



**HAL**  
open science

# SiamFLTP: Siamese Networks Empowered Federated Learning for Trajectory Prediction

Mehdi Salim Benhelal, Badii Jouaber, Hossam Afifi, Hassine MOUNGLA

► **To cite this version:**

Mehdi Salim Benhelal, Badii Jouaber, Hossam Afifi, Hassine MOUNGLA. SiamFLTP: Siamese Networks Empowered Federated Learning for Trajectory Prediction. 2025. hal-04893834

**HAL Id: hal-04893834**

**<https://hal.science/hal-04893834v1>**

Preprint submitted on 17 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SiamFLTP: Siamese Networks Empowered Federated Learning for Trajectory Prediction

Mehdi Salim Benhelal  
SAMOVAR, Telecom SudParis  
Institut Polytechnique de Paris  
Palaiseau, France  
salim.benhelal\_mehdi@telecom-sudparis.eu

Hossam Afifi  
SAMOVAR, Telecom SudParis  
Institut Polytechnique de Paris  
Palaiseau, France  
hossam.afifi@telecom-sudparis.eu

Badii Jouaber  
SAMOVAR, Telecom SudParis  
Institut Polytechnique de Paris  
Palaiseau, France  
badii.jouaber@telecom-sudparis.eu

Hassine Moun gla  
Université Paris Cité,  
Institut Polytechnique de Paris  
Paris, France  
hassine.moun gla@u-paris.fr

**Abstract**—Our main objective in this work is to address the challenge of enhancing the forecasting of agents trajectories for Connected and Autonomous Vehicles (CAVs) while prioritizing privacy. We introduce an innovative approach to Federated Learning tailored to the contextual aspects of trajectory prediction. We employ the Siamese Neural Network (SNN) to capture context similarities between clients’ environments. Subsequent cluster formation employs SNN to group clients with similar static contexts for federated training, enhancing learning efficiency.

Results of our experiments on real-world datasets collected from the highway drone dataset (highD) and the intersection drone dataset (inD) combination, quantified by utilizing well-established metrics such as Average Displacement Error (ADE) and Final Displacement Error (FDE), validate the effectiveness of our approach, obtaining superior trajectory prediction capabilities, showcasing the successful alignment of Federated learning with the intricate challenges of trajectory forecasting, all while prioritizing privacy.

**Index Terms**—Trajectory prediction, Federated learning, Siamese neural network, Connected and autonomous vehicles.

## I. INTRODUCTION

Trajectory prediction stands out as one of the key areas identified by researchers to enhance the capabilities of the Connected and autonomous vehicles (CAVs) [1]. The goal is to forecast future positions of agents (cars, pedestrians, etc.) in the scene, an information to be used for planning future routes of CAVs.

Existing research gathers trajectory data from various scenes and combines them, ignoring the distributive part of the data. So effectively aggregating them is a challenging task and may lead to many privacy violations. Thus, a new plethora of studies [2] introduced the use of Federated learning, which

This research was partially supported by Labex DigiCosme (project ANR11LABEX0045DIGICOSME) operated by ANR as part of the program « Investissement d’Avenir » Idex ParisSaclay (ANR11IDEX000302)

exploits the distributed capabilities of edge computing [3] and preserves data privacy.

However, existing methods for selecting participants for edge computing and Federated learning, mainly consider resources and device capabilities of each participant [4] ignoring the delicate setting of trajectory prediction. Therefore, we introduce a context-tailored Federated learning approach that better captures the intricacies of the problem in a collaborative manner where only scene context (images, etc.) is shared without violating user’s privacy.

An overview of the primary contributions of this work is provided below:

- We introduce a new Federated learning framework that leverages the use of the Siamese networks for a more context oriented learning.
- We show the efficacy of our method using real world datasets.

In order to clearly illustrate our results, the remaining of this paper is arranged as follows:

- In section II we present the related work for each of the Federated learning, trajectory prediction and Siamese Networks.
- In section III we provide an explanation for each step of our approach.
- In section IV we test our method and show our results.
- Lastly, in Section V we wrap up and discuss potential directions for further research.

## II. RELATED WORK

### A. Federated learning

Federated learning is a machine learning paradigm that was first presented by [5]. It allows a model to be trained across several decentralized edge devices or servers, all the

while protecting the privacy of the data on each individual device. Using local stochastic gradient descent (SGD), each local model in this method is trained. The global model then aggregates the local models at each iteration until convergence is reached.

Several research works have opened the path to improvements. [6] proposed to introduce an Adam adaptive optimisation when updating the global model on the server, [7] generalised other adaptive optimisation techniques in the same style and provided convergence guarantees, [8] introduced the use of Adam optimiser for local clients (FedAvg-Adam) where each client uses Adam locally and the server averages the global weights and moments of the clients.

Another aspect of additional enhancement suggests that, instead of using random client selection as in the original algorithm, studies have introduced numerous criteria, mainly based on the capabilities of participants and resource allocation. [9] and [10] proposed a novel client selection strategy based on the resources conditions. Authors of [4] employed a trust-based deep reinforcement learning model in order to select the appropriate autonomous vehicles at each round to improve performance and explainability.

### B. Federated learning and trajectory prediction

The main aim of trajectory prediction is to forecast the future positions of each agent (humans, vehicles, etc.) within a specified scene based on their past lateral and longitudinal positions, additionally some methods also use the scene context [11].

Recently, many trajectory prediction systems have employed Federated learning, where the data is distributed. [12] presented a Spatio-Temporal Prediction framework in Multi-Robot systems, where each agent contributes to the global learning. Another approach [13] designed a trajectory prediction method for vehicles based on Federated learning, taking into consideration both space and time. Adding to this momentum, [14] proposed an automatic human Trajectory Prediction Model Design under a Federated learning Framework. [15] elaborated a Federated learning vehicle trajectory prediction algorithm that tackles the risks of cyberattacks. Additionally, novel client selection methods for trajectory prediction, which mainly consider the explainability of the model, have been adopted [4].

### C. Siamese Neural Network(SNN)

In the field of artificial intelligence research, Siamese networks have become increasingly prominent when used in tasks that need measures of similarity or dissimilarity [16].

In a recent study [17], the authors delineate a novel approach wherein they treat the local and global models of Federated learning as distinct branches within the Siamese Neural Network (SNN) architecture during local training. This approach effectively governs the model update direction by continuously

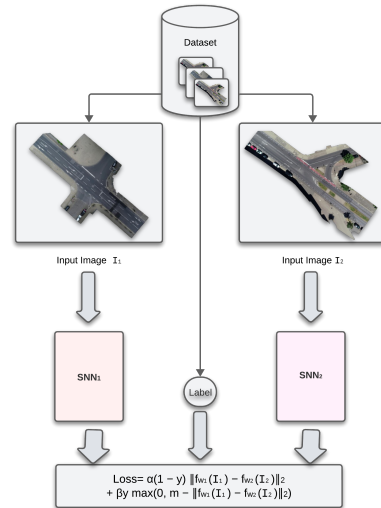


Fig. 1. Preamble training of SNN

adapting the model similarity, thereby personalizing the local model.

## III. METHOD

Our solution consists of a preamble training of SNN as illustrated in Fig. 1, followed by two main parts as shown in Fig. 2, the first part consists of the clusters creation where clients with similar static context are grouped together, and the second part where each cluster constructs its own global model using Federated learning detailed explanation of each component of our framework is provided below:

### A. Preamble training of SNN

Motivated by the goal of effectively learning context similarity between different environments from image data, the Siamese Network is trained beforehand using environment images of scenes from 'inD' [18] and 'highD' [19], more detailed description of these datasets is provided in IV-A. It is important to note that, in the preamble training we exclusively used images and only from specific scenes (to avoid influencing the subsequent framework results) in creating a custom dataset for Siamese Networks training. We utilize the contrastive loss function, which has demonstrated its reliability for embedding learning tasks [20] [21], the mathematical formula is defined as the following:

$$Loss = \alpha(1 - y) \|f_{w_1}(I_1) - f_{w_2}(I_2)\|^2 + \beta y \max(0, m - \|f_{w_1}(I_1) - f_{w_2}(I_2)\|)^2 \quad (1)$$

In this context, we have two inputs:  $I_1$ ,  $I_2$  each one denoting an environment image. The variable  $y$  represents a label indicating whether the environments are similar or not. Additionally, there are two constants,  $\alpha$  and  $\beta$ , which control the trade-off between different terms. In our case, both  $\alpha$  and

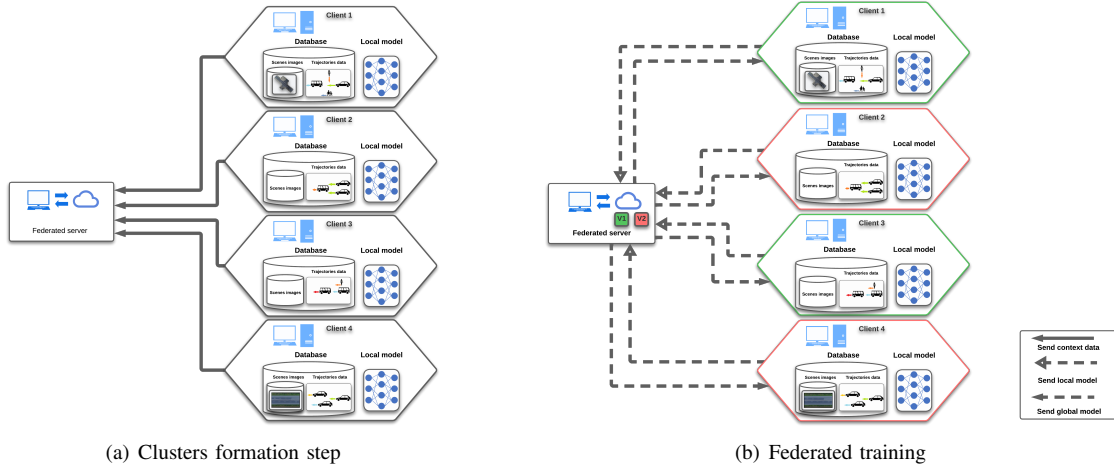


Fig. 2. Two-Step Process: Cluster Creation Based on Static Context and Federated learning for cluster-specific Global Models

$\beta$  are set to 1 to balance the model’s ability to discriminate between similar and dissimilar pairs. Moreover a margin value,  $m$ , which is set to 2 in our scenario.

$\|f_{w_1}(I_1) - f_{w_2}(I_2)\|$  denotes the Euclidean distance calculated in an embedded feature space. Here, ‘ $f$ ’ represents an embedding function that transforms an input context map into a real vector.

### B. Clusters formation

In this part of our framework, we suggest using our trained SNN to form our clusters. Each available client sends its static context to the central server, Subsequently, the server then proceeds to create clusters using the method outlined in Algorithm 1.

At each iteration of our loop, we randomly elect a client from the available clients then compare its environment with the remaining clients, the closer matching clients form a cluster which later used for the Federated learning training.

The resemblance of two clients is calculated using their context as input to the SNN which results in the computation of the Euclidean distance metric, the smaller distance produced indicates a lower dissimilarity between the two clients context and the client will be chosen to join the cluster if the dissimilarity doesn’t exceed the Limit.

### C. Federated training

In this module of our framework, we leverage the existing clusters created by the previous part to complete our federated training. In contrast with existing methods, each cluster will only train its own global model in parallel.

The intricate details of the training process are provided in Algorithm 2.

After the training phase, the testing scenes context images are initially compared to random scenes from each cluster using the SNN to determine the suitable global model for evaluation.

---

### Algorithm 1 Pseudo-code of our Clusters formation algorithm using clients context

---

#### Initialization:

```

Clusters {}
Clients ← get_available_clients()
/* Gather all available clients*/
while Clients.is_not_empty() do
    V {}
    Client E is randomly designated from Clients
    V.add(E)
    for C in Clients do
        if Limit > calculate_distance(E_env, C_env) then
            /* similarity is calculated using the SNN
            between client E and C context images*/
            V.add(C)
        end if
    end for
    Clusters.append(V)
    Clients ← Clients - V
end while

```

---

## IV. EXPERIMENTS

In this experimental section, we opted for MID [22] as our trajectory prediction base model where 8 timesteps are utilized as observed data to predict 12 timesteps. This choice was motivated not only because its fast convergence but also by its precision and the diversity of trajectories it offers. We adapt it to predict trajectories for all agents in the scene, rather than just pedestrians.

For the optimization strategy, we use FedAvg-Adam from [8], where each client employs Adam for local updates, and the server averages the clients models. Furthermore, and to assess the efficiency of our method, we choose to compare our approach of client choice against utilizing all available

---

**Algorithm 2** Pseudo-code of the Federated training process

---

**Main Loop:**

```
Initialize k
/* K is the number of selected clients at each round
for V in Clusters in parallel do
  Create_global_agent_v()
  while not termination_criteria_met(rounds, performance)
  do
    Server-Client Interaction:
    selected_clients ← randomly select K clients from
    available clients in V
    for client in selected_clients do
      send_global_parameters_to_clients(client, global_param
      , optimizer_moments)
      /* Send global parameters to the selected clients */
    end for
    Client-Server Interaction:
    for client in selected_clients do
      (received_params, optimizer_moments) ←
      receive_and_update_local_parameters(client)
      /* Client update local parameters */
      local_epochs_execution(client, received_params)
      /* Client executes local training epoch */
      send_new_data_to_server(client, new_data)
      /* Client send local parameters to server */
    end for
    Update global parameters
  end while
end for
```

---

participants without the process of client selection.

### A. Dataset

To properly evaluate our method, we combine the two following datasets into one rich setup with different scenes from distinct locations: the 'inD' [18] and the 'highD' [19] datasets, examples are showed in Fig. 3.

The 'inD' collection comprises approximately 13,599 roads users (pedestrians, vehicles, etc.) trajectories and scenarios captured from a drone point of view at intersections from four distinct locations. The 'highD' dataset, on the other hand, consists of scenes captured on highways from six distinct sites and includes more than 110,000 vehicles, both datasets are captured at 25 frames per second using a 4k resolution camera.

Note that we preprocessed both datasets similarly using a step size of 12 and a window size of 35.

### B. Setup

We made each client get access to a selection of scenes from a certain location. For a total of 12 clients, we constructed six that employ scenes from the "inD" dataset and the remaining six from the "highD" dataset. TABLE I shows the scenes that are available to each client.

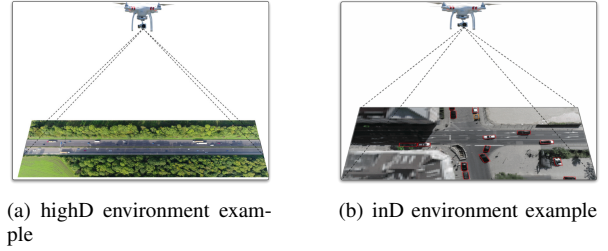


Fig. 3. Visualization of highD and inD Datasets

we divided our testing into two stages. We employed scenes from the "highD" dataset to test our algorithms first, and then we utilized scenes from the "inD" dataset to repeat the testing. We were able to properly assess the effectiveness and generalizability of our approaches across the two datasets and their corresponding contexts thanks to this sequential methodology.

The simulation of the communication process was done programmatically to mimic real-world scenarios where a federated collection of edge devices communicates with one another.

TABLE I  
AVAILABLE CLIENTS

| Dataset | Clients | Scenes   |
|---------|---------|--|
| highD   | C1      | '01', '02', '03'   |
|         | C2      | '15', '16', '17', '18', '19', '20', '21', '22', '23', '24' |
|         | C3      | '11', '12'   |
|         | C4      | '07', '08', '09', '10'                                     |
|         | C5      | '14', '27', '28'   |
|         | C6      | '04', '05', '06'   |
| inD     | C7      | '22', '23'   |
|         | C8      | '07', '08', '09', '10'                                     |
|         | C9      | '25'   |
|         | C10     | '30', '31', '32'   |
|         | C11     | '01', '02', '03', '04', '05', '06'                         |
|         | C12     | '19'   |

### C. Metrics

The selection of appropriate metrics is crucial for objectively assessing the performance of predictive models thus we utilize the popular Average displacement error (ADE) and Final displacement error (FDE) measures.

- ADE : Quantifies the mean Euclidean distance, measured in meters, as in [23] with the following:

$$ADE = \frac{1}{N_A \times H} \sum_{i=1}^{N_A} \sum_{t=1}^H \left| \hat{Y}_t[i] - Y_t[i] \right| \quad (2)$$

Where  $H$  is the prediction horizon,  $\hat{Y}$ ,  $Y$  are the predicted and ground truth trajectories across  $H$  and  $N_A$  is the total number of agents in the scene.

- FDE: indicates the final Euclidean distance in meters where the trajectories are only assessed over the last predicted position T. This calculation is performed as follows:

$$FDE = \frac{1}{N_A} \sum_{i=1}^{N_A} \left| \hat{Y}_T[i] - Y_T[i] \right| \quad (3)$$

We used a server with these specifications:

- GPU/RAM: NVIDIA Quadro RTX 4000/755GiB
- CPU: Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz

#### D. Results

This section presents the results of our experiments. Subsection IV-D1 demonstrates the viability of our SNN cluster formation solution for selecting the most suitable clients for the training.

Subsequently in IV-D2, we assessed our method using highD scenes. Later in Section IV-D3, our evaluation was extended to scenes from the inD dataset.

1) *Client selection*: In this Subsection we present the produced clusters from Algorithm 1 as illustrated in TABLE II, we can notice that the created clients clusters from the static environment follow the same trend as the initial datasets before the reintegration indicating that our method successfully generated meaningful clusters for the training.

TABLE III shows the selected clients at a random round of the training where K clients are drafted from the created clusters distinct from the random selection.

TABLE II  
RESULTING CLUSTERS FROM SNN

| Clusters | Selected clients          |
|----------|---------------------------|
| V1       | C1, C2, C3, C4, C5, C6    |
| V2       | C7, C8, C9, C10, C11, C12 |

TABLE III  
EXAMPLE OF SELECTED CLIENTS FOR A TRAINING ROUND

| Random                   | Ours                      |                        |     |
|--------------------------|---------------------------|------------------------|-----|
|                          | V2                        | V1                     |     |
| C4, C2, C11              | C11, C8, C12              | C2, C4, C5             | K=3 |
| C4, C2, C6, C11, C8, C12 | C7, C12, C9, C10, C11, C8 | C1, C3, C2, C4, C5, C6 | K=6 |

2) *Evaluation on highD*: Our first experiments were conducted on scenes from the highD dataset, more precisely on scenes ('58', '59', '60').

After using the SNN to calculate similarity between the evaluation scenes and random scenes from clusters V1 and V2, the model resulting from training on cluster V1 was

chosen for the evaluation, Fig. 4 and Fig. 5 show a respective comparison between randomly choosing participants for the training and our framework using ADE and FDE while considering different numbers of participating clients. The results clearly indicate that our framework consistently outperforms the random participant selection method across various scenes by achieving lower displacement predictions.

Note that each client trains for one local epoch and the total number of rounds is 55 in all our experiments. The shaded region in the graphs represents the variability values resulting from running the experiments multiple times.

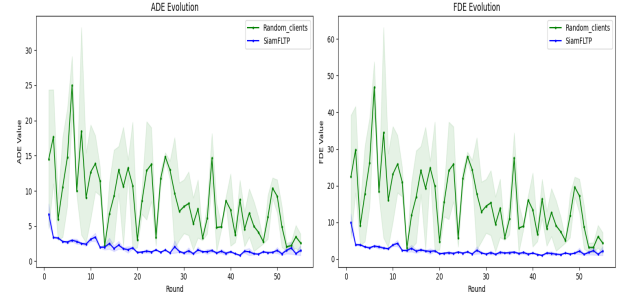


Fig. 4. Results on highD scenes when selecting 3 clients per round

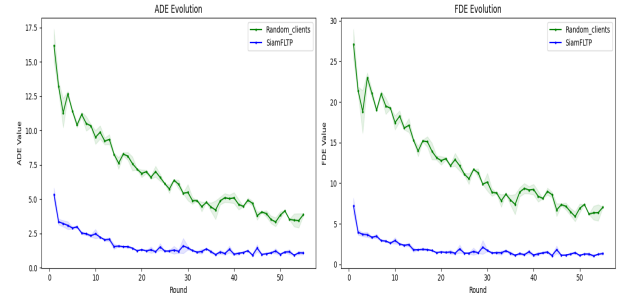


Fig. 5. Results on highD scenes when selecting 6 clients per round

3) *Evaluation on inD*: Our later phase of experiments were conducted on scenes ('13', '14') from the inD dataset, the model resulting from cluster V2 was chosen for the evaluation. Fig. 6 and Fig. 7 confirm respectively that our framework performs better and achieves superior ADE and FDE results in comparison to the default Federated learning framework using random participants choice.

## V. CONCLUSION AND FUTURE WORK

This paper tackles the problem of trajectory prediction for CAVs, highlighting the necessity of practical approaches that take into account the distributed nature of trajectory data while preserving privacy.

We introduced a context-tailored Federated learning framework employing Siamese Networks for a context-oriented learning during the preamble training. Guided by the trained Siamese Networks, the cluster formation process organizes

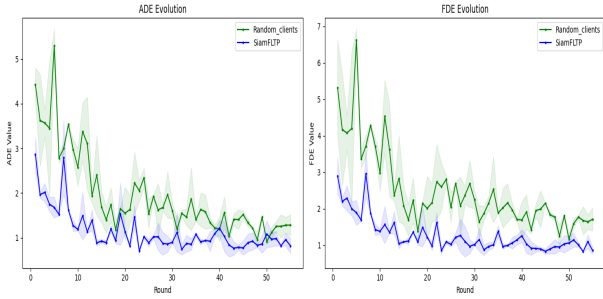


Fig. 6. Results on inD dataset when selecting 3 clients per round

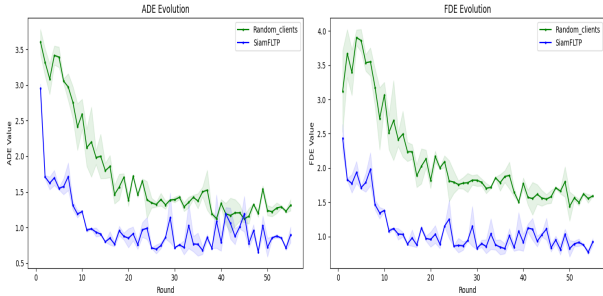


Fig. 7. Results on inD dataset when selecting 6 clients per round

clients with similar static contexts for federated training. Then, the federated training module refines global models in parallel within these clusters, the experiments conducted on real-world datasets, combining scenes from 'inD' and 'highD' demonstrate the efficacy of the proposed method.

Our method helps CAVs make better decisions and plan better routes increasing their navigation efficiency in real-world scenarios.

To increase the contributions that this work has made, we may investigate variants in SNN designs and integrate more sophisticated machine learning approaches. Potential areas for further research include examining the dynamic nature of the scenes and integrating real-time adaptability.

## REFERENCES

- [1] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.
- [2] R. Ni, Y. Lu, B. Yang, C. Yang, and X. Liu, "A federated pedestrian trajectory prediction model with data privacy protection," *Complex Intell. Syst.*, Oct. 2023.
- [3] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.
- [4] G. Rjoub, J. Bentahar, and O. A. Wahab, "Explainable ai-based federated deep reinforcement learning for trusted autonomous driving," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022, pp. 318–323.
- [5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [6] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.
- [7] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.
- [8] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630–641, 2022.
- [9] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.
- [10] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2021.
- [11] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "Car-net: Clairvoyant attentive recurrent network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 151–167.
- [12] N. Majcherczyk, N. Srishankar, and C. Pinciroli, "Flow-fl: Data-driven federated learning for spatio-temporal predictions in multi-robot systems," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 8836–8842.
- [13] X. Zhou, R. Ke, Z. Cui, Q. Liu, and W. Qian, "Stfl: spatio-temporal federated learning for vehicle trajectory prediction," in *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPPI)*, 2022, pp. 1–6.
- [14] C. Wang, X. Chen, J. Wang, and H. Wang, "Atpfl: Automatic trajectory prediction model design under federated learning framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6563–6572.
- [15] Z. Wang and T. Yan, "Federated learning-based vehicle trajectory prediction against cyberattacks," in *2023 IEEE 29th International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2023, pp. 1–6.
- [16] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems*, ser. NIPS'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 737–744.
- [17] M. Yang, Y. Wang, X. Wang, Z. Zhang, X. Wu, and P. Cheng, "Fedsiamda: Dual-aggregated federated learning via siamese networks under non-iid data," *arXiv preprint arXiv:2211.09421*, 2022.
- [18] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.
- [19] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.
- [20] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," *arXiv preprint arXiv:1707.02131*, 2017.
- [21] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.
- [22] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 113–17 122.
- [23] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.