MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Improve Federated Learning Stability for Vehicle Trajectory Prediction

Sun, Youbang; Guo, Jianlin; Parsons, Kieran; Nagai, Yukimasa

TR2024-094 July 09, 2024

Abstract

The crowdsourced information is useful to calibrate Advanced Driver Assistance Systems/Autonomous Driving (ADAS/AD) parameters for automated and autonomous vehicles. However, learning such information in vehicular networks is challenging. On the one hand, data collected by individual vehicle may be not sufficient to train a large scale machine learning model. On the other hand, uploading raw data to cloud server is likewise impractical due to enormous communication bandwidth requirement and data privacy threat. This paper seeks a solution by applying federated learning (FL). We aim to improve FL algorithm stability to increase prediction accuracy. Accordingly, we propose a variance-based and structure-aware FL (VSFL), in which a variance-based model aggregation method is introduced for FL server to make optimal model aggregation and a structureaware model training scheme is provided for vehicle clients to tackle statistical heterogeneity without compromising performance. We first provide theoretical analysis for the proposed VSFL. We then validate the effectiveness of VSFL algorithms on vehicle trajectory prediction using both synthetic data and real data.

International Conference on Ubiquitous and Future Networks (UCIFN) 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Improve Federated Learning Stability for Vehicle Trajectory Prediction

Youbang Sun[1,3], Jianlin Guo[1], Kieran Parsons[1], Yukimasa Nagai[2]
[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
[2]Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa, Japan
[3]Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02215, USA

***Abstract*— The crowdsourced information is useful to calibrate Advanced Driver Assistance Systems/Autonomous Driving (ADAS/AD) parameters for automated and autonomous vehicles. However, learning such information in vehicular networks is challenging. On the one hand, data collected by individual vehicle may be not sufficient to train a large scale machine learning model. On the other hand, uploading raw data to cloud server is likewise impractical due to enormous communication bandwidth requirement and data privacy threat. This paper seeks a solution by applying federated learning (FL). We aim to improve FL algorithm stability to increase prediction accuracy. Accordingly, we propose a variance-based and structure-aware FL (VSFL), in which a variance-based model aggregation method is introduced for FL server to make optimal model aggregation and a structure-aware model training scheme is provided for vehicle clients to tackle statistical heterogeneity without compromising performance. We first provide theoretical analysis for the proposed VSFL. We then validate the effectiveness of VSFL algorithms on vehicle trajectory prediction using both synthetic data and real data.**

I. Introduction

The recent advancement of technology has drastically changed modern vehicles, using a variety of sensors and computational resources, the amount of data captured and the computing capabilities in vehicles have been greatly increased. This has enabled the application of machine learning (ML) algorithms to analyze and learn from the captured data. Vehicular ML takes advantage of the various data collected by sensors and seeks to solve problems related to vehicles such as trajectory prediction [1].

Similar to other ML tasks, vehicular ML is mostly studied for centralized algorithms. In centralized ML, the algorithm requires data available at a central server that requires vehicles to send their raw data to a central server, which is highly impractical due to extensive privacy threats and enormous communication overhead.

Different from traditional distributed ML algorithms, the recently introduced FL termed FederatedAveraging (FedAvg) [2] enables multiple client devices and servers to train a ML model collaboratively without sharing their data. In FL, each of the client devices calculates its local update based on the local data, and the parameters, instead of data, are sent to the central server, the server then calculates a global parameter update as an aggregation of all local parameter updates. The updated global parameters are then sent back to the client devices. This approach protects data privacy, improves communication efficiency and train more robust models in some scenarios.

We state our contributions as follows. Unlike the state-of-the-art FL algorithms that mainly focus on model training, the proposed VSFL tackles both model aggregation and model training by considering fact that FL is collaboratively performed by learning server and learning clients. We first provide theoretical analysis on the variance-based model aggregation and then address the structural heterogeneous issues commonly existing in model training. We evaluate the effectiveness of the proposed VSFL methods on vehicle trajectory prediction using both synthetic data and real data.

The paper is organized as follows. The related works are provided in Section II. The accelerated FL setup is presented in Section III. We introduce the proposed VSFL in Section IV. The effectiveness of the VSFL algorithms are validated in Section V. Lastly, Section VI concludes our paper.

II. Related Works

We have conducted extensive literature survey. This section presents the works of particular interest.

A. Vehicle Trajectory Prediction

The trajectory prediction task mainly uses HD maps and previous vehicle trajectories as the input information to predict future trajectory. Early works such as multimodal trajectory prediction [3] are motivated by computer vision techniques, and use modern convolutional neural networks (CNNs) to extract raster features. Although intuitive, these approaches suffer from computational complexity and information loss. Recent works have proposed another novel approach to extract and encode the map information into nodes in a graph, this line of work is able to achieve state-of-the-art performances while using less parameters. Of particular relevance to this paper is the Prediction via Graph-based Policy (PGP) [1], apart from encoding map information with graphs, PGP also uses a graph-based policy model for trajectory prediction.

B. Federated Learning

The advantages of Federated Learning [2] include improved communication efficiency and privacy protection. Additionally, to address system heterogeneity and statistical heterogeneity and improve the performance of FL, the novel optimization methods have been proposed. FedProx [4] introduces an additional regularization into the local clients to prevent clients from straying far from each other.

This work was done while Youbang Sun was working at Mitsubishi Electric Research Laboratories (MERL) as an intern.

Personalized FL method SCAFFOLD [5] is able to produce a global model that do not suffer from the "client drift" by introducing control variables. FL has been used in autonomous driving tasks such as road actor classification [6] and traffic sign detection/classification [7].

C. Algorithmic Stability and Generalization Error

The works of FL have rarely studied its properties in algorithmic stability and generalization error, two equivalent notions related to prediction accuracy of the ML algorithms. The algorithmic stability was first introduced into FL in [8]. Generally, the algorithmic stability provides a bound on the algorithm output given fluctuations of the input. The generalization error denotes the expected difference between the errors a ML model evaluated on training set and a new data point. The exact formulations and relationships are provided in [9], [10], [11].

FedProx [4] and SCAFFOLD [5] are two of recent works that aim to improve the stability and robustness of FL algorithms. However, the advantages in these algorithms are not consistent. The extensive empirical experiments [12] show that in homogeneous setting, these algorithms in fact exhibit worse performance compared to vanilla FedAvg [2].

III. Accelerated Federated Learning

This section provides a brief introduction to FL with non-IID clients. Additionally, we adapt a momentum-based accelerated FL optimizer in contrast to the standard gradient descent scheme used in vanilla FedAvg [2].

FL was first introduced as a communication-efficient and privacy-preserving algorithm to solve optimization problems in a distributed fashion. For a network of total n clients, we denote the dataset possessed by the i-th client as D_i. Thus, $D := \cup_{i=1}^{n} D_i$ is the global dataset.

The centralized learning seeks to find a set of model parameters x that minimizes the loss function $l(x, D)$ for all clients

$$\arg\min_{x} l(x, D). \tag{1}$$

However, the problem (1) requires all data available at the server, which is impractical for vehicular tasks.

In the decentralized learning, each of the local clients optimizes over its own local version of the objective, while the server finds consensus among all clients. The equivalent decentralized version of problem (1) can be written as

$$\begin{aligned} &\arg \min_{x_1,\ldots,x_n} \sum_{i=1}^{n} l(x_i, D_i), \\ &\text{subject to:} \quad x_i = x_j \text{ initialy}, \quad \text{for all} \quad i, j. \end{aligned} \tag{2}$$

FL algorithm is executed as follows. In the round t, the global model parameters of last round $x_{global}^{(t-1)}$ are distributed from server to all clients, and each client tries to find a local optimizer of the algorithm $x_i^{(t)} = \arg\min_{x_i} l(x_i, D_i)$ using the received $x_{global}^{(t-1)}$ as a starting point. In order to reduce communication, it is common for FL clients to perform multiple optimization steps using the local objective as an approximation of the global objective. After local computation is completed, the server collects updated parameters from clients and aggregates them as the updated global parameters x_{global}^{t}. The server often selects a subset $\mathcal{C}_t$ of n_t clients to participate in aggregation. The server aggregation typically takes the form of weighted average over a simplex $\mathbf{p} = (p_1, ..., p_{n_t})$ as

$$x_{global}^{(t)} = \sum_{i \in \mathcal{C}_t} p_i x_i^{(t)}. \tag{3}$$

The algorithm then proceeds to next round. It can be seen that the determination of $\mathbf{p}$ becomes the key in model aggregation.

We adapt the Adam optimizer [13] in the local client with client optimizer restart. We note that the use of adaptive optimizers in local clients is not solely motivated by the superior empirical performance, the client optimizer state also becomes useful in our algorithm presented in Section IV-A.

IV. Variance-Based and Structure-Aware FL

This section presents the proposed variance-based and structure-aware FL (VSFL).

A. Variance-Based Model Aggregation

1) Mathematical Model: Since FL algorithms, especially those using momentum-based solvers such as Adam, are difficult to analyze directly. We simplify the problem by considering the problem of finding the mean of a Gaussian random vector using data from all clients.

We denote the individual data as $d_{i,j} \in D_i \sim \mathcal{N}(\mu, \sigma_i^2 I_d)$, where μ is the expectation of the distribution and is assumed to be the same across all clients, while σ_i is the standard deviation and σ_i^2 is the variance of the distribution. The objective for the server is to run an FL algorithm to find the best estimation of μ.

When the clients are homogeneous (the datasets D_i follow the same distribution), setting the dataset size based aggregations weights in (3) as

$$p_i = \frac{|D_i|}{\sum_{j \in \{n\}} |D_j|} \tag{4}$$

yields optimal results in terms of excess risk [8]. However, when the data distribution of clients are heterogeneous, finding the optimal aggregation weight is challenging.

Assumption 1: We assume that for client i, the local dataset D_i follows a distribution $\mathcal{D}_i$, where

$$\begin{aligned} &\mathbb{E}_{D_i \sim \mathcal{D}_i}[\nabla l(x, D_i)] = \mathbb{E}_{D_j \sim \mathcal{D}_j}[\nabla l(x, D_j)], \\ &\qquad\qquad \text{for all } i \text{ and } j, \\ &\mathit{Var}_{D_i \sim \mathcal{D}_i}(\nabla l(x, D_i)) = \sigma_i^2 I_d. \end{aligned} \tag{5}$$

Assumption 1 assumes that the gradient evaluated at different clients i share the same expectation, yet the variance of the gradient varies across clients. This assumption is especially common in vehicular data, since the traffic dynamics on the road typically stays same, yet the data captured by different sensors tend to have different quality, therefore causing different variances in data.

The variance-based model aggregation algorithm seeks to minimize the squared error

$$||x_{estimate} - x_{true}||^2 := \frac{\sum_{i=1}^{n} |D_i| \mathbb{E}_{d \in D_i} ||x_{estimate} - d||^2}{\sum_{i=1}^{n} |D_i|},$$

where the estimated mean is denoted by $x_{estimate}$. The global estimation is calculated by the $\mathbf{p}$-average method using simplex $\mathbf{p} = (p_1, ..., p_n)$. We denote the global estimation of x as $x_{global} = \sum_{i=1,...,n} p_i x_i$, hence the solution of the problem is calculated as follows,

$$x_{global} = \sum_{i=1}^{n} p_i x_i = \sum_{i=1}^{n} p_i \frac{\sum_{j=1}^{|D_i|} d_{i,j}}{|D_i|}. \tag{6}$$

In this case, we can calculate algorithmic stability by studying generalization error. We present the following theorem with the proof provided in Appendix.

Theorem 1: For a task that satisfy Assumption 1 where the estimated mean is calculated by (6), the generalization error of x_{global} with respect to the global dataset D, denoted as $gen(\mu, x_{global}|\{D\})$, is minimized when the weight simplex $\mathbf{p} = (p_1, ..., p_n)$ takes the following value

$$p_i = \frac{|D_i|/\sigma_i^2}{\sum_{j=1}^{n} |D_j|/\sigma_j^2}. \tag{7}$$

This theorem states that in order to minimize generalization error, the optimal averaging weight is proportional to the local dataset size and inversely proportional to the variance of the local dataset.

This result also follows intuition, a dataset with less variance in its data distribution appears more stable, and can be relatively more trusted, thus the dataset has a heavier averaging weight in aggregation.

2) Estimating Variance of clients: For the case of Gaussian variables with given variance, Theorem 1 ensures the best-case averaging weights that guarantee best possible algorithmic stability. In the sense of FL algorithms, the analysis becomes much more difficult. Motivated by the theoretical justification of Theorem 1, we then seek to find an estimation of the variance in dataset.

Unfortunately for most modern ML algorithms, calculating the variance of gradients induced by empirical risk minimization is inefficient, especially for large dataset D_i. A recent work by [14] introduced a new interpretation for the Adam optimizer, where the notion of variance and relative variance of Adam has been introduced. The k-th iteration of Adam is calculated as follows

$$\begin{aligned} g^{(k)} &= \nabla_x l(x^{(k)}), \\ m^{(k)} &= (\beta_1 m^{(k-1)} + (1-\beta_1) g^{(k)})/(1-\beta_1^k), \\ v^{(k)} &= (\beta_2 v^{(k-1)} + (1-\beta_2)(g^{(k)})^2)/(1-\beta_2^k), \\ x_i^{(k+1)} &= x^{(k)} - \alpha m^{(k)}/(\sqrt{v^{(k)}} + \epsilon), \end{aligned} \tag{8}$$

where α denotes the stepsize, β_1, β_2 denotes the exponential decay rates for moment estimates and ϵ is a term in Adam used to increase the stability of the algorithm. As stated by the authors of Adam [13], we consider term m_t as the first moment of gradient g_t, therefore, one can estimate the variance of gradient as

$$\hat{\mathbf{Var}}[g_t] = ||g_t - m_t||_2^2. \tag{9}$$

Using (9) as our estimation of gradient variance. we propose the following algorithm 1 named variance-based model aggregation.

Algorithm 1 Variance-Based Model Aggregation

Initialize global parameter $x_{global}^{(0)}$, stepsize α, moment-estimate parameters β_1, β_2.
for $t = 0, 1, ..., T-1$ **do**
  Central server broadcasts $x_{global}^{(t)}$ to all clients.
  for agent $i \in [n]$ **do**
    $x_i^{(t,0)} \leftarrow x_{global}^{(t)}$
    moment estimate $m_i^{(0)} \leftarrow 0, v_i^{(0)} \leftarrow 0$
    variance estimate $s_i^{(0)} \leftarrow 0$
    for $k = 0, 1, ..., K-1$ **do**
      $g_i^{(t,k)} \leftarrow \nabla_x l(x_i^{(t,k)}, D_i)$
      $m_i^{(k)} \leftarrow (\beta_1 m_i^{(k-1)} + (1-\beta_1) g_i^{(t,k)})/(1-\beta_1^k)$
      $v_i^{(k)} \leftarrow (\beta_2 v_i^{(k-1)} + (1-\beta_2)(g_i^{(t,k)})^2)/(1-\beta_2^k)$
      $x_i^{(t,k+1)} \leftarrow x_i^{(t,k)} - \alpha m_i^{(k)}/(\sqrt{v_i^{(k)}} + \epsilon)$
      $s_i^{(k+1)} \leftarrow s_i^{(k)} + ||g_i^{(t,k)} - m_i^{(k)}||^2$
    end for
    if $i \in \mathcal{C}_t$ **then**
      Client i sends local variable $x_i^{(t,K)}$ and variance estimate $s_i^{(K)}$ to server
    end if
  end for
  $x_{global}^{(t+1)} \leftarrow \sum_{i \in \mathcal{C}_t} \frac{|D_i|/s_i^{(K)}}{\sum_{j \in \mathcal{C}_t} |D_j|/s_j^{(K)}} x_i^{(t,K)}$
end for

B. Structure-Aware Model Training

Variance-based aggregation allows the FL server to increase algorithmic stability of the training process. In this sub-section, we focus on the client side and introduce structure-aware model training scheme for heterogeneous FL tasks with inherent structures in model and data. We aim to tackle the heterogeneity without compromising performance.

1) Finding Homogeneity in Heterogeneous Tasks: Considering the structure of ML network, different layers of a complex model often serve different purposes. Take Convolutional Neural Networks (CNNs) in computer vision tasks for instance, it is commonly believed that the lower layers of a CNN serves as a common feature detector which can be kept invariant across different tasks, and the last layers are used to learn specific tasks.

For the task of vehicular trajectory prediction, we extend the centralized PGP [1] model to a distributed FedPGP model. PGP consists of three interacting modules each with unique purpose. Firstly, a **graph encoder** encodes vehicle and map context as node encodings of a directed graph, then a **policy header** learns a discrete policy, and the sampled path is decoded into predicted trajectory by a **trajectory decoder**.

Based on information of the application task and the motivation behind network structure construction, we can determine and classify the model structure into homogeneous and heterogeneous parts. For the PGP model, we consider the **graph encoder** and **trajectory decoder** as generic modules that should not be effected by the statistical heterogeneity induced by clients' dataset, and the **policy**

header is the part of the model that differs across clients.

2) Structure-Aware Model Parameter Update: In order to maximize the advantages and minimize the disadvantages of heterogeneous updates, we propose the client-side structure-aware FL model parameter update. At the start of the algorithm, we classify model parameters into homogeneous set $\mathcal{S}_{\text{Hom}}$ and heterogeneous set $\mathcal{S}_{\text{Het}}$ such that the parameters in **graph encoder** module and **trajectory decoder** module are classified as homogeneous and the parameters in **policy header** module are classified as heterogeneous. In every communication round, each client performs homogeneous update (FedAvg) on $\mathcal{S}_{\text{Hom}}$ and heterogeneous update (FedProx, SCAFFOLD, etc.) on $\mathcal{S}_{\text{Het}}$, the exact structure-aware FL model parameter update is provided in Algorithm 2.

Algorithm 2 Structure-Aware Model Training

Initialize neural network model $\mathcal{M}$ with parameter set $x = \{[x]_1, ...\}$, classified by $\mathcal{S}_{\text{Het}}$ and $\mathcal{S}_{\text{Hom}}$.
for $t = 0, 1, ..., T-1$ **do**
  Central server broadcasts $x_{global}^{(t)}$ to all clients.
  for agent $i \in [n]$ **do**
    $x_i^{(t,0)} \leftarrow x_{global}^{(t)}$
    for $k = 0, 1, ..., K-1$ **do**
      for $[x_i^{(t,k)}]_j \in \mathcal{S}_{\text{Het}}$ **do**
        $[x_i^{(t,k)}]_j \leftarrow \text{HetUpdate}([x_i^{(t,k)}]_j)$
      end for
      for $[x_i^{(t,k)}]_j \in \mathcal{S}_{\text{Hom}}$ **do**
        $[x_i^{(t,k)}]_j \leftarrow \text{HomUpdate}([x_i^{(t,k)}]_j)$
      end for
    end for
    if $i \in \mathcal{C}_t$ **then**
      Client i sends local parameters to server
    end if
  end for
  Server performs global aggregation.
end for

V. Experimental Results

This section validates the performance of the proposed VSFL algorithms. We conducted the extensive experiments. The vanilla FedAvg and the heterogeneous SCAFFOLD are used as baselines.

A. Convergence and Stability Experiments

We first evaluate the effectiveness of Algorithm 1 and verify the effectiveness of Theorem 1 using synthetic data. The task is constructed as a linear regression task with artificial additive Gaussian noises. We consider a network with 10 clients, each with an individually generated dataset with different signal to noise ratio (SNR).

Each client dataset has 1000 data samples with input dimension of 10, and each data sample is generated as the weighted sum of the input features with additive Gaussian noise. In accordance with our assumptions in Section IV-A, the amplitude of noise varies across agents. We construct a regression model to minimize the mean squares error (MSE). It can be easily verified that this synthetic task satisfies Assumption 1.

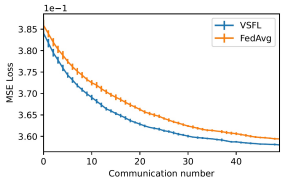

Fig. 1. Mean and variance of the MSE at start stage

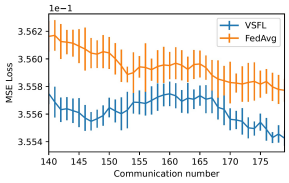

Fig. 2. Mean and variance of the MSE at stable stage

We use a fully connected network for the task and keep the clients to have the same model parameters, hyper parameters and dataset across all synthetic experiments, and compare the performance between FedAvg aggregation and the proposed variance-based aggregation. We show the mean and variance of global verification error evaluated on all clients, and provide performance analysis on the metric score evaluated at the start of experiment and when the algorithms reach steady-state. Fig. 1 shows that from the start of the experiment, Algorithm 1 exhibits faster convergence and yields more stable results across communication rounds. Additionally, we compare two algorithms when both have reached steady-state. Fig. 2, a zooming in view, clearly shows that Algorithm 1 consistently produces lower error while maintaining better consistency across the communication rounds.

B. Trajectory Prediction Experiments

In this section, we verify the empirical performance of our proposed VSFL Algorithms using the trajectory prediction task. All VSFL algorithms and baseline algorithms use the same input of HD map information and vehicle trajectory of previous 2 seconds to predict the future trajectory in next 6 seconds. We use the standard minADE metric, i.e., the minimum average displacement error measured as the average L2 distance between the best predicted trajectory and the ground truth, where the minADE k represents the minADE over the top k predictions.

1) NuScenes Dataset: The NuScenes dataset [15] is a public large-scale autonomous driving dataset commonly used in trajectory prediction works. The NuScenes data is collected as scenes across four different locations in Boston and Singapore. For each individual trajectory, the data also include additional information such as vehicle type and timestamp of data collection.

2) Construction of Split Datasets: To reveal algorithm performance on homogeneous data and heterogeneous data, we use two different rules to split the NuScenes dataset and assign different data segments to clients: i) Split the data randomly for all clients to ensure homogeneous data distribution across clients and ii) Split the data based on the the aforementioned additional information to ensure strong heterogeneity across clients, which we refer to as homogeneous and heterogeneous setups, respectively.

3) The minADE Performance on sampled Datasets: We employ the same model structure on two different setups. For the first setup, all clients possess homogeneous data

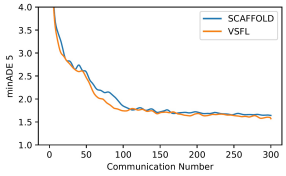

Fig. 3. The minADE 5 performance with homogeneous agents

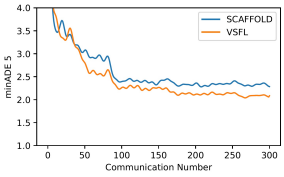

Fig. 4. The minADE 5 performance with heterogeneous agents

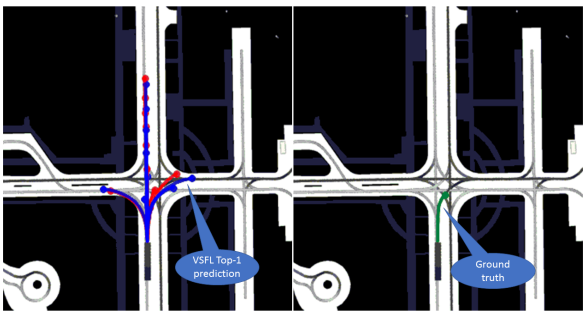

Fig. 5. Top-10 Trajectory Predictions versus Ground Truth

from a universal distribution. For the second setup, all clients use heterogeneous datasets. Each experiment shows averaged results from 3 independent runs.

Fig. 3 shows that for homogeneous case, the application of VSFL slightly improves the performance. However, Fig. 4 demonstrates that the application of VSFL yields significantly better prediction accuracy when heterogeneity is presented in the task. This acts as a confirmation that VSFL addresses the heterogeneity issue of the problem.

4) The minADE Performance on Full Dataset: We also evaluated the proposed algorithms on full NuScenes dataset and list the MinADE 5 results in Table I. It can be also seen that our VSFL model has clear performance improvement compared with baseline FL models in terms of vehicle trajectory prediction, 9.38% improvement over FedAvg and 7.57% improvement over SCAFFOLD. Furthermore, the MinADE 5 result of 1.295 carried out by our distributed VSFL model is close to 1.27 given by the centralized ML model, which further confirms the robustness of the proposed VSFL model. We stress that applying centralized ML model in vehicular networks is impractical since uploading data to cloud server requires enormous communication bandwidth requirement and also risks data privacy. Our purpose is to compare the performance of the proposed VSFL model with the distributed FL baselines.

TABLE I

TRAJECTORY PREDICTION PERFORMANCE COMPARISON AMONG DISTRIBUTED FEDERATED LEARNING MODELS.

Algorithm	FedAvg	SCAFFOLD	**Proposed VSFL**
MinADE 5	1.429	1.401	**1.295**

5) Trajectory Prediction Demonstration: Lastly, we demonstrate the trajectory prediction by our VSFL model. To show performance of the proposed VSFL model under the complex traffic condition, we selected an intersection as illustration point and compared trajectory predictions of the VSFL model with the ground truth and the predictions of the centralized PGP model. Fig. 5 shows a snapshot of trajectory predictions, where right half shows the ground truth, blue and red trajectories in left half are top-10 predictions by the VSFL model and the PGP model, respectively. It can be seen that the top-1 prediction by our VSFL model matches ground truth well.

VI. CONCLUSIONS

Modern vehicles are packed with various on-board sensors to accomplish higher automation levels. However, how to efficiently utilize data collected presents challenges in vehicular networks due to the data privacy threat and communication bandwidth limitation. This paper proposes a variance-based and structure-aware federated learning model to address aforementioned issues. A variance-based model aggregation method is proposed for learning server to select optimal model aggregation weights and a structure-aware model training method is provided for vehicle clients to take full advantages of model parameter homogeneity and heterogeneity in model parameter update. Compared with the FedAvg and the SCAFFOLD baselines, the proposed FL algorithms can improve vehicle trajectory prediction accuracy by 9.38% and 7.57%, respectively.

REFERENCES

[1] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Conference on Robot Learning*. PMLR, 2022, pp. 203–212.

[2] B. McMahan and E. Moore et al, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] H. Cui and V. Radosavljevic et al, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[4] T. Li and A. K. Sahu et al, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[5] S. P. Karimireddy and S. Kale et al, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[6] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Vehicular Communications*, vol. 33, p. 100396, 2022.

[7] K. Xie and Z. Zhang et al, "Efficient federated learning with spike neural networks for traffic sign recognition," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9980–9992, 2022.

[8] S. Chen and Q. Zheng et al, "A theorem of the alternative for personalized federated learning," *arXiv preprint arXiv:2103.01901*, 2021.

[9] O. Bousquet and A. Elisseeff, "Stability and generalization," *The Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.

[10] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Learnability, stability and uniform convergence," *The Journal of Machine Learning Research*, vol. 11, pp. 2635–2670, 2010.

[11] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *International conference on machine learning*. PMLR, 2016, pp. 1225–1234.

[12] Y. Huang and L. Chu et al, "Personalized cross-silo federated learning on non-iid data." in *AAAI*, 2021, pp. 7865–7873.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] L. Balles and P. Hennig, "Dissecting adam: The sign, magnitude and variance of stochastic gradients," in *International Conference on Machine Learning*. PMLR, 2018, pp. 404–413.

[15] H. Caesar and V. Bankiti et al, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

APPENDIX

A. Proof of Theorem 1: Calculation and Minimization of Generalization Error

We consider $\tilde{d}$ as an independent copy of d, with x_{global} denoting the global mean estimation as described is Section IV-A, then

$$\begin{aligned}
gen(\mu, x_{global}|\{D_i\}) =& \mathbb{E}_{x_{global},\{D_i\}}[L_\mu(x_{global}) - L_S(x_{global})] \\
=& \mathbb{E}_{\tilde{d},\{D_i\}}[L_\mu(x_{global})] - \mathbb{E}_{\{D_i\}}[L_S(x_{global})] \\
=& \mathbb{E}_{\tilde{d},\{D_i\}}[\sum_{i=1}^{n} p_i \frac{\sum_{j=1}^{|D_i|} ||\tilde{d}_{i,j} - x_{global}||^2}{|D_i|}] - \mathbb{E}_{\{D_i\}}[\sum_{i=1}^{n} p_i \frac{\sum_{j=1}^{|D_i|} ||d_{i,j} - x_{global}||^2}{|D_i|}] \\
=& \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \mathbb{E}_{\tilde{d},\{D_i\}}[||\tilde{d}_{i,j} - x_{global}||^2] - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \mathbb{E}_{\{D_i\}}[||d_{i,j} - x_{global}||^2] \\
=& \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \big(Tr(Cov(\tilde{d}_{i,j})) + Tr(Cov(x_{global}))\big) - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \mathbb{E}_{\{D_i\}}[||d_{i,j} - x_{global}||^2]
\end{aligned} \tag{10}$$

If we denote the dimension of x as m, then,

$$\begin{aligned}
gen(\mu, x_{global}|\{D_i\}) =& \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \big(\sigma_i^2 m\big) + Tr(Cov(x_{global})) - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \mathbb{E}_{\{D_i\}}[||d_{i,j} - x_{global}||^2] \\
=& \sum_{i=1}^{n} p_i \sigma_i^2 m + \sum_{i=1}^{n} \frac{p_i^2}{|D_i|} \sigma_i^2 m - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \mathbb{E}_{\{D_i\}}[||d_{i,j} - x_{global}||^2] \\
=& \sum_{i=1}^{n} p_i \sigma_i^2 m + \sum_{i=1}^{n} \frac{p_i^2}{|D_i|} \sigma_i^2 m \\
& - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \Big(\big(\sum_{k=1,k\neq i}^{n} \frac{p_k^2}{|D_k|} \sigma_k^2 m \big) + \frac{p_i^2(|D_i|-1)}{|D_i|^2} \sigma_i^2 m + (1 - \frac{p_i}{|D_i|})^2 \sigma_i^2 m \Big) \\
=& \sum_{i=1}^{n} p_i \sigma_i^2 m + \sum_{i=1}^{n} \frac{p_i^2}{|D_i|} \sigma_i^2 m - \sum_{k=1}^{n} \frac{p_k^2}{|D_k|} \sigma_k^2 m \\
& - \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sum_{j=1}^{|D_i|} \Big(- \frac{p_i^2}{|D_i|^2} \sigma_i^2 m + (1 - \frac{p_i}{|D_i|})^2 \sigma_i^2 m \Big) \\
=& \sum_{i=1}^{n} \frac{p_i}{|D_i|} \sigma_i^2 m - \sum_{i=1}^{n} p_i \sum_{j=1}^{|D_i|} \big(1 - \frac{2p_i}{|D_i|}\big) \sigma_i^2 m \\
=& \sum_{i=1}^{n} \frac{2p_i^2}{|D_i|} \sigma_i^2 m
\end{aligned} \tag{11}$$

Additionally we know that $\mathbf{p} = (p_1, ..., p_n)$ is constrained on the unit simplex, hence we can find the optimal set of $\mathbf{p}$ to minimize The following optimization problem,

$$\begin{aligned}
&\max_{\mathbf{p}=[p_1,\ldots,p_n]} \sum_{i=1}^{n} \frac{2p_i^2}{|D_i|} \sigma_i^2 m \\
&\text{Subject to: } \sum_{i=1}^{n} p_i = 1
\end{aligned}$$

Which admits a closed-form solution of

$$p_i = \frac{\frac{|D_i|}{\sigma_i^2}}{\sum_{j=1}^{n} \frac{|D_j|}{\sigma_j^2}}$$

the result shows that the optimal weight factor is proportional to the number of data the client holds, and inversely proportional to the variance of the client.