Secure Federated Learning with Efficient Communication in Vehicle Network

Yinglong Li[1], Zhenjiang Zhang[2], Zhiyuan Zhang[1], Yi-Chih Kao[3]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, China
[2] School of Software Engineering, Beijing Jiaotong University, China
[3] Information Technology Service Center, National Chiao Tung University, Taiwan
ylongli@bjtu.edu.cn, zhangzhenjiang@bjtu.edu.cn, zhangzhiyuan@bjtu.edu.cn, ykao@mail.nctu.edu.tw

Abstract

Internet of Vehicles (IoV) generates large amounts of data at the network edge. Machine learning models are often built on these data, to enable the detection, classification, and prediction of traffic events. Due to network bandwidth, storage, and especially privacy concerns, it is often impossible to send all the IoV data to the edge server for centralized model training. Federated learning is a promising paradigm for distributed machine learning, which enables edge nodes to train models locally. As vehicle usually has unreliable and relatively slow network connection, reducing the communication overhead is importance. In this paper, we propose a secure federated learning with efficient communication (SFLEC) scheme in vehicle network. To protect the privacy of local update, we upload the updated parameters of the model with local differential privacy. We further propose a client selection approach that identifies relevant updates trained by vehicles and prevents irrelevant updates from being uploaded for reduced network footprint to achieve efficient communication. Then we prove the loss function of the trained FL in our scheme exits a theoretical convergence. Finally, we evaluate our scheme on two datasets and compare with basic FL. Our proposed scheme improves the communication efficiency, while preserves the data privacy.

Keyword: Edge computing, Federated learning, Privacy preservation, Client selection

1 Introduction

The development of 5G and edge computing has brought new vitality into smart city, resulting in an exponential growth of data generated by the internet of vehicles. In the vehicle networks, it is a challenge for vehicle to use the massive data for providing better services, such as autonomous driving and traffic prediction, due to computing resource and the bandwidth of wireless networks constraints [1]. To solve this problem, Mobile Edge Computing (MEC) is envisioned as a potential solution. Mobile Edge Computing make it possible for vehicles, which are equipped with computing and storage capability, to store and process data locally. In [2], the authors propose a new VEC offloading scheme, which consider the sharing of the backup server resources between the VEC servers. In [3], the authors studied a multi-user multi edge-node computation offloading problem and proposed a model-free reinforcement learning offloading mechanism to maximize the long-term utilities. In [4], the authors addressed the resource allocation problem using convex and quasi-convex optimization techniques, and proposed a novel heuristic algorithm to the task offloading problem.

Generally, MEC framework suppose that all data are transferred from clients (vehicles, wireless sensors and IoT devices) to cloud computing servers through cellular networks to process their data. However, when the data contains personal privacy, such as health information, website visit history and phone calls record, the clients are unwilling to upload data to the cloud server. To address this privacy concern, as a decentralized machine learning technique, Federated Learning (FL) has recently been presented by ML community [5]. Instead of training model with the dataset of clients in central server, Federated Learning assigns the training work to distributed users. In order to protect the data privacy, each client trains their local model based on local training dataset which is never uploaded to the central server. Instead, each client computes an update to the current global model maintained by the server, and only this update is communicated.

In this paper, we focus on the Implementation of federated learning in practical MEC frameworks of vehicle network and propose a Secure Federated Learning with Efficient Communication scheme in vehicle edge network. Our main contributions are as follow:

We add artificial noise into gradient descent training

*Corresponding Author: Zhenjiang Zhang; E-mail: zhangzhenjiang@bjtu.edu.cn
DOI: 10.3966/160792642020122107022

process in order to prevent the parameters of updated model leakage in federated learning. And we propose a client selection approach that identifies relevant updates trained by vehicles and prevents irrelevant updates from being uploaded for reduced network footprint to achieve efficient communication for federated learning. Specifically, our scheme provides vehicles with the feedback information regarding the global tendency of model updating. Each vehicle checks if its update aligns with this global tendency and is similar enough to global update, which makes the whole training process efficient and reduces the communication costs. This is a client selection problem that determines which vehicles participate. Then, we put forward a convergence bound on the loss function of the trained FL model in our scheme with Gaussian noise. Finally, compared with basic Federated Learning, we evaluate our scheme on two datasets and evaluation results demonstrate that our scheme reduces communication overhead, outperforming the basic FL by 4.0x.

The remaining of this paper is organized as follows. In Section II, we present the related work. In Section III, we introduce the federated learning and formulate our problem. In Section IV, our Secure Federated Learning with Efficient Communication scheme is provided in detail and analyze the convergence property of SFLEC. In Section V, we present the numerical results of our proposed scheme on MNIST and CIFAR-10 datasets. Finally, we summarize this paper in Section VI.

2 Related Work

The concept of federated learning was first proposed in [5], which showed its effectiveness through experiments on various datasets. Based on the comparison of synchronous and asynchronous methods of distributed gradient descent in [6], it is proposed that federated learning should use the synchronous approach because it is more efficient than asynchronous approaches..

In order to prevent information leakage, a popular approach is differential privacy (DP) [7] which adding artificial noise to the privacy information. Local differential privacy (LDP) [8] is a recently proposed approach which can provide strong guarantees of privacy to the users. Different from traditional differential privacy [7] which provides guarantee in data analysis part, LDP focuses on the privacy in data collection process. The authors in [9] considered distributed estimation at the server over uploaded data from clients while providing protections on these data with LDP. An algorithm for user-level differentially private training of large neural networks was proposed in [10]. The authors in [11] improved the computational efficiency of DP based SGD by tracking detailed information about the privacy loss. A novel DP based SGD algorithms was proposed in [12] and the authors also analyzed their performance bounds which were shown to be related to privacy levels and the sizes of datasets. The work in [13] proposed an FL algorithm with the consideration on preserving clients' privacy which can achieve good training performance at a given privacy level, especially when there is a sufficiently large number of participating clients.

Due to the limited edge device resources, it is a challenge to perform distributed machine learning on each client. To reduce the communication overhead, the work in [14] proposed Deep Gradient Compression (DGC) to greatly reduce the communication bandwidth. which explored model compression techniques for efficient communications However, data compression results in information loss of training updates, which may harm the learning accuracy and usually come with no convergence guarantees. The work in [15] proposed two ways to reduce the uplink communication costs. One is structured updates, where they directly learn an update from a restricted space parametrized using a smaller number of variables, the other is sketched updates, where they learn a full model update and then compress it using a combination of quantization, random rotations, and subsampling before sending it to the server. The work in [16] a client selection problem that determines which clients participate in the training process and when each client has to complete the process while considering the computation and communication resource constraints imposed by the client.

3 Federated Learning

In this section, we briefly introduce the original FL framework [5] in vehicle edge network. Then, we present the problems formulation that will be discussed in our following analysis.

3.1 Federated Learning

In the vehicle edge network, the amounts of mobile vehicles individually have data that they want to keep as a secret, such as the privacy information of vehicles for traffic prediction and cityscape images captured by autonomous vehicles. If a vehicle edge computing server collects all the distributed data, a high-performance machine learning model can be trained on these data. However, it is not acceptable for vehicles to reveal their data due to privacy issues.

Federated learning is a decentralized approach for training Machine Learning model that intent to solve the abovementioned problem. The training process of a FL system which is shown in Figure 1 contains the following four steps:

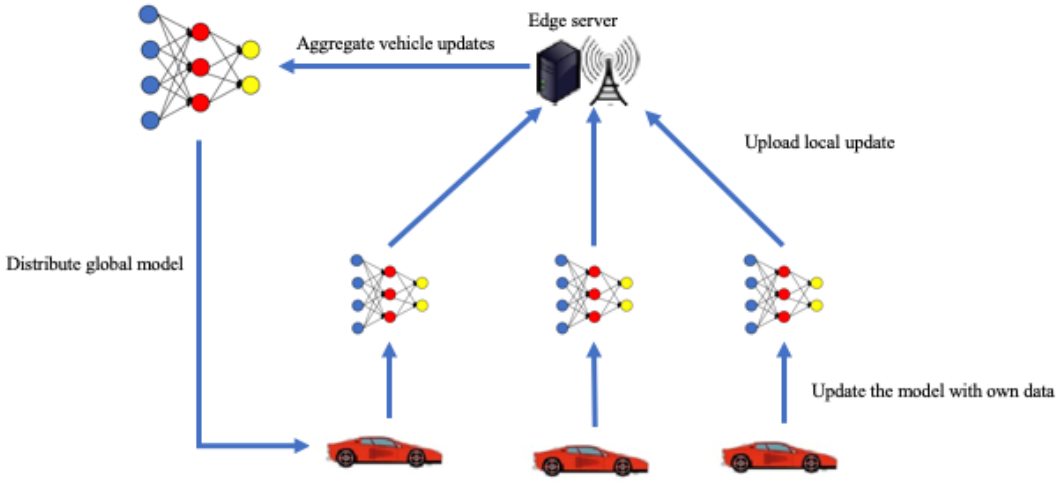

Figure 1. Federated learning in vehicle edge network

(1) Distribute global model. The edge server distributes the parameters of global model to the K vehicles.

(2) Update the model with own data. All the vehicles train the ML model with their own data locally.

(3) Upload local update. Each vehicle adds artificial noise into the updated parameters of the model and then upload to the edge server.

(4) Aggregate vehicle updates. The edge server performs secure aggregation over the uploaded parameters from K vehicles to obtain the global model and tests the performance of the model.

3.2 Problem Formulation

We consider applying federated learning in vehicle edge networks, while solving the security and privacy concerns. The set of vehicles is described as $V=\{v_1, v_2, v_3, \ldots, v_K\}$, and the set of whole training dataset is denoted as $D=\{D_1, D_2, D_3, \ldots, D_K\}$. Each vehicle v_k has its own dataset $D_K=\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x_1 is the input for models and y_i is the labels of x_i. The vehicle communicates with the vehicle edge server when it locals in the coverage area. The goal is to learn a global model over the training set D. Formally, let $n_k=|D_k|$ be the number of the sample in dataset D_k. For each vehicle v_k, its loss function is

$$\min_{\omega\in R^d} F_k(\omega)=\frac{1}{n_k}\sum_{i}^{n_k} f_i(\omega), \tag{1}$$

where $f_i(\omega)$ is the loss function for i-th data sample (x_i, y_i) with the model parameters ω.

The objective function $F(\omega)$ is defined as follow:

$$F(\omega)=\sum_{k}^{K}\frac{n_k}{n} f_k(\omega). \tag{2}$$

where $F(\omega)$ is the total loss function for the K dataset, $n=|D|$ is the number of the training set D.

In the vehicle edge network, the goal of the federated learning is to train a global model, which is an optimization problem to minimize $F(\omega)$. That is,

$$h(\omega)=\arg\min_{\omega\in\{\omega_t\}} F(\omega), \tag{3}$$

where ω_t is the parameter set of the aggregated model at round t, and ω_t is defined as follow:

$$\omega_{t+1}=\omega_t-\sum_{k=1}^{K}\frac{n_k}{n}u_{k,t} \tag{4}$$

where $u_{k,t}$ is the local update from vehicle v_k in the *t-th* round.

4 Secure Federated Learning with Efficient Communication Scheme

The goal of our federated learning is to obtain a set of optimal parameters for the model which minimizes the loss function with efficient communication. In this section, we introduce our proposed SFLEC in details and analyze the convergence property of SFLEC.

4.1 Add Gaussian Noise in Local Update

We establish the federated optimization in Eq. (1) through Stochastic Gradient Descent (SGD) [17], which is an effective optimization to minimizes the objective function $F(\omega)$ by iterating the local update. For a vehicle v_k, the goal of local training is to obtain the parameters ω_k of the model. In iteration t, a local model parameter $\omega_{k,t}$ is computing according to Eq. (5):

$$\omega_{k,t}=\omega_{k,t-1}-\eta_t\cdot\nabla F_k(\omega_{k,t})=\omega_{k,t-1}+u_{k,t} \tag{5}$$

where ∇F_k and η_t denote the gradient function and the learning rate, and $u_{k,t}$ denotes the local update of vehicle k in the t^{th} iteration given by $u_{k,t}=-\eta_t\cdot\nabla F_k(\omega_{k,t})$.

We adopt local differential privacy to protect the

privacy of the local update by adding noise to perturb the parameters, which is described as follow:

$$\tilde{u}_{k,t} = u_{k,t} + n_t, \tag{6}$$

$$\tilde{\omega}_{k,t} = \omega_{k,t-1} + \tilde{u}_{k,t}, \tag{7}$$

where n_t is the additive noise. In iteration t, before uploading the local update to edge computing server, vehicle v_k adds Gaussian noise in $u_{k,t}$ locally. To ensure that the noise distribution $n \sim N(0, \sigma^2)$ preserves (ε, δ)-DP, where N represents the Gaussian distribution, we set the noise scale, represented by the standard deviation of the additive Gaussian noise, as $\sigma = c\Delta s / \varepsilon$ where $\varepsilon \in (0,1)$ and the constant $c = \sqrt{21n(1.25/\delta)}$. And Δs is the sensitivity the function given by

$$\Delta s = \min_{D_k, D_k'} \| s(D_k) - s(D_k') \|, \tag{8}$$

In order to measure the sensitivity Δs, we assume that the batch size in the local training is equal to the number of the local dataset and then define the local training process in the k-th vehicle by

$$\begin{aligned} s(D_k) \triangleq u_k &= \omega_0 - \arg\min_{\omega} F_k(\omega, D_k) \\ &= \omega_0 - \frac{1}{n}\sum_{i=1}^{n_k} \arg\min_{\omega} F_k(\omega, D_k), \end{aligned} \tag{9}$$

where $n_k = |D_k|$ is number of the sample in the dataset D_k and $D_{k,i}$ is the i-th sample in D_k. For the upload process, using a clipping technique, we ensure that $\| u_k \| < U$ where u_k describes the local parameters update from the t k-th vehicle without perturbation and U is a clipping threshold for bounding u_k. Thus, the sensitivity Δs can be calculated by

$$\begin{aligned} \Delta s(D_k) &= \max_{D_k, D_k'} \| s(D_k) - s(D_k') \| \\ &= \max_{D_k, D_k'} \| \omega_0 - \frac{1}{n_k}\sum_{i=1}^{n_k} \arg\min_{\omega} F_k(\omega, D_{k,i}) \| \\ &\quad - (\omega_0 - \frac{1}{n_k}\sum_{i=1}^{n_k} \arg\min_{\omega} F_k(\omega, D'_{k,i})) = \frac{2U}{n_k} \end{aligned} \tag{10}$$

where compared with D_k, D_k' has the same size but only differ by one sample, $D'_{k,i}$ is the i-th sample in $D'_{k,i}$. From the above result, the sensitivity for all vehicles can be defined by

$$\Delta s \triangleq \max\{\Delta s(D_k)\}, \tag{11}$$

To obtain the sensitivity, we define the minimum size of the local datasets by

$$\mathrm{m} = \min\{n_k\}, \tag{12}$$

Thus, we obtain the sensitivity $\Delta s = \frac{2U}{n_{\min}}$ and then the noise scale $\sigma = \frac{2cU}{m\varepsilon}$.

4.2 Communication-Efficient Federated Learning

In order to improve communication efficiency for federated learning, we consider reducing the irrelevant local updates uploaded to edge computing server. Following this idea, each vehicle need to know the total optimization tendency in the global aggregation. To solve this problem, in each iteration, vehicles should compare their local updates with the aggregated global update to know whether their updates are relevant. However, it is a challenge that the global update cannot be known before all of local updates have been uploaded and aggregated in the current iteration.

To address this problem, Wang [18] proposes that the global update in the current iteration can be estimated by that in the previous iteration. Specifically, given two global updates $Update_t$ and $Update_{t+1}$, they describe that normalized difference as

$$\Delta Update_t = \frac{\| |Update_{t+1} - Update_t| \|}{\| Update_t \|}, \tag{13}$$

where $\|\cdot\|$ is the Euclidean norm of a given vector. This means that the smaller the normalized difference is, the less the two updates diverge from each other. They have verified their insight that the global update aggregated in the current iteration can be estimation for that to be aggregated in previous iteration.

Given the global update in the current iteration, we propose an efficient metric to measure the relevance of global update to local update. First, as a model update is actually a gradient vector of model parameter, we compute the total number of parameters of the same symbol in the two updates, which describes the percentage of same-sign parameters in the two updates. Then, we compute the sum of the relative ratio and normalize the result by the total number of parameters, which we will use to measure the relevance between the global update and the local update. Specifically, let $u_k = \{u_k^1, u_k^2, ..., u_k^N\}$ be the update of vehicle k over N model parameters. Let u describe the global update. We compute the relevance of local update u_k to global update u as

$$r_d(u_k, u) = \frac{1}{N}\sum_{I}^{N} I(\mathrm{sgn}(u_k^i) = \mathrm{sgn}(u_k^i)), \tag{14}$$

$$r_s(u_k, u) = \frac{1}{N}\sum_{I}^{N} \frac{u_k^i}{u^i} \cdot I(\mathrm{sgn}(u_k^i) = \mathrm{sgn}(u^i)), \tag{15}$$

where $I(\text{sgn}(u_k^i) = \text{sgn}(u^i)) = 1$ describes the sign of u_k^i and ω are same, and $\frac{u_k^i}{u^i}$ describes the relative ratio between the update of vehicle k and the global update.

Intuitively, the sign of a parameter in the update describes the direction to which the model should be trained along the dimension of that parameter. And the relative ratio between the local update and the global update determines the speed of parameter change in one iteration for a model.

Based on this idea, our approach can find relevant local update and excludes those irrelevant local updates. An update is considered relevant if relevance measure (14) is more than a predefined threshold. On the meantime, in order to ensure the stability of the model, the relative ratio (15) should smaller than a predefined threshold.

Algorithm 1 describes the process of Communication-Efficient Federated Learning with Client Selection.

Algorithm 1. Secure Federated Learning with Efficient Communication

1. Global Aggregate:
2. Input: Vehicle set $V = \{v_1, v_2, v_3, \ldots, v_K\}$
3. Initialize the global model ω_0 and the global update u_0
4. for each iteration $t = 1, 2, ..., T$ do
5. for all vehicle $v_k \in V$ do in parallel
6. $u_{k,t} \leftarrow \text{LocalUpdate}\ (k, u_{t-1})$
7. $S_t \leftarrow \{u_{k,t}\}$
8. $u_t = \frac{1}{|S_t|}\Sigma_{u_{k,t}} u_{k,t}$
9. $\omega_t = \omega_{t-1} + u_t$
10. Local Update:
11. Input Vehicle index k, Global Model ω_{t-1} and Global Update u_{t-1}
12. Execute the local training and update the local
13. parameters $\omega_{k,t}$ as
14. $\omega_{k,t} = \arg\min_{\omega_k}(F_k(\omega_t) + \frac{\mu}{2}\|\omega_k - \omega_{t-1}\|^2)$
15. Calculate the local update $u_{k,t}$ as
16. $u_{k,t} = \omega_{t-1} - \omega_{k,t}$
17. Calculate $r_d(u_{k,t}, u_{t-1})$ and $r_s(u_{k,t}, u_{t-1})$
18. Follow Eq.(8) and Eq.(9)
19. if $r_d(u_{k,t}, u_{t-1}) < \alpha(t)$ then
20. if $r_s(u_{k,t}, u_{t-1}) < \beta(t)$ then
21. Add noise $\tilde{u}_{k,t} = u_{k,t} + n_{k,t}$
22. return $\tilde{u}_{k,t}$ to server

4.3 Convergence Analysis on SFLEC

In this section, we focus on deriving the convergence property of out method under the (ε, δ)-DP requirement and show that the proposed SFLEC convergence to a global optimum.

We make the following assumptions on the function $F_1, F_2, \ldots, F_K$ where $K = |S|$ and S is a set of vehicles which would upload the local update to the edge server. And we defined the global loss function $F(\omega)$ by $F(\omega) = \Sigma_k^K \frac{F_k(\omega)}{K}$.

Assumption 1: $F_k(\omega)$ is convex and satisfies the Polyak-Lojasiewicz condition with positive parameter l: for all ω, $F(\omega) - F(\omega^*) \le \frac{1}{2l}\|\nabla F(\omega)\|^2$, and $F(\omega_0) - F(\omega^*) = C$, where ω^* is the optimal result.

Assumption 2: $F_k(\omega)$ *is* γ *-Lipschitz: for any* ω, ω', $\|F_k(\omega) - F_k(\omega')\| \le \gamma\|\omega - \omega'\|$.

Assumption 3: $F_k(\omega)$ is ρ-Lipschitz: for any ω, ω', $\|\nabla F_k(\omega) - \nabla F_k(\omega')\| \le \rho\|\omega - \omega'\|$; where ρ is a constant determined by practical loss function.

Assumption 4: *For any k and* ω, $\|\nabla F_k(\omega) - \nabla F(\omega)\| \le \varepsilon_k$, *where* ε_k *is the divergence metric.*

According to assumption 1, 2, 3 and 4, we put forward the following lemma.

Lemma 1 (C-dissimilarity of various Clients): for any ω*, there exits C satisfying*

$$\frac{\mathbb{E}\{\|\nabla F_k(\omega)\|^2\}}{\mathbb{E}\{\|\nabla F(\omega)\|^2\}} \le C^2, \tag{16}$$

According to the assumption 4 of divergence metric, we proposed the Lemma 1 which shows the statistical heterogeneity of all vehicles. To analyze the convergence property of our scheme, we first use the Lemma 2 which has been proved in [19] to obtain that the expected difference in the loss function between adjacent global aggregation has an upper bound.

Lemma 2 (Expected Increment in the Loss Function): In the (t+1)-th iteration, after receiving updates, the expected difference in the loss function can be upper-bounded by

$$\begin{aligned}\mathbb{E}\{F(\tilde{\omega}_{t+1}) - F(\tilde{\omega}_t)\} &\le \lambda_2\mathbb{E}\{\|\nabla F(\tilde{\omega}_t)\|^2\} \\ &+ \lambda_1\mathbb{E}\{\|n_{t+1}\|\|\nabla F(\tilde{\omega}_t)\|\} + \lambda_0\mathbb{E}\{\|n_{t+1}\|^2\}\end{aligned} \tag{17}$$

where $\lambda_0 = \frac{\rho}{2}$, $\lambda_1 = \frac{1}{\mu} + \frac{\rho c}{\mu}$, $\lambda_2 = \frac{1}{\mu} + \frac{\rho c}{\mu^2}, \frac{\rho c^2}{2\mu^2}$, and n_t is the noise term imposed on the parameters given by $n_t = \sum_i^K \frac{1}{K} n_{i,t}$ in the t-th iteration.

In lemma 2, the additive noise n_t satisfies the following Gaussian distribution $n \sim N(0, \sigma^2)$, and we can obtain $\sigma = \frac{2cU}{m\varepsilon}$ from Section A. Next, we will analyze the convergence property of our proposed SFLEC.

Theorem 1: In the t-th iteration, the convergence upper bound of our scheme with the protection level ε is defined by

$$\text{c} \tag{18}$$

where $R = 1 + 2l\lambda_2$ *and* $Q = \frac{\lambda_1 rcU}{m\varepsilon}\sqrt{\frac{2}{\pi}} + \frac{\lambda_0 c^2 U^2}{Km^2\varepsilon^2}$.

As we can see from the theorem 1, the convergence of Algorithm 1 relies on the setting of the privacy protection level ε. By increasing the ε, which means that relaxing the privacy level, the performance of SFLEC algorithm will improve.

5 Evaluation

In order to show how our scheme works effectively, we simulate a MEC environment and conduct experiments of Machine Learning tasks using public datasets. We evaluate the reduced communication overhead by applying our approach to basic FL [5] via simulations. And we compare SFLEC against baseline: basic FL. In order to provide comparability with basic FL, we choose a machine learning model and adopt two realistic object classification tasks using publicly available large-scale image datasets which are MNIST and CIFAR-10.

In out simulation, we adopt pytorch to build model and use a 40-client to simulate edge vehicles in practiced FL. In order to do a thorough analysis for SFLEC and basic FL despite of the influence of the threshold, we test various threshold values to identify the relevance and relative ratio of local updates and choose the threshold values with the best performance for training model. Specifically, as shown in Figure 2 and Figure 3, we test a set of 13 relevance threshold values for SFLEC: {0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9}, and another set of 13 relative ratio threshold values: {0.05, 0.1, 0.2, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.8, 0.9} in both two datasets. For MNIST, in the 800 rounds, the best performance is obtained when setting the relevance threshold value as 0.8 and the relative ratio threshold value as 0.65. And, these two values are tuned as 0.7 and 0.5 for the CIFAR-10 dataset to get the best performance.

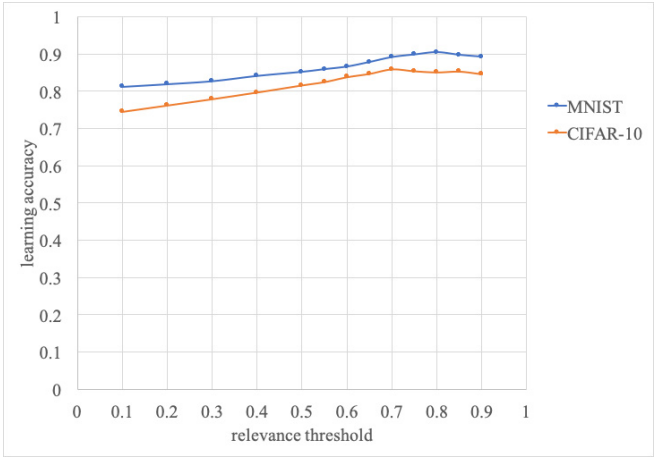

Figure 2. Comparison of accuracy under different relevance threshold on MNIST and CIFAR-10

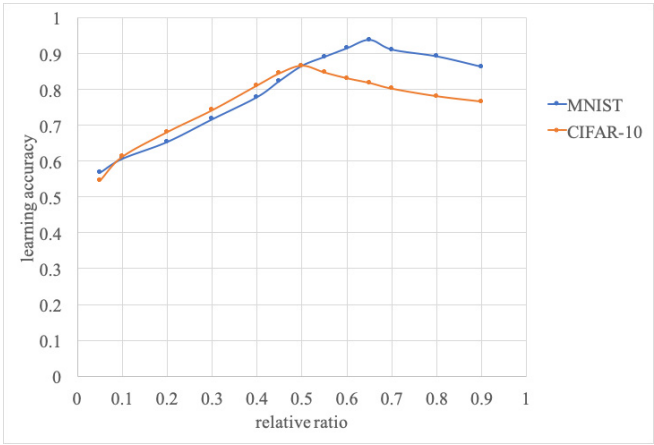

Figure 3. Comparison of accuracy under different relative ratio on MNIST and CIFAR-10

We first compare the learning accuracy of both basic FL and SFLEC on MNIST datasets. Specifically, In Figure 4, we observe that the accuracy of SFLEC is more than the basic FL in the same accumulated communication rounds，and when the rounds reach 800, the accuracy of our scheme is statistically good with an average value of 0.938, but the accuracy of basic FL only raises to 0.809 which means this model still needs training. Further, we measure the accuracy of both basic FL and SFLEC on CIFAR-10 datasets. In Figure 5, the result also shows that the accuracy of basic FL is less than our scheme under the same accumulated communication rounds.

In particular, for MNIST, when the learning accuracy raises to 60%, the basic FL costs 400 communication rounds as shown in Table 1. On the other hand, our scheme substantially reduces the required communication rounds to 110, providing a saving of 3.63. Furthermore, when the learning accuracy reaches nearly the highest value, i.e., 80%,

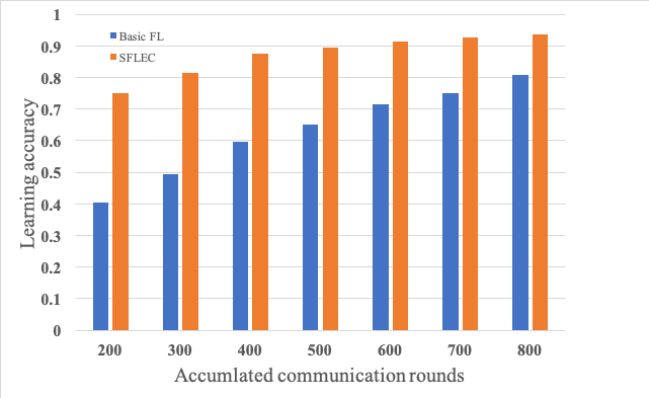

Figure 4. The learning accuracy of SFLEC compared to basic FL on MNIST

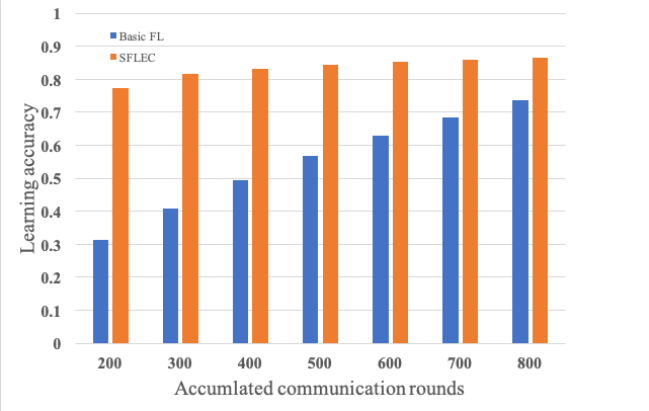

Figure 5. The learning accuracy of SFLEC compared to basic FL on CIFAR-10

Table 1. Summary of the rounds for different learning accuracies in MNIST and CIFAR-10

	Basic FL	SFLEC
MNIST 60% accuracy	400	110
MNIST 80% accuracy	800	225
CIFAR-10 60% accuracy	584	144
CIFAR-10 80% accuracy	952	232

the basic FL take 800 rounds, respectively. Our SFLEC costs only 225 rounds, reducing the network footprints by 3.55x. For the CIFAR-10 datasets, the communication overhead increases in basic FL and SFLEC. The basic FL uses 584 rounds to obtain a training model with the accuracy 60%. However, SFLEC provides the saving 4.05 with the relevance threshold tuned as 0.7 and the relative ratio threshold tuned as 0.5, reducing the required number of communication rounds to 144. Moreover, our scheme reduces the communication rounds from 952 to 238 when requiring the learning accuracy as 80% with the saving of 4.11.

In summary, our scheme consistently outperforms basic FL in improving the communication efficiency for FL under various learning accuracies. As we see in Table 1, SECFL keeps outperforms basic FL by more than 3.5x in MNIST and more than 4.0x in CIFAR-10.

6 Conclusion

In this article, we proposed a Secure Federated Learning with Efficient Communication scheme for edge computing in vehicular networks. To protect the updated models of each vehicle, we added artificial noise in local training process with local differential privacy. Due to uploading the local updates of each vehicle will cause excessive communication overhead, we proposed a client selection approach that identifies relevant updates trained by vehicles and prevents irrelevant updates from being uploaded for reduced network footprint to achieve efficient communication for federated learning. We have shown in theory that our scheme is guaranteed to converge. Evaluations demonstrate that our scheme reduces the network footprint compared with the basic FL.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NO. 61772064) and the National Key Research and Development Program of China, grant number 2018YFC0831900.

Reference

[1] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics, *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 3, pp. 2134-2143, March, 2020.

[2] K. Zhang, Y. Mao, S. Leng, S. Maharjan, Y. Zhang, Optimal Delay Constrained Offloading for Vehicular Edge Computing Networks, *IEEE International Conference on Communications*, Paris, France, 2017, pp. 1-6.

[3] T. Q. Dinh, Q. D. La, T. Q. S. Quek, H. Shin, Learning for Computation Offloading in Mobile Edge Computing, *IEEE Transactions on Communications*, Vol. 66, No. 12, pp. 6353-6367, December, 2018.

[4] T. X. Tran, D. Pompili, Joint Task Offloading and Resource Allocation for Multi-server Mobile-edge Computing Networks, *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 1, pp. 856-868, January, 2019.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient Learning of Deep Networks from Decentralized Data, *20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, America, 2017, pp. 1273-1282.

[6] M. Teng, F. Wood, Bayesian Distributed Stochastic Gradient Descent, *Advances in Neural Information Processing Systems*, Montreal, Canada, 2018, pp. 6380-6390.

[7] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, *Foundations and Trends in Theoretical Computer Science*, Vol. 9, No. 3-4, pp. 211-407, August, 2014.

[8] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, K. Ren, Generating Synthetic Decentralized Social Graphs with Local Differential Privacy, *2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, TX, USA, 2017, pp. 425-438.

[9] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, W. Yang, Local Differential Private Data Aggregation for Discrete Distribution Estimation, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 30, No. 9, pp. 2046-2059, September, 2019.

[10] H. Brendan McMahan, D. Ramage, K. Talwar, L. Zhang, Learning Differentially Private Recurrent Language Models, *6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018, pp. 1-14.

[11] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, L. Zhang, Deep Learning with Differential Privacy, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, 2016, pp. 308-318.

[12] N. Wu, F. Farokhi, D. Smith, M. Ali Kaafar, *The Value of Collaboration in Convex Machine Learning with Differential Privacy*, http://apxiv.org/abs/1906.09679, June, 2019.

[13] R. C. Geyer, T. Klein, M. Nabi, *Differentially Private Federated Learning: A Client Level Perspective*, http://apxiv.org/abs/1712.07557, December, 2017.

[14] Y. Lin, S. Han, H. Mao, Y. Wang, W. J. Dally, Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training, *6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018, pp. 1-14.

[15] J. Konecnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, D. Bacon, *Federated Learning: Strategies for Improving Communication EFficiency*, https://arxiv.org/abs/1610.05492, October, 2016.

[16] T. Nishio, R. Yonetani, *Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge*, https://arxiv.org/abs/1804.08333, April, 2018.

[17] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, *Nature*, Vol. 521, No. 7553, pp. 436-444, May, 2015.

[18] L. P. Wang, W. Wang, B. Li, Cmfl: Mitigating Communication Overhead for Federated Learning, *IEEE 39th International Conference on Distributed Computing Systems*, Dallas, TX, USA, 2019, pp. 954-964.

[19] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, H. V. Poor, Federated Learning with Differential Privacy: Algorithms and Performance Analysis, *IEEE Transactions on Information Forensics and Security*, Vol, 15, pp. 3454-3469, April, 2020.

Biographies

Yinglong Li received the B.S. degree in communication engineering from Chongqing University in 2017. Since 2019, he has been studying for his doctor's degree in information and communication engineering at School of Electronic and Information Engineering. His research interesting includes edge computing and privacy present protection.

Zhenjiang Zhang received the Ph.D. degree in communication and information systems from Beijing Jiaotong University. He has been a Professor in BJTU since 2014. He is currently served as the vice dean of School of Software Engineering in BJTU. His research interests include cognitive radio, and wireless sensor networks.

Zhiyuan Zhang received his Ph.D. degree in Beijing Jiaotong University. Currently, he is a teacher in the School of Electronics and Information Engineering at Beijing Jiaotong University. His current research fields include social network analysis, recommender system, machine learning, data mining and artificial intelligence.

Yi-Chih Kao is currently the deputy director of the Information Technology Service Center at National Chiao Tung University (NCTU). He received his Ph.D. degree in Industrial Engineering and Management from NCTU. He is also certified in both Cisco CCIE and EC-Council CEH. His research interests include cyber security, network performance, software-defined networking, and IT service design.

Appendix A

Proof of Lemma 1

According to the assumption 4, we have

$$\mathbb{E}\{\| \nabla F_k(\omega) - \nabla F_k(\omega) \|^2\} \le \mathbb{E}\{\varepsilon_k{}^2\} \tag{19}$$

and

$$\begin{aligned} &\mathbb{E}\{\| \nabla F_k(\omega) - \nabla F_k(\omega) \|^2\} \\ &= \mathbb{E}\{\| \nabla F_k(\omega) \|^2\} + 2\mathbb{E}\{\nabla F_k(\omega)^T\} \nabla F_k(\omega) + \| \nabla F_k(\omega)^2 \|^2 \\ &= \mathbb{E}\{\| \nabla F_k(\omega) \|^2\} + \| \nabla F_k(\omega) \|^2 . \end{aligned} \tag{20}$$

Substituting (20) into (19) and $\nabla F(\omega) = \mathbb{E}\{\nabla F_k(\omega)\}$, we obtain

$$\mathbb{E}\{\| \nabla F_k(\omega) \|^2\} \le \| \nabla F(\omega) \|^2 + \mathbb{E}\{\varepsilon_k{}^2\}. \tag{21}$$

When $\| \nabla F(\omega) \| \ne 0,$ we have

$$\begin{aligned} \| \nabla F_k(\omega) \|^2 \ \mathbb{E}\{\varepsilon_k{}^2\} &= \| \nabla F(\omega) \|^2 \ (1 + \frac{\mathbb{E}\{\varepsilon_k{}^2\}}{\| \nabla F(\omega) \|^2}) \\ &= \| \nabla F(\omega) \|^2 \ C(\omega)^2, \end{aligned} \tag{22}$$

where we set

$$C(\omega) = \sqrt{1 + \frac{\mathbb{E}\{\varepsilon_k{}^2\}}{\| \nabla F(\omega) \|^2}}. \tag{23}$$

Observing (23), we can notice that when $C(\omega)$ approach 1, $\mathbb{E}\{\varepsilon_k{}^2\}$ will approach 0, which means that the local loss functions are similar with the global loss function. When all the local loss functions are the same, then $C(\omega) = 1.$. Therefore, we can obtain

$$\frac{\mathbb{E}\{\| \nabla F_k(\omega) \|^2\}}{\mathbb{E}\{\| \nabla F_k(\omega) \|^2\}} \le C^2, \tag{24}$$

where $\mathbb{E}\{\| \nabla F(\omega) \|^2\} = \| \nabla F(\omega) \|^2$ and C is the upper bound of $C(\omega)$. Hence, the Lemma 1 has been proved.

Appendix B

Proof of Theorem 1

We assume that F satisfies the Polyak-Lojasiewicz inequality with positive l, which means that

$$\mathbb{E}\{\nabla F(\tilde{\omega}_t) - F(\omega^*)\} \le \frac{1}{2l} \| \nabla F(\tilde{\omega}_t) \|^2, \tag{25}$$

where $F(\omega^*)$ is the loss function corresponding to the optimal parameters ω^*. Then, subtract $\mathbb{E}\{F(\omega^*)\}$ in both sides of (17), we obtain

$$\begin{aligned} &\mathbb{E}\{F(\tilde{\omega}_{t+1}) - F_k(\omega^*)\} \\ &\le \mathbb{E}\{F(\tilde{\omega}_t) - F(\omega^*)\} + \lambda_2 \mathbb{E}\{\| F(\tilde{\omega}_t) \|^2\} \\ &+ \lambda_1 \mathbb{E}\{\| n_{t+1} \| \| \nabla F(\tilde{\omega}_t) \|^2\} + \lambda_0 \mathbb{E}\{\| n_{t+1} \|^2\}, \end{aligned} \tag{26}$$

Considering $\| \nabla F(\omega_t) \| \le \gamma$ and (18), we know

$$\begin{aligned} &\mathbb{E}\{F(\tilde{\omega}_{t+1}) - F_k(\omega^*)\} \\ &\le \mathbb{E}(1 + 2l\lambda_2) \mathbb{E}\{F(\tilde{\omega}_t) - F(\omega^*)\} \\ &+ \lambda_1 \gamma \mathbb{E}\{\| n_{t+1} \| \| \nabla F(\tilde{\omega}_t) \|\} + \lambda_0 \mathbb{E}\{\| n_{t+1} \|^2\}, \end{aligned} \tag{27}$$

Due to the same and independent distribution of additive noise terms, we define $\mathbb{E}\{\| n_t \|\} = \mathbb{E}\{\| n \|\}$ and $\mathbb{E}\{\| n_t \|^2\} = \mathbb{E}\{\| n \|^2\}$, for $0 \le t \le T$. Applying recursion to (19), we obtain

$$\begin{aligned} &\mathbb{E}\{F(\tilde{\omega}_T) - F_k(\omega^*)\} \\ &\le (1 + 2l\lambda_2)^T \mathbb{E}\{F(\omega_0) - F(\omega^*)\} \\ &+ (\lambda_1 \gamma \mathbb{E}\{\| n \|\| + \lambda_0 \mathbb{E}\{\| n \|^2\}) \sum_{t=0}^{T-1} (1 + 2l\lambda_2)^t \\ &= (1 + 2l\lambda_2)^t \mathbb{E}\{F(\omega_0) - F(\omega^*)\} \\ &+ (\lambda_1 \gamma \mathbb{E}\{\| n \|\| + \lambda_0 \mathbb{E}\{\| n \|^2\}) \frac{(1 + 2l\lambda_2)^T - 1}{2l\lambda_2}. \end{aligned} \tag{28}$$

Considering $\sigma = \frac{2cU}{m\varepsilon}$, we can obtain

$$\begin{aligned} \mathbb{E}\{\| n \|\} &= \frac{cU}{m\varepsilon} \sqrt{\frac{2}{\pi}} \\ \mathbb{E}\{\| n \|^2\} &= \frac{c^2 U^2}{K m^2 \varepsilon^2}, \end{aligned} \tag{29}$$

Substituting (21) into (20), setting $F(\omega_0) - F(\omega^*) = C$, we have

$$\begin{aligned} &\mathbb{E}\{F(\tilde{\omega}_T) - F_k(\omega^*)\} \\ &\le (1 + 2l\lambda_2)^T C \\ &+ (\frac{\lambda_1 \gamma cU}{m\varepsilon} \sqrt{\frac{2}{\pi}} + \frac{\lambda_0 c^2 U^2}{K m^2 \varepsilon^2}) \frac{(1 + 2l\lambda_2)^T - 1}{2l\lambda_2} \\ &= \frac{Q}{R-1} + R^T (C - \frac{Q}{R-1}), \end{aligned} \tag{30}$$

where $R = 1 + 2l\lambda_2$, $Q = \frac{\lambda_1 rcU}{m\varepsilon} \sqrt{\frac{2}{\pi}} + \sigma = \frac{\lambda_0 c^2 U^2}{k m^2 \varepsilon^2}$. Hence, the Theorem 1 has been proved.