacked by malicious clients. However, these aggregations are mainly conducted on the parameter space, and the parameter distances cannot reflect the data distribution divergences between clients. Therefore, existing Byzantine-resistant aggregations cannot defend against backdoor injection by malicious attackers in federated natural language tasks. In this paper, we propose the client embedding for malicious client detection to enhance Byzantine-resistant aggregations. The distances between client embeddings are required to reflect the data distribution divergences of the corresponding clients. Experimental results validate the effectiveness of the proposed client embeddings.

1 Introduction

Byzantine attacks are a kind of threat to federated learning security, and therefore a line of Byzantine-resistant aggregation algorithms (Blanchard et al., 2017; Mhamdi et al., 2018; Zhang et al., 2022) are designed to defend against Byzantine attacks.

The core of Byzantine-resistant aggregations is to detect poisonous clients and discard them to ensure that the global model is not poisoned or attacked by malicious clients. These aggregations are mainly conducted on the parameter space, namely, these aggregations determines suspected poisonous clients based on the distances between client parameters. Existing Byzantine-resistant aggregations can defend against adversaries caused by software bugs, hardware bugs, network asynchrony, or datasets biases (Blanchard et al., 2017; Mhamdi et al., 2018), while Zhang et al. (2022) point out that they cannot defend against backdoor injection by malicious attackers in federated natural language tasks. Zhang et al. (2022) point out the limitation of the malicious client detection in the parameter space.

In this paper, we argue that for better detection of malicious clients, we should not apply Byzantine-resistant aggregations in the parameter space directly, because the parameter distances of clients cannot directly reflect the distribution divergences between clients. Therefore, we propose the client embedding for malicious client detection. We assume that the distances between client embeddings can reflect the distribution divergences of the corresponding client data distributions. According to this assumption, we propose to sovle the low-dimension embeddings according to Proposition 1.

As demonstrated in Fig. 1, enhanced with the proposed client embedding, Byzantine-resistant aggregations detect malicious clients according to embedding distances instead of parameter distances. Aggregation algorithms can better detect malicious clients enhanced with client embedding, since malicious and clean clients are easier to distinguish in the embedding space than the parameter space.

To validate the effectiveness of the proposed client embedding, we conduct the defense and detection experiments algorithms on typical NLP Byzantine attacks, including adversaries (Blanchard et al., 2017; Mhamdi et al., 2018) and backdoors (Chen et al., 2020b; Dai et al., 2019). Experimental results show that defense performance of existing Byzantine-resistant aggregations can be improved enhanced with client embedding. Furthermore, the results of malicious detection also show that the client embedding can prove the detection performance of Byzantine-resistant aggregations, which indicates that the improvement of defense performance does come from better detection performance brought by client embedding. The detection ability of client embedding comes from the ability of client embeddings to model the dataset distributions of clients.

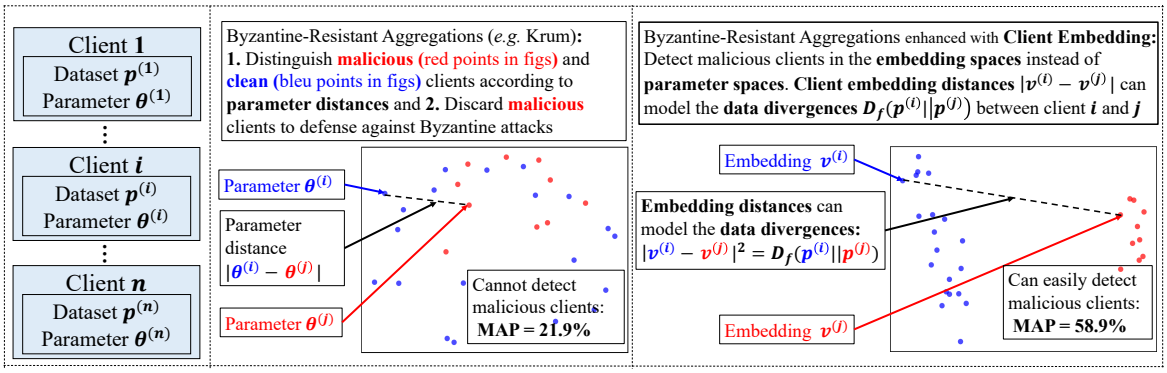


Figure 1: Illustration of the client embedding. Byzantine-resistant aggregations enhanced with client embedding detect malicious clients in embedding spaces instead of parameter spaces (adopted by traditional Byzantine-resistant aggregations, *e.g.*, Krum) because embedding distances model the data distribution divergences between clients.

2 Background and Related Work

In this section, we first introduce the concept of federated learning and Byzantine-resistant aggregations. Then we introduce some NLP Byzantine attacks adopted in experiments.

2.1 Federated Learning Paradigm

Suppose the federated learning process includes T learning rounds. In every round, suppose there are n **clients** training the local model with parameters $\boldsymbol{\theta}_t^{(i)}$ on client $i \in [1, n]$, then the **server** needs to update the global $\boldsymbol{\theta}_t^{\text{Server}}$ according to $\{\boldsymbol{\theta}_t^{(i)}\}_{t=1}^{T}$.

Byzantine attack means that malicious clients among n clients send poisonous clients for malicious purposes. To defend against byzantine attacks, Byzantine-resistant aggregations (Blanchard et al., 2017; Mhamdi et al., 2018; Zhang et al., 2022) choose a secure client index set S and only aggregate secure clients to update the global model:

$$\boldsymbol{\theta}_t^{\text{Server}} = \frac{1}{|S|} \sum_{i \in S} \boldsymbol{\theta}_t^{(i)}. \tag{1}$$

2.2 Byzantine-Resistant Aggregations

Traditional Byzantine-resistant aggregations (Blanchard et al., 2017; Mhamdi et al., 2018; Zhang et al., 2022) detect malicious clients and choose the set S according to parameters distances or other metrics in the parameter space. In this section, we take the classic multi-Krum (Blanchard et al., 2017) algorithm as an instance.

Suppose $d_{ij} = \|\boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^{(j)}\|$ denotes the parameter distance of client i and j, N_i denotes the neighbors of client i which includes $\lceil \frac{n+1}{2} \rceil$ clients with the smallest distances d_{ij} (including client i itself). Suppose i^* denotes the client with the smallest distance sum of its neighbors $\mathcal{N}_i$:

$$i^* = \arg\min_i \sum_{j \in \mathcal{N}_i} d_{ij}. \tag{2}$$

The multi-Krum algorithm trusts the neighbors of i^*, namely chooses $S = \mathcal{N}_i$. Other Byzantine-resistant aggregations adopt different algorithms to determine the set S, but all according to parameters distances d_{ij} in parameter space.

We choose four Byzantine-resistant aggregations as baselines: they are **Krum** (Blanchard et al., 2017), **Multi-Krum** (Blanchard et al., 2017), **Bulyan** (Mhamdi et al., 2018), and **Dim-Krum** (Zhang et al., 2022) algorithms. In addition to Byzantine-resistant aggregations, there are also a line of other robust aggregations without explicitly detecting malicious clients and choosing the set S. In our experiments, we also adopt the statistical median (**Median**) (Chen et al., 2020a; Yin et al., 2018), the geometric median (**RFA**) (Pillutla et al., 2019), certifiably robust federated learning (**CRFL**) (Xie et al., 2021), **FoolsGold** (Fung et al., 2020), and Residual-based (**Residual**) (Fu et al., 2019) algorithms as baselines.

2.3 NLP Byzantine Attacks

Blanchard et al. (2017) first consider adversaries as Byzantine attacks: the attacker can add a **Guassian** noise (Blanchard et al., 2017) or fixed **bias** (Blanchard et al., 2017) on the parameters. In our experiments, we adopt both **Guassian** and **bias** attacks as adversaries. Beside adversaries, attacker can also poison the local dataset (Muñoz-González et al., 2017; Chen et al., 2017) to inject backdoors (Gu et al., 2019) to control the model's behaviors for malicious purposes. We two typical federated NLP

backdoor attacks: **BadWord** (Chen et al., 2020b) and **BadSent** (Dai et al., 2019; Chen et al., 2020b).

3 Federated Client Embedding

Suppose $\mathbf{v}^{(i)}$ denotes the federated client embedding of client i, and $\mathbf{E} = [\mathbf{v}^{(1)}; \mathbf{v}^{(2)}; \cdots; \mathbf{v}^{(n)}]$ is the embedding matrix. We have only one assumption for client embedding:

Assumption 1. *The embedding distances can model the data divergences:*

$$\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^2 = D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)}), \quad (3)$$

here $D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)})$ denotes the f-divergences of data distributions $\mathbf{p}^{(i)}$ and $\mathbf{p}^{(j)}$ on client i and j, and we adopt the f-divergence indicator (Zhang et al., 2024) to estimate it.

In Assumption 1, if we assume $\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^p = D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)})$, to ensure the linearity of embeddings, namely the corresponding embedding of mixed $\mathbf{p}^* = \alpha\mathbf{p}^{(i)} + (1-\alpha)\mathbf{p}^{(j)}$ is approximately $\alpha\mathbf{v}^{(i)} + (1-\alpha)\mathbf{v}^{(j)}$, we can only choose $p = 2$ since $D_f(\mathbf{p}^*||\mathbf{p}^{(j)}) \approx \alpha^2 D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)})$.

Suppose the matrix $\mathbf{F}$ denotes the divergence matrix, namely $\mathbf{F}_{ij} \approx D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)})$ is the f-divergence indicator (Zhang et al., 2024). In Proposition 1, we prove we can find $(n-1)$-dimension embeddings satisfying Assumption 1.

Proposition 1. *There exists an $(n-1)$-dimension solution for $\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^2 = \mathbf{F}_{ij}$, which can be solved with the following Cholesky decomposition (Dereniowski and Kubale, 2004):*

$$\mathbf{E}^T\mathbf{E} = \hat{\mathbf{F}} := \frac{\mathbf{FJ} + \mathbf{JF} - \mathbf{F} - \mathbf{JFJ}}{2}, \quad (4)$$

where $\mathbf{1}$ is an n-dimension vector full of ones, $\mathbf{J} = \frac{\mathbf{1}\mathbf{1}^T}{n}$, and rank$(\hat{\mathbf{F}}) \leq n-1$.

Proposition 1 guides us how to solve a low-dimensional client embeddings $\mathbf{E}$ to enhance Byzantine-resistant aggregations. More theoretical details are deferred to Appendix.A. For example, multi-Krum algorithm enhanced with client embedding chooses S according to $d_{ij} = \|\mathbf{v}_t^{(i)} - \mathbf{v}_t^{(j)}\|$ in embedding spaces instead of $d_{ij} = \|\boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^{(j)}\|$ in parameter spaces. The direct advantage of our proposed method is that embedding distances can directly reflect the data distribution divergence between clients according to Assumption 1.

4 Experiments

Our experiments include **defense** and **detection** experiments. The target of **defense** is to train a model with low Attack Success Rate (ASR) or high Accuracy (ACC) under Byzantine attacks, and the target of **detection** is to detect malicious clients precisely in one training round. We introduce experiment setup and report experimental results in this section. Due to space limit, supplementary results are reported in Appendix B.

4.1 Experiment Setups

We train an LSTM on two typical text classification tasks, *i.e.*, *SST-2* (Stanford Sentiment Treebank) (Socher et al., 2013) and *Amazon* (Amazon reviews) (Blitzer et al., 2007).

As introduced in Sec. 2.2, we adopt four **Byzantine-resistant baselines**: *Krum* (Blanchard et al., 2017), *Multi-Krum* (Blanchard et al., 2017), *Bulyan* (Mhamdi et al., 2018), *Dim-Krum* (Zhang et al., 2022); and other **aggregations**: *FedAvg* (McMahan et al., 2017), *Median* (Chen et al., 2020a; Yin et al., 2018), *RFA* (Pillutla et al., 2019), *CRFL* (Xie et al., 2021), *FoolsGold* (Fung et al., 2020), and *Residual* (Fu et al., 2019). In Dim-Krum, we choose the ratio as $\rho = 10^{-3}$ and the adaptive noise scale $\lambda = 2$.

As introduced in Sec. 2.3, we adopt two **adversaries** (*i.e.*, *Guassian* (Blanchard et al., 2017) and *bias* (Mhamdi et al., 2018)) and two **backdoor** attacks (*i.e. BadWord* (Chen et al., 2020b) and *BadSent* (Chen et al., 2020b; Dai et al., 2019)) as the Byzantine attacks. In BadWord, the trigger words are "cf", "mn", "bb", "tq" and "mb". In BadSent, the trigger sentence is "I watched this 3d movie". The target label is label 0.

In adversary attacks, the defense target is to avoid the Accuracy (**ACC with adversaries**) decrease with adversaries. In backdoor attacks, the defense target is to gain a low Backdoored Attack Success Rate (*Backdoored ASR*) with less Backdoored Accuracy (*Backdoored ACC*) decreases. The detection target of all six Byzantine attacks is to detect malicious precisely, and we adopt some detection metrics to evaluate different aggregations: *FAR* (false acceptance rate), *FRR* (false rejection rate), *Precision*, *Recall*, *F1-score*, *ACC* (accuracy), *MR* (mean rank), and *MAP* (mean average precision).

We adopt Adam (Kingma and Ba, 2015) optimizer with a learning rate of 10^{-3} and a batch size of 32. The attacker number and total client number

Aggregations	Metric	Clean Training	FedAvg	Median	FoolsGold	RFA	CRFL	Residual
Baseline	ACC with **adversaries**	86.16	50.32	86.14	50.05	86.34	74.86	86.24
	Backdoored ACC	86.16	86.03	85.71	86.15	86.23	75.00	86.14
	Backdoored ASR	16.22	98.86	97.95	98.35	98.66	95.92	99.00
Aggregations	**Metric**	Clean Training	FedAvg	Krum	Multi-Krum	Bulyan	Dim-Krum	**Byzant. Ave.**
Byzantine-resistant	ACC with **adversaries**	86.16	50.32	79.56	86.27	85.79	85.27	84.22
	Backdoored ACC	86.16	86.03	77.82	86.07	85.80	85.12	83.70
	Backdoored ASR	16.22	98.86	99.95	98.78	98.84	46.58	86.04
Aggregations	**Metric**	Clean Training	FedAvg	Krum	Multi-Krum	Bulyan	Dim-Krum	**Byzant. Ave.**
Byzantine-resistant, enhanced with **Client Embedding**	ACC with **adversaries**	86.16	50.32	**83.77**	86.21	85.46	84.96	**85.10**
	Backdoored ACC	86.16	86.03	**83.23**	86.20	86.01	84.82	**85.07**
	Backdoored ASR	16.22	98.86	**18.17**	**26.05**	**64.55**	**36.02**	**36.20**

Table 1: Average defense results of baseline aggregations and Byzantine-resistant aggregations enhanced with client embedding. Higher ACCs with adversaries and Backdoored ACCs are better, while lower Backdoored ASRs are better. Results of Byzantine-resistant aggregations enhanced with client embedding are in bold if they have statistically significant improvements.

Aggregations	FAR	FRR	Precision	Recall	F1-score	ACC	MR	MAP
Krum/Multi-Krum/Bulyan	15.1	94.1	8.3	5.8	6.8	70.0	23.5	21.9
enhanced with **Client Embedding**	**10.0**	**72.5**	**38.9**	**27.4**	**32.1**	**78.2**	**12.3**	**58.9**
Dim-Krum	10.1	72.6	38.9	27.5	32.2	78.2	16.5	57.4
enhanced with **Client Embedding**	**8.4**	**65.7**	**48.6**	**34.3**	**40.2**	**80.7**	**11.5**	**70.9**

Table 2: Average detection results of Byzantine-resistant aggregations and those enhanced with client embedding. Higher metrics except FARs, FRRs, and MRs are better. Results of Byzantine-resistant aggregations enhanced with client embedding are in bold if they have statistically significant improvements.

is 1 and 10 in the defense experiments. In detection experiments, the client number is 30, and we enumerate the attacker number from 1 to 10. We report the average metrics and the metrics with statistically significant improvement are in bold.

4.2 Defense Results and Analysis

It is maybe because adversaries are conducted on the parameter spaces directly (Blanchard et al., 2017; Mhamdi et al., 2018), and thus easy to detect for traditional Byzantine-resistant aggregations. However, the backdoor attacks are more stealthy, since they poison the clients' dataset and hard to detect according to the parameter space. Enhanced with our proposed client embedding, Byzantine-resistant aggregations can model the distributional variations between clients, and thus can detect malicious clients with poisonous dataset.

4.3 Detection Results and Analysis

Existing Byzantine-resistant aggregations calculate parameter distances $d_{ij} = \|\boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^{(j)}\|$ directly (Krum (Blanchard et al., 2017), Multi-Krum (Blanchard et al., 2017), and Bulyan (Mhamdi et al., 2018)), or calculate parameter distances on some suspicious dimensions like Dim-Krum (Zhang et al., 2022). We try both in our experiments and label the discarded clients in Byzantine-resistant aggregations as the malicious clients.

As shown in Table 2, enhanced with client embedding, existing Byzantine-resistant aggregations, the detection performance of both parameter distance calculating mechanisms improve. It shows that the improvement of defense performance does come from better detection performance brought by our proposed client embedding that can model the clients' data divergences more accurately.

5 Conclusion

In this paper, we propose the client embedding to enhance Byzantine-resistant aggregations. The proposed client embedding can model the dataset distributions of corresponding clients, namely the embedding distances can model the data divergences of clients. Experimental results show that the defense performance of Byzantine-resistant aggregations can be improved enhanced with client embedding. Further analyzes show that the improvements of defense performance come from better detection performance of client embedding, which demonstrates that the proposed client embedding can model the data divergences of clients.

Ethical Considerations and Limitations

In this paper, the proposed client embeddings are proposed to enhance the Byzantine-Resistant aggregations for more secure federated language learning. At the same time, since the proposed client embeddings have lower dimensions, the risk of privacy leakage is much lower.

This paper focuses on explaining the theoretical motivation and preliminary experimental validation. Although experimental results show that the proposed client embeddings can enhance existing Byzantine-Resistant aggregations, we only validate the proposed client embeddings on several classic NLP models and tasks. Further detailed experiments on the latest NLP model architectures, especially large language models, need to be conducted in the future work.

References

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 119–129.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Xiangyi Chen, Tiancong Chen, Haoran Sun, Zhiwei Steven Wu, and Mingyi Hong. 2020a. Distributed training with heterogeneous data: Bridging median- and mean-based algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2020b. Badnl: Backdoor attacks against nlp models. *arXiv preprint arXiv:2006.01043*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Dariusz Dereniowski and Marek Kubale. 2004. Cholesky factorization of matrices in parallel and ranking of graphs. In *Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, Czestochowa, Poland, September 7-10, 2003. Revised Papers 5*, pages 985–992. Springer.

Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. 2019. Attack-resistant federated learning with residual-based reweighting. *CoRR*, abs/1912.11464.

Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2020. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2020, San Sebastian, Spain, October 14-15, 2020*, pages 301–316. USENIX Association.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3518–3527. PMLR.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 27–38. ACM.

Venkata Krishna Pillutla, Sham M. Kakade, and Zaïd Harchaoui. 2019. Robust aggregation for federated learning. *CoRR*, abs/1912.13445.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,*

EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1631–1642. ACL.

Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. 2021. CRFL: certifiably robust federated learning against backdoor attacks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11372–11382. PMLR.

Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR.

Zhiyuan Zhang, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2024. Fed-fa: Theoretically modeling client data divergence for federated language backdoor defense. *Advances in Neural Information Processing Systems*, 36.

Zhiyuan Zhang, Qi Su, and Xu Sun. 2022. Dim-krum: Backdoor-resistant federated learning for NLP with dimension-wise krum-based aggregation. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 339–354. Association for Computational Linguistics.

A Theoretical Details

In this section, we introduce theoretical details.

A.1 The Estimation of Data Divergences

In Assumption 1, we assume:

$$\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^2 = D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)}). \tag{5}$$

Following Zhang et al. (2024), we use the f-divergence indicator $\mathbf{F}_{ij}$ to estimate the f-divergence $D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)})$ of $\mathbf{p}^{(i)}$ and $\mathbf{p}^{(j)}$:

$$D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)}) \propto \mathbf{F}_{ij} := \mathbf{\Delta}_{ij}^{\mathrm{T}}\mathbf{H}\mathbf{\Delta}_{ij}, \tag{6}$$

where $\mathbf{\Delta}_{ij} = \boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^{(j)}$, and $\mathbf{H}$ denotes the estimated diagonal Hessian matrix. We follow the same estimating methods as Zhang et al. (2024). Since scaling our embeddings $\mathbf{E}$ into $\alpha\mathbf{E}$ does not affect the detection, we assume $\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^2 = D_f(\mathbf{p}^{(i)}||\mathbf{p}^{(j)}) \approx \mathbf{F}_{ij}$.

A.2 The Proof of Proposition 1

Proof. The objective of Proposition 1 is to find the embedding matrix $\mathbf{E} = [\mathbf{v}^{(1)}; \mathbf{v}^{(2)}; \cdots; \mathbf{v}^{(n)}]$ satisfying $\|\mathbf{v}^{(i)} - \mathbf{v}^{(j)}\|^2 = \mathbf{F}_{ij}, 1 \le i, j \le n$.

Denote $\mathbf{a} \cdot \mathbf{b}$ as the inner product of vectors $\mathbf{a}, \mathbf{b}$, namely $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}}\mathbf{b}$. Denote $\bar{\mathbf{v}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}^{(i)}$, $\overline{\mathbf{v}\cdot\mathbf{v}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}^{(i)}\cdot\mathbf{v}^{(i)}$. Note that $\overline{\mathbf{v}\cdot\mathbf{v}} \neq \bar{\mathbf{v}}\cdot\bar{\mathbf{v}}$. Denote $\mathbf{F}_{*j} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{F}_{ij}$, $\mathbf{F}_{i*} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{F}_{ij}$, and $\mathbf{F}_{**} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{F}_{ij}$. We have:

$$\mathbf{F}_{ij} = \mathbf{v}^{(i)}\cdot\mathbf{v}^{(i)} + \mathbf{v}^{(j)}\cdot\mathbf{v}^{(j)} - 2\mathbf{v}^{(i)}\cdot\mathbf{v}^{(j)}, \tag{7}$$

$$\mathbf{F}_{*j} = \overline{\mathbf{v}\cdot\mathbf{v}} + \mathbf{v}^{(j)}\cdot\mathbf{v}^{(j)} - 2\bar{\mathbf{v}}\cdot\mathbf{v}^{(j)}, \tag{8}$$

$$\mathbf{F}_{i*} = \mathbf{v}^{(i)}\cdot\mathbf{v}^{(i)} + \overline{\mathbf{v}\cdot\mathbf{v}} - 2\mathbf{v}^{(i)}\cdot\bar{\mathbf{v}}, \tag{9}$$

$$\mathbf{F}_{**} = 2\overline{\mathbf{v}\cdot\mathbf{v}} - 2\bar{\mathbf{v}}\cdot\bar{\mathbf{v}}. \tag{10}$$

Therefore,

$$\frac{\mathbf{F}_{i*} + \mathbf{F}_{*j} - \mathbf{F}_{ij} - \mathbf{F}_{**}}{2} \tag{11}$$

$$= (\mathbf{v}^{(i)} - \bar{\mathbf{v}})\cdot(\mathbf{v}^{(j)} - \bar{\mathbf{v}}). \tag{12}$$

Since moving our embeddings $\mathbf{v}^{(i)}$ into $\mathbf{v}^{(i)} + \mathbf{v}^{\text{Delta}}$ does not affect the detection, we also assume that $\bar{\mathbf{v}})$ is the zero vector. Therefore,

$$\mathbf{E}^{\mathrm{T}}\mathbf{E} = \hat{\mathbf{F}} := \frac{\mathbf{FJ} + \mathbf{JF} - \mathbf{F} - \mathbf{JFJ}}{2}, \tag{13}$$

where $\hat{\mathbf{F}}_{ij} = \frac{\mathbf{F}_{i*}+\mathbf{F}_{*j}-\mathbf{F}_{ij}-\mathbf{F}_{**}}{2} \in \mathbb{R}^{n\times n}$.

Then we prove that $\mathrm{rank}(\hat{\mathbf{F}}) \le n-1$, then we can solve the $(n-1)$-dimension embeddings $\mathbf{E}$ with Cholesky decomposition.

$$2\hat{\mathbf{F}} \tag{14}$$

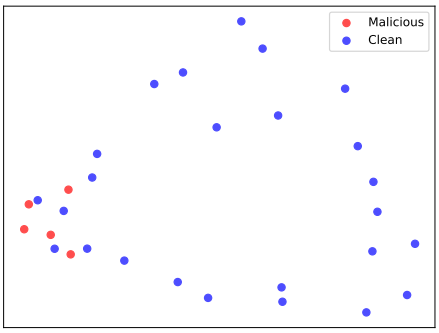
$$= \mathbf{FJ} + \mathbf{JF} - \mathbf{F} - \mathbf{JFJ} \tag{15}$$

$$= (\mathbf{I} - \mathbf{J})(-\mathbf{F})(\mathbf{I} - \mathbf{J}), \tag{16}$$

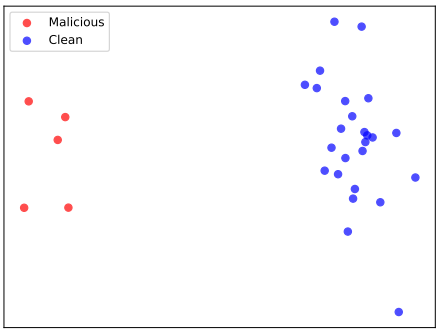
since $(\mathbf{I} - \mathbf{J})\mathbf{1} = \mathbf{1} - \mathbf{1} = 0 \cdot \mathbf{1}$, it is easy to prove that the matrix $\mathbf{I} - \mathbf{J}$ has an eigenvalue of 0, namely $\mathrm{rank}(\mathbf{I}-\mathbf{J}) \le n-1$. Therefore, $\mathrm{rank}(\hat{\mathbf{F}}) \le n-1$. □

B Supplementary Experimental Results

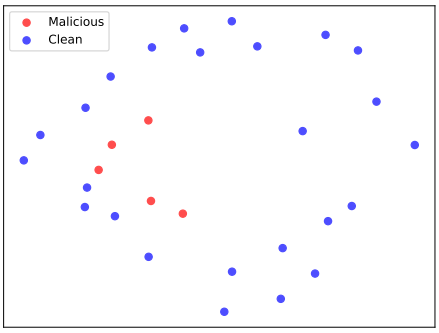
We provide supplementary experimental results in Fig. 2. It can be concluded that, enhanced with client embedding, Byzantine-resistant aggregations, take Krum (Blanchard et al., 2017) algorithm as an instance can better distinguish clean and malicious clients, thus resulting in better defense and detection performances.



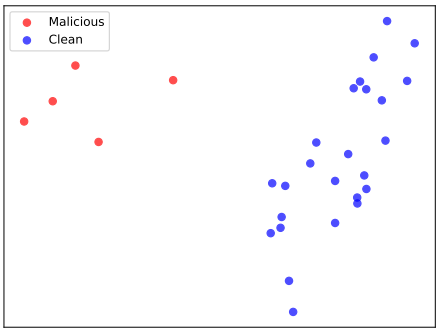
(a) Krum (5/30, run 1).



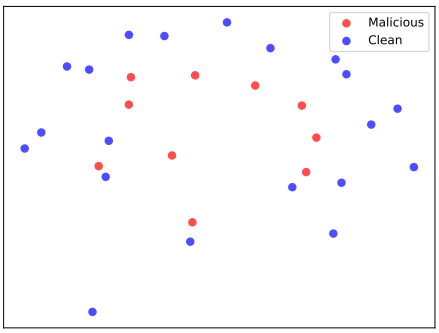
(b) Krum Enhanced with Client Embedding (5/30, run 1).



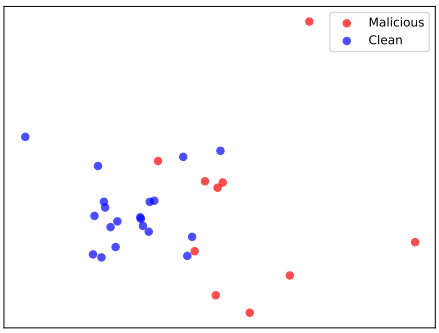
(c) Krum (5/30, run 2).



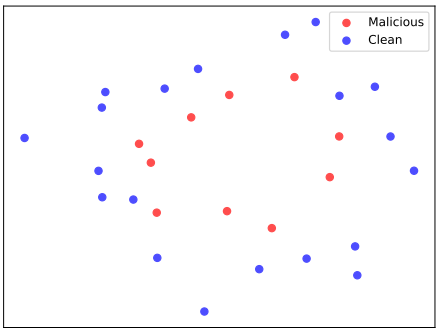
(d) Krum Enhanced with Client Embedding (5/30, run 2).



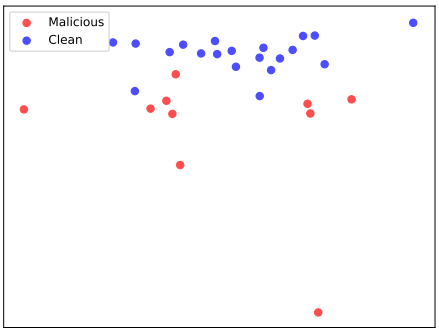
(e) Krum (10/30, run 1).



(f) Krum Enhanced with Client Embedding (10/30, run 1).



(g) Krum (10/30, run 2).



(h) Krum Enhanced with Client Embedding (10/30, run 2).

Figure 2: Illustrations of parameter (a,c,e,g) and embedding spaces (b,d,f,h). Malicious/total clients are provided.