FEDERATED LEARNING FOR DIGITAL TWIN-BASED VEHICULAR NETWORKS: ARCHITECTURE AND CHALLENGES

Latif U. Khan, Ehzaz Mustafa, Junaid Shuja, Faisal Rehman, Kashif Bilal, Zhu Han, and Choong Seon Hong

ABSTRACT
A digital twin uses a virtual model of the physical system to fulfill the diverse requirements (e.g., latency, reliability, quality of physical experience) for emerging vehicular network applications. Although a twin-based implementation of vehicular networks can offer performance optimization, modeling a digital twin is a significantly challenging task. Federated learning (FL) is a better privacy-preserving, distributed learning scheme that can be used to model twin models. Although FL can offer performance enhancement, it requires careful design. Therefore, in this article, we present an overview of FL for a twin-based vehicular network. A general architecture for FL-enabled digital twins for a vehicular network is presented. Our proposed architecture consists of two spaces, such as twin space and the physical space. A physical space consists of all the physical entities (e.g., cars and edge servers) required for vehicular networks, whereas, the twin space refers to the logical space that is used for the deployment of twins. A twin space can be implemented either using edge servers and cloud servers. We also outline a few use cases of FL for the twin-based vehicular network. Finally, the article is concluded and an outlook on open challenges is presented.

INTRODUCTION
Vehicular networks enable many applications (e.g., congestion control and accident reporting) for intelligent transportation systems (ITSs). Vehicular network applications are based on diverse requirements (e.g., latency and reliability) and user-defined characteristics (e.g., quality of physical experience), which are difficult to be fulfilled by the existing wireless technologies [1]. There is a need to deploy a vehicular network using two design trends, such as self-sustaining wireless systems and proactive intelligent analytics (https://www2.deloitte.com/us/en/insights/focus/tech-trends/2020/digital-twin-applications-bridging-the-physical-and-digital.html; https://intellias.com/creating-digital-replicas-using-iot-how-digital-twin-technology-works-in-practice/). For meeting the diverse requirements [2]. Self-sustaining wireless systems will enable ITS with minimum possible intervention from the operators/users. This approach will make the system more autonomous to adapt to the varying nature of the environment, and thus more preferable for enabling emerging applications. On the other hand, proactive online learning-based wireless systems will enable proactive (i.e., prior to request) optimization of the wireless resources for ensuring the quality of service (QoS) of various ITS applications with strict latency requirements. Digital twins can be a good solution to deploy ITS applications by enabling the features of self-sustaining wireless systems and proactive online learning based systems [3].

A digital twin uses a virtual model of the physical system to enable proactive online learning and self-sustainability. For virtual modeling of the physical system, mathematical modeling (i.e., optimization, game theory, graph theory, etc.) can be used which is based on several assumptions, and thus might not truly reflect the physical phenomena. Additionally, there may be certain phenomena that cannot be modeled using mathematical modeling [4]. Coping with this challenge, machine learning (ML) can model various challenging phenomena of vehicular networks [5]. For instance, consider infotainment services in the car, where every user has data about their requested contents. Such data can be used to train twins deployed at the edge to enable proactive caching, ML can be based on either centralized training or distributed training. In training, centralized training transfers distributed device data to a centralized location, and thus results in privacy leakage because vehicles are reluctant to dispense their confidential data. To tackle this limitation of centralized ML, federated learning (FL) was proposed to use distributed devices to learn a global FL model without moving the data from devices to a centralized location for training, and thus better preserves privacy compared to centralized ML. In literature, various works considered digital twins [3, 6–8] and ML [4, 5, 9] for effectively enabling wireless system applications (e.g., ITSs). In [5], the authors presented a dispersed FL (DFL) concept for enabling a scalable and fast converging learning for wireless systems. Another study...
in [4] considered the role of efficiently enabling wireless systems by machine learning. On the other hand, the work in [3] presented the vision of digital twin-based wireless systems. Additionally, they outlined key requirements and proposed a general architecture for digital twin-based wireless systems. The work in [9] proposed an intelligent digital twin hierarchical (EDT) routing for vehicular networks. Another work [6] proposed the use of digital twins in assisting task offloading for aerial edge computing networks. Other works [7] and [8] considered digital twin along with reinforcement learning and digital twin-based defect control for vehicular edge computing and hotrolled coil, respectively.

To the best of our knowledge, different from [3–9], our work is the first one to review FL for digital twin-based vehicular networks. In contrast to the works in [3–9], our work focuses on role of FL in digital twin-based vehicular networks and present a general architecture. Also, we present an example scenario of FL toward enabling digital twin-based vehicular networks. Additionally, we present use cases of FL in twin-based vehicular networks. More specifically, our contributions include constructing a general, detailed architecture of FL for digital twin-based vehicular networks. An example scenario of FL for a twin-based vehicular network is presented. Furthermore, we present the sequence diagrams for FL in twin-based vehicular networks. We also discuss the use cases, such as intelligent analytics, edge caching, and intelligent resource management of FL toward enabling twin-based vehicular networks. Finally, we present open research challenges with their possible solutions.

**Architecture of FL and Digital Twin Based Vehicular Networks**

In this section, we present a high-level architecture of FL-enabled digital twin for vehicular networks, as shown in Fig. 1. There are two main phases in a digital twin-based vehicular network: offline training, and online operation [3]. In offline training, one can proactively train twin models prior to user requests, whereas online operation is based on instructing cars, sensors, and roadside units (RSUs) to serve end-users, as shown in Fig. 2. The architecture consists of two main layers: twins layer, and physical interaction layer. A physical interaction layer consists of all entities (e.g., autonomous cars, edge-based RSUs, and unmanned aerial vehicles (UAVs)) necessary for implementing digital twin-based vehicular networks. The architecture consists of two main layers: twins layer, and physical interaction layer. A physical interaction layer consists of all entities (e.g., autonomous cars, edge-based RSUs, and unmanned aerial vehicles (UAVs)) necessary for implementing digital twin-based vehicular networks.
One prominent challenge is the requirement of a large number of communication rounds for reaching a desirable convergence. Also, training a twin model using a massive amount of data is challenging in the case of a traditional FL performance.

Although FL can offer better privacy preservation compared to centralized ML, FL has some challenges. One prominent challenge is the requirement of a large number of communication rounds for reaching a desirable convergence. Also, training a twin model using a massive amount of data is challenging in the case of a traditional FL performance. Additionally, a centralized aggregation server might suffer from malfunctioning issues, which results in the degradation of FL. Therefore, to improve the convergence and single point of failure in FL, a DFL can be utilized to deploy distributed aggregations in digital twin-based vehicular networks [5]. Meanwhile, one can use efficient resource optimization schemes that can result in high throughput, and thus low latency communication between devices and aggregators deployed at the twin layer. In DFL, first of all, the computation of sub-global models takes place in groups in a similar fashion as that of traditional FL. After computing the sub-global model in DFL, the process of sharing sub-global models takes place, and finally, global models are obtained by aggregating sub-global models, as shown in Fig. 2b. Note that we can compute sub-global models in various ways for vehicular networks. For instance, we can compute sub-global models for enabling infotainment inside every car with multiple users. For other scenarios (e.g., traffic congestion control and lane change guidance) where data from a car is treated as a single local dataset. One can compute a sub-global model for set of cars connected to a RSU. A single RSU can be used to serve a single group for computing the sub-global model. Next, the sub-global models computed at various RSUs are aggregated to get a global model. On the other hand, if there are few RSUs and it is not possible for them to provide coverage to all autonomous cars. For a such case, one can deploy UAVs to serve cars. Multiple cars can be associated with a single UAV where sub-global model is computed. Table 1 discusses the comparison of computing sub-global models using various ways, i.e., within a single car, multiple cars using RSU, and multiple cars using a UAV. We use management complexity, communication cost, and mobility management complexity for comparison of various schemes used to compute sub-global models for DFL in vehicular networks. The management complexity of single car-based sub-global model computation is low compared to multiple cars. Cost of communication cost (i.e., access network resources) is high for UAV-based sub-global model computation and RSU-based sub-global model computation compared to single car-based sub-global model computation. The reason for highest cost for UAV-based implementation is due to the fact that UAV uses access network resources for communication between UAVs and cars. Additionally, UAVs will use access network resource for connectivity with RSUs. Mobility management complexity is highest for UAV-based implementation due to fact that both UAVs and cars are mobile.

Next, we discuss the deployment of twin objects. For strict latency applications with low computing power requirements, one can deploy twin objects at the network edge, whereas for other applications that require high computing power requirements, one can deploy twin objects at a remote cloud. Note that edge and cloud support containers/virtual machines that run twin objects. Mostly, the intelligent transportation applications (e.g., accident reporting and lane change assistance) have strict latency requirements. For such strict latency applications, it is a desirable to deploy twin objects near the devices at network edge. However, there are storage and computing power limitations at network edge. On the other hand, cloud has more storage with computing power capacity compared to edge, but at the cost of latency. Therefore, we should make a trade-off between available computing power and latency. Another approach is to use a hybrid scheme that uses both cloud-based twin objects and edge-based twin objects [1]. End-devices should be served (e.g., provide computing power and control signaling) by the edge-based twin objects until
maximum limit of available computing power of edge is reached. Beyond the available computing at the edge, twin objects deployed at cloud can be used for serving the end-devices.

**Example Scenario**

We present an example scenario of a digital twin-based infotainment system for autonomous cars. To enable infotainment services for autonomous cars with strict latency requirements, there is a need to deploy caches at the network edge (i.e., RSUs). To do so, we deploy twin objects at the network edge for serving users. Twin objects will decide which content to cache and which to move to the cloud because the edge has storage and computing power limitations compared to the cloud. To train such twin objects, one can use DFL, as shown in Fig. 3. Inside every autonomous vehicle, an access point can be installed and can act for sub-global aggregation. Inside the vehicle, cameras installed on their mobile devices can take images that can be used by a local learning model of DFL to yield a local model. There should be a certain effective local model that enables us to find the relevant infotainment item for the particular user based on passenger age, emotion, and gender. Within every autonomous vehicle, all the local models are aggregated to yield a sub-global twin model. Next to sub-global twin model computation, every autonomous vehicle can send its sub-global twin model to RSUs. Then RSUs share the sub-global twin models and perform aggregations at all RSUs to yield global twin models. This process takes place iteratively until convergence. Such a twin model at the network edge will be used by twin objects to decide which content to cache at particular RSUs. Additionally, twin objects will control the communication and computation required for enabling infotainment in autonomous vehicles.

**Use Cases**

**Secure and Robust Federated Analytics**

Autonomous vehicles share data with each other to get more information about the environment, which improves traffic management and reduces the risk of traffic accidents. However, sharing of data may cause serious privacy risks. Therefore, instead of sharing data, one can send a function of data with the edge/cloud server using federated analytics [10]. On the other hand, one can use the data generated in autonomous vehicles to train various ML models. Training based on centralized ML for traffic prediction and network manage-

| Description                        | Management complexity | Communication cost | Mobility management complexity |
|------------------------------------|-----------------------|--------------------|--------------------------------|
| Sub-global model for users         | Low                   | Lowest             | Lowest                         |
| within an autonomous car           |                       |                    |                                |
| Sub-global model for multiple cars | High                  | Low                | High (i.e., for both mobile    |
| and single RSU                      |                       |                    | cars and mobile UAVs)          |
| Sub-global model for UAVs and cars | High                  | High               |                                |

**TABLE 1.** Comparison of various schemes for computing sub-global models for vehicular networks.

![FIGURE 3.](image)  

**FIGURE 3.** FL for digital twin-based infotainment in autonomous cars.

A massive number of autonomous cars will request content from a remote cloud which will cause an increase in service requests on a vehicular network. Fetching contents from a remote cloud in ITS for various applications will suffer from high latency.
In twins, the contents are stored at the edge based on their popularity. Upon request from the users, the twin will provide a user with the content (i.e., content retrieval) in an efficient way by performing effective resource allocation.

To address the escalating content requests, edge-based caching can be used that reduces service latency and network traffic in vehicular networks. However, mobility in vehicular network setup highly influences prediction and content popularity. Vehicles connected to the edge server keep moving in transportation systems making the cached contents to be out of date. The caching process has three major tasks: content placement in the cache, content popularity prediction, and content retrieval [11]. To deploy caching at the network edge, one can use twins deployed at the edge that will perform all three tasks of caching. In twins, the contents are stored at the edge based on their popularity. Upon request from the users, the twin will provide a user with the content (i.e., content retrieval) in an efficient way by performing effective resource allocation. For resource allocation, the twin uses ML, optimization theory, game theory, and graph theory. There can be many ways for predicting the content prior to storage at the edge by twins. ML-based content popularity prediction effectively works in a mobile scenario. However, these schemes based on a centralized training need to upload and centrally process user data at a central server. Autonomous cars frequently generate data whose transfer to the centralized server is challenging. To address this limitation, one can use deep reinforcement learning (DRL)-based on FL. Figure 5 illustrates the DRL for twins-enabled intelligent caching. One can deploy twins (i.e., DRL agents) either at the network edge or on devices. Twin agents deployed at devices can perform training followed by transmitting the trained twin model to the edge for aggregation to yield a global twin model. Although this approach can send only updates to the edge server, and thus can better preserve privacy compared to centralized training of twin agents, mostly the end-devices have computing power (i.e., CPU-cycles/sec) constraints. To address this challenge, devices can send their data to the edge for training a twin agent.

Although this approach can easily train twin agents; however, at the cost of a slight loss in privacy due to the moving of devices’ data to the edge server. Therefore, a trade-off needs to be made between performance and privacy preservation.

**Intelligent Resource Management**

There are two main aspects of resource management for computing and communication resources in digital twin-based vehicular networks: Resource management for twin signaling, and resource management for ITS. Twin signaling is necessary to carry out the control signals for a twin-based system, whereas resource management is for applications. The allocation and management of computational and communication resources are complex and challenging goals in vehicular edge computing. To efficiently manage computing and wireless resources, one can efficiently use twins. A twin may use optimization theory for performing resource management. However, there are some resource management functions in the intelligent transportation system, that can not be well modeled using mathematical optimization. FL can be preferably used to tackle this issue. Twin based on FL will enable vehicular networks with efficient management of network resources by offline analysis and online control. In offline analysis, autonomous vehicles/devices train local models and shared them with the twin deployed at the edge for global aggregation to yield a global twin model. This process will occur iteratively to train a global twin model. A blockchain network can be used to store these pre-trained models for future use. The twin will retrieve the pre-trained model to serve the users via efficient resource management. The main reason for performance improvement using digital twins for vehicular networks will be due to its feature of enabling proactive intelligent analytics [3]. Such analytics will offer us proactive analysis prior to user requests. Additionally, twin-based implementation

---

**FIGURE 4** Blockchain-based FL for autonomous cars
will migrate the vehicular networks toward the self-sustaining wireless systems that are one of the key design aspects of the 6G wireless systems [12].

**Open Challenges**

**Mobility-Aware Association of Autonomous Vehicles**
How does one efficiently manage the mobility-aware association of autonomous cars for FL to learn twin models in vehicular networks? In a digital twin-based vehicular network, autonomous vehicles have high mobility, and thus may be difficult to get seamless connectivity with RSUs. On the other hand, seamless connectivity between vehicles and RSUs is required during FL. Therefore, we must effectively manage the mobility of vehicles. To address this challenge, one can use mobility management schemes based on deep learning. Such a deep learning-based mobility management scheme will predict the future locations of mobile vehicles, and thus helps associate the sets of vehicles to RSUs for FL (i.e., more specifically, groups in DFL) that can remain seamlessly connected for more time compared to other vehicles.

**Vehicular Twin Objects Migration**
How do we perform efficient migration of FL-based twin objects in high mobility vehicular networks? Twin objects serving a set of vehicles might not be able to get seamless connectivity due to the high mobility of autonomous cars. Therefore, one must migrate the twin objects deployed in the twin layer. Similar to a mobility-aware association, we can propose mobility management schemes to enable efficient and effective migration of the twin objects as per the mobility of the devices. Such a mobility management scheme can use deep learning to predict future locations of vehicles, and thus can efficiently migrate twin objects to relevant edge/cloud servers.

**Fairness-Enabled FL for Twin-Based Vehicular Networks**
How do we enable fairness-enabled FL for training twin models in high mobility vehicular networks? In vehicular networks, due to increased mobility and wireless channel degradation, needs to be made cars will experience poor performance compared to other cars. Additionally, due to locally available datasets on cars, some cars might not perform well. Such poor-performing cars will less influence the global FL model compared to better (i.e., both local model and wireless channel) performing nodes. Therefore, there will be fairness issues that need to be resolved. Fairness issues due to a wireless channel can be resolved by applying fairness-enabled association and resource allocation schemes (e.g., whale optimization-based scheme [13]). On the other hand, fairness issues due to learning algorithms can be resolved by applying fairness-based FL schemes [1].

**Personalized Twin Models**
How does one train a twin model based on FL that can effectively model particular end-user functions/behavior? Generally, training a twin model based on FL results in the general model that may not fit the particular end-device twin model. However, getting a global twin model using FL can be used for getting a personalized device FL-based twin model. Such a personalized twin model can be easily obtained with fewer efforts for further training at end-devices. To do so, one can use a Model-Agnostic Meta-Learning (MAML) [15]. The MAML-based implementation finds an initial point (i.e., initial point derived in a distributed manner) that is shared among all end-devices. Every device then performs well after updating the given model by using a few steps of a gradient-based method.

**Scalable Twin Models**
How do we enable a massive number of vehicles/vehicular devices to effectively train twin models deployed at network edge/cloud? Enabling a massive number of cars/devices in cars to participate in the training of a twin model requires a significant amount of communication resources and computation resources as well as seamless communication. Computing and communication resources are needed for local training and transferring learning model updates between devices and aggregation servers, respectively. For seamless communication, one should properly manage the mobility of cars. On the other hand, for efficient communication one must propose schemes that can effectively manage communication resources. To do so, one can use heuristic schemes. Although heuristic schemes check all possible options, and thus provide better results.
but at the cost of high computational complexity. To resolve this issue, one can use decomposition-relaxation-based schemes. However, these schemes will suffer from approximation errors. To address the limitations of the aforementioned challenges, one can use matching theory-based schemes.

Conclusions
In this article, we have presented the role of FL toward enabling digital twin-based vehicular networks. A general architecture with role of FL in enabling digital twin-based vehicular networks is also proposed. Furthermore, use cases of FL for digital twin-based vehicular network are also outlined. Finally, open challenges with their causes and possible solutions are presented. We concluded that FL is a promising candidate to enable a digital twin-based vehicular network.

Acknowledgment
This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. No. 2020R1A4A1018607), Institute of Information & communications Technology Planning & Evaluation (IITP) Grant funded by MSIT (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068, IITP grant funded by MSIT (No.2019-0-01287), Evolvable Deep Learning Model Generation Platform for Edge Computing), and IITP grant funded by MSIT (No.RS-2022-0155911, Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University). Dr. CS Hong is the corresponding author.

References
[1] L. U. Khan et al., “Digital Twin of Wireless Systems: Overview, Taxonomy, Challenges, and Opportunities,” arXiv preprint arXiv:2202.02559, 2022.
[2] L. U. Khan et al., “Metaverse for Wireless Systems: Vision, Enablers, Architecture, and Future Directions,” arXiv preprint arXiv:2207.00413, 2022.
[3] L. U. Khan et al., “Digital-Twin-Enabled 6G Vision, Architectural Trends, and Future Directions,” IEEE Commun. Mag., vol. 60, no. 1, 2022, pp. 74–80.
[4] S. Ali et al., “6G White Paper on Machine Learning in Wireless Communication Networks,” arXiv preprint arXiv:2004.11875, 2020.
[5] L. U. Khan et al., “Dispersed Federated Learning: Vision, Taxonomy, and Future Directions,” IEEE Wireless Commun., vol. 28, no. 5, 2021, pp. 192–98.
[6] B. Li et al., “Digital Twin Assisted Task Offloading for Aerial Edge Computing and Networks,” IEEE Trans. Vehicular Technology, vol. 71, no. 10, 2022, pp. 10,863–77.
[7] K. Zhang, J. Cao, and Y. Zhang, “Adaptive Digital Twin and Multisegment Deep Reinforcement Learning for Vehicular Edge Computing and Networks,” IEEE Trans. Industrial Informatics, vol. 18, no. 2, 2021, pp. 1405–13.
[8] F. Xiang et al., “Digital Twin Driven End-Face Defect Control Method for Hot-Rolled Coil With Cloud-Edge Collaboration,” IEEE Trans. Industrial Informatics, vol. 19, no. 2, 2022, pp. 1674–82.
[9] L. Zhao et al., “Elite: An Intelligent Digital Twin-Based Hierarchical Routing Scheme for Softwarized Vehicular Networks,” IEEE Trans. Mobile Computing, 2022.
[10] D. Chen et al., “Digital Twin for Federated Analytics Using a Bayesian Approach,” IEEE Internet of Things J., vol. 8, no. 22, 2021, pp. 16,301–12.
[11] M. A. Javed and S. Zeadally, “AI-Empowered Content Caching in Vehicular Edge Computing: Opportunities and Challenges,” IEEE Network, vol. 35, no. 3, 2021, pp. 109–15.
[12] W. Saad, M. Bennis, and M. Chen, “A Vision of 6g Wireless Systems: Applications, Trends, Technologies, and Open Research Problems,” IEEE Network, vol. 34, no. 3, 2019, pp. 134–42.
[13] Q.-V. Pham et al., “Whale Optimization Algorithm With Applications to Resource Allocation in Wireless Networks,” IEEE Trans. Vehicular Technology, vol. 69, no. 4, 2020, pp. 4285–97.
[14] Y. H. Ezzeldin et al., “Fairly: Enabling Group Fairness in Federated Learning,” arXiv preprint arXiv:2110.00857, 2021.
[15] A. Fallah et al., “Personalized Federated Learning: A Meta-Learning Approach,” arXiv preprint arXiv:2002.07948, 2020.

Biographies
Late U. Khan received his Ph.D. degree in Computer Engineering at Kyung Hee University (KHU), South Korea. Prior to that, he received his M.S (Electrical Engineering) degree with distinction from University of Engineering and Technology, Peshawar, Pakistan in 2017. His research interests include analytical techniques of optimization and game theory to edge computing and end-to-end network slicing.

Ehzaq Mustafa is a Ph.D. student at COMSATS University Islamabad, Abbottabad Campus, Pakistan. He is also serving as lecturer at Government College Abbottabad. His research interest include Mobile Edge Computing, Machine learning, and Wireless networks.

Junaid Shuja completed his Ph.D. from University of Malaya, Malaysia in 2017 under the BrightSpark scholarship program. He received “Graduate on Time” award from University of Malaya. He is an Assistant Professor at CUI, Abbottabad Campus, Pakistan. His research interests include application of machine learning techniques in edge computing, energy efficient cloud data centers, and mobile cloud computing. He has published research in more than 50 International journals and conferences.

Kashif Bilal received his PhD from North Dakota State University USA. He is currently working as Associate Professor at COMSATS University Islamabad, Pakistan. His research interests include cloud computing, energy efficient high speed networks, UAVs communications, and crowdsourced multimedia. Kashif was awarded CoE Student Researcher of the year 2014 based on his research contributions during his doctoral studies at North Dakota State University.

Faisal Rehman received the M.S. and Ph.D. degrees in computer science from COMSATS University Islamabad at Abbottabad, Pakistan, in 2010 and 2018, respectively. He is currently an Assistant Professor with COMSATS University Islamabad. His research interests include recommender systems, green computing, and wired and wireless networks.

Zuoy Han [S’19, M’20, SM’21] received his Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park. Currently, he is a professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. He is an AAAS fellow since 2019. He is a one-percent highly cited researcher since 2017 according to Web of Science.

Chong Seon Hong [S’95, M’97, SM’11] is working as a professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include Internet, ad hoc networks, network management, and network security. He was an Associate Editor of the IEEE Trans. Network and Service Management, J. Communications and Networks and an Associate Technical Editor of the IEEE Commun. Mag.