

# Mobility Digital Twin: Concept, Architecture, Case Study, and Future Challenges

Ziran Wang<sup>ID</sup>, Member, IEEE, Rohit Gupta, Kyungtae Han<sup>ID</sup>, Senior Member, IEEE,  
Haoxin Wang, Member, IEEE, Akila Ganlath, Nejib Ammar, and Prashant Tiwari

**Abstract**—A Digital Twin is a digital replica of a living or nonliving physical entity, and this emerging technology attracted extensive attention from different industries during the past decade. Although a few Digital Twin studies have been conducted in the transportation domain very recently, there is no systematic research with a holistic framework connecting various mobility entities together. In this study, a mobility digital twin (MDT) framework is developed, which is defined as an artificial intelligence (AI)-based data-driven cloud–edge–device framework for mobility services. This MDT consists of three building blocks in the physical space (namely, *Human*, *Vehicle*, and *Traffic*), and their associated Digital Twins in the digital space. An example cloud–edge architecture is built with Amazon Web Services (AWS) to accommodate the proposed MDT framework and to fulfill its digital functionalities of storage, modeling, learning, simulation, and prediction. A case study of the personalized adaptive cruise control (P-ACC) system is conducted, which integrates the key microservices of all three digital building blocks of the MDT framework: 1) the *Human Digital Twin* with user management and driver type classification; 2) the *Vehicle Digital Twin* with cloud-based advanced driver-assistance systems (ADAS); and 3) the *Traffic Digital Twin* with traffic flow monitoring and variable speed limit. Future challenges of the proposed MDT framework are discussed toward the end of the article, including standardization, AI for computing, public or private cloud service, and network heterogeneity.

**Index Terms**—Amazon Web services (AWS), cloud computing, connected vehicles, digital twin, edge computing.

## I. INTRODUCTION

THE RECENT development of the Internet of Things (IoT) has been facilitating all kinds of cutting-edge technologies, where their application scenarios are rooted both in the user level (namely, individual consumer or private company), and the system level (namely, commercial or industrial sector). From the user's perspective, the introduction of the IoT will play a leading role in scenarios like assisted living, e-health, and enhanced learning. From the system's perspective, the most apparent scenarios will be industrial manufacturing, logistics, business/process management, and intelligent transportation of people and goods [1].

Manuscript received 17 October 2021; revised 30 January 2022; accepted 26 February 2022. Date of publication 2 March 2022; date of current version 7 September 2022. (Corresponding author: Ziran Wang.)

The authors are with InfoTech Labs, Toyota Motor North America Research and Development, Mountain View, CA 94043 USA (e-mail: ryanwang11@hotmail.com; rohit.gupta@toyota.com; kyungtae.han@toyota.com; haoxin.wang@toyota.com; akila.ganlath@toyota.com; nejib.ammarr@toyota.com; prashant.tiwari@toyota.com).

Digital Object Identifier 10.1109/JIOT.2022.3156028

The Digital Twin, as an emerging representation of the cyber–physical systems (CPS), attracted increasing attention over the past decade [2]. Based on a market research report, the global Digital Twin market size was valued at U.S. \$ 5 billion in 2020, and is projected to expand to 86 billion in 2028 at a compound annual growth rate of 42.7% during this period [3]. It was also pointed out in their report that, the automotive and transportation industry takes one of the largest shares in the global Digital Twin market in 2020, among other end-uses like manufacturing, energy, and healthcare.

Although the definitions of the Digital Twin vary in different versions [4], [5], the basic concepts are essentially the same: A Digital Twin is a digital replica of a living or nonliving physical entity. Digital Twin technology paves the way to real-time monitoring and synchronization of real-world activities with the virtual counterparts [6]. The Digital Twin concept was first born in the aerospace domain when the National Aeronautics and Space Administration (NASA) adopted that as a key element in its 2010 technology roadmap. Along with its rapid development in different domains during the past decade, including aeronautics and space [4], [7], robotics [8], [9], manufacturing [10], [11], and informatics [12], the Digital Twin also has a huge potential in the transportation domain.

The emergence of connected vehicle technology introduces another platform to implement the Digital Twin. Since the level of connectivity within our vehicles has greatly improved, these equipped vehicles are able to “talk” with other entities, such as with other connected vehicles through vehicle-to-vehicle (V2V) communications, with traffic infrastructures through vehicle-to-infrastructure (V2I) communications, and with cloud servers through vehicle-to-cloud (V2C) communications [13]–[16]. Specifically, V2C communications allow connected vehicles to: 1) upload their data to the cloud server, enabling Digital Twins to be built in the digital (cyber) world based on their counterparts in the physical world and 2) offload their onboard computations to the cloud server, enabling Digital Twins to build models and calculate guidance information through powerful cloud computing, which can then be fed back to connected vehicles.

Very recently, a few Digital Twin studies have been conducted in the transportation domain [17]–[19], but none of them has a holistic framework connecting various mobility entities (i.e., human, vehicle, and traffic) together. In this study, a mobility digital twin (MDT) framework is proposed, which is defined as an artificial intelligence (AI)-based data-driven cloud–edge–device framework for mobility services.

The MDT framework is built on top of three different planes: 1) the physical space that has human beings, vehicles, and traffic infrastructures; 2) the digital space that has the digital replicas of aforementioned physical entities; and 3) the communication plane between these two spaces. Given the connectivity nature of this framework, it transforms connected vehicles into Internet of Vehicles (IoV) by leveraging IoT technologies. The cloud–edge architecture is built with Amazon Web Services (AWS) to accommodate the proposed MDT framework and implement its digital functionalities of storage, modeling, learning, simulation, and prediction. The effectiveness of the MDT framework is shown through the case study of three digital building blocks with their key microservices: the *Human Digital Twin* with user management and driver type classification, the *Vehicle Digital Twin* with cloud-based advanced driver-assistance systems (ADAS), and the *Traffic Digital Twin* with traffic flow monitoring and variable speed limit.

Since traditional mobility system frameworks heavily rely on onboard storage and computing, their functionalities are limited by multiple constraints, such as computing power, accessibility to big data, and easiness of deployments and modifications. On the contrary, the proposed MDT framework in this study addresses these constraints by making the following contributions.

- 1) *Powerful*: The MDT framework allows users to rapidly adjust cloud resources to meet fluctuating/unpredictable demands, providing high computing power at certain periods of peak demand.
- 2) *Shareable*: Bulk data generated by an end user is offloaded and stored on the cloud (and/or edge), which can be retrieved and utilized by the same user at a later time frame, or shared with other end users for microservices on demand.
- 3) *Manageable*: The MDT framework allows users to get their microservices up and running faster on the cloud platform, with more manageability and less maintenance. Over-the-air (OTA) updates are also available to the MDT framework.
- 4) *Extendable*: Arbitrary mobility microservices can be easily implemented to the MDT framework with minimal changes on the cloud–edge architecture and data structure.

The remainder of this study is organized as follows. Section II conducts a literature review regarding cloud computing and Digital Twins in the context of connected vehicles. Section III introduces the concept of this proposed MDT framework with a detailed explanation of the communication plane and data workflow, the physical space, and the digital space. Then, the cloud–edge architecture based on AWS is developed in Section IV, which accommodates the proposed MDT framework. A case study is conducted in Section V, where a personalized adaptive cruise control (P-ACC) system integrates multiple microservices of the MDT framework and outperforms traditional ACC systems. Finally, this study is finished with a discussion about future challenges in Section VI and a brief conclusion in Section VII.

## II. LITERATURE REVIEW

### A. Transportation Applications With Cloud Computing

The emergence of commercial cloud computing services, such as AWS [20], Microsoft Azure [21], Google Cloud Platform (GCP) [22], and Alibaba Cloud [23], has facilitated many applications in the domain of vehicular/transportation CPS. Such services always provide a variety of basic abstract technical infrastructure and building blocks for distributed computing. Taking AWS as an example, which has the largest market share among all competitors in 2020, it comprises over 200 products and services for computing, storage, networking, database, analytics, IoT, and so on [20]. All these features of cloud computing services, together with their advantage of scalability, enable connected vehicles to offload their data and onboard computing demand to the cloud.

Guerrero-Ibanez *et al.* [24] demonstrated that cloud computing can be integrated with intelligent transportation systems to address issues faced by the transportation sector, such as traffic congestion, roadway safety, and pollutant emissions. Specifically, the concept of vehicular cloud can enhance transportation systems by storing and processing the collected data (including traffic lights, parking meters, camera images, etc.), and creating a historical registry of various data sources [25]. Therefore, the transportation authorities who own these entities can make informed decisions on when to change traffic directions, install new traffic lights, and remodel/repair road segments. However, the detailed cloud architecture design is not covered in these studies, and the vehicular cloud applications are introduced only on the conceptual level without conducting case studies.

During the past decade, various transportation applications were proposed by leveraging the capability of cloud computing [26], [27]. A navigation-assisted route optimizer was developed by Gerla, where the navigator server collects information from connected vehicles, and then computes the optimal routes by constructing a traffic load map and traffic pattern matrix, estimating road segment loads and delays [28]. A bus smart sensor prototype was designed and implemented by Herrera-Quintero *et al.* [29] using the serverless and microservice cloud architecture, where GCP Firebase was used for storage and AWS Lambda was used for computation. A vehicular pollutant emission detection system was developed by Bhatnagar *et al.* [30], where AWS IoT and Amazon DynamoDB were integrated to send notifications to the vehicle driver if the emission sensor detects a gas leakage. A vehicle-based traffic surveillance application was developed by Deng *et al.* [31], where the AWS-based serverless cloud architecture was proved to be feasible for real-time transportation applications through a field implementation. However, aforementioned studies focus more on individual transportation application that provides solutions within a very limited domain, while none of them designs a holistic framework that connects various mobility entities, benefiting human, vehicle, and traffic at the same time.

### B. Digital Twins for Connected Vehicles

The Digital Twin concept has been loosely defined and adopted in the transportation domain since its emergence, partly due to its similarity and connection with other technologies. In particular, the confusion among the roles of iteration (i.e., switches back and forth between the physical and digital spaces), model-based design (i.e., starts with digital components and incrementally swaps in physical components), and Digital Twins (i.e., maintain synchronized versions of a physical system and its digital counterpart) in connected vehicles are well discussed in the survey paper by Schwarz and Wang [32]. Nonetheless, many previous efforts related to the IoT and CPS in the automotive industry envision the development of the Digital Twin, since the majority of those proposed methodologies and/or algorithms were developed on multilayer system frameworks with physical entities (i.e., vehicles) and their replicas (simulation models/environments).

Alam and Saddik developed a Digital Twin framework reference model for the cloud-based CPS, where a telematics-based driving assistance application was proposed for the vehicular CPS with three parts: 1) computation, 2) control, and 3) sensors and services fusion [17]. Kumar *et al.* [18] proposed a Digital Twin-centric approach with machine learning, edge computing, 5G communication, and data lake, aiming for driver intention prediction and traffic congestion avoidance. Chen *et al.* [19] proposed a “Digital Behavior Twin” framework in which behavioral models of drivers are shared among connected vehicles to predict future actions of neighboring vehicles and hence improve driving safety. This idea was extended to two subsequent patent applications by the same authors [33], [34]. However, the term of “Digital Twin” is used more like a simple concept in these studies, where their presented technologies are agnostic to any IoT technologies besides the Digital Twin. They do not systematically integrate the Digital Twin in any cloud architecture design, nor do they have clear data structures or workflows.

Related literature in the fields of “parallel driving” or “parallel transportation” has direct implications to the Digital Twin framework of our study. In 2010, Wang brought up the parallel transportation concept for the first time, where he defined parallel control and management of transportation as “a data-driven approach for modeling, analysis, and decision making that considers both the engineering and social complexity in its processes” [35]. Many subsequent works were conducted in this research domain afterward, including the parallel driving framework proposed by Wang *et al.* [36]. In this cloud-based cyber-physical-social system framework, the physical world, mental world, and artificial world are modeled as three parallel levels, considering interactions among connected vehicles, human drivers, and information. However, many applications shown in these studies do not come up with a cloud-based framework, and in such cases their capabilities of storage, learning, and prediction are well limited by vehicle onboard resources.

More relevant studies have been conducted very recently by the authors of this study in the context of the Digital Twin for connected vehicles. Wang *et al.* [37] proposed a

Digital Twin paradigm for an ADAS of connected vehicles. In this paradigm, onboard devices on connected vehicles collect and upload data to the cloud server through cellular-based V2C communication, where the cloud server can create digital replicas of entities in the real world (i.e., roads, vehicles, and drivers) based on the received data. All proposed models and algorithms are applied to these digital copies with cloud computing, where their results are propagated back to the real connected vehicles through V2C communication for ADAS, assisting the decision making of drivers in real time. A subsequent field implementation using this Digital Twin paradigm was conducted by Liao *et al.* [38], where three human-driven passenger vehicles performed ramp merging cooperatively, showing the benefits of safety and environmental sustainability compared to the traditional ramp merging scenario. A Digital Twin simulation architecture was also proposed later to allow researchers to implement this paradigm in a fully virtual environment of connected and automated vehicles with the Unity game engine [39].

Visualization of the Digital Twin information from the cloud remains a challenging issue, where Liu *et al.* [40] developed a data-fusion methodology to overlay the Digital Twin information for the driver’s field of view with the help of camera (RGB and depth) images, assisting the driver to make lane-change prediction of neighboring vehicles. On top of this study, Wang *et al.* [41] designed a cooperative driving system for connected vehicles, where nonline-of-sight vehicles are visualized as “Digital Twin slots” on the augmented reality (AR)-based head-up display (HUD) of the ego vehicle, guiding it to cross nonsignalized intersections without any collision or unnecessary full stop. However, none of these recent studies, which are from the authors of this study as well, designs a holistic system framework that connects mobility entities (i.e., human, vehicle, and traffic) together, and neither do they develop any cloud architecture with detailed data structures or workflows.

It needs to be noted that, many studies consider the Digital Twin simply as a high-fidelity modeling and simulation environment of real-world entities. Although this statement is partially correct, our understanding of the Digital Twin covers wider than merely modeling and simulation, namely, sampling and actuation in the physical space, and storage, modeling, learning, simulation, and prediction in the digital space. Related literature with the limited definition of the Digital Twin is not reviewed in this study.

### III. MOBILITY DIGITAL TWIN CONCEPT

The MDT framework proposed in this study, as shown in Fig. 1, consists of three planes: 1) the lower plane, highlighted in yellow, stands for the physical space where human beings, vehicles, and traffic infrastructures reside; 2) the upper plane, highlighted in blue, represents the digital space where the digital replicas of those physical entities are located at; and 3) between these two planes, the communication plane (in gray) plays a crucial role in this framework to allow real-time and nonreal-time data streaming for both upstream and downstream.

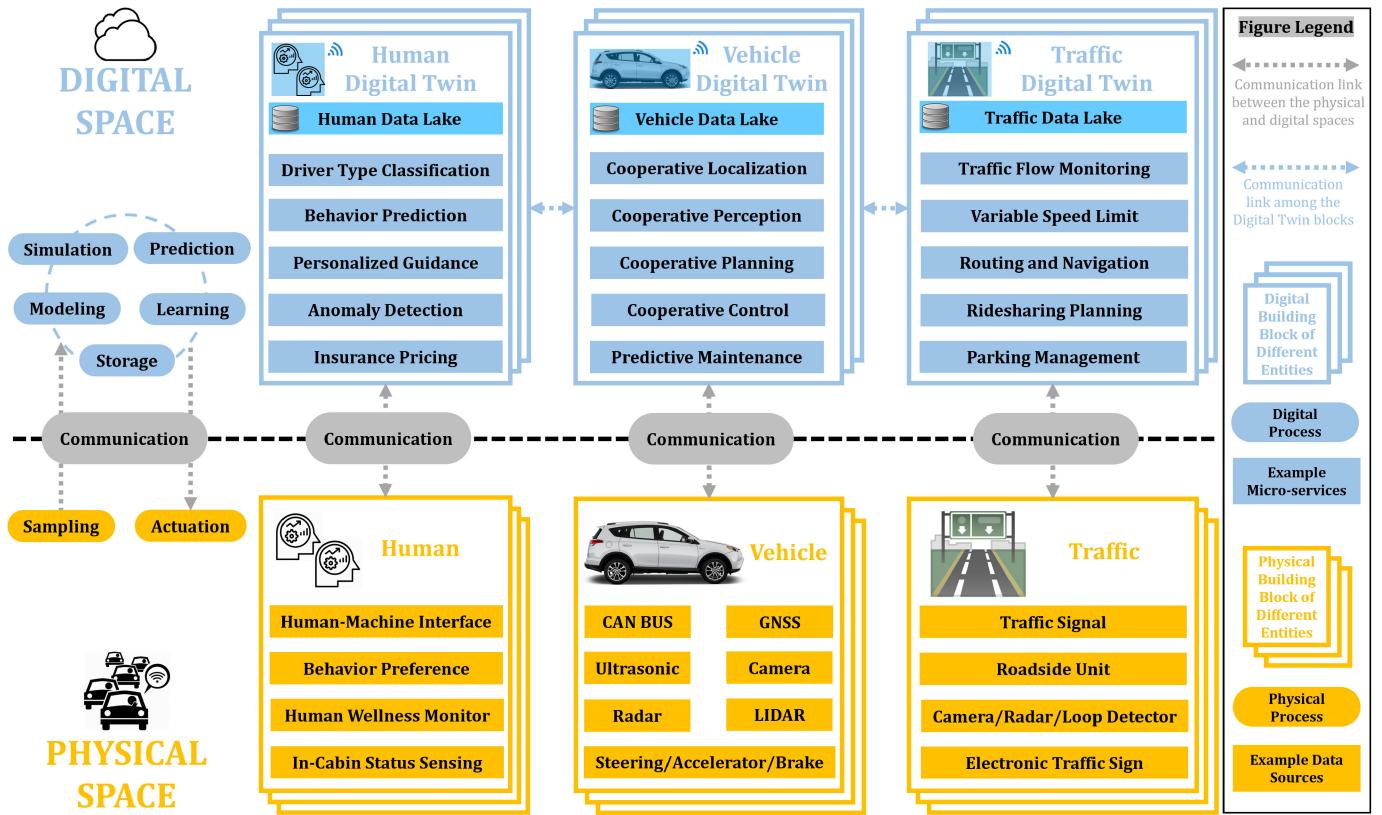


Fig. 1. Illustration of the MDT framework for connected vehicles, which consists of a physical space, a digital space, and a communication plane between two spaces.

Three entities are considered in this MDT framework: *Human*, *Vehicle*, and *Traffic*. Given the existence of the communication plane, each of the entity can be connected to the digital space (e.g., Internet) and exchanges data with each other. Therefore, this MDT framework is a good representation of the IoT, and it allows connected vehicles to act as IoVs with IoT technologies. In this section, we provide a deep dive into this MDT framework regarding all three aforementioned planes, introducing their building blocks with respect to their concepts and functionalities.

#### A. Communication Plane and Data Workflow

The communication plane of this MDT framework sits between the physical space and digital space, and it provides seamless connections between these two spaces. This MDT framework's end-to-end process starts from sampling data in the physical space, where all or part of the data is transmitted upstream to the digital space via the communication plane. Those data will go through one or multiple processes in the digital space internally, including storage, modeling, learning, simulation, and prediction, and the resulting data are transmitted downstream to the physical space via the communication plane. Those data, upon receiving, is applied by the actuators of the physical space to fulfill the end-to-end process.

The major difference between a Digital Twin framework with an iteration framework, or a model-based design framework is that, a Digital Twin framework always maintains synchronized versions of the physical system and its digital

counterpart [32]. In the proposed MDT framework of our study, this is guaranteed by the communication plane between the physical and digital spaces. Without the communication plane, data cannot be transmitted between these two spaces to enable their interactions and synchronizations, hence Digital Twins cannot be formed.

Since cloud computing is leveraged in this MDT framework, the digital space of the framework is deployed fully or partially on the commercial and/or private cloud. Therefore, the communication module needs to provide access to the cloud for the physical space, which is either direct access or indirect access (via edges). The MDT framework does not necessarily require any specific wireless communication technology (dedicated short-range communication (DSRC) [42], C-V2X [43], or something else in the future) to be served as the communication plane, as long as it can be applied to transmit data between the physical space and the digital space.

#### B. Physical Space

If we consider this MDT framework as an end-to-end framework, then the physical space is in charge of both ends of this framework, namely, sampling and actuation. We assume no (or only minimal) computing work needs to be conducted in the physical space, since all (or majority) of that is offloaded to the digital space through the communication plane.

For sampling, sensors in the physical space detect the dynamic status, operating process, or event occurrences, and then aggregate these measurements under various resolutions

for their transmission to the digital space. On the other hand, once the processed results are received from the digital space, actuation can be made by physical entities to fulfill this end-to-end framework. Generally, the physical space is defined on a world coordinate, which may contain all the transportation-related physical entities, and can be classified into three building blocks: *Human*, *Vehicle*, and *Traffic*.

1) *Human*: In this framework, all human beings involved in the transportation system are considered, which include not only drivers but also passengers, pedestrians, cyclists, etc. The sampling process can be accomplished by the human-machine interface in an active manner, or by the in-cabin status sensing (e.g., camera, seat sensor, etc.), human wellness monitor (e.g., smartwatch, electrocardiogram, etc.), and other perception sensors in a passive manner. The preferences of a human's behavior can also be set actively (e.g., a driver manually sets the preferred cruise control speed), or be measured passively (e.g., a pedestrian's preferred trajectory of crossing a crosswalk is recorded by the vehicle/intersection camera), where both of them are considered as the sampling process.

The actuation process of the *Human* block in this MDT framework is mainly conducted by drivers. In the foreseeable future, our transportation system will remain in a mixed autonomy traffic environment, where only part of all vehicles will be fully autonomous vehicles (with SAE level-5 automation), but the majority are still driven by human drivers (with no degree or a certain degree of automation). Therefore, if drivers can be provided with additional information from the digital space of this MDT framework, such as adjacent vehicle's lane-change possibility [44] or upcoming signal phase and timing [45], their actuation will be more accurate and in turn benefit other entities in the transportation system.

2) *Vehicle*: *Vehicle* is the core of this MDT framework, as it is the host of drivers and passengers, and also the fundamental component of traffic. As can be seen from Fig. 1, all modules in the physical space, not only the ones in the *Vehicle* block itself but also those in the *Human* block and the *Traffic* block, are serving for vehicle-related activities.

Specifically for the *Vehicle* block, the localization module (i.e., GNSS), the perception sensors (i.e., ultrasonic, camera, radar, and/or LiDAR), together with vehicle CAN BUS are in charge of the sampling process. Related data, such as positions, speeds, and accelerations of the ego vehicle and its surrounding vehicles can be sampled from these physical components, and then be propagated to the digital space through communication.

The actuation process of the *Vehicle* block in this MDT framework is conducted by the vehicle steering system, accelerator, and brake. These physical components are able to actuate any lateral or longitudinal control command received from the digital space, and therefore allow the vehicle to achieve its desired motion.

3) *Traffic*: Many existing intelligent vehicle platforms and applications, such as ADAS or autonomous driving systems (ADSs), only focus on their performances on the ego vehicle without considering their interactions with the large-scale traffic network. However, as can be seen from Fig. 1, *Traffic* is indeed a crucial building block of our MDT framework for

connected vehicles. The beneficiaries of this MDT framework include not only connected vehicles and their occupants but also the whole traffic network on a wider scope.

Particularly, the *Traffic* block in the physical space includes various traffic infrastructures, such as traffic signals, roadside units, camera/radar/loop detectors, and electronic traffic signs. These physical components are able to either generate data (e.g., signal phase and timing) by themselves, or measure data (e.g., traffic count and traffic flow) generated by other traffic entities. Such data are sampled and sent to the digital space through the communication plane, benefiting other building blocks of this MDT framework.

On the other hand, guidance or adjustment received from the digital space can also be actuated by the *Traffic* block to improve the safety and efficiency of the large-scale traffic network. For example, the signal phase and timing of traffic lights can be adjusted to better serve different traffic flows under different situations [46]. Guidance or warning information can be broadcast to connected vehicles via roadside units, and to all traffic entities via electronic traffic signs.

### C. Digital Space

The aforementioned physical space of this MDT framework handles both ends of this end-to-end framework (i.e., sampling and actuation). On the other hand, the digital space is in charge of the processes between both ends: storage, modeling, learning, simulation, and prediction.

One of the biggest strengths of this MDT framework over traditional mobility system frameworks is the data lake, which is a centralized repository that allows structured or unstructured data at any scale to be stored. Traditionally, mobility data measured by a physical entity is only saved in its onboard data storage due to the lack of communication capability. Such data are only used for the physical entity itself without being shared with other entities, and will be wiped out once the maximum size limit of the onboard data storage is met. However, with the proposed MDT framework, mobility data measured by *Human*, *Vehicle*, and *Traffic* blocks in the physical space can be transmitted to the digital space through the communication plane, and stored in the data lakes of associated Digital Twins for future use. Such data can be used for the microservices not only in the original mobility block but also in other blocks (e.g., traffic signal data measured by the *Traffic* block can be used for both the "real-time monitoring" microservice in the *Traffic Digital Twin* and the "cooperative control" microservice in the *Vehicle Digital Twin*).

Note that there exists a misunderstanding about the Digital Twin in the research community, where some simply consider Digital Twin technology as a modeling and simulation technology. In our MDT framework, modeling and simulation are part of the digital processes that are enhanced by the data lake and data sharing in the digital space, where co-simulation platforms can be built to synchronize data from multiple simulators (such as the Unity-SUMO-AWS integrated platform [47]). However, as shown in Fig. 1, our MDT framework is more than just modeling and simulation, where other

digital processes (i.e., storage, learning, and prediction) play equivalently crucial roles in the digital space. All of the aforementioned digital processes can be applied to mobility microservices, and they are realized in a more powerful, shareable, manageable, and extendable manner by leveraging cloud computing and edge computing.

1) *Human Digital Twin*: *Human Digital Twins* are digital replicas of real humans in the physical space. This building block in the digital space has a human data lake that stores all data sampled from the *Human* block in the physical space, where different humans have their personal databases to be differentiated from others. With real-time data sampling and historical data storage, the *Human Digital Twin* is able to classify drivers into specific driver types by machine learning algorithms like k-nearest neighbors (KNNs), and to provide guidance in a customized or personalized manner [48]. Taking advantage of the data coming from the *Vehicle* block, the *Human Digital Twin* can also predict future behaviors of drivers (e.g., lane-change intention [49]) and detect their anomalies [50]. The results of the aforementioned microservices can be applied to third parties such as insurance companies, where they can further build a microservice to set the insurance pricing for different drivers based on their driving behaviors [51].

2) *Vehicle Digital Twin*: *Vehicle Digital Twins* are the digital replicas of real vehicles in the physical space. Once the sampled data are received from a connected vehicle in the physical space, it can be saved in this particular vehicle's data lake with a unique identification number. Those data in the *Vehicle Digital Twin* about the ego vehicle (e.g., position, speed, and acceleration) and its surrounding environment (perceived by perception sensors) can also be shared with the *Human Digital Twin*, the *Traffic Digital Twin*, or other connected vehicles' *Vehicle Digital Twins* for various microservices.

With massive data storage and data sharing in the digital space, multiple vehicle-related microservices can be enabled, such as the ones requiring cooperation among multiple connected vehicles: cooperative localization, cooperative perception, cooperative planning, and cooperative control. Additionally, microservices that need time-series data can also be benefited from this MDT framework, where one typical example is predictive maintenance: Based on modeling and simulation of the time-series vehicle data that is sampled from the *Vehicle* block in the physical space and stored in the *Vehicle Digital Twin*, the learning process can be conducted in the digital space and predictions can be made regarding potential failures of vehicle components at a future time [52]. Such prediction results can be used by the vehicle owner or manufacturer to schedule onsite maintenance before the components break down.

3) *Traffic Digital Twin*: *Traffic Digital Twins* are the digital replicas of traffic infrastructures, which receive data from the *Traffic* block in the physical space. Such sampled data, such as signal phase and timing, traffic count, and traffic flow, can be stored in the traffic data lake for future reference. It can also be used for multiple traffic microservices in real time, such as monitoring the traffic condition [18], variable speed limit [53],

routing and navigation [54], ridesharing planning [55], and parking management [56].

Similar to the *Human Digital Twin* and *Vehicle Digital Twin*, the *Traffic Digital Twin* can be enhanced by the communication among these Digital Twin blocks. For example, the microservice of routing and navigation can be carried out solely by the real-time traffic flow data sampled from camera/radar/loop detectors in the real world. However, they can be further enhanced if behavior preferences are set by the *Human* block and predictions are made by the *Human Digital Twin* (e.g., a driver/passenger always goes to grocery stores when his/her commute route is highly congested). Additionally, if the *Vehicle* block detects the fuel/battery level is low and sends that to the *Vehicle Digital Twin*, it can also assist the routing and navigation microservice to find a gas/charging station near a user-preferred grocery store along the original route.

#### IV. EXAMPLE ARCHITECTURE WITH CLOUD-EDGE COMPUTING

In this section, we build an example architecture for the proposed MDT concept by leveraging cloud computing and edge computing, enabling both real-time and bulk-batch ingestion, processing, and analytics of mobility data. As shown in Fig. 2, the architecture can be divided into four layers: 1) the cloud layer, which is built on AWS and its Virtual Private Cloud (VPC) [57]; 2) the edge layer, which has a computing component, a communication component, and a storage component; 3) the device layer, which generates data and consumes guidance; 4) the API (Application Programming Interface) layer, which hooks up the cloud layer with external APIs.

The major purpose of designing this architecture is to accommodate the proposed MDT framework shown in Fig. 1, so the Digital Twin does not just remain in the conceptual phase but can also be deployed in the real world with the help of cloud computing and edge computing. Particularly, the physical space of the MDT framework shown in Fig. 1 is represented by the device layer of this example architecture, where mobile apps, simulators, real vehicles and radio control (RC) vehicles sample data from *Human*, *Vehicle* and *Traffic*, and also actuate commands received from the edge and cloud layers. The communication plane of the MDT framework is positioned within the edge layer's communication component of this example architecture. The digital space of the MDT framework, along with its data lakes and microservices, spans the whole part of the cloud and API layers, as well as part of the edge layer (except for its communication component).

As mentioned in Section II, the Digital Twin concept is sometimes inadequately considered as a high-fidelity modeling and simulation environment by many studies, since none of these studies fulfilled all functionalities of the Digital Twin concept (storage, modeling, learning, simulation, and prediction) by building a cloud-edge architecture. Although the example architecture built in this study is not the sole solution, it does provide an idea of deploying the Digital

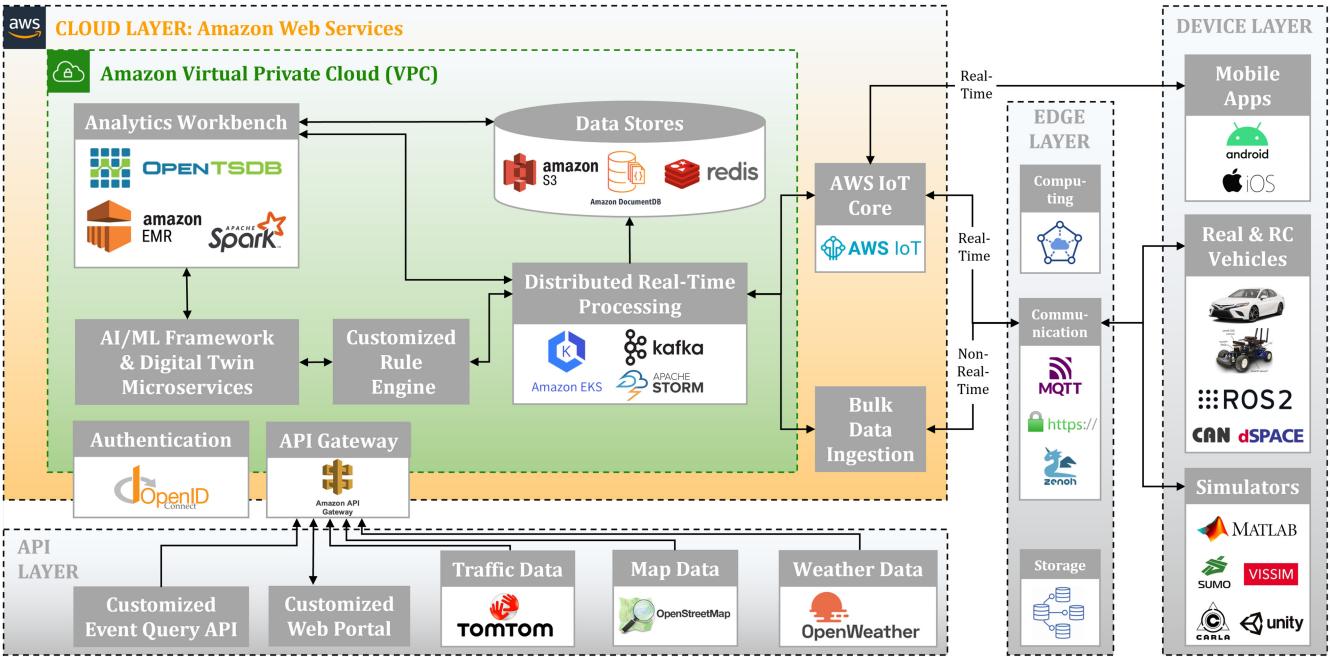


Fig. 2. Example architecture of the proposed MDT with the cloud layer, edge layer, device layer, and API layer.

Twin concept in the real world. Throughout the rest of this section, various modules in this architecture are explained in corresponding sections sorted by the layers they are in.

#### A. Cloud Layer

In Amazon VPC of the cloud layer, there are five different modules: 1) distributed real-time processing; 2) data stores; 3) analytics workbench; 4) rule engine; and 5) AI/ML framework & Digital Twin microservices. Specifically, in Distributed Real-time Processing, we leverage existing tools like Amazon EKS [58], Apache Kafka [59], and Apache Storm [60] to provide real-time processing and analytics. The Analytics Workbench is the workhorse for big data analytics. It consists of OpenTSDB, which is a distributed, scalable, time-series database built on top of Hadoop and HBase [61]. This supports a writing rate of up to millions of entries per second, supports data storage with millisecond-level precision, and preserves data permanently without sacrificing precision. In addition, Apache Spark [62], a distributed processing system, is used to conduct predictive analytics using Amazon EMR clusters [63].

Rule Engine service evaluates the rules configured for entities (e.g., humans and vehicles) on the data received from the Kafka queue, and redirects it to AI/ML Framework and Digital Twin Microservices based on the rule validation result. AI/ML Framework and Digital Twin Microservices are the core of this cloud–edge architecture, where end users are able to implement customized algorithms and applications with various objectives. This module processes time-series data sent from the physical space using statistical techniques, and sends guidance back to the entities in the physical space. The data workflow is triggered via Apache Airflow, an open-source workflow management platform.

Data Stores of our cloud–edge architecture are made up with: 1) Amazon S3, a scalable storage infrastructure to build our Digital Twin data lakes [64]; 2) Amazon DocumentDB (with MongoDB compatibility), a database service that is purpose-built for JSON data to execute flexible, low latency queries to obtain a near real-time record of events in parallel on a massive scale [65]; and 3) Redis, an open-source, highly replicated, nonrelational kind of database and caching server [66].

Outside of Amazon VPC but still inside of the cloud layer sits AWS IoT Core [67], which enables the connection between IoT devices (such as mobile apps, simulators, real vehicles, and RC vehicles in this study) and AWS cloud without the need to provision or manage servers. It supports various devices and messages, and can process/route those messages to AWS endpoints/devices reliably and securely. A Bulk Data Ingestion module is also developed in this cloud–edge architecture, enabling the ingestion of terabytes of data in batch mode into our data lakes. Some scenarios where this module can be triggered are: 1) end-of-vehicle-trip bulk data ingestion; 2) periodic bulk data ingestion; 3) event-triggered data ingestion; and 4) in-vehicle data logging. OpenID Connect is a simple identity layer on top of the OAuth 2.0 authorization protocol, which is adopted in this cloud–edge architecture to verify the identity of end users based on the authentication performed by an authorization server, as well as to obtain the basic profile information about end users [68].

#### B. API Layer

External data sources can be integrated into this architecture through the API layer, which enriches the functionalities of digital microservices. The API layer is connected with the cloud layer of this example architecture through Amazon API

Gateway, which is an AWS managed service that can create, publish, maintain, monitor, and secure APIs at any scale [69]. The API Gateway may act as the “front door” for applications to access data or functionalities from our back-end services.

In the API layer, traffic data (TomTom [70]), map data (OpenStreetMap [71]), and weather data (OpenWeather [72]) can be connected to Amazon API Gateway via HTTPS. With such data, more microservices can be deployed in the *Traffic Digital Twin* of the proposed MDT framework, and hence provide better guidance toward humans and vehicles in the physical space. Additionally, a customized Event Query API is developed to receive notifications of events and fetch data from the cloud layer. Examples of using this API are event-triggered detection of anomalous driving behavior, and notification of data ingestion in the cloud layer at the end of a trip. A customized Web portal is designed to visualize the digital processes on the cloud, and it enables end users to create and modify microservices (to be shown in Section V).

### C. Edge Layer

Although the cloud layer of this example architecture is designed to accommodate most of the digital processes of the proposed MDT framework, this does not necessarily indicate all digital processes shown in Fig. 1 must sit on the cloud. In fact, it gets difficult sometimes for connected vehicles to have the cloud access, since they continuously move around and may lose Internet connection every now and then. Therefore, a hybrid approach with cloud computing and edge computing can meet the requirements of ultralow latency for running safety-critical microservices at the edge (e.g., road-side units), and of extensive resources for running data-driven microservices on the cloud [27]. Edge computing has already been widely researched by various works in the field of connected vehicles [73], [74], and also deployed in the real world in projects like 5G-MOBIX [75] and initiatives like automotive edge computing consortium (AECC) [76].

In this example architecture, an edge layer is designed with three different modules: computing, communication, and storage.

- 1) The computing module provides a hierarchical computing platform for connected vehicles, where heterogeneous computing nodes (e.g., microprocessor, GPU, and TPU) are responsible for processing a variety of tasks offloaded by vehicles. Sophisticated middleware mechanisms are designed for determining which type of computing nodes should be selected to handle specific tasks and services.
- 2) The communication module enables real-time data exchange among the cloud, edge, and device layers. The protocols implemented in the communication module are HTTPS and two lightweight publish-subscribe network protocols, MQTT and Zenoh [77]. Zenoh has been designed to address the needs of applications that deal with data in movement, data at rest and computation in a scalable, efficient and location transparent data manner.
- 3) The storage module is capable of caching temporary data. The cached data could be either downloaded from

the cloud layer or uploaded from the device layer. Some representative mobility applications that could benefit from the storage module include OTA (i.e., caching the update content distributed by the cloud at the edge) and crowdsourcing high-definition (HD) map (i.e., caching uploaded sensor data from vehicles to update the local map fragment accordingly at the edge). In our case study shown in Section V, the storage module of the edge layer also plays a crucial role to store the raw CAN BUS data of the vehicle.

### D. Device Layer

The device layer of this example architecture is shown on the right side of Fig. 2, which stands for the *Human*, *Vehicle*, and *Traffic* building blocks in the physical space of the MDT framework. Mobile apps are designed for both Android and iOS, where end users’ position and speed data (measured by GPS and gyroscope) can be uploaded to AWS IoT Core directly via MQTT in real time.

Real vehicles and RC vehicles are the major data sources in the device layer of this example architecture. Dynamics data from CAN BUS (or ROS2 for RC vehicles), position data from the localization module, together with perception data from perception sensors (e.g., camera, radar, LIDAR), is sampled in real time and pushed to the edge layer. On the other hand, guidance from the digital space of the MDT framework is received from the edge layer, and gets actuated on real vehicles and RC vehicles.

External simulators, such as MATLAB [78], microscopic traffic simulators SUMO [79] and PTV VISSIM [80], and game engine-based simulators CARLA [81] and Unity [82], are used in this architecture to emulate the physical devices of the MDT framework. Note these simulators in the device layer only play the role as physical entities in the physical space, i.e., only for sampling and actuation processes. However, simulators can also be deployed in the cloud layer or the edge layer to play the role as digital entities, i.e., for modeling, learning, simulation and prediction.

## V. EXAMPLE MICROSERVICES AND CASE STUDY

This section first introduces example microservices of each digital building block of the proposed MDT framework, including user management and driver type classification of the *Human Digital Twin*, cloud-based ADAS of the *Vehicle Digital Twin*, and traffic flow monitoring and variable speed limit of the *Traffic Digital Twin*. Then, a case study of the P-ACC system is conducted, which integrates aforementioned example microservices into one application. Real-world experimental results showcase the effectiveness of P-ACC while functioning in the MDT framework, compared to traditional ACC systems that purely rely on onboard sensing and computing.

### A. Human Digital Twin: User Management and Driver Type Classification

This section introduces example microservices of the *Human Digital Twin*. A Web portal is built to enable various data management and visualization functionalities, where



Fig. 3. Real-world demonstration in Mountain View, CA of the cloud-based driver scoring system, used for user management and driver type classification.

related data sampled in the physical space is transmitted and visualized on the Web portal through HTTPS.

To begin with, each human user needs to register a unique account of the MDT through the Web portal, and also to specify the group(s) he/she belongs to: “viewer,” “supervisor,” and/or “admin.” Particularly, the user account that belongs to the “admin” group is granted full access to the human data lake, which can register/delete any account in the digital space, alter the group(s) any account belongs to, and subscribe/unsubscribe microservices for any account. “Supervisor” account can only access its own human data lake with modification right, while “viewer” account has no modification right at all.

Human–vehicle association is also available through the Web portal, which enables the human user to associate his/her vehicle in the vehicle data lake, building the connection between the *Human Digital Twin* and the *Vehicle Digital Twin*. Therefore, once the user account is registered, all the data generated by this human user in the physical space will be stored both in the user data lake (under the particular user ID) and in the vehicle data lake (under the particular vehicle ID).

Besides the aforementioned details regarding user management, the built-in microservices of the *Human Digital Twin* can also be applied. One example is shown as Fig. 3, where different driving scores of a driver (i.e., overall score, eco score, safe score, and comfort score) are calculated by the opensource MOVESTAR model [83] running on the cloud in real time. These driving scores can be further compared with the historical data, which is generated by other drivers and stored in the human data lake to classify this driver into a certain type. An example classification result is visualized on the Web portal (currently shown as “Competent” in Fig. 4), which can be used for other microservices, such as behavior prediction, personalized guidance, and insurance pricing. The historical classification results of this driver can also be retrieved by clicking the detailed trip list shown in Fig. 4, which further validates the power of the *Human Digital Twin* in terms of storage.

#### B. Vehicle Digital Twin: Cloud-Based ADAS

This section introduces example microservices of the *Vehicle Digital Twin* through a cloud-based ADAS, where cloud computing is leveraged to provide visualization guidance and control commands toward connected vehicles. As

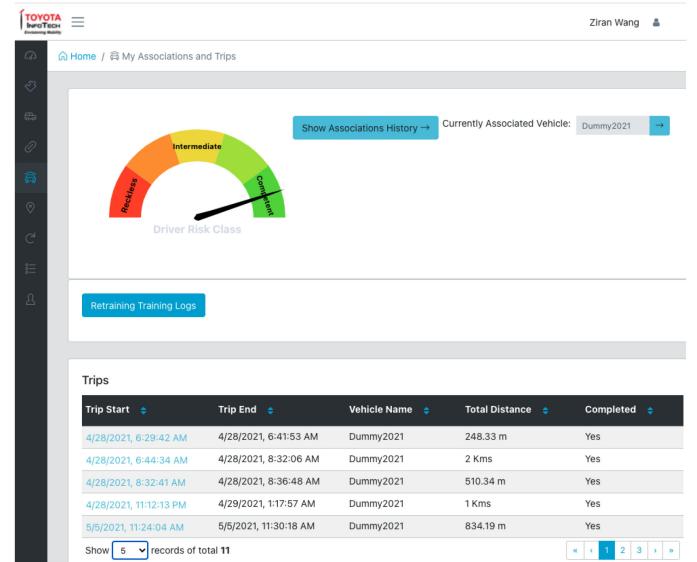


Fig. 4. User management page on the Web portal of the MDT, showing user’s driver risk class (among reckless, intermediate, and competent), associated vehicle (currently and historically), and past trips with clickable links (in light blue fonts) for more details.



Fig. 5. Human-in-the-loop simulation of the cloud-based ADAS based on AWS, the Unity game engine, and the Logitech G29 Driving Force.

shown in Fig. 5, a human-in-the-loop simulation (built with AWS, the Unity game engine, and the Logitech G29 Driving Force) is conducted to emulate the data sampling process from a connected vehicle in the physical space [39]. When a human driver manually controls the vehicle, its data are sampled from the physical components (e.g., CAN BUS, radar, camera) and uploaded to the data lake of the *Vehicle Digital Twin*. This process is shown as the lower-left corner “Unity-AWS Uplink Message” of Fig. 5.

The data stored in the data lake (potentially from all historical trips) of this vehicle is inputted to microservices of the *Vehicle Digital Twin* (e.g., cooperative planning, cooperative control, etc.). Machine learning-based algorithms are implemented to learn the performance and preference of each vehicle and/or driver, where the algorithm outputs may include prediction/guidance of their current status and future behaviors.

Such algorithm outputs are downloaded from the *Vehicle Digital Twin* to the vehicle, shown as the lower-right corner of



Fig. 6. Prediction and guidance information received from the *Vehicle Digital Twin* on the cloud is visualized through an AR HUD, which may include: driving proficiency score and its trend, potential action (e.g., hard braking or lane change) and its possibility, as well driving mood score.

Fig. 5 “AWS-Unity Downlink Message.” By leveraging computer vision technologies, this information is overlaid on top of each vehicle through an AR HUD design, assisting the decision making of other drivers [84]. As shown in Fig. 6, from this driver’s field of view, the following information of the surrounding vehicles and their drivers can be known (from top to bottom of the overlaid information): driving proficiency score and its trend, potential action (e.g., hard braking or lane change) and its possibility, as well as driving mood score.

Compared to a traditional ADAS that relies on pure onboard sensing and processing of the ego vehicle, the key advantages that MDT brings to this cloud-based ADAS are:

- 1) Heavy computations, such as training a machine learning algorithm based on the sampled data, can be offloaded to the cloud to utilize more computing power and hence saves time.
- 2) Additional data sources in the physical space, such as surrounding vehicles and downstream traffic, can be utilized to enhance the functionalities of ADAS.
- 3) The cloud-based ADAS can be easily migrated through the cloud and hence increases accessibility, which can be available on various vehicles for the same driver, or for various drivers on the same vehicle.
- 4) Updates of ADAS algorithms and applications can be conducted much quicker and easier through OTA updates.

### C. Traffic Digital Twin: Traffic Flow Monitoring and Variable Speed Limit

This section introduces example microservices of the *Traffic Digital Twin*: traffic flow monitoring and variable speed limit. A mobile app is built (both in iOS and Android versions) to allow end users to upload their data from the physical space to the digital space, which may include latitude, longitude, and speed. At the initialization step of the app, the user is asked to associate with a vehicle in the digital space, and also set the data push rate. Once “START” button is pressed, the app will get data from the mobile phone, and push that to the *Vehicle Digital Twin* through AWS IoT Core. This app is adopted to generate traffic flow and hence represents connected vehicles traveling in the physical space.

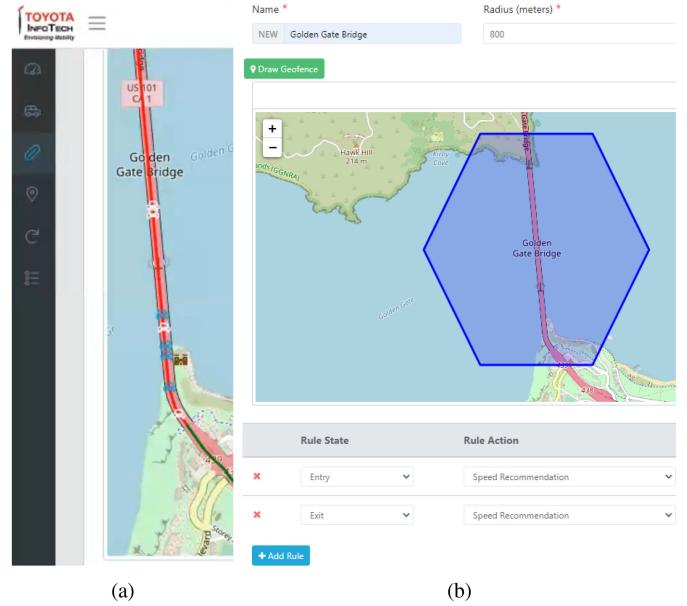


Fig. 7. (a) Traffic flow monitoring page of the Web portal (digital space) that visualizes the data received from end users (physical space) and (b) Variable speed limit based on geo-fences.

When the traffic flow monitoring microservice is turned on, it gets the data from all *Vehicle Digital Twins* and aggregates it in the *Traffic Digital Twin*. A demonstration of this microservice running in motion is shown in the snapshot Fig. 7(a). The count of all vehicles running on the specific link is calculated, and then gets further divided by the link length and number of lanes to get the traffic density value. This value is compared with the predefined ranges of heavy congestion (red), moderate congestion (orange), and no congestion (green) to determine the traffic density on that link and gets represented in the corresponding color on the map. Additionally, the different colors of vehicles visualized on the map indicate their current speeds, where the ones traveling equal or below the average link speed are shown in blue color, and the ones above the average link speed are in white.

In order to better manage traffic flows in the physical space, variable speed limits can be applied to connected vehicles through the *Traffic Digital Twin*. As can be seen from Fig. 7(b), users can easily draw a geo-fence by specifying its name, radius, latitude, and longitude of the center location. A hexagon will then be visualized on the map, and users are prompted to choose the rules that are applied to vehicles, together with their states (“entry” or “exit”) and corresponding values. Users can customize the settings of the variable speed limit on the Web portal, for example, setting the rule state entry with a value “40” and another rule state exit with a value of “60.” This setting indicates that all connected vehicles which are subscribed to this microservice will receive a speed recommendation/limit of 40 mph when entering this geo-fence, and 60 mph while exiting.

### D. Personalized Adaptive Cruise Control With Human/Vehicle/Traffic Digital Twins

This section conducts a case study of P-ACC, a system which integrates aforementioned example microservices from

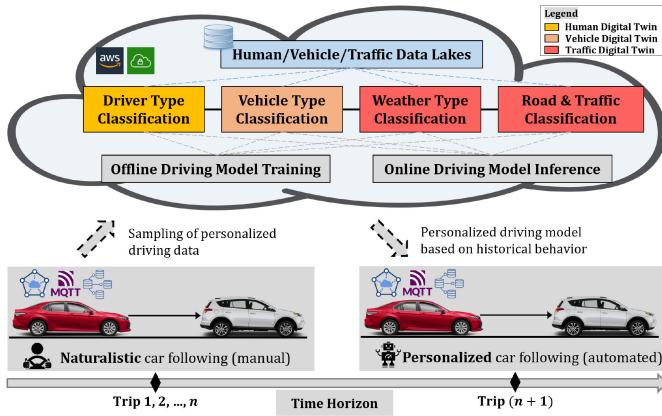


Fig. 8. P-ACC system integrates microservices from the driver digital twin (yellow), the vehicle digital twin (orange), and the traffic digital twin (red).

*Human*, *Vehicle*, and *Traffic* Digital Twins. Traditional ACC systems allow the ego vehicle to travel at a set speed and/or maintain a desirable gap with its preceding vehicle with the help of ranging sensors like radars, however, they require drivers to manually adjust speed and gap settings based on dynamic environments [85]–[87]. They do not consider each driver’s personalized driving style, and cannot account for the changes of environmental factors, such as traffic conditions, road types, and weather types [88].

As shown in Fig. 8, the P-ACC system is proposed with an edge layer on the vehicle and a cloud layer with AWS. While the computer on the edge layer (i.e., Nvidia Jetson TX2) conducts some basic edge computing tasks to filter the raw data generated by the vehicle CAN BUS, most computing tasks are offloaded from the edge layer to the cloud layer (i.e., AWS), where the *Human/Vehicle/Traffic* Digital Twins get deployed.

Once the personalized driving data sampled from a driver’s naturalistic car-following behavior gets filtered on the edge layer and uploaded to the cloud layer, it is further classified by the following Digital Twins.

- 1) The *Human Digital Twin*, with the example microservice of driver type classification (aggressive, neutral, conservative, etc.).
- 2) The *Vehicle Digital Twin*, with the example microservice of vehicle type classification (ego/preceding vehicle being a sedan, SUV, truck, etc.).
- 3) The *Traffic Digital Twin*, with the example microservices of weather type classification (sunny, cloudy, rainy, snowy, foggy, etc.), road-type classification (rural, urban, highway, etc.) and traffic-type classification (congested, moderate, light, etc.).

The aforementioned classifications can be conducted either by simple rule-based algorithms or complex machine learning algorithms [48]. Once the driving data are classified into various clusters, the data in each cluster can be used by machine learning algorithms to train a personalized driving model. In this case study, the maximum entropy inverse reinforcement learning (Max-Ent IRL) algorithm [89] is adopted, where the probability of a trajectory is proportional to the sum of the exponential rewards accumulated along the

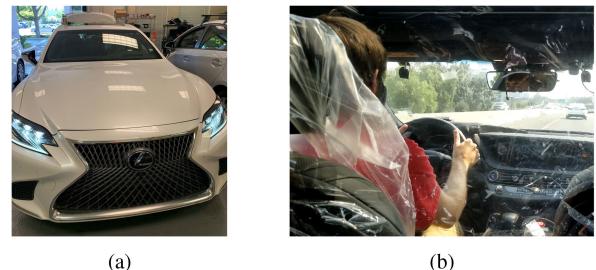


Fig. 9. (a) Mass-produced vehicle that is used for the end-to-end latency test and (b) 10-min trip conducted on Highway 237 in California to collect CAN BUS data from the vehicle.

trajectory

$$\begin{aligned} p(\xi|\alpha) &= \frac{1}{Z(\alpha)} \exp\left(\sum_t R_\alpha(s_t)\right) \\ &= \frac{1}{Z(\alpha)} \exp\left(\sum_t \alpha^T \Phi(s_t)\right) \end{aligned} \quad (1)$$

where  $Z(\alpha)$ , called partition function, equals to  $\sum_\xi \exp(\sum_t R_\alpha(t))$ . To recover the reward function, the maximum log likelihood method is used at the demonstration trajectories with respect to the weight of the reward function.

$$L(\alpha|\xi) = \max_\alpha \sum_\xi \log p(\xi|\alpha) \quad (2)$$

Then, the gradient of the weight  $\alpha$  can be written in the following form:

$$\nabla_\alpha L = \sum_\xi p(\xi) \sum_{s \in \xi} \Phi(s) - \sum_\xi D_s \sum_{s \in \xi} \Phi(s) \quad (3)$$

The first term  $\sum_\xi p(\xi) \sum_{s \in \xi} \Phi(s) = \tilde{f}$  is named expected feature count. The second term  $\sum_\xi D_s \sum_{s \in \xi} \Phi(s) = \bar{f}$  is named empirical feature count, and  $D_s$  is the state visitation frequency.

Based on the recovered reward function from the Max-Ent IRL algorithm, the particular driver’s desired car-following gap (e.g., 70 m) at each speed value (e.g., 35 m/s) can be calculated by the following equation and then put into a look-up table:

$$d_{\text{des}}(v) = \arg \max_g r(v) \quad (4)$$

This  $d_{\text{des}}-v$  table is the trained driving model for the certain cluster of driving data, which can be inferred in real time once the driver/vehicle/weather/road/traffic type match.

A real-world end-to-end test is conducted, which utilizes the CAN BUS data generated from a Lexus LS test vehicle shown as Fig. 9(a), including 15 data fields, such as position, speed, acceleration, and so on. Time-series data generated by naturalistic driving [i.e., Fig. 9(b)] gets filtered on the edge and sent to AWS through edge communication module every 10 min. When the driver turns on the ACC feature, the cloud layer infers the particular driving model trained with the data in the particular driver/vehicle/weather/road/traffic type, and pushes that model to the vehicle through the edge communication module. ACC at this time will satisfy this driver’s personalized car-following preference, and hence acts as a P-ACC.

TABLE I  
TAKEOVER RESULTS OF P-ACC COMPARED TO TRADITIONAL ACC  
(PERCENTAGE OF TAKEOVER DURATION DURING A TRIP)

Driver	P-ACC (Proposed)	ACC (Baseline)	Improvement
A	3.5%	18.5%	81.1%
B	1.5%	17.4%	91.4%
C	1.6%	12.0%	86.7%
Average	2.2%	16%	<b>86.4%</b>

TABLE II  
LATENCY RESULTS OF PROCESSING 10-MIN CAN BUS  
DATA AND QUERYING THE DATA LAKES

Data Frequency	Uplink Data Total Size	Uplink Latency	Cloud Computing+ Downlink Latency	Overall Latency
0.5 Hz	12 MB	1.4 s	16.1 s	17.5 s
1 Hz	24 MB	2.1 s	16.1 s	18.2 s
5 Hz	120 MB	10.6 s	16.1 s	26.7 s
10 Hz	240 MB	21.1 s	16.1 s	33.2 s

The benefit of the proposed P-ACC with the MDT framework is quantitatively measured using the takeover percentage. Takeover denotes the status where the driver steps onto the acceleration pedal or brake pedal when he/she feels uncomfortable. The takeover percentage is the takeover time divided by the overall P-ACC (or ACC in the baseline) activation time, and the results are presented in Table I. Based on the test results from three different drivers in different environments, the takeover event happens 86.4% less frequently when drivers use the proposed IRL-based P-ACC, compared to a traditional PID-based ACC that purely relies on real-time onboard computing.

The end-to-end latency results are shown in Table II, indicating that under different sampling frequency of the CAN BUS data, the uplink latency is relatively proportional to the uplink batch size, with a minimum of 1.4 s for 12 MB and a maximum of 21.1 s for 240 MB. However, the majority of the end-to-end latency is made up by the aforementioned cloud computing processes, which roughly takes 16 s under different data frequency. This means once the P-ACC feature is switched on by a driver, it needs a 16 s warm-up period to get the corresponding IRL-based personalized model. During this warm-up period, the baseline PID model can be run to guarantee car-following safety.

In a nutshell, the end-to-end testing showcases the capability of the proposed MDT framework to execute complex cloud computing processes based on bulk uplink data within a rational range of latency, and the computing results improve the performance of commercially available vehicle applications. In real-time mobility applications like the ones shown in the *Human Digital Twin* as Fig. 3, and in the *Vehicle Digital Twin* as Figs. 5 and 6, where only a limited number of cloud computing processes are executed based on the uplink data stream, our MDT framework guarantees 80 ms as the medium end-to-end latency to enable safety-critical and time-critical applications [37].

## VI. FUTURE CHALLENGES

In the future development and utilization of the Digital Twin technology in both academia and industry, together with the involvements of the connected vehicle technology and cloud-edge computing, numerous challenges need to be addressed from the perspectives of both research and engineering. Some of the major challenges are discussed in this section with open questions for future studies to solve.

### A. Digital Twin Standardization

Although Digital Twin technology has gained momentum in various domains during the past decade, there is no universal definition of this technology, let alone existing standardization. Currently, the joint technical committee “IoT and Digital Twin” of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) is still developing the standards for the Digital Twin in terms of concepts and terminology [90], as well as use cases [91]. Additionally, the specific standards for the Digital Twin manufacturing framework are also under development by the ISO technical committee “Industrial Data,” with focuses on overview and general principles [92], reference architecture [93], digital representation of manufacturing elements [94], and information exchange [95].

Similar to manufacturing, specific standards need to be developed for the transportation domain, so Digital Twin technology can be fully deployed on the connected vehicles we will be riding in the future. Such standards can be used by different organizations to define APIs for Digital Twin data access, enabling different transportation entities (e.g., human drivers, vehicles, and traffic infrastructures) to securely and reliably store, manage, and retrieve records. Related standards can also help developers design human machine interfaces (HMIs) to enable better interactions between physical and digital spaces of the Digital Twin.

However, the standardization of the Digital Twin in the transportation domain can face numerous challenges, since the consensus may be difficult to reach across the public sector (e.g., transportation agencies) and the private sector (e.g., automotive manufacturers, suppliers, and network providers), similar to the everlasting debate between DSRC and Cellular Vehicle-to-Everything (C-V2X) communication for the communication technology of connected vehicles.

### B. AI for Cloud–Edge Computing Systems

Breakthroughs in AI techniques, including advanced machine learning algorithms and the availability of high performance computing platforms, have recently received much attention as a key enabler for edge AI, next-generation cloud computing, and large-scale intelligent transportation systems. Meanwhile, machine learning techniques, such as federated learning, provide many new opportunities in the way we optimize complex systems and manage different connected services and system resources [96], [97]. However, the evolution toward learning-based cloud–edge computing systems is still in its early stage, and the realization of its promised

benefits in our proposed MDT framework requires thorough research and development.

For example, directly applying AI techniques in system management might lead to significant performance degradation due to the unpredictable management actions during the training phase. Therefore, fundamental questions such as how AI techniques can significantly complement the system design and management of the MDT framework still remain. Other open research challenges are how to collect high-quality data that contain both system profiles and performance metrics for training high-fidelity Digital Twins; how to design AI and machine learning solutions with high generalization ability that can adapt in the high-variance and dynamic traffic circumstances; and how to enhance the robustness of AI-based Digital Twin methods in real-world environments.

### C. Public Cloud or Private Cloud

In this study, a public cloud-edge architecture has been designed and deployed on AWS. However, this does not necessarily mean our proposed MDT framework can only work on AWS instead of other cloud platforms. Other commercial platforms like Microsoft Azure (especially with its “Azure Digital Twins”), GCP and Alibaba Cloud could also be the alternatives to accommodate the MDT framework.

Nevertheless, a cloud platform must be trustworthy to deploy any of the microservices mentioned in this study (especially for the ones related to the *Human Digital Twin*), as end user information and related data need to be secured from being compromised. The nature of public cloud platforms inevitably gives away the control of resources to some extent, which may introduce cybersecurity and privacy risks for end users.

A private cloud platform, on the other hand, consists of cloud computing resources used exclusively by one business or organization. Since the services and infrastructure are always maintained on a private network without sharing with others, a private cloud platform can address the security and privacy issues faced by a public cloud platform. Private cloud platforms also make it easier for end users to customize cloud resources to meet specific requirements and implement specific functions, which was proved in our private cloud-based cooperative ramp merging experiment [38].

Although private cloud platforms are more secured and flexible, it has several disadvantages compared to public cloud platforms. In general, private cloud platforms are more expensive, since hardware and software should be dedicated solely to particular organizations that they serve (and hence paid solely by those organizations). In terms of scalability and reliability, private cloud platforms are also outperformed by public cloud platforms, because the public ones provide on-demand resources to meet various organizations’ needs, and also provide a vast network of servers to ensure against failure.

Therefore, based on specific requirements and needs of end users, choices can be made between public and private cloud platforms, considering their advantages and disadvantages described above. Additionally, building Digital Twins with a hybrid cloud approach (combining public and private

cloud platforms, potentially with edge/fog computing) provides another possibility.

### D. Heterogeneous Wireless Communication Environment

Future vehicular networks tend to be highly heterogeneous, where a variety of radio access technologies, such as DSRC, Wi-Fi, long-term evolution (LTE), and C-V2X, will co-exist in the vehicular environment. DSRC has been widely deployed in the past decades, which enables vehicles to communicate with each other and road-side infrastructures directly, without involving cellular or additional infrastructures. But its drawbacks like low peak data rate and limited coverage become the bottleneck for satisfying the Quality-of-Service (QoS) requirements for many emerging microservices of our proposed MDT framework, such as cooperative perception, HD map, and anytime OTA update, that require transmitting a high volume of data in a short time window.

Although WiFi and LTE are capable of providing a larger bandwidth and throughput, they require more sophisticated mechanisms to tackle MDT management challenges incurred by one of the traits of vehicles - high mobility. In contrast, the more recent C-V2X technology is compatible with 5G and could address the communication challenges due to vehicles’ high mobility. However, C-V2X is not affordable and widely deployed compared with DSRC [98].

Therefore, network heterogeneity might be a promising solution to tackle the challenges presented above and to enable connected vehicles to achieve diverse QoS requirements for our proposed MDT framework. However, multiple future challenges might be imposed by the network heterogeneity:

- 1) Maintaining a seamless connectivity within a heterogeneous network environment is challenging. Sophisticated mechanisms for managing the connectivity across different radio access technologies is desirable.
- 2) Balancing the tradeoff between latency and cost is crucial. How to achieve a satisfactory latency with the lowest overhead (e.g., data transmission cost and energy consumption) is challenging.
- 3) Connected vehicles heavily rely on data that are shared with surrounding intelligent nodes, such as roadside units, pedestrians’ devices, other vehicles. Open issues like how to secure these shared data with cost-efficient solutions, and what might be the role of cloud-edge computing systems still remain.

## VII. CONCLUSION

In this study, a MDT framework has been developed for connected vehicles with the help of cloud computing and edge computing. It has been found from the literature review that, although the Digital Twin concept has been recently studied in the transportation domain, there is no related literature that leverages this technology together with cloud-edge computing to benefit connected vehicles with detailed microservices. In this study, the proposed MDT framework has been demonstrated with details regarding all of its components: *Human*, *Vehicle*, and *Traffic*, together with their associated *Human Digital Twin*, *Vehicle Digital Twin*, and *Traffic Digital Twin*.

The public cloud service AWS has been adopted in this study to design the cloud–edge architecture, which deploys the MDT framework and makes it a reality rather than a concept. To showcase the effectiveness of the MDT framework, a case study of P-ACC has been conducted leveraging the key microservices of the MDT framework: user management and driver type classification of the *Human Digital Twin*, cloud-based ADAS of the *Vehicle Digital Twin*, and traffic flow monitoring and variable speed limit of the *Traffic Digital Twin*.

In conclusion, scalability, reliability, security, cost, latency, and fidelity are the key factors to consider when designing any Digital Twin frameworks. Along with the rapid development of the Digital Twin, it can be envisioned that more related studies will be conducted in the mobility domain to tackle the real-world challenges in the near future.

#### ACKNOWLEDGMENT

The authors want to sincerely thank the Atos team for their technical supports in building the Connected Car Cloud (CCC), and also thank Shalini Keshavamurthy, Dr. Seyhan Ucar, Haritha Muralidharan, Dr. Takamasa Higuchi, Johnny Wang, Divya Sai Toopran, Kazuhisa Shitanaka, Akio Orii, and Kenichi Murata for their support and guidance.

The contents of this study 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] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] Y. Guo, X. Hu, B. Hu, J. Cheng, M. Zhou, and R. Y. Kwok, “Mobile cyber physical systems: Current challenges and future networking applications,” *IEEE Access*, vol. 6, pp. 12360–12368, 2017.
- [3] “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.” Grand View Research. [Online]. Available: <https://www.grandviewresearch.com/industry-analysis/digital-twin-market> (accessed Dec. 9, 2021).
- [4] E. Glaessgen and D. Stargel, “The digital twin paradigm for future NASA and U.S. Air Force vehicles,” in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf.*, 2012, p. 1818.
- [5] J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manuf. Lett.*, vol. 1, no. 1, pp. 38–41, 2013.
- [6] J. Lee, B. Bagheri, and H.-A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems,” *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.
- [7] E. Tuegel, “The airframe digital twin: Some challenges to realization,” in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf.*, 2012, p. 1812.
- [8] M. Schluse and J. Rossmann, “From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems,” in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, 2016, pp. 1–6.
- [9] Q. Wang, W. Jiao, P. Wang, and Y. Zhang, “Digital twin for human-robot interactive welding and welder behavior analysis,” *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 2, pp. 334–343, Feb. 2021.
- [10] R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, “About the importance of autonomy and digital twins for the future of manufacturing,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.
- [11] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, “Digital twin data modeling with automationml and a communication methodology for data exchange,” *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.
- [12] A. Canedo, “Industrial IoT lifecycle via digital twins,” in *Proc. 11th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, 2016, p. 29.
- [13] 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.
- [14] J. Cheng, G. Yuan, M. Zhou, S. Gao, Z. Huang, and C. Liu, “A connectivity-prediction-based dynamic clustering model for VANET in an urban scene,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8410–8418, Sep. 2020.
- [15] Y. Ma, Z. Wang, H. Yang, and L. Yang, “Artificial intelligence applications in the development of autonomous vehicles: A survey,” *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020.
- [16] L. Silva *et al.*, “Computing paradigms in emerging vehicular environments: A review,” *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 3, pp. 491–511, Mar. 2021.
- [17] K. M. Alam and A. El Saddik, “C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems,” *IEEE Access*, vol. 5, pp. 2050–2062, 2017.
- [18] S. A. Kumar, R. Madhumathi, P. R. Chelliah, L. Tao, and S. Wang, “A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance,” *J. Rel. Intell. Environ.*, vol. 4, no. 4, pp. 199–209, 2018.
- [19] 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.
- [20] “Amazon Web Services (AWS)—Cloud Computing Services.” Amazon. Mar. 3, 2021. [Online]. Available: <https://aws.amazon.com/>
- [21] “Azure Digital Twins.” Microsoft Azure. Jun. 8, 2021. [Online]. Available: <https://azure.microsoft.com/en-us/services/digital-twins/>
- [22] “Cloud Computing Services | Google Cloud.” Google. Mar. 3, 2021. [Online]. Available: <https://cloud.google.com/>
- [23] “Alibaba Cloud: Reliable & Secure Cloud Solutions to Empower Your Global Business.” Alibaba. Mar. 3, 2021. [Online]. Available: <https://www.alibabacloud.com/>
- [24] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, “Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies,” *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.
- [25] S. Olariu, I. Khalil, and M. Abuelela, “Taking VANET to the clouds,” *Int. J. Pervasive Comput. Commun.*, vol. 7, no. 1, pp. 7–21, 2011.
- [26] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, “A survey on vehicular cloud computing,” *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.
- [27] H. Wang *et al.*, “Architectural design alternatives based on cloud/edge/fog computing for connected vehicles,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2349–2377, 4th Quart., 2020.
- [28] M. Gerla, “Vehicular cloud computing,” in *Proc. The 11th Annu. Mediterr. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, 2012, pp. 152–155.
- [29] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. C. Zambrano, “Smart ITS sensor for the transportation planning based on IoT approaches using serverless and microservices architecture,” *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 17–27, 2018.
- [30] A. Bhatnagar, V. Sharma, and G. Raj, “IoT based car pollution detection using AWS,” in *Proc. Int. Conf. Adv. Comput. Commun. Eng. (ICACCE)*, 2018, pp. 306–311.
- [31] H. W. Deng, M. Rahman, M. Chowdhury, M. S. Salek, and M. Shue, “Commercial cloud computing for connected vehicle applications in transportation cyberphysical systems: A case study,” *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, pp. 6–19, 2021.
- [32] C. Schwarz and Z. Wang, “The role of digital twins in connected and automated vehicles,” *IEEE Intell. Transp. Syst. Mag.*, early access, Jan. 5, 2022, doi: [10.1109/MITS.2021.3129524](https://doi.org/10.1109/MITS.2021.3129524).
- [33] S. Shiraishi, Z. Jiang, and B. Kim, “Digital twin for vehicle risk evaluation,” U.S. Patent 16 007 693, Dec. 19, 2019.
- [34] Z. Jiang, S. Shiraishi, and B. Kim, “Collision avoidance for a connected vehicle based on a digital behavioral twin,” U.S. Patent 10 843 689, Nov. 24, 2020.
- [35] F.-Y. Wang, “Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
- [36] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, “Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence,” *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 577–587, Sep. 2017.

- [37] Z. Wang, X. Liao, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems," in *Proc. IEEE 91st Veh. Technol. Conf.*, May 2020, pp. 1–6.
- [38] 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.*, early access, Jul. 30, 2021, doi: [10.1109/TITS.2020.3045123](https://doi.org/10.1109/TITS.2020.3045123).
- [39] 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.
- [40] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. Hansen, "Vision-cloud data fusion for ADAS: A lane change prediction case study," *IEEE Trans. Intell. Veh.*, early access, Aug. 10, 2021, doi: [10.1109/TIV.2021.3103695](https://doi.org/10.1109/TIV.2021.3103695).
- [41] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, early access, Jul. 21, 2021, doi: [10.1109/TIV.2021.3100465](https://doi.org/10.1109/TIV.2021.3100465).
- [42] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
- [43] "V2X." 3GPP. [Online]. Available: <https://www.3gpp.org/v2x> (accessed Jan. 17, 2022).
- [44] X. Liao *et al.*, "Online prediction of lane change with a hierarchical learning-based approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 1–7.
- [45] Z. Wang, G. Wu, and M. J. Barth, "Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2029–2038, May 2020.
- [46] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2018, pp. 2496–2505.
- [47] X. Zhao *et al.*, "Co-simulation platform for modeling and evaluating connected and automated vehicles and human behavior in mixed traffic," *SAE MobilityRxiv*, to be published.
- [48] 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.
- [49] 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.
- [50] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "SafeDrive: Online driving anomaly detection from large-scale vehicle data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2087–2096, Aug. 2017.
- [51] W. Weidner, F. W. Transchel, and R. Weidner, "Telematic driving profile classification in car insurance pricing," *Ann. Actuarial Sci.*, vol. 11, no. 2, pp. 213–236, 2017.
- [52] T. Masuda, B. Kim, and S. Shiraishi, "Proactive vehicle maintenance scheduling based on digital twin simulations," U.S. Patent 15 908 768, Aug. 29, 2019.
- [53] M. Papageorgiou, E. Kosmatopoulos, and I. Papamichail, "Effects of variable speed limits on motorway traffic flow," *Transp. Res. Rec.*, vol. 2047, no. 1, pp. 37–48, 2008. [Online]. Available: [https://journals.sagepub.com/doi/abs/10.3141/2047-05?casa\\_token=MvsF4\\_OAfhoAAAAA:GrGFy7ZpK7f1R-r3FW9uPGXGO-lyPVgWYPA8XQBKUXUi0KvqbGlpsW7Efml\\_kRWHfkswMJuYegUzg](https://journals.sagepub.com/doi/abs/10.3141/2047-05?casa_token=MvsF4_OAfhoAAAAA:GrGFy7ZpK7f1R-r3FW9uPGXGO-lyPVgWYPA8XQBKUXUi0KvqbGlpsW7Efml_kRWHfkswMJuYegUzg)
- [54] Y. Pan, N. Wu, T. Qu, P. Li, K. Zhang, and H. Guo, "Digital-twin-driven production logistics synchronization system for vehicle routing problems with pick-up and delivery in industrial park," *Int. J. Comput. Integr. Manuf.*, vol. 34, nos. 7–8, pp. 1–15, 2020.
- [55] A. K. Tripathy, P. K. Tripathy, A. G. Mohapatra, N. K. Ray, and S. P. Mohanty, "WeDoShare: A ridesharing framework in transportation cyber-physical system for sustainable mobility in smart cities," *IEEE Consum. Electron. Mag.*, vol. 9, no. 4, pp. 41–48, Jul. 2020.
- [56] T. Higuchi, S. Ucar, and O. Altintas, "A collaborative approach to finding available parking spots," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, 2019, pp. 1–5.
- [57] "Amazon Virtual Private Cloud." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/vpc/?vpc-blogs.sort-by=item.additionalFields.createdDate&vpc-blogs.sort-order=desc>
- [58] "Amazon Elastic Kubernetes Service." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/eks/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&eks-blogs.sort-by=item.additionalFields.createdDate&eks-blogs.sort-order=desc>
- [59] "Apache Kafka." Apache. Jun. 5, 2021. [Online]. Available: <https://kafka.apache.org/>
- [60] "Apache Storm." Apache. Jun. 5, 2021. [Online]. Available: <https://storm.apache.org/>
- [61] "The Scalable Time Series Database." OpenTSDB. Jun. 5, 2021. [Online]. Available: <http://opentsdb.net/>
- [62] "Apache Spark." Apache. Jun. 5, 2021. [Online]. Available: <https://spark.apache.org/>
- [63] "Amazon EMR." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/emr/>
- [64] "Amazon S3." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/s3/>
- [65] "Amazon DocumentDB (with MongoDB compatibility)." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/documentdb/>
- [66] "Redis." Jun. 5, 2021. [Online]. Available: <https://redis.io/>
- [67] "AWS IoT Core." Amazon. May 31, 2021. [Online]. Available: <https://aws.amazon.com/iot-core/>
- [68] "Welcome to OpenID Connect." OpenID. [Online]. Available: <https://openid.net/connect/> (Accessed: Jun. 5, 2021).
- [69] "Amazon API Gateway." AWS. Jun. 5, 2021. [Online]. Available: <https://aws.amazon.com/api-gateway/>
- [70] "Real-Time Traffic Data." TomTom. Jun. 5, 2021. [Online]. Available: <https://www.tomtom.com/products/real-time-traffic/>
- [71] "API." OpenStreetMap. Jun. 1, 2021. [Online]. Available: <https://wiki.openstreetmap.org/wiki/API>
- [72] "Weather API." OpenWeather. Jun. 1, 2021. [Online]. Available: <https://openweathermap.org/api>
- [73] J. Santa *et al.*, "MIGRATE: Mobile device virtualisation through state transfer," *IEEE Access*, vol. 8, pp. 25848–25862, 2020.
- [74] J. Santa, P. J. Fernández, J. Ortiz, R. Sanchez-Iborra, and A. F. Skarmeta, "SURROGATES: Virtual OBUs to foster 5G vehicular services," *Electronics*, vol. 8, no. 2, p. 117, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/2/117>
- [75] "Driving forward Connected & Automated Mobility." 5G-MOBIX. Aug. 18, 2021. [Online]. Available: <https://www.5g-mobix.com/>
- [76] "Driving the Vision of Network and Computing Infrastructure for Connected Car Big Data." Automotive Edge Computing Consortium. Aug. 18, 2021. [Online]. Available: <https://aecc.org/>
- [77] "What Is Zenoh." Zenoh. 2022. [Online]. Available: <https://zenoh.io/docs/overview/>
- [78] "MATLAB." MathWorks. [Online]. Available: [https://www.mathworks.com/products/matlab.html?s\\_tid=hp\\_products\\_matlab](https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab) (accessed Aug. 10, 2020).
- [79] "Simulation of Urban Mobility." Eclipse SUMO. [Online]. Available: <https://www.eclipse.org/sumo/> (accessed May 3, 2021).
- [80] "PTV Vissim Is the World's Most Advanced and Flexible Traffic Simulation Software." PTV GROUP. [Online]. Available: <https://www.ptvgroup.com/en/solutions/products/ptv-vissim/> (accessed Aug. 10, 2021).
- [81] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
- [82] "Unity for All." Unity. [Online]. Available: <https://unity.com/> (Accessed: May 3, 2021).
- [83] Z. Wang, G. Wu, and G. Scora, "MOVESTAR: An open-source vehicle fuel and emission model based on USEPA MOVES," 2020, *arXiv:2008.04986*.
- [84] Z. Wang, K. Han, and P. Tiwari, "Augmented reality-based advanced driver-assistance system for connected vehicles," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2020, pp. 752–759.
- [85] "Toyota Safety Sense: The Standard for Safety." Toyota. 2021. [Online]. Available: <https://www.toyota.com/safety-sense/>
- [86] "Adaptive Cruise Control." Ford. 2021. [Online]. Available: <https://www.ford.com/technology/driver-assist-technology/adaptive-cruise-control/>
- [87] "Adaptive Cruise Control." Volkswagen. 2021. [Online]. Available: <https://www.volkswagen-newsroom.com/en/adaptive-cruise-control-access-3664>
- [88] Y. Wang, Z. Wang, K. Han, P. Tiwari, and D. B. Work, "Personalized adaptive cruise control via Gaussian process regression," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, 2021, pp. 1496–1502.
- [89] Z. Zhao, Z. Wang, K. Han, P. Tiwari, G. Wu, and M. Barth, "Personalized car following for autonomous driving with inverse reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 1–7.

- [90] *Digital Twin—Concepts and Terminology*, ISO/IEC AWI Standard 30173, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/81442.html>
- [91] *Digital Twin—Use Cases*, ISO/IEC AWI Standard 30172, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/81578.html>
- [92] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles*, ISO/CDIS Standard 23247-1, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/75066.html>
- [93] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 2: Reference Architecture*, ISO/CDIS Standard 23247-2, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/78743.html>
- [94] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 3: Digital Representation of Manufacturing Elements*, ISO/CDIS Standard 23247-3, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/78744.html>
- [95] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 4: Information Exchange*, ISO/CDIS Standard 23247-4, Jun. 8, 2021. [Online]. Available: <https://www.iso.org/standard/78745.html>
- [96] S. Lu, Y. Yao, and W. Shi, “Collaborative learning on the edges: A case study on connected vehicles,” in *Proc. 2nd USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, 2019, pp. 1–8.
- [97] D. Ye, R. Yu, M. Pan, and Z. Han, “Federated learning in vehicular edge computing: A selective model aggregation approach,” *IEEE Access*, vol. 8, pp. 23920–23935, 2020.
- [98] S. Lu and W. Shi, “The emergence of vehicle computing,” *IEEE Internet Comput.*, vol. 25, no. 3, pp. 18–22, May/Jun. 2021.



**Ziran Wang** (Member, IEEE) received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2015, and the Ph.D. degree from The University of California at Riverside, Riverside, CA, USA, in 2019.

He is currently a Principal Researcher with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA, where he leads the “Digital Twin” project. His research focuses on intelligent vehicle technology, including cooperative automated 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.



**Rohit Gupta** received the Ph.D. degree in mechanical engineering from The University of California at Santa Barbara, Santa Barbara, CA, USA, in 1998.

He is currently a Principal Cloud Architect with the InfoTech Labs, Toyota Motor North America, 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.



**Kyungtae Han** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2006.

He is currently a Senior Principal Scientist with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA. Prior to joining Toyota, he was a Research Scientist with Intel Labs, Santa Clara, CA, USA, and a Director of Locix Inc., San Bruno, CA, USA. His research interests include cyber-physical systems, connected and automated vehicle techniques, and intelligent transportation systems.



**Hauxin Wang** (Member, IEEE) received the B.S. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2015, and the Ph.D. degree in electrical and computer engineering from The University of North Carolina at Charlotte, Charlotte, NC, USA, in 2020.

He is currently a Research Scientist with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA, where he leads the “Edge Computing” project. His research interests include edge computing for connected and autonomous vehicles, applied machine learning for intelligent systems, and low-power and high-assurance cyber-physical systems.



**Akila Ganlath** received the B.S. and M.S. degrees in mechanical engineering from the University of California at Riverside, Riverside, CA, USA, in 2016 and 2019, respectively.

He is currently a Principal Researcher with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA. His research interests include control systems, autonomous driving, and software-driven vehicles.



**Nejib Ammar** received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California at Davis, Davis, CA, USA, in 2005.

He is currently a Senior Principal Scientist with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA. Previously, he served as the Chairman of ATSC 3.0 Waveform Ad-Hoc Group.



**Prashant Tiwari** received the Ph.D. degree in mechanical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2004, and the MBA degree from the University of Chicago, Chicago, IL, USA, in 2016.

He is currently an Executive Director with the InfoTech Labs, Toyota Motor North America, Mountain View, CA, USA. He is highly active in Automotive Edge Computing Consortium and SAE. Prior to joining Toyota, he held several leadership positions of increasing responsibilities at GE and UTC Aerospace Systems.