Driver Digital Twin for Online Prediction of Personalized Lane-Change Behavior

Xishun Liao, *Graduate Student Member, IEEE*, Xuanpeng Zhao, Ziran Wang, *Member, IEEE*, Zhouqiao Zhao, *Graduate Student Member, IEEE*, Kyungtae Han, *Senior Member, IEEE*, Rohit Gupta, Matthew J. Barth, *Fellow, IEEE*, and Guoyuan Wu, *Senior Member, IEEE*

***Abstract*—Connected and automated vehicles (CAVs) are supposed to share the road with human-driven vehicles (HDVs) in a foreseeable future. Therefore, considering the mixed traffic environment is more pragmatic, as the well-planned operation of CAVs may be interrupted by HDVs. In the circumstance that human behaviors have significant impacts, CAVs need to understand HDV behaviors to make safe actions. In this study, we develop a driver digital twin (DDT) for the online prediction of personalized lane-change behavior, allowing CAVs to predict surrounding vehicles' behaviors with the help of the digital twin technology. DDT is deployed on a vehicle-edge–cloud architecture, where the cloud server models the driver behavior for each HDV based on the historical naturalistic driving data, while the edge server processes the real-time data from each driver with his/her digital twin on the cloud to predict the personalized lane-change maneuver. The proposed system is first evaluated on a human-in-the-loop co-simulation platform, and then in a field implementation with three passenger vehicles driving along an on/off ramp segment connecting to the edge server and cloud through the 4G/LTE cellular network. The lane-change intention can be recognized in 6 s on average before the vehicle crosses the lane separation line, and the Mean Euclidean Distance between the predicted trajectory and GPS ground truth is 1.03 m within a 4-s prediction window. Compared to the general model, using a personalized model can improve prediction accuracy by 27.8%. The demonstration video of the proposed system can be watched at https://youtu.be/5cbsabgIOdM.**

***Index Terms*—Connected and automated vehicle (CAV), digital twin, driver behavior modeling, field implementation, lane-change prediction.**

I. Introduction

A. Motivation

THE RAPID development of connected and automated vehicles (CAVs) provides a new perspective on addressing the safety, mobility, and environmental sustainability issues of our transportation systems [1]. However, our transportation systems cannot achieve full automation/connectivity in a foreseeable future, where CAVs have to interact with human-driven vehicles (HDVs) in a mixed traffic environment. To deal with complex interactions, CAVs need to predict HDVs' behaviors, make decisions in response to the actions (in presence or to be taken), and execute the right maneuvers through the planner and controller. Particularly, the prediction of HDVs' behaviors is challenging due to the uncertainties of human drivers. As a result, many researchers [2], [3], [4], [5] include driving behavior modeling in the prediction and planning stage, where driver type classification and identification play an important role. Nevertheless, the improvement of integrating the collective driving behavior is limited, because driving behavior can be diverse among different drivers. Therefore, there is an increasing amount of literature that recognizes the importance of personalized driving behaviors.

To study personalized driving behavior, this article deploys the driver digital twin (DDT) in the real world. DDT is a digital replica of a driver with his or her naturalistic driving data and driving behavior models. Based on real-world data, DDT system in the virtual world provides both online and offline micro services, e.g., interactive prediction, driving style analysis, etc. In addition, to capture the driving preference variation, the evolving driver model will be updated in a certain period (e.g., every five new trips) by consuming the driving data from the real-world vehicle.

In our previous work, the vehicle digital twin is built on an edge server [6], allowing the planning and control of the connected vehicles. However, only using an edge server is sometimes insufficient to fulfill the requirements of personalized behavior study, such as data storage, modeling, learning, simulation, and prediction [7]. Therefore, DDT in this study is performed on a cloud–edge architecture, leveraging cloud computing and personalized profiling, enabling both real-time and bulk-batch ingestion, processing, and analytics of personal data.

As a typical example of personalized driver behavior that can be modeled by DDT, lane changing is a fundamental but challenging task in our daily driving, especially in mandatory lane change situations (such as ramp merging) where the open areas are very limited and levels of risk are higher, compared to discretionary lane changes. In these situations, it is particularly important to accurately predict lane-change intention as well as when and where the lane change occurs, because the

Manuscript received 2 November 2022; revised 27 February 2023; accepted 24 March 2023. Date of publication 27 March 2023; date of current version 25 July 2023. This work was supported by the Toyota Motor North America, InfoTech Labs under the Digital Twin Project. *(Corresponding author: Xishun Liao.)*

Xishun Liao, Xuanpeng Zhao, Zhouqiao Zhao, Matthew J. Barth, and Guoyuan Wu are with CE-CERT, University of California at Riverside, Riverside, CA 92507 USA (e-mail: xliao016@ucr.edu; xzhao094@ucr.edu; zzhao084@ucr.edu; barth@ece.ucr.edu; gywu@cert.ucr.edu).

Ziran Wang is with the College of Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ziran@purdue.edu).

Kyungtae Han and Rohit Gupta are with the Toyota Motor North America, InfoTech Labs, Mountain View, CA 94043 USA (e-mail: kyungtae.han@toyota.com; rohit.gupta@toyota.com).

Digital Object Identifier 10.1109/JIOT.2023.3262484

lane-change maneuver will have a significant impact on the safety and efficiency of the road network. Moreover, online prediction of the lane-change maneuver provides inputs to the motion planner and controller of ego vehicle, allowing it to better cooperate with surrounding vehicles (including HDVs). By building the personalized lane-change model, CAVs can have a better understanding of the specific human drivers and provide a more accurate prediction.

In this study, learning-based algorithms for personalized behavior modeling and online lane-change prediction are developed, and field implementations are carried out on a customized vehicle-edge–cloud platform under the digital twin framework. In the field implementation, the cloud (i.e., Amazon Web Services) analyzes the personalized behavior of HDVs with connectivity (e.g., by cellphone) and stores the learned driver models and historical data, while the edge (i.e., local server) is responsible for the computation of online lane-change prediction.

Compared to the existing literature on prediction and behavior modeling, this study has the following main contributions.

1) A hierarchical learning-based system is developed for personalized driving behavior modeling, online lane-change prediction, and trajectory likelihood estimation.
2) Under the digital twin framework, a vehicle-edge–cloud platform is constructed and demonstrated, enabling real-world data collection and algorithms development.
3) Personalized driver models are trained and validated using the vehicle-edge–cloud platform, as one of the first real-world deployments of DDT in transportation.
4) To validate the proposed algorithm in the field experiments, a portable vision-based human–machine interface (HMI) system is designed to provide the prediction information to the driver supported by edge computing and cloud computing.

B. Specifications and Assumptions

In this article, the target predicted vehicle is a connected HDV, whose historical/real-time data and the trained driver model are accessible through the digital twin. When other connected vehicles detect and recognize this target vehicle, they can download the driver model of the target vehicle to assist in the prediction. Specifically, our prediction algorithm is designed for the on/off-ramp scenario to predict the maneuver and trajectory of the on-ramp vehicle.

To expedite field implementation, some reasonable specifications and assumptions are made in the current stage of online lane-change prediction and personalized driving behavior modeling, as follows.

1) When the target vehicle comes into view, if ego vehicle is able to recognize it (e.g., by computer vision or V2X communications), the associated driver model is then acquired from the cloud server (with permission). Before target vehicle is recognized, a general driver model will be used instead.
2) The driving preference of the same driver is assumed to be long lasting and will not be affected by the mood on the testing day.
3) As designing the perception system (e.g., LiDAR, radar, and camera) is outside the scope of this article, the necessary vehicle information (e.g., location and yaw angle) is uploaded by the target vehicle and shared by the edge server.

C. Organization of This Article

The remainder of this article is organized as follows. In Section II, state-of-the-art research on lane-change prediction, personalized driver behavior modeling and digital twin platform design are reviewed. Section III explains the methodology of the proposed system, including personalized behavior modeling, online lane-change prediction, and map matching for real-world implementation. Section IV elaborates on the field experiment design using the vehicle-edge–cloud architecture. In Section V, we analyze the driving styles of two drivers and evaluate the performance of the proposed algorithm. Finally, this article is concluded with future directions in Section VI.

II. Background and Related Work

A. Lane Change Prediction

Hidden Markov model (HMM) was widely used to infer the lane-change intention [8], [9], [10] and is usually integrated with the Bayesian network [11] to recognize the lane-change behavior. As lane-change intention prediction can be modeled as a classified problem, the multilayer perceptron (MLP) was used as a discriminator [12] in long-term lane-change prediction. Moreover, deep-learning methods, such as long short-term memory (LSTM) model, achieved a precision of 90.5% on time-series problems [14]. To find out relevant features for lane changing in a time series, Scheel et al. [15] integrated a temporal attention mechanism with LSTM to improve the prediction accuracy to 92.6% and provide understandability on feature importance. Besides lane-change intention prediction, lane-change trajectory prediction is also a critical problem. Based on the beam search technique, Park et al. [16] adopted sequence-to-sequence (Seq2seq) LSTM to produce K most likely trajectories on an occupancy grid map. By adding information on traffic level and vehicle types, Xue et al. [17] adopted XGBoost for lane-change decision prediction and LSTM for trajectory prediction, achieving a mean-square error of 6.62 m for trajectory prediction.

However, most of the algorithms ignored vehicular interaction with the surroundings. Furthermore, supervised learning methods were limited by the lack of enough labeled data sets to cover each possible scenario, and very little online validation was carried out as well as real-world validation.

B. Personalized Driver Behavior Modeling

A personalized driver model is usually learned based on the data set from a specific driver (i.e., learning from demonstration) and is mainly used for prediction, planning, and control [18]. Drivers' preferences in their vehicle states are well studied. A personalized driving assistant system developed by Lefevre et al. [19] could identify the current driving maneuver

and predict the steering and acceleration, facilitating control assistance. Considering the occupants' preference for lateral and longitudinal comfort, Bae et al. [20] proposed a personalized control system enabling autonomous vehicles to drive like human beings. Also, algorithms based on Gaussian Mixture Model and HMM are popular for prediction of personalized driving behavior and are deployed in a variety of applications, such as personalized lane-departure prediction [21] and personalized car-following behavior prediction [22].

Not only the vehicle states but also the surroundings and vehicular interaction should be considered in drivers' preference. To integrate the interaction into behavior modeling, Huang et al. [23] included the awareness of the effect of the ego vehicle on the surrounding vehicles into the cost function. The driver's preference over vehicle states and interactions can be expressed by the cost function recovered by inverse reinforcement learning (IRL), which assumed that human behavior was motivated by optimizing the unknown reward function [24]. In [25], the interaction behavior under different conditions was formulated as a cost function with different linear combinations of features and learned by continuous IRL. The cost function with interpretable weights on features opens the black box of behavior modeling, by showing the diversity of feature attention in different scenarios. Furthermore, the cost function can be adjusted as needed to reflect changing driving conditions and can always provide a solution aligning with driver preference in unseen scenarios.

Collecting a personalized data set for interaction study is another major limitation in interaction modeling and driving personalization study. The difficulty of scenario reproduction constrains the driver experiences in similar interactive conditions and the collection of enough data for model training.

C. Digital Twins for Connected and Automated Vehicles

The recent emergence of digital twin technology has attracted a significant amount of attention from both academia and industry. The global digital twin market size was reported to be valued at U.S. $5 billion in 2020, and will be expanded to U.S. $86 billion in 2028, with a compound annual growth rate of 43%. Among all end-users like manufacturing, energy, and health care, the automotive and transportation industry took one of the largest shares in the global digital twin market in 2020 [26].

By a widely adopted definition (with some variations), a digital twin is a digital replica of a living or nonliving physical entity [7]. This concept got to be known by most people in the early 2010s, when NASA adopted it as a key element in its technology roadmap [27]. During the past few years, digital twins have been applied to different vehicular systems. Particularly, Chen et al. [28] developed a "Driver Behavior Twin" to allow driver behavioral models to be shared among multiple connected vehicles to predict future actions of surrounding vehicles. Although this study did not come up with a solid network architecture, its concept did inspire a series of subsequent studies by the authors of our paper.

In 2020, the authors first proposed a digital twin paradigm for advanced driver-assistance systems (ADAS) with a cloud architecture, which enables the communication between real vehicles and their digital twins deployed on the cloud server in real time [29]. This cooperative ramp merging ADAS was later validated in a field implementation with real passenger vehicles and a private cloud server at University of California, Riverside [6]. Later, the authors introduced edge computing to this network architecture by proposing a mobility digital twin framework, which includes not only vehicle digital twins but also human and infrastructure digital twins [7]. Some of the detailed aspects of digital twins for CAVs have also been studied by the authors, such as how to visualize the digital twin information [30], how to leverage the digital twin information for cooperative driving scenarios [31], and how to build a simulation environment to model digital twins [32]. Many of the aforementioned studies have been summarized in a survey paper, where the role of digital twins in CAVs is also compared with the roles of several similar technologies, such as iteration, model-based design, and parallel driving [33].

III. Methodology

A. Personalized Behavior Modeling Process

Based on our previous work [34], we extend the lane-change prediction algorithm into a real-world implementable version and enable personalized behavior modeling. As shown in Fig. 1, the algorithm consists of an offline learning process and an online prediction process. The personalized driving behavior for each driver is learned in the offline phase based on the personalized data set collected from the specific driver. For each driver, a neural network structure Seq2seq structure [35] based on LSTM is adopted to predict the lane-change intention, while the cost functions inferring the driver preference are learned by IRL for evaluating the prediction. In the online prediction process, at each time step, the personalized Seq2seq neural network recognizes the maneuver and selects a proper cost function. Then, the driver's cost function evaluates the probability of candidate trajectories provided by the trajectory generator. Finally, the system outputs are the most probable trajectory and the respective lane-change probability.

B. Lane-Change Maneuver Prediction

In order to analyze the trajectory in detail, we need to recognize the driver's intention. We assume that before planning the trajectory, the human driver first considers high-level tasks (e.g., lane change and lane keeping). Therefore, the lane-change intention prediction is formulated as a time-series classification problem, which predicts vehicle states in future time steps, i.e., either lane change or lane keeping. That is to classify the future T-step actions $A_{t:t+T}$ into $\{a_{\text{change}}, a_{\text{keep}}\}$, given historical vehicle states and the map information.

To model long-term temporal dependencies among time series, LSTM network is chosen, as its time series prediction capability is validated in many existing studies, e.g., [36], [37], and [38]. Since each vehicle state in the time series is highly correlated with its adjacent time steps, the Seq2seq neural network using two LSTMs [39] is adopted for a multistep and

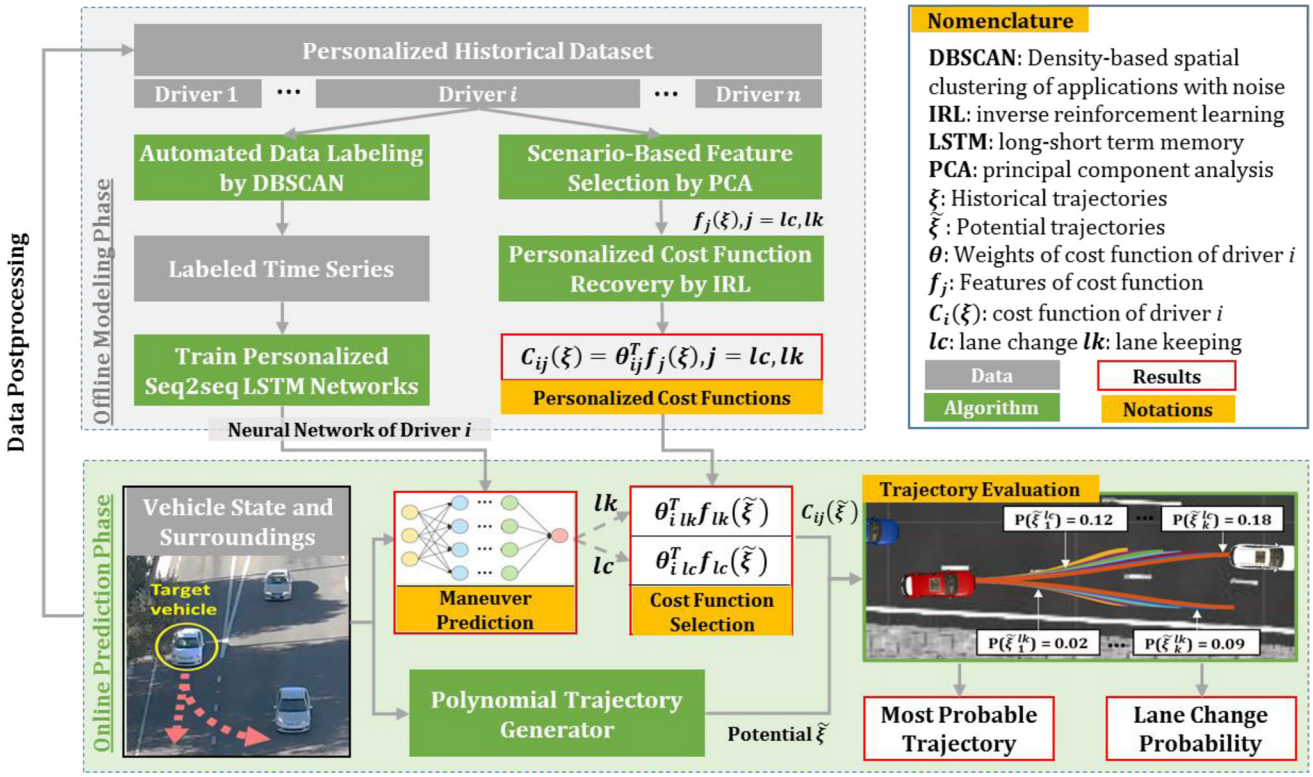


Fig. 1. Personalized lane-change behavior modeling: offline learning and online prediction.

multivariable prediction. The neural network input is a trajectory sequence $\xi = (s_{t-T+1}, \ldots, s_t)$ of the last T steps, and the vehicle states s_t consists of yaw angle, lateral speed, longitudinal speed, and remaining distance for a mandatory lane change. The output is the predicted lane-change action sequence $(A_{t+1}, \ldots, A_{t+T+1})$ for the next T steps.

C. Probability Estimation and Trajectory Prediction Based on IRL

The driver behavior and preference are usually represented by the cost function, and rational drivers are assumed to behave for optimizing their cost functions. Considering the continuity of the trajectory space, this study adopts Continuous IRL with locally optimal examples [40], [41] to recover this unknown cost function from expert demonstrations.

1) Continuous IRL: The cost function is a linear combination of a set of features, i.e., $C_i(\theta_i, \xi) = \theta_i^T f_i(\xi), i = a_{change}, a_{keep}$, where θ_i^T is the weights vector emphasizing the features, and $f_i(\xi) = *f_i(s_1, s_2, \ldots, s_t)*_2$. The goal of the IRL is to figure out the optimal weights θ_i^* to describe each driver's preference, which maximizes the likelihood of the driver's historical trajectories $S = \{\xi_k\}$, shown in

$$\theta_i^* = \arg\max_{\theta_i} P(S \mid \theta_i). \tag{1}$$

According to the principle of maximum entropy, as shown in (2), a trajectory with a low cost has a higher probability, which is proportional to the exponential of its cost

$$P(\xi \mid \theta_i) = \frac{e^{-C_i(\theta_i,\xi)}}{Z(\theta)} = \frac{e^{-\theta_i^T f_i(\xi)}}{\int e^{-\theta_i^T f_i(\xi)} d\xi} \tag{2}$$

where $Z(\theta) = \int e^{-\theta_i^T f_i(\xi)} d\xi$ is the partition function integrating all arbitrary trajectories ξ. To handle the computational complexity in solving the partition function, the continuous IRL approximates (1) and reformulates the problem as a minimization of $-\log P(S \mid \theta_i)$

$$\theta_i^* = \arg\min_{\theta_i} \sum_{k=1}^{K} \frac{1}{2} \mathbf{g}^T(\xi_k) \mathbf{H}^{-1}(\xi_k) \mathbf{g}(\xi_k) - \frac{1}{2} \log \mathbf{H}_{\theta_i}(\xi_k) \tag{3}$$

where $\mathbf{g}^T$ and $\mathbf{H}$ are the gradient and Hessian, respectively. This formula indicates that along the expert demonstration, the recovery cost function should have small gradients and large positive Hessians.

2) Cost Function Feature Selection: The selected features present the vehicle state in an interpretable way and can capture the preference of the driver. We select the following features to calculate the cost function, based on the available inertial measurement units (IMUs) and global navigation satellite systems (GNSSs) information.

1) *Lane-Change Risk f_{thw}:* Ego vehicle is projected to its adjacent lane and calculates the time headway to its potential leading vehicle and the time headway from its following vehicle.
2) *Lane-Change Urgency f_{urge}:* If the ego vehicle needs to perform a mandatory lane change, the remaining time distance should be considered.
3) *Mobility f_m:* Drivers have different preferences on mobility. The difference between current speed and the speed limit $(v_{\lim})$ is used to evaluate this preference.

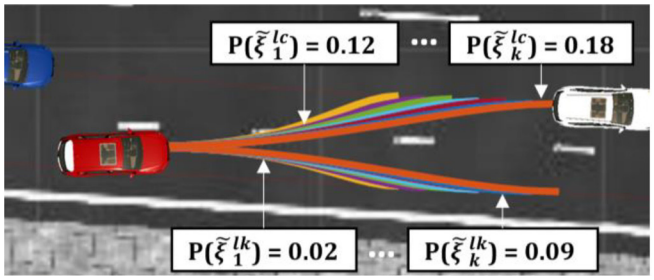


Fig. 2. Polynomial trajectory generator.

4) *Comfort f_a and f_ω:* The absolute value of the longitudinal acceleration a_{lon} and the yaw rate ω is used to gauge comfort preference.
5) *Lane Deviation f_{dev}:* We also include lateral distance into the cost function to evaluate the imperfection of driving along the centerline of the lane even in the lane-keeping stage.

3) Trajectory Evaluation: To execute the decision of lane change or lane keeping, planning the trajectory is essential. Considering the real-time performance, instead of exploring arbitrary trajectory, we adopt a polynomial trajectory generator [23] to plan the candidate trajectories ξ_k; as in Fig. 2, at each time step, this trajectory generator takes the vehicle's state x, y, v, a, yaw , as inputs and generates multiple trajectories within a prediction window. With integrated vehicle kinematics, the trajectory generation ensures that the prediction result is realistic and reachable. In this study, we set the planning time window as 4 s.

Based on (4), the cost function $C_i(\theta_i, \xi_k)$ is used to evaluate the probability of each possible trajectory ξ_k, and select the most probable trajectory. The probability of the lane-change maneuver prediction is evaluated by (5), i.e., the probability of lane change equals the sum of the probabilities of all sampled lane-change trajectories

$$P(\xi_k \mid \theta_i^*) = \frac{e^{-C_i(\theta_i^*, \xi_k)}}{\sum_{k=1}^{K} e^{-c(\theta_i^*, \xi_k)}} \tag{4}$$

$$P(a_i) = \sum_{k=1}^{K} P(\xi_k \mid \theta_i^*). \tag{5}$$

D. Hidden Markov Model Map Matching

Map matching is an important component in a field implementation to reduce the effect of noisy GPS measurements. HMM exploits the road connectivity information and time-sequence feasibility to solve the problem. As shown in Algorithm 1, we briefly introduce the HMM map-matching algorithm, which was proposed by Newson and Krumm [42]. For the HMM map-matching process, we first predefine a road network with a set of geographical coordinate pairs (i.e., latitude and longitude) of the road segments based on the real-world situation. The input is the GPS measurement points from the current time step and the last time step. The goal of map matching is to find the best-fitted road segment from all candidate road segments. The candidate road segments are selected by the great-circle distance between the measurement points and projection points on the road segments. Among candidates, we find the one which maximizes the product of the measurement probability and transition probability. Then, we output the projection point of the current GPS measurements on the best-fitted road segment as the matching point and the distance between them.

Algorithm 1 HMM Map-Matching Algorithm

Input: 1. Current GPS point (z_t). 2. Road segments on map (r_i)
Output: 1. Matching point (r_{best}). 2. Distance to road
1: if (t = 0) Initialize z_{t-1}
2: Find candidate road segments (r_i) based on z_t
3: Calculate the projection $x_{t,i}$ of z_t on each r_i
4: Calculate the *probability of the measurement* $(z_t|r_i)$ by

$$(z_t|r_i) = \frac{1}{\sqrt{2\pi\sigma_z}} \exp\left(-0.5 \left(\frac{\|z_t - x_{t,i}\|_{great\ circle}}{\sigma}\right)^2\right)$$

5: Calculate the *transition probability* $p(d_t)$ by

$$p(d_t) = \frac{1}{\beta} \exp\left(-\frac{\|z_t - z_{t-1}\|_{great\ circle} - \|x_{t,i} - x_{t-1,j}\|_{route}}{\beta}\right)$$

6: Find the best-fitted road segment by

$$r_{best} = \underset{i}{argmax}(p(z_t|r_i) * p(d_t))$$

7: Find the projection $x_{t,best}$ of z_t on r_{best}
8: Calculate the distance between z_t and $x_{t,best}$ by

$$\|z_t - x_{t,best}\|_{great\ circle}$$

9: Update state for next coming measurement $z_{t-1} = z_t$
Return r_{best} and $\|z_t - x_{t,best}\|_{great\ circle}$

E. Algorithm Validation in Simulation

Before modeling the personalized driving behavior in the real world, the prediction capability of the proposed algorithm is validated on a general driving behavior model in simulation [34]. In the human-in-the-loop co-simulation platform [43], a real-world test track is programmed in the Unity game engine as the digital version of the real-world testbed to be introduced in Section IV. In the simulation, a mixed traffic flow is generated by SUMO, and human input is consumed by Unity with Logitech driving set, as shown in Fig. 3(a), allowing various drivers to conduct human-in-the-loop simulations in an immersive traffic environment, where drivers can experience mixed traffic with different CAV penetration rate and congestion levels. To model the general lane-changing and lane-keeping behavior, 59 trips are collected from ramp drivers within the on-ramp/off-ramp area, under a volume-to-capacity (V/C) ratio of 0.6.

The real-time predicted lane-change probabilities and trajectories are visualized in Fig. 3(b), and the proposed algorithm recognizes the lane-change maneuver in 3 s before the vehicle crosses the lane-separation line. Moreover, the mean Euclidean distance (MED) [44] is used to quantify the accuracy of trajectory prediction. This general model achieves a mean MED of 0.39 m on average within a 4-s prediction window for ten test

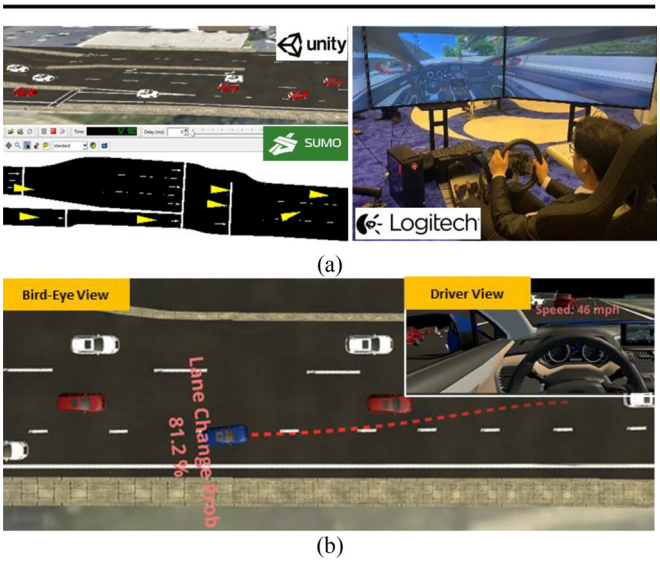


Fig. 3. Algorithm validation on UCR's human-in-the-loop co-simulation platform: (a) major elements in the platform and (b) predictions for lane change and most probable trajectory.

trips, outperforming the IRL-based prediction method in [41], which achieves a mean MED of 0.62 m in a 3-s prediction window. In the field implementation, we further study how the personalized model improves the prediction.

IV. Description of Field Operational System

A. System Setup

The vehicle-edge–cloud architecture is explained in Fig. 4, including key components, hardware for the real-world implementation, and the information flow between them. Under this architecture, vehicles are considered service consumers who store their personalized data set on the cloud server and share the real-time perception information with the edge server, from which vehicles receive driving assistance and support services to facilitate automation. The edge server creates the vehicle digital twin and serves as the bridge between the cloud server and vehicles. DDT is created on a cloud server by the offline-modeling algorithms described in Fig. 1.

To be specific, the first step of building the vehicle digital twin on the edge server is to know the position of each vehicle in the network. After building the connection between edge and vehicles, the perception information and GPS coordinates of each vehicle are uploaded to the edge server, where the map-matching algorithm updates the vehicle's positions in real time. Furthermore, algorithms for predictions, planning, and control can be provided upon request for different purposes. With the connection to the cloud server, the edge server receives service requests from vehicles and passes on those requests and the unique login information to the cloud. In response to the request, the cloud can feed back edge server with personalized driving behavior models of specific drivers. Due to the high communication latency between the cloud and vehicles, it is hard to implement real-time services provided directly from the cloud to customers. To address this problem, edge gateway is adopted to handle the data exchange and provide services to vehicles in real time.

The concern of privacy is also addressed, as only unidentifiable information is transmitted. Cloud server is responsible for driver model training, storage of personalized data and models, and microservices support (e.g., energy consumption analysis). Therefore, we take advantage of the strong computational power, high-speed data processing, and secure data storing features of AWS. Historical personal driving data are archived in the personal folder and can only be accessed by the personal login information. Service consumers can also request post-processed driving behavior reports, (e.g., energy consumption analysis, to better understand or improve their driving skills.

Hardware for the real-world implementation is shown in Fig. 4. All three vehicles are 2012 Corolla LE models, with 1.8-liter internal combustion engines. Each vehicle is equipped with a Wi-Fi hotspot (Netgear MiFi) to establish a wireless connection with edge server, a GPS unit to collect accurate position information of vehicles, and a portable HMI device using Galaxy A7 tablet. The U-blox C102-F9R is adopted as the GPS unit, which is a multiband GNSS with real-time kinematic positioning (RTK) function and sensor fusion technologies. It can achieve a 2-D (horizontal) accuracy of 46.7 cm and 4.0 cm with the "RTK Fixed" mode [6].

The GPS unit and the HMI device are connected (wired) with serial communications. The Wi-Fi hotspot handles the wireless communications between the tablet and other infrastructures, such as the base station and edge server. Depending on the request of the driver, the HMI device receives and displays the speed guidance or lane-change prediction information from the edge server. Moreover, to enable the RTK function in the U-blox and get high-accuracy GPS measurements, the HMI device shares the calibration correction message received from the base station to its paired GPS unit.

The edge gateway is running on a customized edge server installed at CE-CERT, University of California, Riverside, which is 1.2 km away from the testing site. A Dell R630 server, with two Xeon 2.4-GHz (6-core) CPUs, 64-GB RAM, 1-TB solid-state drive, and 14-TB hard disk drive, is adopted as the edge server in this implementation.

B. Experiment Plan

As shown in the vehicle level of Fig. 4, the real-world implementation is conducted by three passenger vehicles which are mainline vehicle 1 (MV1), mainline vehicle 2 (MV2), and ramp vehicle (RV), respectively. Specifically, RV is the target predicted RV, MV1 is the conflicting vehicle with RV. While MV2 drives behind MV1 and is potentially affected by the lane-change maneuver of RV, MV2 observes and predicts the whole lane-change process. To ensure MV1 and RV can encounter each other, speed guidance is provided to MV1 only before reaching an observable point, from which MV1 and RV can see each other for the rest of the trip. The speed guidance [6] is calculated by edge server, where RV has its virtual projection on the mainline, and MV1 is assigned to follow the virtual projection without an intervehicle gap. By

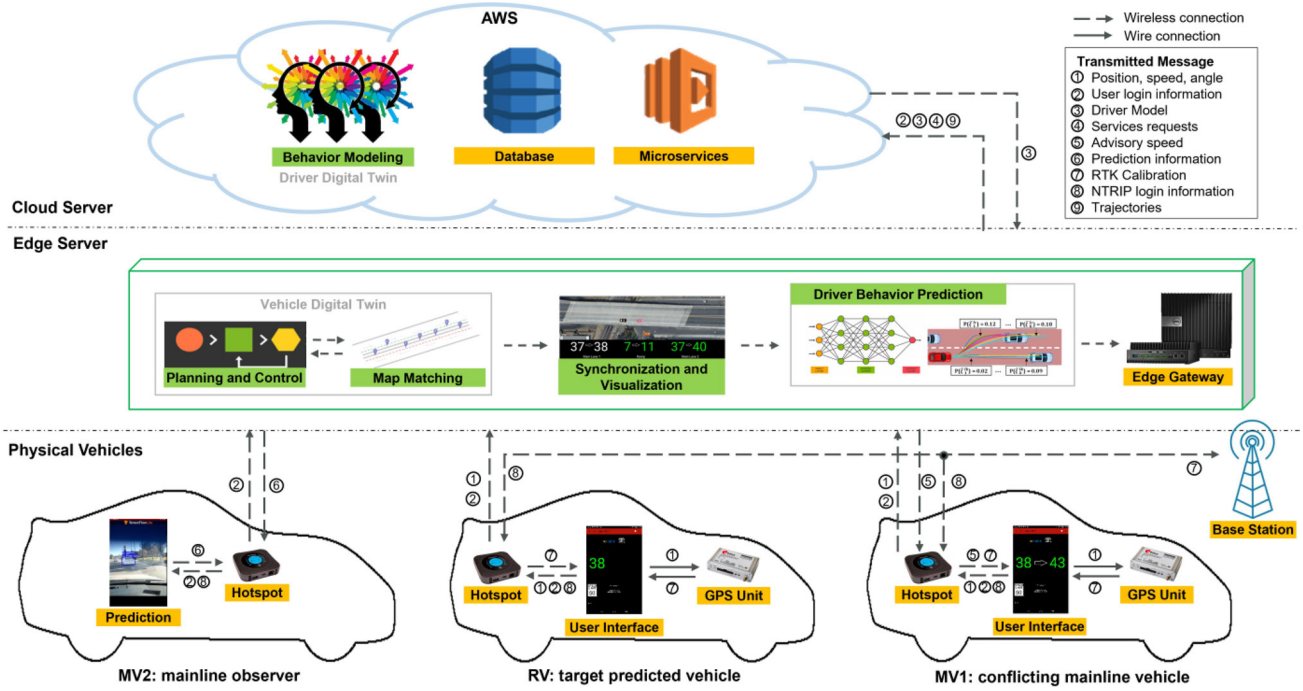


Fig. 4. Vehicle-edge–cloud architecture validated in the field experiment.

(a) (b)

Fig. 5. Vehicle HMI design. (a) Speed guidance and (b) detection for lane-changing prediction.

Fig. 6. Bird-eye's view application interface: digital twin of vehicles running on the edge server in real time.

following the speed guidance, MV1 can reach the observable point at the same time as RV, and for each experiment, nearly the same condition of merging conflict is guaranteed. Different from MV1, to collect naturalistic driving data of the RV driver, RV receives no interference from the system during the whole trip. During the conflict creation process, MV2 is not involved since it plays the role of an observer and has no impact on the lane change. If MV2 successfully detects and recognizes RV, MV2 sends a request and downloads the driver behavior model of RV from the cloud to facilitate the lane-change prediction.

To meet the above demands, the HMI guides MV1 with speed guidance which shows the current speed on the left and the target speed on the right as shown in Fig. 5(a). For MV2, we leverage the benefits from TensorFlow Lite Object Detection, an open-source Android application, to detect the RV and visualize the predicted lane-changing probability with a user ID on the top of a bounding box as indicated in Fig. 5(b).

Before starting the vehicle, each driver logs in with a unique ID and password to access his/her personalized digital twin. In addition, to obtain personalized services, drivers need to agree with sharing information with edge server.

A bird-eye view Android application is designed for visualizing the vehicle Digital Twins running on the edge server. MV1 and RV are the orange vehicles and the blue vehicle, respectively, in Fig. 6. Moreover, the lane-changing probability and the predicted trajectory of RV are provided for a better understanding of the entire field implementation process.

To build DDT on a cloud server, a personalized data set for each RV driver is collected. In total, 20 trips entering the mainline and 20 trips driving off-ramp are used for behavior modeling. The average duration of each trip is 35 s, with an average update rate of 5 Hz. In each time step, both the trajectories of MV1 and RV are recorded and synchronized at

TABLE I
STATISTIC RESULT OF COMMUNICATION DELAY AND COMPUTATION TIME

	Average	75th Percentile	99th Percentile	Maximum
Cloud-Edge	800 ms	915 ms	1221 ms	1261 ms
Edge-Vehicle	80 ms	88 ms	247 ms	3861 ms
Computation	28 ms	14 ms	36 ms	71 ms

TABLE II
WEIGHTS OF RECOVERED COST FUNCTIONS

Scene	Driver	f_a	f_{thw}	f_{dev}	f_{yaw}	f_m	f_{urge}
LC	#1	0.696	0.151	0	0	0.023	0.126
	#2	0.425	0.240	0	0	0.182	0.151
	General	0.529	0.284	0	0	0.179	0.008
LK	#1	0.388	0	0.246	0.348	0.017	0
	#2	0.323	0	0.222	0.043	0.412	0
	General	0.356	0	0.311	0.309	0.023	0

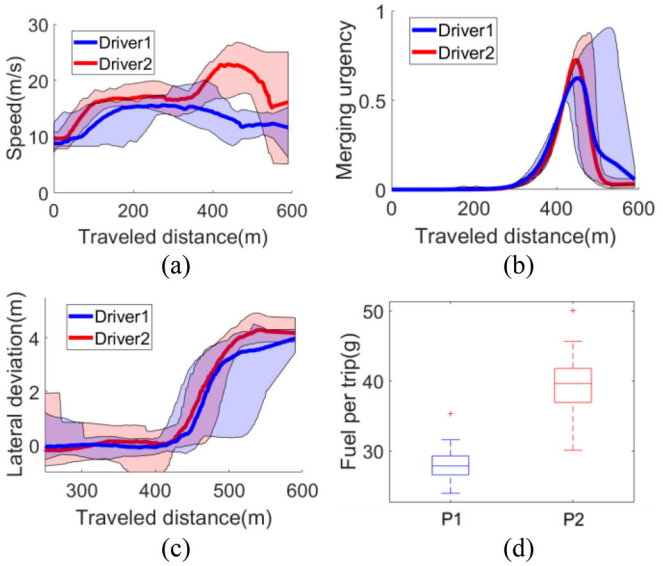


Fig. 7. Driving pattern comparison during a lane-change process: (a) longitudinal speed of RV, (b) mandatory lane-change urgency, (c) lateral behavior, and (d) fuel consumption.

edge server, and the data set on the cloud server is updated at the end of the trip.

Moreover, the communication delay and computation time are considered for real-world implementation. The back-and-forth communication delays of cloud–edge communication (model download), the edge-vehicle communication (speed guidance and online prediction), and the computation time are measured and shown, respectively, in Table I.

V. RESULT ANALYSIS

A. Cost Function and Driving Pattern Analysis

Personalized models are trained using drivers' own data sets (i.e., 19 trips collected from Driver 1 and 20 trips from Driver 2). For two drivers, a general model is trained using the aggregated data set from both drivers, standing for collective driving behavior. The weights in the cost functions can reflect drivers' preference when large weights penalize high values, as shown in Table II. For lane-change maneuvers, Driver 1 cares more about longitudinal comfort and penalizes large acceleration with 0.696 on f_a, compared with 0.425 for Driver 2. Driver 2 prefers a smaller time gap during lane change as he/she puts a larger weight on f_{thw} than Driver 1 to penalize the time headway. Driver 1 does not care about the mobility f_m and drives slower than the speed limit, while Driver 2 prefers to drive faster with a weight of 0.182. For lane-keeping maneuvers, two drivers show the same preference for longitudinal comfort f_a but significant differences in lateral comfort and mobility. Driver 1 tends to keep the vehicle stable (i.e., putting a large weight on f_{yaw} penalizing unstable yaw movements) and drives slowly. On the contrary, Driver 2 pays little attention to the stability of the vehicle pose and seeks high speed. According to the analysis of cost function weights, Driver 1 is more likely to be a cautious driver while Driver 2 tends to be more aggressive. For lane changing, the general model puts the least weight on urgency f_{urge} and penalizes large time headway. For lane keeping, the general model is sensitive to lane deviation f_{dev} the most.

Besides the cost function analysis, the driving pattern can also be recognized via the overall lane-change behavior, as shown in Fig. 7.

The interaction with the mainline vehicle can be reflected by the longitudinal speed of the lane-change process in Fig. 7(a), which displays the median speed at each location. The observable point (at 320 m) is 100 m before the lane-change point, where mainline and ramp drivers can see each other for the first time during the lane change. After observing the conflict with MV1, Driver 1 chooses to slow down and yield to MV1 for lane change behind, but Driver 2 accelerates to surpass in order to cut in front of MV1. The lane-change urgency in Fig. 7(b) is used to measure how the driver deals with a mandatory lane change. The urgency value grows when RV comes closer to the end of the lane-change area without changing the lane. Once the lane change is completed, the urgency will decrease shortly. Driver 1 has a smaller peak value than Driver 2 for lane-change urgency; however, the urgency pattern of Driver 2 is more consistent, as the urgency variation (red area) is smaller than the one of Driver 1 (blue area).

The lateral movement preference is captured by the lateral deviation shown in Fig. 7(c), where the lane change is completed once the lateral deviation reaches 4 m or above. Although two drivers have different preferences for lane-change sequences, their lane change starting points are close to each other. Moreover, two types of slopes (i.e., lane change speed) in the blue line are observed during the lane-change process of the driver. In the first segment, Driver 1 merely crosses the lane-separation line, and in the second segment, he/she approaches the center line slowly after confirming safety.

A similar conclusion to cost function analysis can be made that Driver 1 is more cautious than Driver 2. As a result, as shown in Fig. 7(d), the average fuel consumption of Driver 1

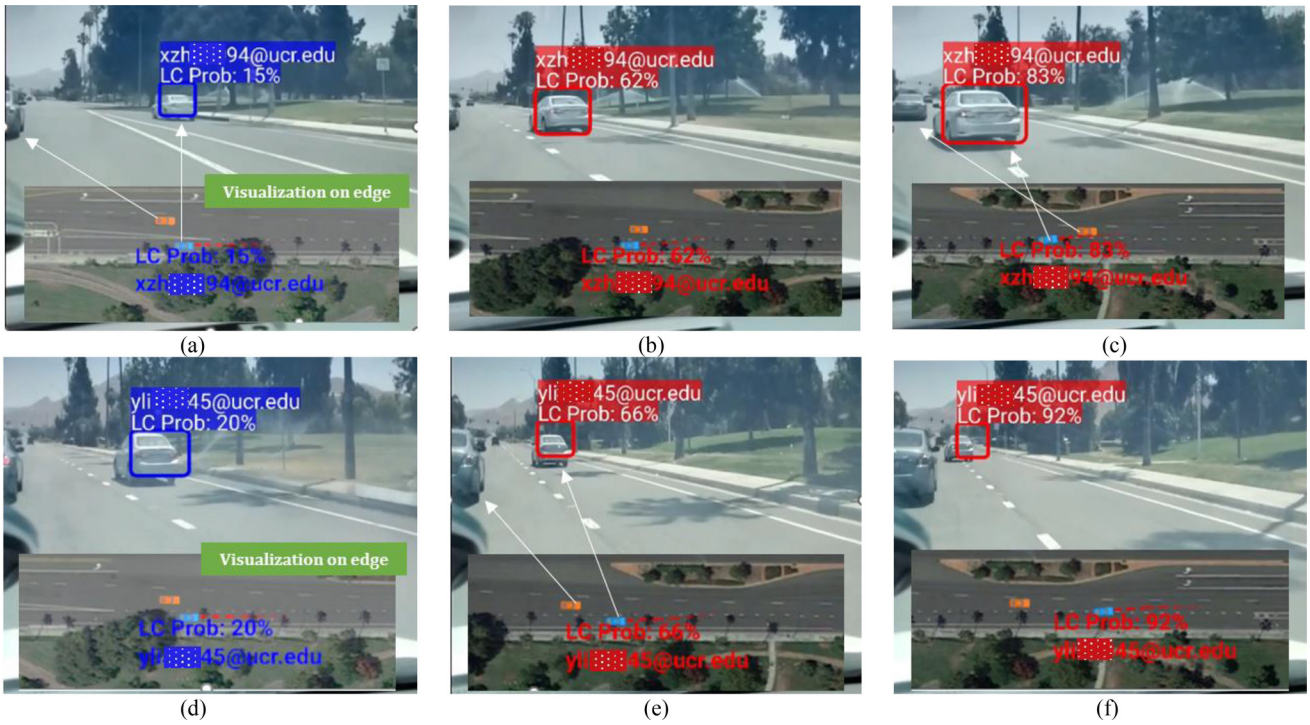


Fig. 8. Field implementation for online lane-change prediction with visualization on camera and edge server: (a)–(c) Prediction for Driver 1, changing the lane behind MV1. (d)–(f) Prediction for Driver 2, changing the lane in front of MV1.

(28.2 g) is 28.9% less than the one of Driver 2 (39.7 g), in each lane change.

B. Case Study

As presented in Fig. 8, the online prediction combines probability estimation and trajectory prediction. In stage 1 see Fig. 8(a), RV enters the interacting zone and is recognized by MV2 (observer). At the same time, MV2 sends a request to edge server, and then edge server receives the driver model from the cloud. Then, the driver ID and lane-change probability are visualized on both observer's view and edge. Meanwhile, the lane-change probability shown above the blue bounding box is low, and the predicted trajectory (visualized on edge in red dashed line) is straight, indicating the lane-keeping maneuver. In stage 2 see Fig. 8(b), RV slows down to yield MV1, which is captured by MV2. At the moment, the predicted trajectory points to the lane-separation line, and the lane-change probability increases with the color of the bounding box changing from blue to red for warning of a potential lane change. In stage 3 see Fig. 8(c), Driver 1 perceives that the gap is large enough, and the lane change begins. MV2 is more certain about the prediction, showing a high probability and a predicted trajectory pointing to the center of the mainline. Fig. 8(d)–(f) present a similar process for Driver 2's lane change. One noticeable difference from Driver 1's model is that Driver 2's personalized model pays attention to speeding up rather than slowing down.

Fig. 9 elaborates on the whole prediction process of the same trips in Fig. 8. Specifically, Fig. 9 (a) presents the prediction result of one lane change performed by Driver 1. The top subfigure depicts the prediction result and probability estimation of lane change during a trip, reflecting the intention of the driver. In the bottom subfigure, each time step of the ground-truth trajectory is labeled by the neural network in real time as lane change (red dots) or lane keeping (blue dots), when the predicted trajectory of each time step is shown in the green line.

The maneuver prediction can be corrected by probability estimation. In the top subfigure of Fig. 9(b), when the lane change has been completed, lane change is still predicted by LSTM (orange dashed line) for the time steps from 177 to 180, but the probability of lane change (blue solid line) is estimated only 20%. According to the bottom subfigure, Driver 2 has completed the lane change at the 177th step, and the cost function-based probability estimation corrects the neural network prediction.

Personalized models can only be used on a specific person. Fig. 9(c) illustrates how will the prediction result be when the model is mismatched (using Driver 1's model on Driver 2's trip), where the predictions are not accurate for lane-change probability, action, and trajectory.

The prediction needs to be evaluated for both maneuver and trajectory. For lane-change intention recognition, the predictive capability can be quantified as the time between the moment of recognizing the lane-change intention and the moment of the vehicle crossing the lane-separation line. The personalized model of Driver 1 recognizes the lane-change intention in 6.08 s on average, with a 1.96-s standard deviation (STD), while Driver 2's model achieves an average of 3.73 s with an STD of 1.29 s. In this study, output of intention recognition layer can be corrected and modified by the downstream probability estimation. Therefore, we propose a heuristic method to evaluate to accuracy of intention

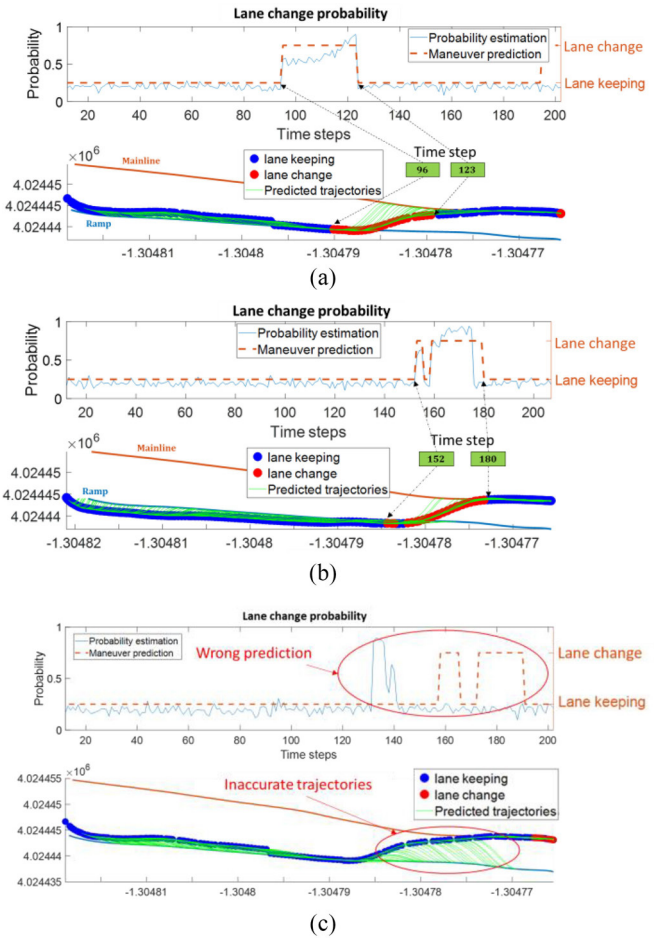


Fig. 9. Prediction result analysis of using personalized models. (a) Online prediction process for Driver 1. (b) Online prediction process for Driver 2. (c) Offline prediction using a mismatched driver model.

recognition based on the lane-change probability. It should be noted that the accuracy can be evaluated in a more sophisticated way. If the probability of the moment when the vehicle touches the lane-separation line (P_s) is significantly larger than the probability during lane keeping (P_k), i.e., $\sigma \cdot P_s > P_k$, the lane change is successfully predicted. The significant factor σ can be chosen as 0.5. The proposed general model predicts 90.4% lane change events, while the personalized model for Driver 1 exhibits an even higher level of performance with a prediction accuracy of 95.2%.

For trajectory prediction, the MED is measured to analyze the accuracy. Although we cannot decouple the GPS error and prediction error, the model achieves a good result. The quantitative evaluation is shown in Table III, which presents the MED of baseline, general model, and personalized model. Inspired by [16], we adopt Seq2seq LSTM as the baseline model for the entire pipeline, while the proposed method employs the Seq2seq LSTM solely for the intention recognition part in the overall pipeline. Within a 4-s prediction window, the general model of the proposed algorithm outperforms the baseline.

Using either a general or personalized model, the prediction of Driver 1 is better than Driver 2. The MEDs between the predicted trajectory and GPS points are 1.03 m (STD: 0.4 m) for Driver 1 and 1.48 m (STD: 1.05 m) for Driver 2 in a 4-s prediction window. Compared to results from the general model, the personalized model improves Driver 2's results the most, by 27.8% on average. Since Driver 1 is more predictable, the improvement of using a personalized model is limited (by 1.9%), but the prediction variation is reduced by 42%.

TABLE III
PREDICTION ACCURACY COMPARISON IN A 4-S PREDICTION WINDOW

MED (m)		Seq2seq LSTM	General	Personalized	Improvement in the general model
Driver 1	Mean	6.76	1.05	1.03	1.9%
	STD	1.16	0.69	0.4	42%
Driver 2	Mean	8.28	2.05	1.48	27.8%
	STD	1.99	1.17	1.05	10%

VI. CONCLUSION AND FUTURE WORK

This article has proposed an online lane-change prediction algorithm based on personalized driving behavior modeling, which is validated on a vehicle-edge–cloud testbed under the DDT concept. Specifically, a Seq2seq LSTM neural network has been used to predict the lane-change intention, and personalized driving behaviors have been modeled for different drivers, whose preferences are learned and analyzed by IRL based on the historical data stored on the cloud server. Supported by the personalized models, an online lane-change prediction system has been developed and validated with real-world field implementation. The system is able to recognize the target driver's lane-change intention at 6.08 s before the vehicle crosses the lane-separation line, and the MED between the predicted trajectory and ground truth (based on the measurements from an RTK-enabled GPS unit) is 1.03 m within a 4-s prediction window. Using a personalized model can improve the prediction accuracy by 27.8%.

As one of the first few research projects looking into personalized driving behavior, there are still some limitations of this implementation, and improvements can be made alongside the future development. The prediction algorithm is specifically designed for on/off-ramp mandatory lane changing. The mechanism behind discretionary lane-changing maneuvers can be different, which requires adjustment on feature selection of the cost function. Another major constraint on studying personalized behavior is data availability, which can be relieved by transferring the model learned from simulation. Besides the research on personalized behavior modeling, incorporating personalized prediction with planning is also an important future step, as it allows CAVs to drive like HDVs, and improve their user acceptance and trust.

ACKNOWLEDGMENT

The authors are grateful to Yejia Liao for his contributions to the field implementation. The contents of this article only reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Toyota Motor North America.

REFERENCES

[1] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 12, no. 1, pp. 4–24, 2020.

[2] Z. Wang et al., "Driver behavior modeling using game engine and real vehicle: A learning-based approach," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 738–749, Dec. 2020.

[3] X. Liao, Z. Wang, K. Han, X. Zhao, M. Barth, and G. Wu, *15 Driver Behavior-Aware Cooperative Ramp Merging for Intelligent Vehicles*, vol. 3. Berlin, Germany: Walter de Gruyter GmbH & Co KG, 2023.

[4] X. Liao, S. Mehrotra, S. Ho, Y. Gorospe, X. Wu, and T. Misu, "Driver profile modeling based on driving style, personality traits, and mood states," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2022, pp. 709–716.

[5] W. Xu, J. Wang, T. Fu, H. Gong, and A. Sobhani, "Aggressive driving behavior prediction considering driver's intention based on multivariate-temporal feature data," *Accident Anal. Prevent.*, vol. 164, Jan. 2022, Art. no. 106477.

[6] X. Liao et al., "Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud communication," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4490–4500, May 2022.

[7] Z. Wang et al., "Mobility digital twin: Concept, architecture, case study, and future challenges," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 17452–17467, Sep. 2022.

[8] Y. Xing et al., "Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4377–4390, May 2019.

[9] K. Schmidt, M. Beggiato, K. H. Hoffmann, and J. F. Krems, "A mathematical model for predicting lane changes using the steering wheel angle," *J. Safety Res.*, vol. 49, pp. 85–90, Jun. 2014.

[10] L. Y. Du, W. Chen, J. Ji, Z. H. Pei, B. M. Tong, and H. J. Zheng, "A novel intelligent approach to lane-change behavior prediction for intelligent and connected vehicles," *Comput. Intell. Neurosci.*, vol. 2022, Jan. 2022, Art. no. 9516218.

[11] F. Li, W. Wang, G. Feng, and W. Guo, "Driving intention inference based on dynamic Bayesian networks," in *Practical Applications of Intelligent Systems*, vol. 279, Z. Wen and T. Li, Eds. Berlin, Germany: Springer, 2014, pp. 1109–1119.

[12] Z. Shou, Z. Wang, K. Han, Y. Liu, P. Tiwari, and X. Di, "Long-term prediction of lane change maneuver through a multilayer perceptron," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2020, pp. 246–252.

[13] H. Woo et al., "Dynamic potential-model-based feature for lane change prediction," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, 2016, pp. 838–843.

[14] J. Ashesh, H. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture," 2016, *arXiv:1601.00740*.

[15] O. Scheel, N. S. Nagaraja, L. Schwarz, N. Navab, and F. Tombari, "Attention-based lane change prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 8655–8661.

[16] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder–decoder architecture," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Changshu, China, 2018, pp. 1672–1678.

[17] Q. Xue, Y. Xing, and J. Lu, "An integrated lane change prediction model incorporating traffic context based on trajectory data," *Transp. Res. C Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103738.

[18] M. Hasenjäger, M. Heckmann, and H. Wersing, "A survey of personalization for advanced driver assistance systems," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 335–344, Jun. 2020.

[19] S. Lefevre, A. Carvalho, Y. Gao, H. E. Tseng, and F. Borrelli, "Driver models for personalised driving assistance," *Veh. Syst. Dyn.*, vol. 53, no. 12, pp. 1705–1720, 2015.

[20] I. Bae, J. Y. Moon, and S. Kim, "Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles," 2020, *arXiv:2001.03908*.

[21] W. Wang, D. Zhao, W. Han, and J. Xi, "A learning-based approach for lane departure warning systems with a personalized driver model," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9145–9157, Oct. 2018.

[22] W. Wang, J. Xi, and J. K. Hedrick, "A learning-based personalized driver model using bounded generalized Gaussian mixture models," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11679–11690, Dec. 2019.

[23] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10239–10251, Aug. 2022.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[25] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, "Analyzing the suitability of cost functions for explaining and imitating human driving behavior based on inverse reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 5481–5487.

[26] Grand View Research. "Digital Twin Market Size, Share & Trends Analysis Report by End-Use (Automotive & Transport, Retail & Consumer Goods, Agriculture, Manufacturing, Energy & Utilities), by Region, and Segment Forecasts, 2021–2028." Accessed: Dec. 9, 2021. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/digital-twin-market

[27] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US air force vehicles," in *Proc. 14th AIAA 53rd AIAA/ASME/ASCE/AHS/ASC Structures Structural Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Structures Conf.*, 2012, p. 1818.

[28] X. Chen, E. Kang, S. Shiraishi, V. M. Preciado, and Z. Jiang, "Digital behavioral twins for safe connected cars," in *Proc. 21th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst.*, 2018, pp. 144–153.

[29] Z. Wang et al., "A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems," in *Proc. IEEE 91st Veh. Technol. Conf.*, May 2020, pp. 1–6.

[30] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. H. L. Hansen, "Vision-cloud data fusion for ADAS: A lane change prediction case study," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 210–220, Jun. 2022.

[31] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 198–209, Jun. 2022.

[32] Z. Wang, K. Han, and P. Tiwari, "Digital twin simulation of connected and automated vehicles with the unity game engine," in *Proc. IEEE 1st Int. Conf. Digit. Twins Parallel Intell. (DTPI)*, 2021, pp. 1–4.

[33] C. Schwarz and Z. Wang, "The role of digital twins in connected and automated vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 6, pp. 41–51, Nov./Dec. 2022.

[34] X. Liao et al., "Online prediction of lane change with a hierarchical learning-based approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 948–954.

[35] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017, pp. 353–359.

[38] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *J. Adv. Transp.*, vol. 2019, pp. 1–10, Feb. 2019.

[39] W. Kim, Y. Han, K. J. Kim, and K.-W. Song, "Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems," *Energy Rep.*, vol. 6, pp. 2604–2618, Nov. 2020.

[40] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," 2012, *arXiv:1206.4617*.

[41] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2111–2117.

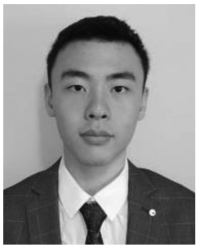
[42] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2009, pp. 336–343.

[43] X. Zhao et al., "Co-simulation platform for modeling and evaluating connected and automated vehicles and human behavior in mixed traffic," *SAE Int.* vol. 5, no. 4, pp. 313–326, 2022.

[44] J. Quehl, H. Hu, Ö. S. Tas, E. Rehder, and M. Lauer, "How good is my prediction? Finding a similarity measure for trajectory prediction evaluation," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017, pp. 1–6.

Xishun Liao (Graduate Student Member, IEEE) received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the M.Eng. degree from the University of Maryland, College Park, MD, USA, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Riverside, Riverside, CA, USA.

His research focuses on motion planning, driver behavior, and connected and automated vehicle technology.



Xuanpeng Zhao received the B.E. degree from Shanghai Maritime University, Shanghai, China, in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Riverside, Riverside, CA, USA.

His research focuses on embedded system, computer vision, and connected and automated vehicle technology.

Ziran Wang (Member, IEEE) received the Ph.D. degree from the University of California at Riverside, Riverside, CA, USA, in 2019.

He is currently an Assistant Professor with Purdue University, West Lafayette, IN, USA, where he leads the Purdue Digital Twin Lab. Prior to joining Purdue, he was a Principal Researcher with Toyota Motor North America, InfoTech Labs, Mountain View, CA, USA. His research focuses on intelligent vehicle technology, including autonomous driving, driver behavior modeling, and vehicular cyber–physical systems.

Dr. Wang serves as a Founding Chair for IEEE TECHNICAL COMMITTEE ON INTERNET OF THINGS IN INTELLIGENT TRANSPORTATION SYSTEMS, an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT VEHICLES and *SAE International Journal of Connected and Automated Vehicles*.

Zhouqiao Zhao (Graduate Student Member, IEEE) received the B.S degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2015, and the M.S. degree in electrical engineering from Ohio State University, Columbus, OH, USA, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with the University of California at Riverside, Riverside, CA, USA.

He works as a Graduate Student Researcher with the College of Engineering—Center for Environmental Research and Technology, University of California at Riverside. His research focuses on cooperative automation and connected and automated vehicle.

Kyungtae (KT) Han (Senior Member, IEEE) received the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA, in 2006.

He is currently a Senior Principal Scientist with Toyota Motor North America, InfoTech Labs, Mountain View, CA, USA. His research interests include cyber–physical systems, connected and automated vehicles, and intelligent transportation systems.

Rohit Gupta received the Ph.D. degree in mechanical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, where he did the research for "high speed" ship design and turbulent flow modeling of gas-liquid interfaces.

He is currently a Principal Cloud Architect with Toyota Motor North America, InfoTech Labs, Mountain View, CA, USA. Prior to joining Toyota, he worked in various industries: semiconductors, telecommunications, finance, and software. His research interests include cloud and edge computing architectures, connected and autonomous vehicles, AI/ML, "Digital Twin" of human–vehicle interactions, and high-speed mobility.

Matthew J. Barth (Fellow, IEEE) received the M.S. and Ph.D. degrees from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1985 and 1990, respectively.

He is currently the Yeager Families Professor and an Associate Dean with College of Engineering, University of California at Riverside, Riverside, CA, USA. His current research interests include ITS and the environment, transportation/emissions modeling, and advanced sensing and control.

Dr. Barth has been a Senior Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He served as the IEEE INTELLIGENT TRANSPORTATION SYSTEMS SOCIETY President for 2014 and 2015 and is currently the IEEE INTELLIGENT TRANSPORTATION SYSTEMS SOCIETY Vice President for Educational Activities.

Guoyuan Wu (Senior Member, IEEE) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2010.

He currently holds an Associate Researcher and an Associate Adjunct Professor position with Bourns College of Engineering—Center for Environmental Research and Technology and Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His research focuses on development and evaluation of sustainable and intelligent transportation system technologies.

Dr. Wu is an Associate Editor of *SAE International Journal of Connected and Automated Vehicles* and IEEE OPEN JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS.