Differential Privacy Approach to Solve Gradient Leakage Attack in a Federated Machine Learning Environment

Krishna Yadav[1], B. B. Gupta[1(✉)], Kwok Tai Chui[2], and Konstantinos Psannis[3]

[1] Department of Computer Engineering, National Institute of Technology Kurukshetra, Kurukshetra, Haryana, India
krishna.nitkkr1@gmail.com, gupta.brij@gmail.com
[2] The Open University of Hong Kong, Kowloon, Hongkong, China
jktchui@ouhk.edu.hk
[3] University of Macedonia, Thessaloniki, Macedonia, Greece
kpsannis@uom.edu.gr

Abstract. The growth of federated machine learning in recent times has dramatically leveraged the traditional machine learning technique for intrusion detection. Keeping the dataset for training at decentralized nodes, federated machine learning have kept the people's data private; however, federated machine learning mechanism still suffers from gradient leakage attacks. Adversaries are now taking advantage of those gradients and can reconstruct the people's private data with greater accuracy. Adversaries are using these private network data later on to launch more devastating attacks against users. At this time, it becomes essential to develop a solution that prevents these attacks. This paper has introduced differential privacy, which uses Gaussian and Laplace mechanisms to secure updated gradients during the communication. Our result shows that clients can achieve a significant level of accuracy with differentially private gradients.

Keywords: Federated learning · Machine learning · Intrusion detection · Differential privacy · Gradient leakage

1 Introduction

In the past few years, we can see the massive increment in the computational resources in smartphones. Modern days smartphones are powered up by high GPU and memory spaces. Due to this computational infrastructure, researchers and developers have embedded several smartphone applications, which are fueled up by the machine and deep learning algorithms. The evolution of machine learning algorithms has extensively leveraged modern apps intelligence, making people's life a lot easier. These algorithms need large amounts of data to be trained to make an intelligent decision. Initially, people were not concerned about their data being exported to large organizations, but as time progressed, several data breaches have occurred, leading to the transfer of people's private data to the adversaries [1, 2]. Adversaries are then using personal data to

S. Chellappan et al. (Eds.): CSoNet 2020, LNCS 12575, pp. 378–385, 2020.
https://doi.org/10.1007/978-3-030-66046-8_31

launch phishing attacks and even blackmail the people [3]. In the case of the Intrusion detection system (IDS), researchers are exporting people's network traffic datasets to build an intelligent IDS for early detection of malicious traffic in people's smartphones [4]. When these types of network traffic datasets get leaked during a data breach, they have served as a baseline to launch more devastating attacks to the users [5]. To solve these problems, and keep people's data private in 2017, Google introduced federated machine learning [6]. Federated machine learning does not export the people's data to the server and trains the machine learning models in their smartphone with their locally generated dataset. Results show that federated machine learning provided the same level of intelligent decision making as with traditional machine learning. Federated machine learning was introduced to solve people's privacy issues; however, this approach was entirely not private. If the adversaries have control over the server where the unencrypted updated gradients from the clients were sent to be averaged or merely intercepting the gradients during federated communications, adversaries were able to reconstruct the dataset present across people's smartphones. Authors at [7] developed a gradient attack algorithm that reconstructed the MNIST dataset with significant accuracy. Moreover, the author at [8] used cosine similarity loss and some optimization methods to rebuild the images by taking the knowledge of shared gradients to the server during federated learning.

This paper uses a differential privacy approach to add noise to the gradients being shared with the server for averaging. We have used two noises, i.e., Gaussian and Laplace noise. In our experimentation, we added three noise levels, i.e., low noise, medium noise, and high noise to the gradients, and saw their accuracy. We have further compared how a differential privacy approach affects a model's accuracy than the non-differential method. We have also discussed how our differential privacy approach gives the adversaries no chance to perform gradient leakage attacks.

The rest of the paper is organized as follows: Sect. 2 gives an overview of work that has been done to solve gradient leakage attacks in federated machine learning. Section 3 gives the reader an introductory background on differential privacy. Section 4 provides an overview of our proposed approach. Section 5 and Sect. 6 discusses the experimentation and results. Finally, Sect. 7 concludes the paper.

2 Related Work

In this paper, we experiment with different possible noise mechanisms that can be used to achieve differential privacy in a federated machine learning environment; however, in literature, we can find some amount of work that is being carried out in differential privacy. Authors at [9] have proposed two mechanisms, i.e., Random sub-sampling and Distorting, to hide the gradients of each client, which later is sent to the server for aggregation. They followed the Gaussian mechanism to distort the sum of all the updated gradients at the client-side. Their proposed differentially private approach achieved more than 90% accuracy which was only 5% less than the non-differential privacy approach. Authors at [9] achieved a reasonable amount of accuracy; however, authors at [10] stated that the traditional differential privacy approach carried out by [9] led to significantly decreased accuracy, so they came with a hybrid approach where they combined differential privacy and secure multiparty computation (SMC) to mask the gradients. Their

proposed approach is more reliable than the author at [9] since combining differential privacy with SMC will reduce the growth of noise injected during federated training as the number of clients increases. Authors at [11] realized that the traditional differential privacy approach creates a lot of noise in the updated gradients in federated training, which have 10s of digits after a decimal point. To solve this problem they came with a LDP-Fed mechanism that first provides a differential privacy guarantee to the updated gradients and then selects and filters the perturbing gradients in the server. Their approach is very effective for precision sensitive gradients value; however, filtering the gradients will require extra effort and computation and may lead to the delay in communication during federated training. Authors at [12] have used the bayesian differential privacy mechanism for preserving the gradients. Their proposed mechanism leads to the addition of a lower amount of noise, which ultimately has increased the accuracy of the model with reduced communication rounds. Authors at [13] have proposed a differential privacy mechanism (NbAFL), which adds the artificial noise to the gradients being updated. They first proved that their proposed mechanism satisfied the principles of differential privacy and did various performance analysis of the model after the gradients were obfuscated with noise.

Most of the literature's work introduces the differential privacy on image, text, and health datasets. Our research work is a pioneer in the literature that applies differential privacy in the intrusion detection dataset to the best of our knowledge. Detecting intrusion is a sensitive and rapid process. Unlike most of the work described in the literature, which requires higher communication rounds to achieve reasonable accuracy after adding noise to the updated gradients, our proposed approach does not require higher communication rounds, perturbing techniques as described in [11] and can achieve a good amount of accuracy in detecting intrusion.

3 Background

In this section, we introduce the definition of differential privacy, give a brief discussion about the Gaussian and Laplace mechanism. We further will discuss the use of these mechanisms in achieving differential privacy.

3.1 Differential Privacy

Differential privacy was introduced to permit statistical analysis of the dataset without revealing the private information of an individual inside a dataset [14]. For any two neighboring dataset D_1 and D_2, a randomized mechanism M is said to be differentially private if the outcome, i.e., C from a mechanism M such that $C \in \text{Range}(M)$ and $\|D_1 - D_2\| \leq 1$ where $\|D_1 - D_2\|$ is the distance between two datasets D_1 and D_2. Equation 1 below represents a differentially private equation. Here, D_1 is a measure of the dataset's size and $\|D_1 - D_2\|$ gives an idea about the difference in the record between two datasets, D_1, and D_2.

$$Pr\big[M(D1) \in C\big] \leq exp(\varepsilon)\, Pr\big[M(D2) \in C\big] + \delta \tag{1}$$

In the above equation, δ controls the amount of additive noise. There is a theoretical difference between (ε, 0) and (ε, δ). (ε, 0) means that for every D_1, the output $M(D_1)$ observed is likely to be equally observed for all the data in a dataset. When the observed output is expected to be the same, an adversary may infer valuable information from the output, and we will not be able to achieve higher privacy. (ε, δ) for dataset D_1 the output with the mechanism $M(D_1)$ is unlikely to be observed for the dataset D_2. When δ has a higher value greater than zero, we may add a significant level of noise, and a reasonable amount of privacy of the data can be achieved. To keep track of the amount of privacy one can reach by varying the value of δ, we have a privacy loss parameter. For a given output $\varepsilon \backsim M(D_1)$, adversaries may produce a dataset D_2 such that $D_1, D2 \in D$, then the privacy loss is defined as:

$$\mathcal{L} exp(\varepsilon)\, M(D1) \,||\, M(D2) \;=\; ln\left(\frac{Pr[M(D1)=\varepsilon]}{Pr[M(D2)=\varepsilon]}\right) \tag{2}$$

Privacy loss can also be defined as an essential random variable that keeps track of noise being added from any random mechanism.

In differential privacy, we can use any Randomized algorithm(M) with dataset D and range R such that M: $D \rightarrow \Delta(R)$ such that on input, let's say, $d \in D$, the randomized algorithm M produces output $M(d) = r$ with probability $(M(d))_r$ for each $r \in R$. Randomization algorithms such as Gaussian, Laplace, exponential, and Bayesian can be used to achieve differential privacy, however, in our approach, we have used only two of those algorithms, i.e., Gaussian and Laplace which is discussed below.

3.2 Laplace Mechanism

Laplace mechanism is derived from Laplace distribution where for a value x, Laplace distribution is $\text{Lap}(x|\mu, b) = \frac{1}{2b} \exp\left(\frac{-|x-\mu|}{b}\right)$. In the equation, μ tells us about the position of distribution, i.e., whether positive or negative, and b is an exponential scale parameter which gives an idea about the distribution of a laplacian noise [14]. The parameter b depends upon two things, i.e., ε and Δf. Here Δf is a sensitivity, which is the change in output obtained when the same function is applied for both dataset D_1 and D_2.

$$\Delta f \;= max\, D1,\; D2|f(D1) - f(D2)| \tag{3}$$

For a randomized function f: $D \rightarrow R$, the laplace mechanism is defined as: M(d, f(.), ε) = f(d) + Lap(Δf/ε), where Lap(Δf/ε) is a laplacian noise.

3.3 Gaussian Mechanism

Let f: $D \rightarrow R$ be a randomized function with its sensitivity Δf. The Gaussian mechanism with a parameter σ adds noise $N(0, I\sigma^2)$. Then gaussian mechanics is defined as M(d) = f(d) + $N(0, I\sigma^2)$ where $\sigma > \Delta f \sqrt{2\, log \frac{1.25}{b}} \Big/ \varepsilon$. The noise $N(0, I\sigma^2)$ is obtained from a Gaussian distribution. For additional information on the Gaussian mechanism and Gaussian distribution, we encourage the readers to [14].

4 Proposed Approach

4.1 Differentially Private SGD Algorithm

Our proposed algorithm is described in Algorithm 1. Our central intuition in the proposed algorithm was to mask the gradients generated by each device with noise obtained from differential privacy mechanisms in federated communication. In algorithm 1, initially, all the nodes receive the global model initialized with some weight to their respective parameters. In the case of the SGD classifier, the parameter is a learning rate η. The global dataset D is divided into local datasets across each node, which is denoted by d_n. The received global model is trained on each client's local dataset, i.e., d_n, and the weight of the parameter, i.e., W_{weight} is obtained. The obtained weight is then masked with the noise factor F_{noise} produced by mechanisms, such as Gaussian and Laplace. Each client's masked weights are then sent to the server for averaging, and $W_{average}$ denotes the average weight. The $W_{average}$ is again obfuscated at the server side with differential privacy mechanisms and sent to different clients. The obfuscation of weight at the server side will prevent the adversaries from gaining knowledge from the gradients if they can intercept the gradients during federated communication.

Algorithm 1. Federated averaging along with the addition of noise

Procedure Server()
- W_{all_nodes}= Receive_weight()
- $W_{average} \leftarrow \frac{1}{n}\sum_{k=1}^{n} W_k$
- Set weight of $M_{global} \leftarrow W_{average} + F_{noise}$
- Send(M_{global})

Procedure Node()
- D← Local dataset divided into mini datasets
- M_{global}=Receive(global_model)
- **for** all each node in K in parallel **do**
  - Train M_{global} with their respective dataset d_n
  - $W_{weight} = W_{weight} - \eta\Delta l(W_{weight,}\ d_n)$
  - $W_{updated\_weight} \leftarrow W_{weight} + F_{noise}$
  - Send_weight($W_{updated\_weight}$)

5 Experimentation

Our experimentation was performed on a machine having a core i5 processor with a clock speed of 3.4 GHz, 8 GB of RAM, and 2 GB of the graphics card. The experimentation was fully performed in Python. We have used the Diffprivlib library developed by IBM to implement differentially private mechanisms [15]. To validate our approach, we take the NSL-KDD dataset for federated machine learning. NSL-KDD dataset is considered as a benchmark dataset for building an intrusion detection system [16]. NSL-KDD contains

125,973 records in KDDTrain+ file. We divided records in this file among ten nodes, and the distribution of our dataset was performed in a Non-IID manner. The distribution of the dataset among ten nodes is described in Table 1. We only made ten nodes in our experimentation and went up to 100 communication rounds due to computational infrastructure limitations. However, the result obtained is very reliable.

Table 1. Distribution of KDDTrain+ among 10 nodes.

Dataset	Attack type	Number of instances
Node1	Neptune, smurf	40466
Node2	Land, teardrop	6107
Node3	Pod, back	6315
Node4	Portsweep, nmap	8938
Node5	Ipsweep, satan	11135
Node6	Imap, warezmaster	5420
Node7	Ftp_write, guess_passwd	5442
Node8	Multihop, spy	5402
Node9	Phf, warezclient, buffer_overflow	6135
Node10	Loadmodule, perl, rootkit	5411

6 Result and Discussion

We added three levels of noise during federated communication, i.e., low, medium, and high to achieve differential privacy. The epsilon value was set to 0.05, 0.01, 0.005 to achieve low, medium, and high noise levels. The obtained accuracy with two different noise mechanisms is presented in Table 1. As gaussian noise depends upon three parameters, i.e., sensitivity(Δf), delta(δ), and epsilon(ε), we set the $\Delta f = 1$ and threshold value to $\delta = 0.7$. For Laplace noise and gaussian noise, the experiment was conducted without any change in parameter value mentioned above. Table 1 shows that we can achieve the highest accuracy of 96.37% by adding a sufficiently adequate amount of noise to the gradients. We also see that to achieve the highest level of privacy, we can use a high amount of noise to the gradients, but it will significantly decrease the accuracy and even requires a large communication round for loss function to be converged. The addition of the low or high amount of noise to the gradients depends upon the type of machine learning problem we are dealing with. A medical health record in federated machine learning may need the highest amount of differential privacy. In contrast, privacy will not be the highest preference in a movie or text recommendation system. Table 2 represents the differential privacy result obtained with the Laplace mechanism; however, there was no significant difference in accuracy and communication round for these two mechanisms, i.e., Laplace and Gaussian. Hence, we can choose any noise mechanism to achieve differential privacy in a federated environment.

Table 2. Accuracy of nodes under different level of noise

Noise	Communication round	Accuracy
Gaussian mechanism		
Low	29	96.37
Medium	64	92.24
High	95	86.39
Laplace mechanism		
Low	26	95.64
Medium	60	92.01
High	89	86.08

We also experimented with the Gaussian mechanism with different epsilon(ε) values to get knowledge about the obtained accuracy by varying noise levels. We set the range $\varepsilon = \{10^{-3}, 10^{-2}, 110^{-1}, 1\}$, and the accuracy obtained is shown in Fig. 1. Figure 1 depicts the lower value of ε leads to higher differential privacy, but lower accuracy and increase in the value of ε compromises the differential privacy; however, we can get sufficient accuracy. Figure 1 also gives a comparison of the variation of accuracy with differentially private and non-private approaches.

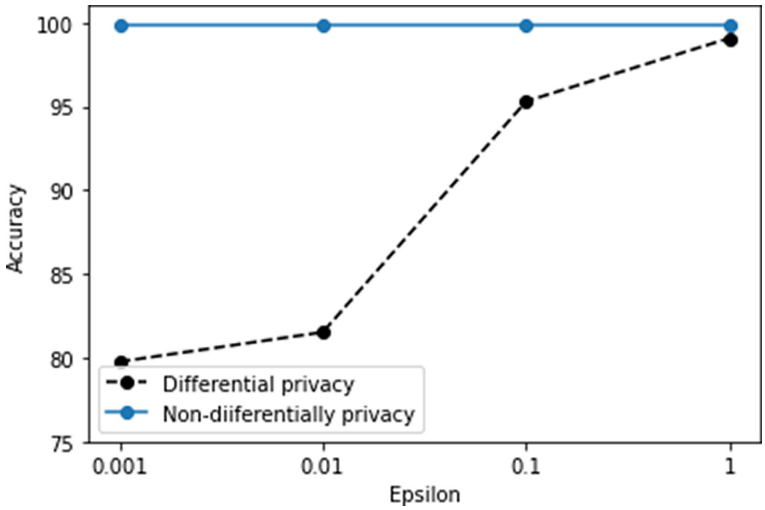

Fig. 1. Accuracy versus epsilon(ε) for a differentially private and non-private approach.

7 Conclusion

In this paper, we discussed two differentially private mechanisms, i.e., Gaussian and Laplace, to achieve differential privacy in a federated environment. The result shows that the mechanisms used produces a significant amount of accuracy without giving the adversaries the chance to share the gradients during federated communication. However, in an intrusion detection system built using a federated approach for smartphones, a false negative of even 5% can lead to a huge loss. In this type of system where accuracy is an utmost preference, in the coming future, we are planning to develop a sophisticated technique that can solve the gradient leakage attack without compromising the accuracy during federated communication.

References

1. Gupta, S., Gupta, B.B.: Detection, avoidance, and attack pattern mechanisms in modern web application vulnerabilities: present and future challenges. Int. J. Cloud Appl. Comput. (IJCAC) **7**(3), 1–43 (2017)
2. Kumar, A.: Design of secure image fusion technique using cloud for privacy-preserving and copyright protection. Int. J. Cloud Appl. Comput. (IJCAC) **9**(3), 22–36 (2019)
3. Jain, A.K., Gupta, B.B.: Towards detection of phishing websites on client-side using machine learning based approach. Telecommun. Syst. **68**(4), 687–700 (2018)
4. Jiang, F., et al.: Deep learning based multi-channel intelligent attack detection for data security. IEEE Trans. Sustain. Comput. **5**, 204–212 (2018)
5. Ramos, J., Nedjah, N., de Macedo Mourelle, L., et al.: Visual data mining for crowd anomaly detection using artificial bacteria colony. Multimed. Tools Appl. **77**(14), 17755–17777 (2018)
6. Cheng, K., et al.: Secureboost: a lossless federated learning framework. arXiv preprint arXiv:1901.08755 (2019)
7. Wei, W., et al.: A framework for evaluating gradient leakage attacks in federated learning. arXiv preprint arXiv:2004.10397 (2020)
8. Geiping, J., et al.: Inverting gradients–how easy is it to break privacy in federated learning?. arXiv preprint arXiv:2003.14053 (2020)
9. Geyer, R.C., Tassilo, K., Moin, N.: Differentially private federated learning: a client level perspective. arXiv preprint arXiv:1712.07557 (2017)
10. Truex, S., et al.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (2019)
11. Truex, S., et al.: LDP-fed: federated learning with local differential privacy. In: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking (2020)
12. Triastcyn, A., Boi, F.: Federated learning with Bayesian differential privacy. In: 2019 IEEE International Conference on Big Data (Big Data). IEEE (2019)
13. Wei, K., et al.: Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans. Inf. Forensics Secur. **15**, 3454–3469 (2020). https://doi.org/10.1109/TIFS.2020.2988575
14. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014)
15. Holohan, N., et al.: Diffprivlib: the IBM differential privacy library. arXiv preprint arXiv:1907.02444 (2019)
16. Dhanabal, L., Shantharajah, S.P.: A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. **4**(6), 446–452 (2015)