Vol.2021-DPS-189 No.10
2021/12/21

Difficulty of detecting overstated dataset size in Federated Learning

Hideaki Takahashi[1,a)] Kohei Ichikawa[2,b)] Keichi Takahashi[2,c)]

Abstract: Federated learning is a distributed learning method in which multiple clients cooperate to train a model. Each client sends the gradient of its locally trained model to the server, and the server aggregates the received gradients to build a global model. Since federated learning requires many clients to train a high-performance model, researchers have designed incentive mechanisms that distribute rewards to clients to motivate their participation. While most incentive mechanisms distribute rewards according to the contribution of each client often defined by the number of data, little research has been done on the risk that clients try to claim more rewards by overstating the number of data. This paper proposes three possible methods to exaggerate the size of a local dataset: simple exaggeration of the reported number, modification of the batch size during training, and exaggeration of the dataset by Data Augmentation. Using a variety of models and datasets, we show the inadequacy of current anomaly detection methods in identifying such exaggerations.

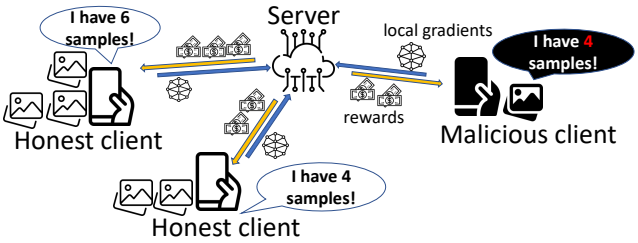

Fig. 1 Clients who overstate their local dataset size

1. Introduction

Federated learning allows multiple data owners to collaboratively train a global model without sharing their data. It has attracted much attention in recent years from the perspective of privacy protection and scalability. In typical federated learning, a central server first sends an initial model to the clients, and each client trains a local model with its data. Next, each client sends its calculated gradient to the central server, and the server updates the global model by aggregating the received gradients. By repeating this process, each client can train a highly accurate model without sharing its data with the central server or other clients [1].

One of the possible ways to motivate more clients to participate in federated learning is to distribute rewards according to the contribution of each client, *i.e.*, how much they contribute to improving the performance of the global model [2], [3]. However, there is an information asymmetry in federated learning, where clients have more information about their computational resources and data quality than the server, making it difficult for the server to estimate the contribution of each client correctly [1]. For example, many incentive mechanisms use dataset size to calculate contribution, but the server cannot know the actual sample size of each client.

Given the large number of clients cheating in distributed systems and crowdsourcing systems that are already in operation [4], [5], it is easy to imagine that some clients will try to trick the server and get paid without effort. For example, [6] showed that free-riders can participate in federated learning sessions and steal rewards and global models even though they have no data.

On the other hand, little research has been done on the risk that clients with a small amount of data may overreport the number of data to claim more rewards (Fig. 1). Suppose some clients overstate the dataset size. In that case, the server may suffer from problems such as the inability to distribute rewards appropriately and degradation of the accuracy of the global model.

In summary, we make the following contributions:

- To the best of our knowledge, this paper is the first study to demonstrate the dangers of clients overstating the number of data they report in federated learning.
- We propose Simple Overstatement, Overstatement with modified batch size, and Overstatement with Data Augmentation as possible strategies for attackers.
- We experimentally show that current anomaly detection methods for federated learning fail to detect overreported local dataset size.

The rest of this paper is organized as follows. In section 2, we first review the current status of Incentive Mechanisms and anomaly detection methods in federated learning. Next, in Section 3, we propose three methods for overstating the number of data: Simple Overstatement, Overstatement with modified batch

1 The University of Tokyo, Bunkyo, Tokyo 113–8654, Japan
2 Nara Institute of Science and Technology, Ikoma, Nara 113–8654, Japan
a) takahashi-hideaki567@g.ecc.u-tokyo.ac.jp
b) ichikawa@is.naist.jp
c) keichi@is.naist.jp

size, Overstatement with Data Augmentation. We also describe the existing anomaly detection methods used in this study. In Section 4, we evaluate the performance of existing anomaly detection methods against these overstatement methods on various datasets. In Section 5, based on the evaluation results of each technique, we show that the existing defense methods struggle with the overstatement of the dataset size.

2. Related work

Since data owners may not voluntarily participate in federated learning due to privacy concerns or computational costs, research on incentive mechanisms that reward data owners based on their contribution is in progress [2], [3]. Many incentive mechanisms often use the amount of data held by each client to calculate the contribution, as the accuracy of the model generally increases with the amount of data [7], [8], [9], [10]. On the other hand, many of these mechanisms do not validate the number of data reported by the client, suggesting that a malicious client may report a value that is larger than the amount of data it has and claim more rewards.

Malicious clients have been a well-known problem in distributed systems and crowdsourcing. For example, [4] reported that as many as 70% of users in Gnutella, a protocol for distributed file sharing, are free-riders. An empirical study of crowdsourcing for online surveys showed that 70% of users are untrustworthy clients [5]. Therefore, countermeasures against malicious clients are also essential in federated learning.

In the following, we discuss the current research on abnormal client behavior detection in federated learning and summarize the techniques for verifying the number of data reported by each client and more general anomaly detection methods in federated learning.

2.0.0.1 Estimating and guaranteeing the number of data

[11] suggested checking if the training time of each client is too fast to verify the number of data reported by each client. However, attackers can easily falsify the time taken for training by leaving an interval between the end of training and the submission of the model. In addition, [12] proposed a method to prevent malicious clients from altering the client software using Intel SGX. Still, it does not address the case where an attacker inflates the dataset in advance using techniques such as Data Augmentation.

2.0.0.2 Popular anomaly detection methods in federated learning

While the amount of research dedicated to overstating the local dataset size is still limited, many studies proposed more general methods for anomaly detection and contribution calculation in federated learning. This paper classifies them into the following four types by reference to [13].

- **Test/Self-Reported Based Detection**: The most naive way to calculate client contribution is to have each client report metrics related to the performance of the model, such as the number of data and local loss [7], [8], [9], [10], [14], [15]. However, these methods, by their very definition, do not assume a malicious client.
- **Marginal Loss Based Detection**: [16] and [17] measured the contribution of a client by the difference in performance and output of the global model with and without a client.
- **Similarity-Based Detection**: [18] and [19] exposed anomalous clients by measuring their similarity, such as the cosine similarity between the gradients submitted by each client.
- **ML-Based Detection**: [6] and [20] applied machine learning algorithms to gradients and other data received from clients to detect abnormal clients.

3. Method

To reveal the danger of local dataset size overstatement in federated learning, we propose a simple overstatement of declared local sample size and more sophisticated local dataset size overstatement methods. Then, we evaluate whether the proposed local dataset size overstatement methods can be detected by existing anomaly detection methods in federated learning and demonstrate the characteristics of each technique.

Section 3.1 describes the Simple Overstatement and the proposed more sophisticated methods of overstating the number of data. Section 3.2 describes the evaluation method of the existing anomaly detection methods in federated learning.

3.1 Proposed data overstatement methods

We propose three methods to overstate the number of data shown in Fig. 2: Simple Overstatement, Overstatement with modified batch size, and Overstatement with Data Augmentation.

3.1.0.1 Simple Overstatement

The simplest way to overstate the local dataset size is to declare a value to the server larger than the actual sample size while the attackers perform training as an honest client does. However, our preliminary experiments with FedAvg indicated that the standard deviation of the gradient is proportional to the actual number of data used for training (Appendix Fig. A·1). Thus, we believe that monitoring the standard deviation of the gradient allows the server to distinguish between honest and malicious clients easily. Also, if the attacker possesses only a tiny amount of data, the calculated gradient is expected to be biased.

3.1.0.2 Overstatement with modified batch size

Even if the server attempts to find abnormal clients based on the linear relationship between the actual train dataset size and the standard deviation of the gradient, [18] showed that clients could increase the standard deviation of the gradients of the model by reducing the batch size during training. Thus, attackers can make anomaly detection difficult by reducing the batch size to a value smaller than requested by the server.

3.1.0.3 Overstatement with Data Augmentation

Since Simple Overstatement and Overstatement with modified batch size require modification of the client program distributed by the server, Trusted Execution Environments such as Intel SGX might completely prevent these attacks [12], [13]. However, a client can still overreport its local dataset size without using a tampered program by inflating its local dataset in advance using Data Augmentation. This may also mitigate the bias caused by the overly small size of the local dataset and further hinder detection.

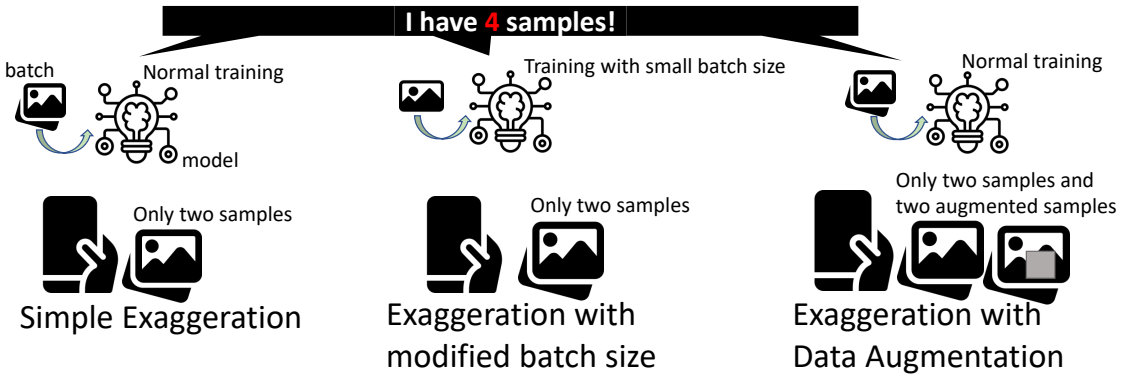

Fig. 2 Proposed data overstatement methods

3.2 Anomaly detection methods and evaluation metric

In this section, we discuss the evaluation metric and anomaly detection methods used in this study. We use Area Under the ROC Curve (AUC), a typical metric used in anomaly detection [21], to measure the performance of existing anomaly detection methods. For the anomaly detection method, we use the three existing methods shown in Fig. 3. We implement each algorithm with FedML [22], a framework for federated learning.

3.2.0.1 STD-NUM-DAGMM

[6] identified free-riders with STD-DAGMM, which used the gradient and its standard deviation sent by the client as features of the anomaly detection method named DAGMM. The original STD-DAGMM algorithm used the compressed gradient Z_c, the standard deviation of the gradient Z_{std}, and the Euclidean and Cosine distance Z_r between the input and output of the Auto Encoder which generated Z_c as features of the estimation network that fitted the Gaussian mixture model. In addition to these features, STD-NUM-DAGMM used in this paper also uses the reported local dataset size as a feature.

3.2.0.2 FoolsGold

[18] proposed a Similarity-Based Detection Method called FoolsGold, which assumes that model updates between malicious clients are similar. Based on this assumption, FoolsGold thinks that the angle θ between the update vectors of the malicious clients is smaller than the angle γ between the malicious clients and the honest clients. FoolsGold can prevent the Data Poisoning Attack, an attack that degrades the accuracy of the global model by mixing specially crafted data with the training data, by discovering those clients whose gradients have an abnormally high cosine similarity.

3.2.0.3 Quality Inference (QI)

Quality Inference (QI) [17] is a Marginal Loss Based Detection Method that uses the change in the performance of a global model between rounds where a client participates in training and rounds where it does not. QI has achieved high performance in various tasks such as data quality estimation and free-rider detection.

4. Experiment

This section examines the performance of existing anomaly detection methods against the dataset size overstatement method proposed in Section 3 on various datasets and models.

4.1 Datasets and models

4.1.0.1 CIFAR-10

First, we conduct experiments using CIFAR-10, one of the significant datasets in the image field. This dataset consists of 32×32 pixel color images belonging to 10 labels, with 50,000 training data and 10,000 test data. As a classification model, we use ResNet-56, a popular Deep Learning model in the image domain [23].

4.1.0.2 Shakespeare

We also experiment with a language model that predicts the next letter in a sentence using The Complete Works of William Shakespeare. The version used in this paper [22] consists of 16,068 training data and 2356 test data, with a vocabulary of 90. We use the LSTM model used in [14] for prediction.

4.2 Experimental setup

4.2.0.1 Attack strategy

For all overstatement methods, we set the number of attackers N_a to 5% or 20% of the total clients, and the overstatement factor M to either 2× or 10×. For Overstatement with modified batch size, the batch size B_a used by the attacker is $\frac{B}{M}$, where B is the batch size determined by the server. For Overstatement with Data Augmentation, we experiment only with CIFAR-10 and augment the dataset with random cropping and random horizontal flipping, which are basic Data Augmentation methods for images [24]. For the sake of simplicity, we assume that the top N_a clients with the least number of data are attackers.

4.2.0.2 Data splits

We set the total number of clients N to 20 or 50 in both datasets. We divide the CIFAR-10 dataset so that the sample size for each client S_i follows the power law, which is close to the distribution of real-world datasets [14]. As in the previous study [14], [25], we divide the Shakespeare dataset so that each client has lines of one character in a script to satisfy the power law and uses the top N people with the largest datasets. After this division, we multiply the sample sizes of attackers by $\frac{1}{M}$ to ensure that the total number of data recognized by the server is the same in all settings.

4.2.0.3 Hyper-parameters

We train ResNet-56 with Adam Optimizer with a default batch size of 20 and a learning rate of 0.001 and train LSTM with SGD with a default batch size of 10 and a learning rate of 1.47. Each

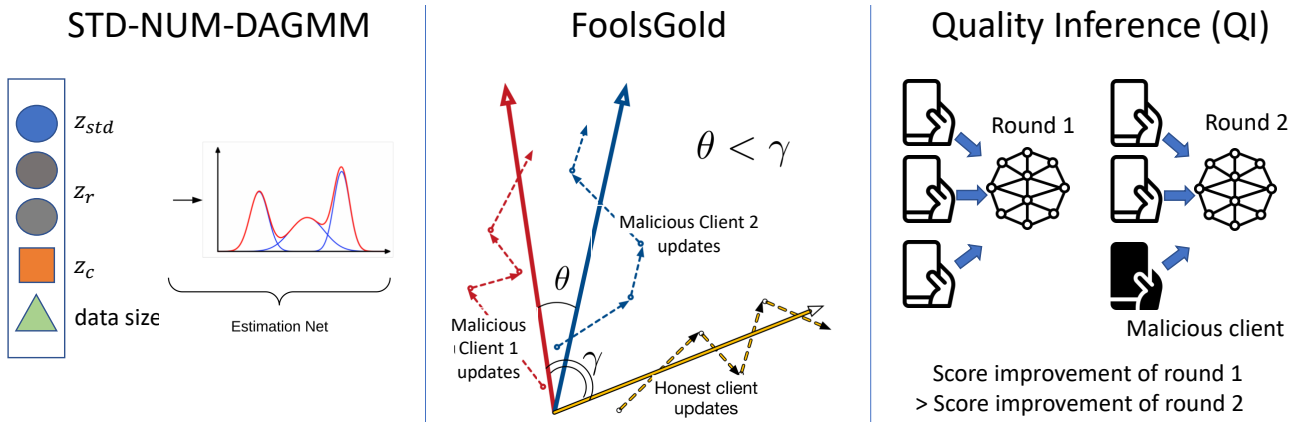

Fig. 3 Existing anomaly detection methods [6], [18]

client trains its model locally for five epochs and then sends the gradient to the server. This procedure is repeated for 50 rounds. STD-NUM-DAGMM is trained for five epochs each round with Adam Optimizer with a learning rate of 0.01. We set the confidence value k of FoolsGold to 0.005; QI samples half of the clients for training.

4.3 Experimental results

4.3.0.1 Simple Overstatement

Figure 4 shows the performance of each anomaly detection method against Simple Overstatement: the average AUC of the final ten rounds in CIFAR-10 was 0.838 for STD-NUM-DAGMM, 0.678 for QI, and 0.010 for FoolsGold, with only STD-NUM-DAGMM being above 0.7, which is the minimum AUC value considered meaningful [26]. The trend was the same for Shakespeare, where the average AUC for the final ten rounds was 0.843 for STD-NUM-DAGMM, 0.413 for QI, and 0.014 for FoolsGold.

4.3.0.2 Overstatement with modified batch size

Figure 5 shows the performance of STD-NUM-DAGMM when the attacker reduces the batch size for learning its model. Compared to Simple Overstatement, the number of rounds where AUC is below 0.5 increased, and the performance became unstable. The average AUC of STD-NUM-DAGMM for the final ten rounds significantly decreased to 0.684 for CIFAR-10 and 0.632 for Shakespeare. Compared to STD-NUM-DAGMM, the performance change of other detection methods was relatively small, where the score of QI was 0.723 for CIFAR-10 and 0.393 for Shakespeare, and the score of FoolsGold was 0.005 for CIFAR-10 and 0.016 for Shakespeare.

4.3.0.3 Overstatement with Data Augmentation

Figure 6 shows the performance of each method against Overstatement with Data Augmentation. The average AUC for the final ten rounds was 0.723 for STD-NUM-DAGMMM, 0.677 for QI, and 0.164 for FoolsGold.

5. Discussion

This section discusses the limitation of each anomaly detection method based on the results obtained in Section 4.

Only the AUC of STD-NUM-DAGMM showed high performance against Simple Overstatement, exceeding 0.7. This result is probably because the model learns the linear relationship between the number of data and the standard deviation of gradient and sees that the standard deviation of the gradients of the attackers is abnormally tiny relative to the reported dataset size. However, when the attacker reduces the batch size, the standard deviation of the gradient becomes larger, and the detection performance of STD-NUM-DAGMM deteriorates significantly, with the average AUC below 0.7. The Overstatement with Data Augmentation also degraded the performance of STD-NUM-DAGMM by more than 0.1. The possible reason is that artificially expanding the dataset may raise the standard deviation of the gradient and eliminate the gradient bias.

The AUC of FoolsGold eventually reaches almost zero in all settings. This score means that FoolsGold classifies honest clients as abnormal clients, and vice versa. The reason for this is that although FoolsGold assumes that the gradient cosine similarity between attackers is high, the tiny samples sizes of the attackers cause the distributions of their datasets to deviate from those of other clients. This deviation causes relatively low similarities between attackers. If the server knows this fact, it may try to identify the attacker by reversing the evaluation criteria of FoolsGold and judging those with low similarity with other clients as malicious clients. However, suppose the attacker has sufficient data, and there is no discrepancy between the data distribution of attackers and the distribution of the entire dataset. In that case, FoolsGold cannot correctly identify the attacker because the similarity between the gradients of the attacker and other clients will be high. When we conduct the experiment assuming that the attacker has the median number of data across all clients, the AUC of FoolsGold stays around 0.5, which is equal to the chance level, as shown in Fig. 7.

The poor detection performance of QI is probably because the model performance degradation due to overstatement of the number of data is slight, unlike attack methods such as free-riders and Data Poisoning. Although QI worked better against Overstatement with modified batch size than Simple Overstatement on CIFAR-10, the result on Shakespeare was the opposite. The possible reason is that the smaller batch size decreased the performance of the local models of attackers on CIFAR-10 but improved on Shakespeare.

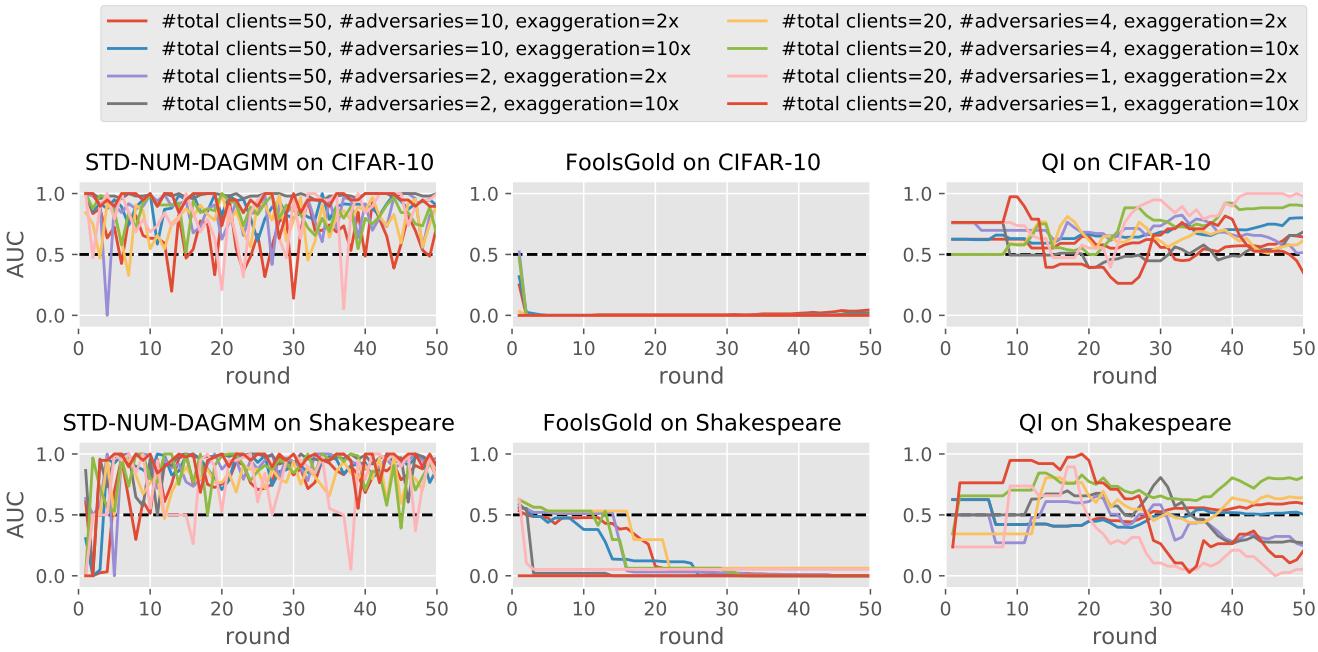

Fig. 4 Simple Overstatement

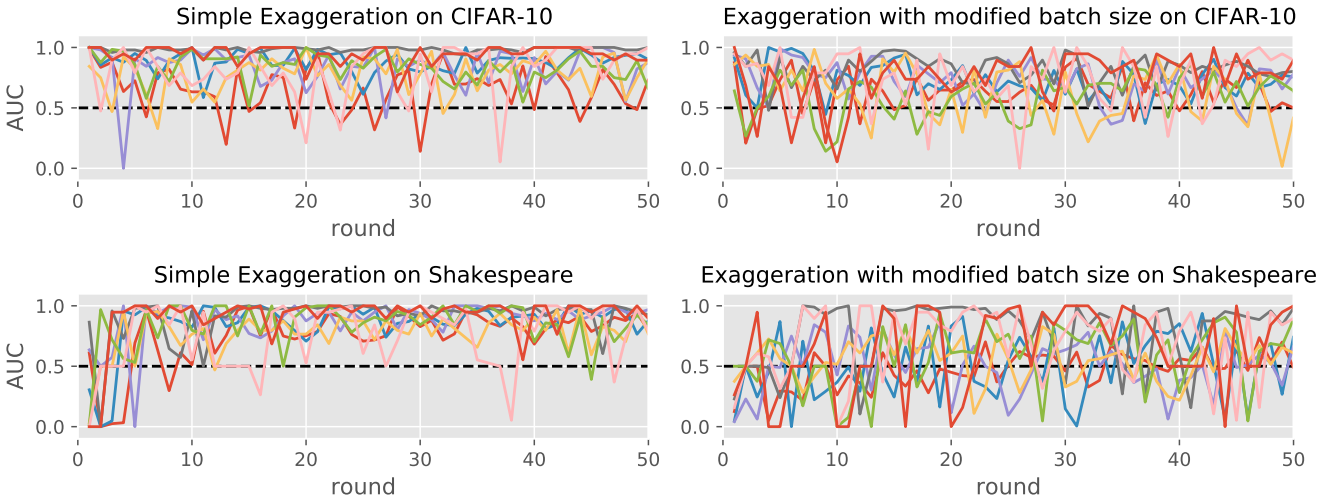

Fig. 5 Overstatement with modified batch size

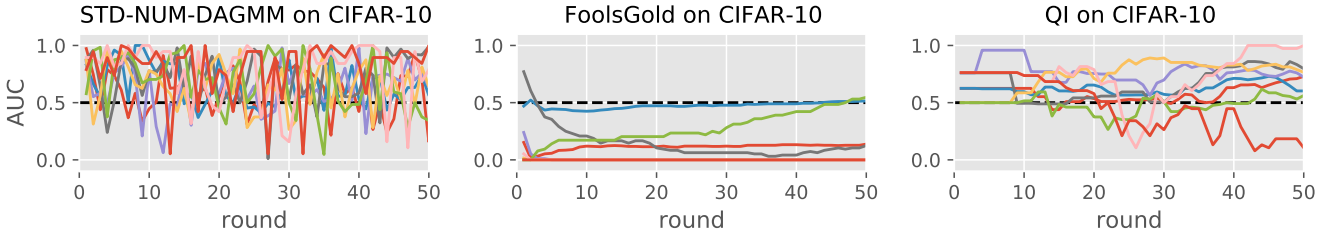

Fig. 6 Overstatement with Data Augmentation

6. Conclusion and future work

Federated Learning is a distributed learning method attracting attention from the viewpoint of privacy protection and scalability. In this paper, we raised the issue of clients overstating their sample size to gain more rewards and proposed three simple strategies that attackers can take: Simple Overstatement, Overstatement with modified batch size, and Overstatement with Data Augmentation. Experiments on image and natural language datasets showed that anomaly detection methods such as STD-NUM-DAGMM, FoolsGold, and QI were unable to prevent sophisticated attacks based on Overstatement with modified batch

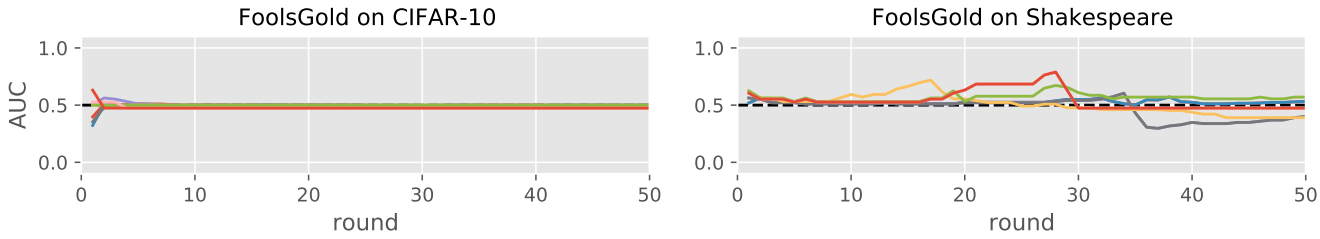

Fig. 7 FoolsGold against attackers with a lot of data

size and Overstatement with Data Augmentation.

Future research directions include theoretical clarification of dataset size overstatement, development of effective anomaly detection methods, and anomaly detection when the server cannot observe the gradient of each client by secure-aggregation [27].

References

[1] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D. and Miao, C.: Federated learning in mobile edge networks: A comprehensive survey, *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 3, pp. 2031–2063 (2020).

[2] Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P. and Guo, S.: A Survey of Incentive Mechanism Design for Federated Learning, *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1 (online), DOI: 10.1109/TETC.2021.3063517 (2021).

[3] Zeng, R., Zeng, C., Wang, X., Li, B. and Chu, X.: A Comprehensive Survey of Incentive Mechanism for Federated Learning, *CoRR*, Vol. abs/2106.15406 (online), available from ⟨https://arxiv.org/abs/2106.15406⟩ (2021).

[4] Adar, E. and Huberman, B. A.: Free riding on gnutella (2000).

[5] Gadiraju, U., Kawase, R., Dietze, S. and Demartini, G.: Understanding malicious behavior in crowdsourcing platforms: The case of online surveys, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1631–1640 (2015).

[6] Lin, J., Du, M. and Liu, J.: Free-riders in federated learning: Attacks and defenses, *arXiv preprint arXiv:1911.12560* (2019).

[7] Zhan, Y., Li, P., Wang, K., Guo, S. and Xia, Y.: Big Data Analytics by CrowdLearning: Architecture and Mechanism Design, *IEEE Network*, Vol. 34, No. 3, pp. 143–147 (online), DOI: 10.1109/MNET.001.1900286 (2020).

[8] Zhan, Y., Zhang, J., Li, P. and Xia, Y.: Crowdtraining: Architecture and Incentive Mechanism for Deep Learning Training in the Internet of Things, *IEEE Network*, Vol. 33, No. 5, pp. 89–95 (online), DOI: 10.1109/MNET.001.1800498 (2019).

[9] Zhan, Y., Li, P., Qu, Z., Zeng, D. and Guo, S.: A Learning-Based Incentive Mechanism for Federated Learning, *IEEE Internet of Things Journal*, Vol. 7, No. 7, pp. 6360–6368 (online), DOI: 10.1109/JIOT.2020.2967772 (2020).

[10] Feng, S., Niyato, D., Wang, P., Kim, D. I. and Liang, Y.-C.: Joint service pricing and cooperative relay communication for federated learning, *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, pp. 815–820 (2019).

[11] Kang, J., Xiong, Z., Niyato, D., Zou, Y., Zhang, Y. and Guizani, M.: Reliable federated learning for mobile networks, *IEEE Wireless Communications*, Vol. 27, No. 2, pp. 72–80 (2020).

[12] Kim, H., Park, J., Bennis, M. and Kim, S.-L.: Blockchained on-device federated learning, *IEEE Communications Letters*, Vol. 24, No. 6, pp. 1279–1283 (2019).

[13] Huang, J., Talbi, R., Zhao, Z., Boucchenak, S., Chen, L. Y. and Roos, S.: An Exploratory Analysis on Users' Contributions in Federated Learning, *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, IEEE, pp. 20–29 (2020).

[14] McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B. A.: Communication-efficient learning of deep networks from decentralized data, *Artificial intelligence and statistics*, PMLR, pp. 1273–1282 (2017).

[15] Kang, J., Xiong, Z., Niyato, D., Yu, H., Liang, Y.-C. and Kim, D. I.: Incentive design for efficient federated learning in mobile networks: A contract theory approach, *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, IEEE, pp. 1–5 (2019).

[16] Wang, G., Dang, C. X. and Zhou, Z.: Measure contribution of participants in federated learning, *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 2597–2604 (2019).

[17] Pejó, B.: The Good, The Bad, and The Ugly: Quality Inference in Federated Learning, *CoRR*, Vol. abs/2007.06236 (online), available from ⟨https://arxiv.org/abs/2007.06236⟩ (2020).

[18] Fung, C., Yoon, C. J. and Beschastnikh, I.: Mitigating sybils in federated learning poisoning, *arXiv preprint arXiv:1808.04866* (2018).

[19] Xu, X. and Lyu, L.: Towards Building a Robust and Fair Federated Learning System, *CoRR*, Vol. abs/2011.10464 (online), available from ⟨https://arxiv.org/abs/2011.10464⟩ (2020).

[20] Li, S., Cheng, Y., Wang, W., Liu, Y. and Chen, T.: Learning to detect malicious clients for robust federated learning, *arXiv preprint arXiv:2002.00211* (2020).

[21] Moumena, A.: Anomalies Detection Based on the ROC Analysis using Classifiers in Tactical Cognitive Radio Systems: A survey, *IAES International Journal of Artificial Intelligence (IJ-AI)*, Vol. 4, p. 12 (online), DOI: 10.11591/ij-ai.v4i3.1415 (2016).

[22] He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H. et al.: Fedml: A research library and benchmark for federated machine learning, *arXiv preprint arXiv:2007.13518* (2020).

[23] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016).

[24] Shorten, C. and Khoshgoftaar, T. M.: A survey on image data augmentation for deep learning, *Journal of Big Data*, Vol. 6, No. 1, pp. 1–48 (2019).

[25] Fraboni, Y., Vidal, R. and Lorenzi, M.: Free-rider attacks on model aggregation in federated learning, *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 1846–1854 (2021).

[26] Hosmer Jr, D. W., Lemeshow, S. and Sturdivant, R. X.: *Applied logistic regression*, Vol. 398, John Wiley & Sons (2013).

[27] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A. and Seth, K.: Practical secure aggregation for privacy-preserving machine learning, *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191 (2017).

Appendix

A.1 Appendix

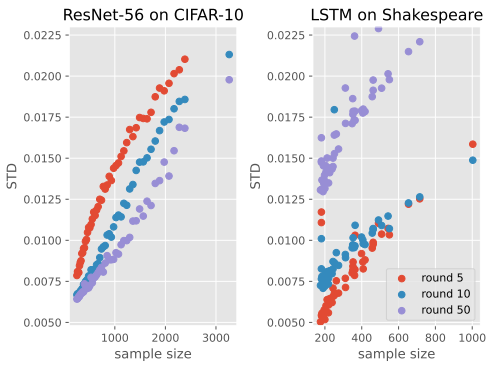

Fig. A·1 Relationship between number of data and standard deviation of gradients