Federated Learning for Lane-Change Prediction

Lilit Yenokyan[1], William Lindskog-Muenzing[2], Christian Prehofer[3](✉), and Matthias Schubert[1]

[1] LMU Munich, Munich, Germany
schubert@dbs.ifi.lmu.de
[2] Flower Labs and Technical University of Munich, Munich, Germany
william.lindskog@tum.de
[3] DENSO Automotive, Eching, Germany
c.prehofer@eu.denso.com

Abstract. In this paper, we present a data-driven approach for predicting lane changes of vehicles in a highway scenario based on observing position, speed, and movements of surrounding vehicles. To train a prediction model, we employ federated learning (FL) with various locations acting as clients. The study employs Long Short-Term Memory (LSTM) networks that utilize 1 s of historical data to forecast lane changes over a 1, 3 and 5-s prediction horizon. We show that personalized FL performs well for a distributed setup without data sharing. The findings demonstrate FL's potential in automotive safety applications, nearly matching centralized performance while significantly improving data security and privacy across distributed locations. This study supports using federated learning as a viable and robust solution for privacy-preserving predictive tasks in dynamic environments.

Keywords: Federated Learning · Lane-Change Prediction · Automotive · Privacy

1 Introduction

Accurate prediction of lane-change scenarios is a crucial factor for highway safety. Research indicates that current adaptive cruise control (ACC) driver assistance systems can exhibit unpredictable behavior in irregular traffic conditions [2]. Furthermore, studies reveal that only about half of drivers use turn signals when changing lanes, with rates as low as 44% in China [29] and 52% in the United States [20]. This lack of turn signal usage significantly contributes to the number of incidents caused by lane changes. According to statistics from the National Highway Traffic Safety Administration (NHTSA), lane-change maneuvers result in up to 610,000 traffic accidents annually in the U.S., leading to at least 60,000 injuries [8].

W. Lindskog-Muenzing—Research carried out at DENSO Automotive.

A. Kocian et al. (Eds.): INTSYS 2024, LNICST 608, pp. 311–331, 2025.
https://doi.org/10.1007/978-3-031-86370-7_19

In this paper, we propose a data-driven approach for predicting lane changes of vehicles in highway scenarios by analyzing the position, speed, and movement of surrounding vehicles. A critical aspect of implementing data-driven solutions is addressing the privacy of driving data, particularly the need for data exchange among different organizations. This exchange involves gathering data from various highways and entities to develop collaborative, high-quality machine learning models. However, mobility data can be sensitive, raising concerns about confidentiality in light of privacy regulations.

To tackle these challenges, we utilize federated learning (FL) [9,16], a privacy-preserving machine learning framework that allows organizations to collaboratively train models without sharing raw data. In this approach, the raw data remain local, and FL enables model training across multiple clients by transmitting model updates after each training step. Research has shown that FL can lead to effective solutions while safeguarding sensitive information. By sharing only local model updates, FL not only protects individual data privacy but also has the potential to significantly reduce network traffic.

We demonstrate that federated learning can effectively leverage data collected from different vehicles to train robust predictive models for lane-change prediction. In our study, we utilize movement data from various highway locations, treating each location as a separate client within the FL framework. This approach allows us to train a global model on decentralized data without the need for raw data transfer.

In summary, our research contributions are as follows:

- We present a data-driven approach for lane change detection with well-defined steps, creating a reproducible framework that enables accurate comparisons for future studies.
- We demonstrate that federated learning achieves strong performance while preserving data privacy, making it a viable alternative for privacy-sensitive applications.
- We show that personalized federated learning improves accuracy by adapting models to specific locations, outperforming standard FL by effectively handling local variations.
- We highlight the benefits of FL over local models for data-scarce locations, providing a strong incentive for client participation in federated training.

In the subsequent sections, we review related work on federated learning and the dataset utilized for lane-change prediction. We then present our modeling and machine learning setup, followed by a comparative analysis of results from centralized and federated learning approaches. We identify federated learning methods and algorithms that are particularly effective for our use case, with data from different location. We focus on establishing a well-defined and reproducible data-driven framework for lane-change detection using federated learning.

2 Related Work and Lane-Change Definitions

Lane-change (LC) maneuver prediction tasks have been extensively addressed using machine learning approaches based on observed data. These methods typically categorize three fundamental maneuvers: changing to the left lane (LLC), changing to the right lane (RLC), and lane keeping (LK) [23,24]. Consequently, LC prediction can be treated as a classification task aimed at estimating the class or probability of the upcoming maneuver. In terms of data utilization, features commonly employed in LC prediction algorithms include geographic positioning system (GPS) traces, external camera or drone recordings, and vehicle sensor readings, such as those from cameras and radars [17,19,26].

In this section, we review related work and the state of the art in lane-change prediction.

Classification and Labeling. Regarding the research on lane-change prediction, definitions of lane-change vary. Some studies refer to the start of the lane-changing process, while others define it as the moment a vehicle crosses a lane marking. For example, [15] states “a lane change is fully executed if all points ... lie within the observed section of the highway.” On the other hand, [13] defines lane-change as “Crossing lane markings and staying in a new lane.” A more precise definition is provided in [22]: “For the moment of the lane change, we are using the point in time when the vehicle center has just crossed the lane marking”. Our work follows the actual Highway Drone (highD) [13] dataset's annotations, indicating when the vehicle crosses the lane marking (see [27] for more details).

Specifically, [15] introduces an automatic labeling approach that uses density-based clustering to detect lane changes. An SVM then learns the cluster boundaries to label new data, which is subsequently fed into an LSTM model for classification. Trained on the highD dataset, this model achieved 88% accuracy for a 3-s prediction horizon with only 1 s of historical data. However, the absence of a precise lane-change point complicates comparisons with other studies.

In a comprehensive analysis, [28] compares LSTM, SVM, XGBoost, and LightGBM methods for LC prediction, using the CitySim dataset with a 5-s input time duration. It achieves around 95% classification accuracy with LSTM and 98% accuracy with LightGBM. However, the report lacks clarity on the definition of lane-change, the prediction horizon, and the target value used in the modeling.

Other works define lane change as a process [3,13], concluding when the vehicle stabilizes in the new lane. This definition directly affects the prediction horizon, as the lane change must occur within that horizon, making comparisons across studies challenging.

Trajectory Prediction. In addition to maneuver classification, many studies adopt a two-step approach: first, they predict vehicle trajectories, and then they use classifiers to identify maneuvers. These maneuvers are classified either into aforementioned multiple categories or as a binary option (lane-change or lane-keep). For instance, [11] evaluates two kinds of ANN over two different datasets for trajectory prediction, followed by an SVM for binary classification of the vehicle's maneuvers. The lane-change point (LCP), defined as the vehicle's center crossing the line between two lanes, is predicted with an 85% F1 score for a 3-s prediction horizon and 93% for 1 s. There are also hybrid approaches based on both deterministic and probabilistic methods. [25] aims to detect the LCs by applying the potential field method for trajectory planning, taking into account the presence of surrounding vehicles to reduce false alarms caused by zigzag driving. Afterwards, SVM is used for vehicle intent classification. With the use of the Federal Highway Administration dataset, they detect LC on average 1.74 s before the vehicle crosses the center-line with a 98.1% accuracy.

Prediction of Time Until Lane-Change. Other studies aim to predict the precise time until the lane-change (TTLC). [22] develops a predictive model for estimating TTLC using LSTM-based RNNs. By using 3 s of historical data from the highD dataset, the authors claim that accurate predictions can be made as early as 3.5 s before the lane-change, with a median error of less than 0.25 s. This model's root mean squared error (RMSE) is later improved by 0.2 s by [18]. They propose a multi-task model to simultaneously predict the LC maneuvers and estimate TTLC for automated driving systems. The authors report that their model outperforms the SOTA on the highD dataset, achieving approximately 85% recall for a 3-s prediction horizon and a 60% recall for the defined maximum prediction horizon of 5.2 s. [5] offers a solution with a richer dataset, predicting the TTLC with an error of only 0.3 s at a prediction horizon of 3 s and a history of 3 s using LSTM. This method uses information about the driver's state along with general traffic information. While this information can be measured at many points in the scene, the driver state measurement makes the model only applicable to predict ego vehicle maneuvers.

Another relevant work, [12], conducts a comparative analysis of various machine learning approaches alongside a rule-based method for predicting lane-change maneuvers 2 s before a vehicle crosses the highway centerline. The results demonstrate that artificial neural networks (ANN) significantly outperform the MOBIL model. Additionally, the error rates in predicting LLC, RLC, and LK using ANN for 4 s in advance are 12.2%, 20.5%, and 0.5%, respectively.

Lane-Change Prediction in Federated Learning. Regarding LC prediction in the context of FL, there is limited research addressing this specific combination. The only notable study we found is by [7], which proposes a clustering-based personalized federated learning (CPFL) framework. It uses in-cabin driver monitoring data, such as head rotation, from 5 drivers in simulation scenarios to predict the intention to change lanes. However, its reliance on limited monitoring data does not capture the environmental factors influencing LC prediction, such as road conditions and traffic. Additionally, the small dataset raises ques-

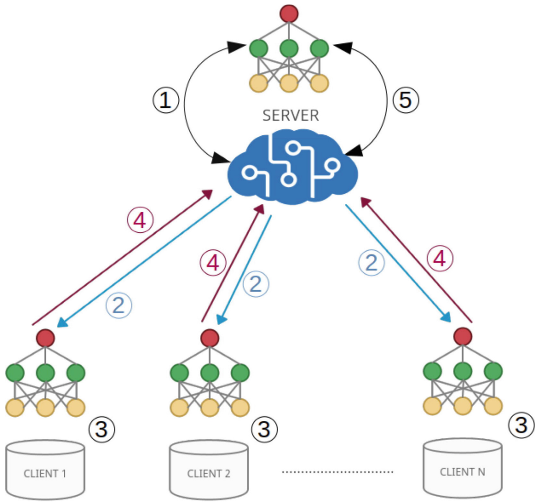

Fig. 1. Scheme of Federated Learning Approach. (1) server creates the model, (2) server sends the model to the clients, (3) clients train the model on local data, (4) clients send model parameters to the server, (5) server aggregates the parameters and updates the model. Adapted from [6].

tions about the generalizability of the results. In contrast, our work focuses on predicting lane changes of surrounding vehicles.

3 Federated Learning

Federated learning (FL) is a distributed, collaborative approach in machine learning that enables models to train directly on decentralized devices. Unlike traditional distributed learning, which aims to parallelize computation by distributing centralized data across nodes followed by aggregation, FL prioritizes privacy protection. In this framework, computation occurs on edge devices where the data is generated, and the centralized server only accesses and aggregates the parameters of locally trained models.

With the increasing emphasis on data privacy and security regulations, such as the General Data Protection Regulation (GDPR), there is a growing need for secure and efficient data processing methods. FL was introduced in 2016 by [16] as a solution to these challenges.

Key components of FL architecture, as shown in Fig. 1, are the **centralized server** and multiple decentralized devices called **clients**. The server acts as the central coordinator of the learning process. It manages the communication with the clients, handling the distribution of the global model and the collection of the local model updates. One of the server's key roles is to aggregate the model parameters from the local models trained on the client devices to get an improved global model.

Algorithm 1: Federated Proximal (FedProx) Algorithm

```
Input: Initial global model weights w_0, Number of communication rounds F,
       Number of clients K, Number of local epochs E, Learning rate η,
       Proximal term coefficient μ
Output: Trained global model weights w_F
for t = 0, 1, ..., F − 1 do
    // Server-side operations
    Select a random subset of clients S_t from K clients;
    Send the global model weights w_t to the selected clients;
    for each client k ∈ S_t in parallel do
        // Client-side operations
        Receive the global model weights w_t from the server;
        Initialize local model weights w^k_{t,0} ← w_t;
        for i = 1, 2, ..., E do
            for each batch b in local data do
                w^k_{t,i} ← w^k_{t,i−1} − η∇ℓ(w^k_{t,i−1}; b) + μ(w^k_{t,i−1} − w_t) // Update
                    local model weights with proximal term
        Send the updated local model weights w^k_{t,E} to the server;
    // Server-side aggregation
    w_{t+1} ← Σ_{k∈S_t} (n_k/n) w^k_{t,E} // Aggregate local updates to update global
        model
```

The clients' primary role is to perform the actual training on their local datasets. For privacy preservation, these datasets remain on clients' respective devices and are not shared with the server or other clients. Clients send their locally trained model updates (typically the gradients or updated weights) to the central server and receive the updated global model in return. This communication is typically done at the end of each training round.

The basic operation of FL is illustrated in Fig. 1. The training begins with the central server initializing the global model. During each communication round, a random subset (or all) of clients is selected to participate in training. These clients perform local training on their data and send the updated model parameters back to the central server, where the updates are aggregated to form an improved global model. In the simplest case, the aggregation process involves averaging the model parameters across clients, as in the FedAvg algorithm.

The **FedProx** algorithm (Algorithm 1) extends the classic FedAvg algorithm by introducing a proximal term to the local objective function [14]. This proximal term penalizes large deviations from the global model during local updates, thereby stabilizing the training process and promoting smoother convergence in heterogeneous data environments.

The **FedPer** strategy introduces a personalization layer within the neural networks, which is learned locally and allows for tailored model adaptation [1]. While federated learning (FL) mechanisms enable clients to benefit from knowl-

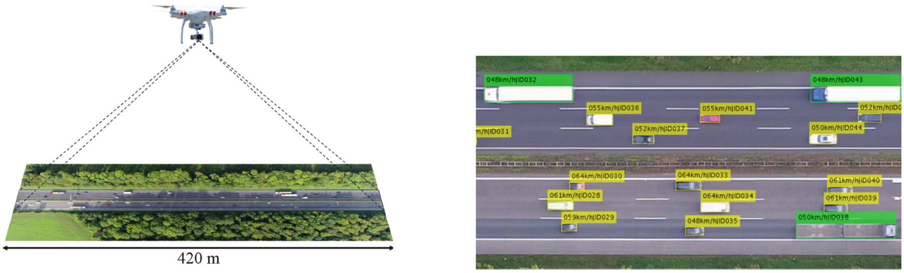

Fig. 2. Highway Drone Dataset. Adapted from [13]

edge shared among them, incorporating local personalization can also be advantageous. In the FedPer framework, the neural network is divided into two distinct components: the base layers and the personalization layers. During optimization, the learning process in clients must determine which aspects of the model should be treated locally in the personalization layer and which should be shared globally in the base layers. This distinction becomes crucial when clients have varying data distributions. In our setting, personalization is particularly important for adapting to different locations, each with its unique characteristics.

4 Lane-Change Dataset Overview

Two primary datasets commonly used in lane-change research are the NGSIM (Next Generation Simulation) Dataset [21] and the highD (Highway Drone Dataset) [13]. While the NGSIM dataset has been widely referenced, it suffers from limitations related to data quality and accuracy [4]. In light of these issues, we have chosen to focus our extensive analysis on the highD dataset.

The Highway Drone Dataset, called highD, is a comprehensive collection of naturalistic vehicle trajectories recorded on German highways and offers a detailed view of traffic patterns and vehicle behavior. Each recording spans a highway segment approximately 420 m in length, captured from an aerial perspective that eliminates occlusions, as illustrated in Fig. 2. The dataset encompasses traffic recordings from six distinct locations, including data from over 110,500 vehicles, covering 44,500 driven kilometers and 147 h of driving time. Vehicles are typically visible for a median duration of 13.6 s due to the high speeds on highways.

The data collection is conducted using drones with cinematic capabilities, which hover above the highway. The highD dataset includes four files for each of its 60 recordings, ensuring convenient data handling.

The **recordingMeta** file has general information about each specific recording, regardless of the individual vehicles captured. It contains entries for the recording's identifier, frame rate, location identifier, speed limit, date of the recording, day of the week, start time, duration, total distance driven, total time driven, the total number of vehicles recorded, and the y-coordinates of upper and lower lane markings. The **tracksMeta** file provides metadata about each recorded vehicle. It includes information such as the vehicle's identifier (id), width, height, initial etc. The **tracks** file comprises the actual movement data of the vehicles. Each row corresponds to a vehicle's position at a specific time frame, including data such as the frame index, vehicle identifier, coordinates (x, y), width, height, velocities, accelerations, front and back sight distances, distance headway, time headway, and time to collision with the leader vehicle.

Figure 3 shows an example of the trajectories from location 1 with three lanes for each driving direction. The `x` markings indicate lane-changes where the lanes are shown with dashed lines.

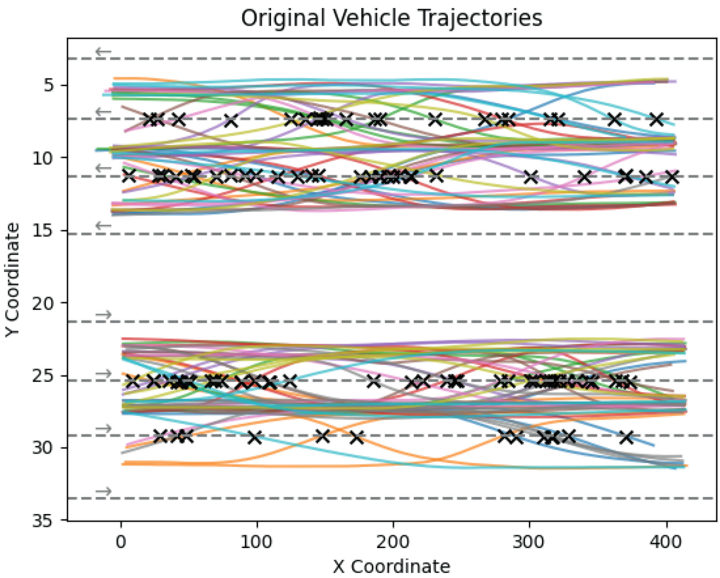

Fig. 3. Vehicles from Location 1

Next, we explore key descriptive statistics to understand the basic properties of the dataset. Table 1 provides a summary of the vehicle data collected at six different locations. It includes the number of recordings, trajectories, trucks, and cars observed at each location, along with the totals across all locations. The most significant observation is that the majority of the recordings (more than 60%) are from location 1. In addition, location 1 features 77% of recorded vehicles. Another noteworthy observation is that there are two types of vehicles, cars, and trucks, and the majority of the vehicles are from cars compared to trucks with a 4:1 ratio. Locations 2, 4, 5, and 6 have no speed limit, and locations 1 and 3 have 33.33 m/s (120 km/h) and 36.11 m/s (130 km/h) speed limits, respectively. This is already leading to some variations in the data distribution.

Table 1. Summary of Vehicle Data by Location

Locations	1	2	3	4	5	6	Total
# Recordings	37	3	3	4	10	3	60
# Vehicles	85962	3074	3747	4751	10079	2903	110516
# Trucks	16211	674	1037	952	1887	616	21377
# Cars	69751	2400	2710	3799	8192	2287	88939
% of vehicles	77.76%	2.78%	3.39%	4.30%	9.12%	2.63%	100%

Data Labeling

The dataset provides labeled *lane IDs* for each trajectory. We assign labels to frames where the *lane ID* changes as either *left lane-change (LLC)* or *right lane-change (RLC)*, while the remaining frames are marked as *lane-keep (LK)*. Figure 4 illustrates this process, using 1 s of historical data. The data is collected at 25 frames per second, enabling our predictions of lane-changes with a precision of 0.04 s. However, to accommodate the inherent variability in predictions, we extend the positive labels (LLC, RLC) by 12 frames before and after the original lane-change frame. This extension allows the model to predict lane changes within a broader 24 frames range (about 1 s).

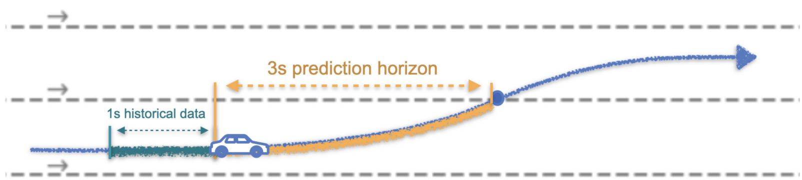

Fig. 4. Illustration of Labeling and historical Data

5 Model Development and Training

In this section, we delve into the development and training of our lane-change prediction model. A Long Short-Term Memory (LSTM) [10] network is chosen for lane-change prediction due to its capability to handle sequential data and capture long-term dependencies.

Before moving to the experiments and results, it is essential to understand the types of training setups used in this study. There are three learning configurations: centralized learning, federated learning, and local learning. The model trained in the centralized setup serves as a baseline to compare and validate the results of the federated learning. This comparison allows us to quantify the trade-off between privacy protection and prediction accuracy. Additionally, examining the model performance from local data learning is crucial to understanding the motivation behind collaboration.

Centralized Training. In the centralized setup, all training data is aggregated and processed on a central server. All sixty recordings are utilized to create the training, validation, and test sets. To ensure fair representation from each location, the data is first split into three subsets per location (see Fig. 5).

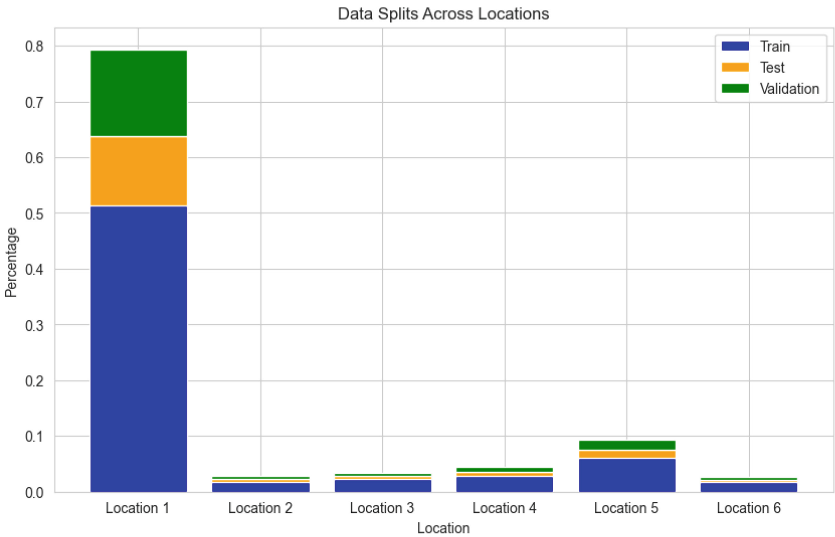

Fig. 5. Centralized: Data from all locations is merged into unified train, test, and validation sets. **Federated**: Train and validation sets are combined per location for training, with test sets remaining unchanged. **Local**: Each location uses its own train, validation, and test sets independently.

Each subset of sequences undergoes preprocessing before being vertically merged, shuffled, and divided into smaller batches. This method guarantees that approximately 20% of the data from each location is allocated for testing the model's generalization capability.

To mitigate overfitting, we employ early stopping alongside model-embedded regularization techniques. Training is halted if the model's performance on the validation set does not improve for a specified number of epochs, referred to as the *patience parameter*. This approach helps to maintain the model's robustness while avoiding unnecessary computation.

In summary, the centralized training in addition to the model-specific **hyperparameters** uses patience for early stopping, batch size, and number of training epochs. These hyperparameters are optimized to enhance overall model performance.

Federated Training. In FL, training data is distributed across multiple agents, and the model is collaboratively trained while preserving the data privacy for each client. In our case, the data is organically split between the six clients representing different highways. As introduced in Sect. 4, the samples per client vary significantly, with Client 1 providing almost 80% of the data. This creates inherent differences between the locations. In addition to quantitative differences, naturally, there are also variations in data distributions.

Similar to centralized setups, the FL training requires tuning **hyperparameters** such as learning rate, batch size, and local epochs to optimize performance. Additionally, determining the optimal *rounds* parameter is crucial, as it specifies the number of communication rounds between the server and clients. FL introduces specialized parameters for aggregation strategies, such as the *proximal* μ for **FedProx** and the partitioning of the neural network into base and personalization layers for **FedPer**.

Local Training. In local training, each agent independently trains its model using only local data, ensuring complete privacy and specialization to its location's characteristics, while following the same steps as the centralized setup without data sharing or model updates between agents.

6 Experimental Evaluation

This chapter presents the outcomes of our experiments, detailing the performance and evaluation of the FL approach for lane-change prediction. The subsections below cover the results of the experiments conducted across centralized, local, and federated setups, focusing on prediction horizons (PH) of 1, 3, and 5 s. The experiments were conducted using the hardware and software configuration as detailed in the Appendix. Further details on the evaluation can also be found in the thesis of the first author [27].

6.1 Centralized Training Results

Using the hyperparameter values outlined in the Appendix, Table 9, we trained centralized models for prediction horizons of 1, 3, and 5 s. The results in Table 2 and Fig. 6 demonstrate varying levels of accuracy, with the most notable performance observed in the 3-s prediction horizon, achieving an accuracy of 84.9% using only 1 s of historical data.

6.2 Local Training Results

The accuracy and F1 scores of local training on all 6 locations with prediction horizons 1, 3, and 5 s are presented in Table 3.

The first observation we make is that the prediction results significantly vary across different locations. The model performs exceptionally well in location 1, having the highest scores in all metrics. Compared to other locations, it also has the lowest validation loss and the most stable training process, indicating effective model performance and good data quality.

Table 2. Centralized Learning: Testing and validation results for 1 s, 3 s, and 5 s prediction horizons. The training values are in parentheses

Prediction Horizon	1 s	3 s	5 s
Testing			
Accuracy	0.941 (0.931)	**0.849** (0.836)	0.589 (0.648)
F1 Scores (LK, LLC, RLC)	0.910, 0.941, 0.938	0.798, 0.863, 0.858	0.587, 0.540, 0.644
Validation			
Accuracy	0.944	0.831	0.528
F1 Scores (LK, LLC, RLC)	0.921, 0.954, 0.939	0.787, 0.854, 0.837	0.499, 0.442, 0.626
Loss	0.190	0.428	0.901

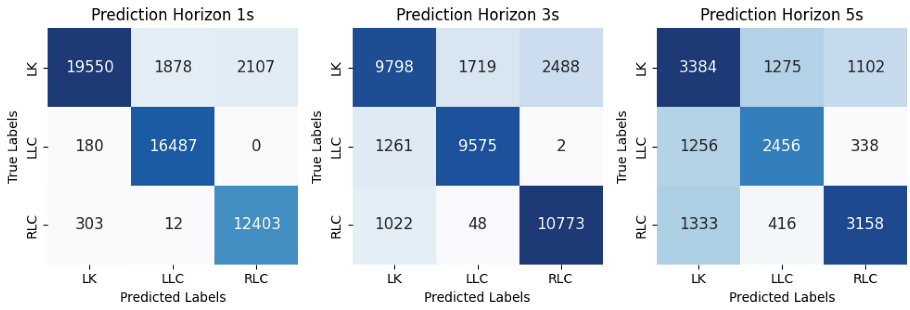

Fig. 6. Centralized Learning: Confusion matrices

Location 5 comes in second place, maintaining relatively good performance even for longer horizons, with smaller gaps between training and validation losses. The F1 scores for location 5 are also relatively high across all classes, showing its ability to make consistent predictions.

All locations perform fairly well for the 1 s PH according to the evaluation scores. However, the training plots show that the validation losses transition from horizontal lines to an upward trend throughout all 150 epochs, indicating the models' struggles to generalize. This pattern of increasing validation loss, indicative of overfitting, is particularly noticeable for longer prediction horizons.

For the 3 s PH, the gap between training and validation loss consistently increases in locations 2 and 3, highlighting overfitting and suggesting that the models are capturing noise in the training data. For the 5 s PH, the gap increases in all six locations with very large oscillations, demonstrating significant difficulties in making longer-term predictions. The confusion matrices validate our earlier observations. Notably, across all locations, the models exhibit almost perfect predictions when it comes to distinguishing between left lane-change and right lane-change. Across all locations, there are minimal instances where LLC is misclassified as RLC or vice versa.

Table 3. Local Learning: Testing results for 1 s, 3 s, and 5 s prediction horizons across 6 locations. The training values are in parentheses.

Location	Metric	1 s	3 s	5 s
Location 1	Accuracy	0.955 (0.945)	0.851 (0.825)	0.610 (0.672)
	F1 Scores (LK, LLC, RLC)	0.936, 0.950, 0.958	0.823, 0.880, 0.864	0.567, 0.577, 0.655
Location 2	Accuracy	0.860 (0.959)	0.546 (0.919)	0.630 (0.873)
	F1 Scores (LK, LLC, RLC)	0.858, 0.896, 0.860	0.453, 0.541, 0.678	0.534, 0.541, 0.802
Location 3	Accuracy	0.782 (0.934)	0.594 (0.93)	0.484 (0.885)
	F1 Scores (LK, LLC, RLC)	0.789, 0.781, 0.826	0.580, 0.551, 0.659	0.410, 0.490, 0.470
Location 4	Accuracy	0.821 (0.942)	0.677 (0.936)	0.402 (0.925)
	F1 Scores (LK, LLC, RLC)	0.800, 0.858, 0.820	0.661, 0.704, 0.675	0.257, 0.311, 0.579
Location 5	Accuracy	0.872 (0.943)	0.703 (0.896)	0.618 (0.800)
	F1 Scores (LK, LLC, RLC)	0.864, 0.855, 0.909	0.683, 0.726, 0.715	0.524, 0.637, 0.693
Location 6	Accuracy	0.792 (0.932)	0.615 (0.904)	0.409 (0.910)
	F1 Scores (LK, LLC, RLC)	0.788, 0.810, 0.784	0.644, 0.597, 0.555	0.498, 0.466, 0.097

Table 4. Accuracies of Federated Learning Using FedAvg, FedProx and FedPer: Testing results for 1 s, 3 s, and 5 s prediction horizons over all 6 locations

FL Algorithm	Accuracy for prediction horizon		
	1 s	3 s	5 s
FedAvg	0.927	0.818	0.576
FedProx	0.923	0.831	0.576
Fedper	0.931	0.841	0.567

6.3 Federated Learning Results

The federated learning was used to train one model per prediction horizon for three local epochs and 30 federated rounds. The training accuracy for all three algorithms, FedAvg, FedProx and FedPer are show in Fig. 4. Then, the models were evaluated on the client side at each location, and the results of these evaluations are presented in Table 5 for FedAvg, in Table 6 for FedProx and in Table 7 for FedPer.

For the main results of the three algorithms—FedAvg, FedProx, and FedPer, we refer to Table 4, which shows that FedProx and FedPer outperform FedAvg.

Table 5. Federated Learning Using FedAvg: Testing results for 1 s, 3 s, and 5 s prediction horizons across 6 locations, including F1 scores for each horizon.

Location	Metric	1 s	3 s	5 s
Location 1	Accuracy	0.935	0.824	0.575
	F1 Scores LK, LLC, RLC	0.907, 0.960, 0.956	0.739, 0.874, 0.891	0.539, 0.572, 0.618
Location 2	Accuracy	0.943	0.764	0.636
	F1 Scores LK, LLC, RLC	0.920, 0.973, 0.954	0.681, 0.854, 0.832	0.534, 0.652, 0.735
Location 3	Accuracy	0.823	0.804	0.453
	F1 Scores LK, LLC, RLC	0.783, 0.921, 0.793	0.734, 0.869, 0.854	0.585, 0.341, 0.472
Location 4	Accuracy	0.933	0.824	0.638
	F1 Scores LK, LLC, RLC	0.908, 0.962, 0.950	0.752, 0.885, 0.902	0.682, 0.541, 0.732
Location 5	Accuracy	0.939	0.829	0.695
	F1 Scores LK, LLC, RLC	0.912, 0.954, 0.966	0.757, 0.891, 0.893	0.638, 0.686, 0.742
Location 6	Accuracy	0.916	0.754	0.522
	F1 Scores LK, LLC, RLC	0.881, 0.916, 0.971	0.692, 0.790, 0.833	0.588, 0.423, 0.492
Overall	Accuracy	**0.927**	**0.818**	**0.576**

6.4 Discussion

Centralized models generally outperformed **local models** due to their access to a more diverse and comprehensive dataset, achieving a high accuracy of 94.1% compared to a weighted average of 91.4% from local models for the 1-s prediction horizon. For the 3-s prediction horizon, centralized models achieved 84.9% accuracy versus 79.1% from local models, and for the 5-s horizon, centralized models had an accuracy of 58.9% compared to 56.2% from local models. In contrast, local models, constrained to data from specific locations, often fail to capture the full spectrum of driving behaviors and conditions, leading to lower performance. This effect is especially pronounced in our case, as location 1 has substantially more data than the other locations, causing a notable drop in accuracy for less represented areas.

Centralized models typically achieve higher accuracy than **federated models** due to their direct access to a the full dataset. In our work, for the 1 s horizon, CL models reached an accuracy of 94.1%, compared to 93.1% from federated models (FedPer). Similarly, for the 3 s PH, centralized models achieved 84.9% accuracy compared to 84.1% from federated models, and for 5 s PH 58.9%, compared to 56.7%.

The accuracy difference between federated and centralized models is small, with federated models showing up to a 1% decrease in accuracy compared to

Table 6. Federated Learning Using FedProx: Testing results for 1 s, 3 s, and 5 s prediction horizons across 6 locations, including F1 scores for each horizon.

Location	Metric	1 s	3 s	5 s
Location 1	Accuracy	0.935	0.843	0.575
	F1 Scores LK, LLC, RLC	0.914, 0.962, 0.960	0.761, 0.887, 0.895	0.564, 0.563, 0.662
Location 2	Accuracy	0.926	0.755	0.585
	F1 Scores LK, LLC, RLC	0.900, 0.954, 0.954	0.658, 0.846, 0.786	0.453, 0.618, 0.750
Location 3	Accuracy	0.792	0.805	0.545
	F1 Scores LK, LLC, RLC	0.758, 0.893, 0.769	0.742, 0.854, 0.866	0.553, 0.464, 0.657
Location 4	Accuracy	0.917	0.815	0.565
	F1 Scores LK, LLC, RLC	0.891, 0.949, 0.943	0.725, 0.884, 0.867	0.571, 0.582, 0.609
Location 5	Accuracy	0.946	0.825	0.645
	F1 Scores LK, LLC, RLC	0.932, 0.961, 0.975	0.768, 0.885, 0.902	0.475, 0.679, 0.764
Location 6	Accuracy	0.909	0.765	0.495
	F1 Scores LK, LLC, RLC	0.881, 0.928, 0.958	0.684, 0.800, 0.836	0.577, 0.411, 0.505
Overall	Accuracy	**0.923**	**0.831**	**0.576**

centralized models for shorter PH. Although federated learning is often benchmarked against centralized solutions, in many cases, centralized solutions are impractical or unfeasible when data cannot be consolidated at a central location, often due to privacy or confidentiality requirements.

Comparing the results of locally trained models with those trained collaboratively in a federated setting is particularly interesting as it goes into the motivation behind local clients choosing to participate in FL. By doing so, clients contribute to and benefit from a collective intelligence that usually outperforms isolated local efforts.

Federated learning models (FedPer) demonstrate superior overall performance compared to the average performance of local models. The accuracies for FedPer and the average local accuracies are 93.1%, 84.1%, and 56.7% versus 91.4%, 79.1%, and 56.2% for the 1 s, 3 s, and 5 s prediction horizons, respectively. This indicates that FedPer models consistently outperform local models on average.

A more detailed analysis per location reveals that FL significantly improves short-term prediction accuracy (1 s and 3 s PH) for all locations besides 1. For example, locations 2, 4, and 6 see around a 10% increase in accuracy for the 1 s PH under FL. For the 3 s PH, the improvement is even more pronounced, with some locations experiencing up to a 15% increase. Specifically, location 2 sees an impressive 26% increase in accuracy for the 3-s PH. This substantial

Table 7. Federated Learning Using FedPer: Testing results for 1 s, 3 s, and 5 s prediction horizons across 6 locations, including F1 scores for each horizon.

Location	Metric	1 s	3 s	5 s
Location 1	Accuracy	0.939	0.860	0.581
	F1 Scores (LK, LLC, RLC)	0.901, 0.951, 0.960	0.775, 0.899, 0.895	0.312, 0.604, 0.694
Location 2	Accuracy	0.942	0.809	0.526
	F1 Scores (LK, LLC, RLC)	0.909, 0.957, 0.959	0.735, 0.903, 0.795	0.464, 0.435, 0.800
Location 3	Accuracy	0.767	0.749	0.476
	F1 Scores (LK, LLC, RLC)	0.720, 0.869, 0.724	0.672, 0.771, 0.822	0.356, 0.388, 0.589
Location 4	Accuracy	0.944	0.829	0.486
	F1 Scores (LK, LLC, RLC)	0.912, 0.955, 0.965	0.754, 0.892, 0.845	0.527, 0.431, 0.687
Location 5	Accuracy	0.928	0.831	0.599
	F1 Scores (LK, LLC, RLC)	0.885, 0.948, 0.944	0.758, 0.894, 0.832	0.113, 0.621, 0.759
Location 6	Accuracy	0.899	0.718	0.495
	F1 Scores (LK, LLC, RLC)	0.850, 0.909, 0.948	0.658, 0.794, 0.712	0.184, 0.642, 0.794
Overall	Accuracy	**0.931**	**0.841**	**0.567**

improvement likely results from the similarity in data across different locations, allowing the federated model to benefit from the shared information.

For the 5-s horizon, the overall performance of FL models is better than that of local models, but individual location results are mixed, with some locations showing slight increases and others slight decreases in accuracy. However, a notable advantage of FL is the more stable learning process, as evidenced by the accuracy plots. FL reduces the variability seen in individual locations, leading to more consistent performance improvements.

While the performance quality for the 1 s and 5 s horizons drops slightly in location 1, this decline is minimal. The slight decrease can be attributed to the aggregation of model parameters from all locations, which introduces a mix of diverse data characteristics. In a non-IID setup with significant quantity skewness, clients that carry the majority of data, such as location 1, may experience a minor reduction in performance. However, this reduction is marginal compared to the significant gains observed in locations with poorer data quality.

This indicates that FL is beneficial for all participating locations. Locations with poorer data and lower performance see substantial improvements, making the overall system more robust and reliable. At the same time, locations with better data quality, like location 1, do not experience significant degradation in their performance. This ensures that all participants benefit from the collaborative training process.

Comparison with Existing Literature. To validate our results for LC prediction, we compare them to existing works in the field. As existing literature does not apply FL for this specific task there is no direct comparison as machine learning for distributed data is more difficult. For reference, we compare both our CL and FL models to existing centralized works. The most comparable work is [15], which uses automatic labeling by clustering LC and LK maneuvers. They then train an SVM to learn the boundaries of these clustered labels, which are subsequently used as inputs for an LSTM model to predict the maneuver class.

This work also utilizes the highD dataset, predicting maneuvers using 1-s historical data with prediction horizons of 0.5, 1, 2, and 3 s. Their approach differs from ours in several ways. Firstly, they use only one recording from location 1 for training and sample additional trajectories from all locations for testing. In contrast, we train our models across multiple multilane locations, leveraging a more diverse dataset.

Another significant difference is in the feature set used for prediction. While they focus primarily on lateral velocity and acceleration, our approach incorporates both lateral and longitudinal values as well as features from surrounding vehicles. This difference in the feature set provides a more comprehensive understanding of vehicle dynamics in our model. Moreover, their model does not distinguish between left and right LC, whereas our approach explicitly accounts for the direction of LC.

Most importantly, the labeling process also differs significantly. They employ a complex clustering technique for automatic labeling, whereas we use a simpler approach by adding a 0.5-s tolerance around the exact LC frame. Their clustering and learning approach with SVM is heavily based on local data from a specific location, which might limit the generalizability of their model to other environments or datasets. In contrast, our approach uses a fixed ground truth for labeling, which can be applied universally across various locations and conditions, ensuring consistent and comparable results.

While there are major differences in privacy assumptions, lane change marking and data used, our FL models achieve competitive performance metrics compared to [15]. For instance, our centralized model achieves an accuracy of 94.1% for the 1 s PH, which is not far off the reported performance of 97.6%. This demonstrates that our simpler labeling approach and diverse dataset can effectively match the accuracy of more complex methods.

Our results demonstrate that both centralized and federated models are competitive with existing state-of-the-art approaches, with the added benefit that FL preserves data privacy and security. The minimal performance gap between centralized and federated models, coupled with the significant privacy advantages, underscores the potential of FL as a robust alternative for LC prediction tasks.

In summary, centralized models generally outperform local models due to a richer and more varied dataset, yet they may not always be optimal. High-performing locations might be disadvantaged when their specific high-quality data is diluted in a larger, more diverse dataset. FL effectively bridges this gap by leveraging diverse datasets from multiple locations while maintaining data privacy. This collaborative approach ensures that locations with poorer data and lower performance see substantial improvements, while high-performing locations experience only minimal performance degradation. This balanced and equitable solution enhances overall model accuracy and provides stability, making FL a compelling alternative that benefits all participants.

7 Conclusion

In this paper, we have presented a data-driven, federated learning framework to address the critical task of lane-change prediction, with the overarching goal of enhancing highway safety. After reviewing the existing approaches with different definitions of lane change and labeling, we started with a data-driven approach without trajectory prediction, AI-based labeling or similar. We developed and implemented a preprocessing pipeline tailored for handling multiple multivariate time series datasets, for both centralized and federated learning systems.

Our framework performs similarly to existing "classic" centralized learning approaches, where all data is centralized. As discussed, variations in definitions, labeling, and other factors underscore the need for standardized definitions and reproducible benchmarks. Specifically, we found that some local models degrade significantly due to limited amount of data. Additionally, differences in location characteristics, like speed limits or lane numbers, contribute to non-IID distributions [30], a known challenge in federated learning.

For the federated learning case, we have achieved the following main results:

- Federated learning performs, using all 6 locations, very close to centralized case, while preserving and protecting local data.
- Federated learning is able to improve the results for the locations with very little data. This clearly shows that federated learning enables the transfer the learnings between heterogeneous locations successfully.
- Personalization in federated learning, here applied to different locations, is shown to improve the results. This indicates that the differences in the locations can be handled better in this case.

In summary, our study demonstrates that federated learning can be effective for applications like lane keeping, representing an important step toward larger-scale use of AI-based, data-driven methods. This approach enables leveraging data from diverse, heterogeneous locations that cannot be centralized, often due to privacy, legal, or company confidentiality constraints.

A Appendix: Hardware, Software and Hyperparameter Setup

Hardware and Software Setup

- NVIDIA GeForce RTX 4090 GPU with 24,564 MiB memory (Driver Version: 550.54.15, CUDA Version: 12.4)
- Operating System: Ubuntu 20.04
- Programming Language: Python 3.9
- Deep Learning Framework: PyTorch
- Federated Learning Framework: Flower
- Data Processing and Visualization Libraries: Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib.

Flower orchestrated the client-server interactions in a synchronous manner, utilizing the server's GPU to accelerate training processes and manage the distribution, aggregation, and communication. This setup provided efficient and scalable FL, with consistent configuration across centralized, local, and federated models to maintain fairness in comparisons.

The following hyperparameters were varied across specified ranges (Table 8):

Table 8. Hyperparameter Ranges used in Grid Search

Hyperparameter	Values
Hidden Size	[32, 64]
Number of Layers	[2, 3, 4]
Learning Rate	[0.00001, 0.0001, 0.001, 0.01]
Batch Size	[16, 32]

The table below summarizes the best-performing models and hyperparameters selected through grid search for each prediction horizon based on their independent test set performance metrics. These exact values are used throughout our work for each prediction horizon regardless of the training setup to ensure fair comparisons of the model performance.

Table 9. Best Hyperparameters

Prediction Horizon(s)	Hidden Size	Number of Layers	Learning Rate	Batch Size
1	64	2	0.001	32
3	32	2	0.001	16
5	64	4	0.0001	32

In addition to the hyperparameters tuned through grid search, several other critical parameters were determined through a combination of literature review and empirical testing. These include a historical window size of 1 s, which was optimal after testing 0.5, 1, 2, and 3 s, the dropout rate of 0.3, weight decay of 0.01, patience for early stopping set to 10 epochs, and the FedProx parameter (proximal μ) of 0.7. While these parameters were not exhaustively tuned through grid search, they were carefully selected to complement the model's architecture and training regime.

References

1. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)
2. Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., Borrelli, F.: Automated driving: the role of forecasts and uncertainty-a control perspective. Eur. J. Control. **24**, 14–32 (2015)
3. Chauhan, P., Kanagaraj, V., Asaithambi, G.: Understanding the mechanism of lane changing process and dynamics using microscopic traffic data. Phys. A: Stat. Mech. Appl. **593**, 126981 (2022). https://doi.org/10.1016/j.physa.2022.126981. https://www.sciencedirect.com/science/article/pii/S0378437122000735
4. Coifman, B., Li, L.: A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset. Transp. Res. Part B: Methodol. **105**, 362–377 (2017)
5. Dang, H.Q., Fürnkranz, J., Biedermann, A., Hoepfl, M.: Time-to-lane-change prediction with deep learning. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–7. IEEE (2017)
6. Díaz, J.S.P., García, Á.L.: Study of the performance and scalability of federated learning for medical imaging with intermittent clients. Neurocomputing **518**, 142–154 (2023)
7. Du, R., Han, K., Gupta, R., Chen, S., Labi, S., Wang, Z.: Driver monitoring-based lane-change prediction: a personalized federated learning framework. In: 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1–7. IEEE (2023)
8. Fitch, G., Lee, S., Klauer, S., Hankey, J., Sudweeks, J., Dingus, T.: Analysis of lane-change crashes and near-crashes. US Department of Transportation, National Highway Traffic Safety Administration (2009)
9. Hard, A., et al.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Izquierdo, R., Parra, I., Muñoz-Bulnes, J., Fernández-Llorca, D., Sotelo, M.: Vehicle trajectory and lane change prediction using ANN and SVM classifiers. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE (2017)
12. Khelfa, B., Ba, I., Tordeux, A.: Predicting highway lane-changing maneuvers: a benchmark analysis of machine and ensemble learning algorithms. Phys. A **612**, 128471 (2023)
13. Krajewski, R., Bock, J., Kloeker, L., Eckstein, L.: 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), pp. 2118–2125. IEEE (2018)

14. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. **2**, 429–450 (2020)
15. Mahajan, V., Katrakazas, C., Antoniou, C.: Prediction of lane-changing maneuvers with automatic labeling and deep learning. Transp. Res. Rec. **2674**(7), 336–347 (2020)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
17. Morris, B., Doshi, A., Trivedi, M.: Lane change intent prediction for driver assistance: on-road design and evaluation. In: 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 895–901. IEEE (2011)
18. Mozaffari, S., Arnold, E., Dianati, M., Fallah, S.: Early lane change prediction for automated driving systems using multi-task attention-based convolutional neural networks. IEEE Trans. Intell. Veh. **7**(3), 758–770 (2022)
19. Ozguner, U., Stiller, C., Redmill, K.: Systems for safety and autonomous behavior in cars: the DARPA grand challenge experience. Proc. IEEE **95**(2), 397–412 (2007)
20. Ponziani, R.: Turn signal usage rate results: a comprehensive field study of 12,000 observed turning vehicles. Technical report, SAE Technical Paper (2012)
21. U.S. Department of Transportation Federal Highway Administration: Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data. Dataset (2016). https://doi.org/10.21949/1504477
22. Wirthmüller, F., Klimke, M., Schlechtriemen, J., Hipp, J., Reichert, M.: Predicting the time until a vehicle changes the lane using LSTM-based recurrent neural networks. IEEE Robot. Autom. Lett. **6**(2), 2357–2364 (2021)
23. Wirthmüller, F., Schlechtriemen, J., Hipp, J., Reichert, M.: Teaching vehicles to anticipate: a systematic study on probabilistic behavior prediction using large data sets. IEEE Trans. Intell. Transp. Syst. **22**(11), 7129–7144 (2020)
24. Wissing, C., Nattermann, T., Glander, K.H., Hass, C., Bertram, T.: Lane change prediction by combining movement and situation based probabilities. IFAC-PapersOnLine **50**(1), 3554–3559 (2017)
25. Woo, H., et al.: Lane-change detection based on vehicle-trajectory prediction. IEEE Robot. Autom. Lett. **2**(2), 1109–1116 (2017)
26. Yang, X., Tang, L., Stewart, K., Dong, Z., Zhang, X., Li, Q.: Automatic change detection in lane-level road networks using GPS trajectories. Int. J. Geogr. Inf. Sci. **32**(3), 601–621 (2018)
27. Yenokyan, L.: Personalised federated learning: lane-change prediction. Master's thesis, LMU Munich (2024)
28. Yuan, R.: A comparative analysis of machine learning methods for lane change intention recognition using vehicle trajectory data. arXiv preprint arXiv:2307.15625 (2023)
29. Zhang, W., Huang, Y.H., Roetting, M., Wang, Y., Wei, H.: Driver's views and behaviors about safety in China-what do they not know about driving? Accid. Anal. Prev. **38**(1), 22–27 (2006)
30. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)