Abstract—Today’s wireless systems are posing key challenges in terms of quality of service and quality of physical experience. Metaverse has the potential to reshape, transform, and add innovations to the existing wireless systems. A metaverse is a collective virtual open space that can enable wireless systems using digital twins, digital avatars, and interactive experience technologies. Machine learning (ML) is indispensable for modeling twins, avatars, and deploying interactive experience technologies. In this paper, we present the role of ML in enabling metaverse-based wireless systems. We identify and discuss a set of key requirements for advancing ML in the metaverse-based wireless systems. Moreover, we present a case study of distributed split federated learning for efficiently training meta-space models. Finally, we discuss the future challenges along with potential solutions.

Index Terms—Metaverse, digital twins, avatars, machine learning, blockchain.

I. INTRODUCTION

Emerging wireless applications, such as digital healthcare, intelligent transportation systems, Industry 4.0, and smart homes, are characterized by a wide variety of diverse user-defined metrics (e.g., quality of physical experience) and traditional metrics (e.g., latency and reliability) [1]. The existing wireless systems fall short of meeting these diverse requirements [2]. A metaverse can be used to enable a wireless system to meet a variety of requirements [1]. The metaverse has the potential to effectively enable proactive online-learning and self-sustainability for wireless systems. Self-sustainability will help wireless systems operate with the least possible intervention from operators/users, whereas proactive learning will help in efficient optimization of wireless system resources for various applications (e.g., healthcare systems for lung diseases)/functions (e.g., wireless resource allocation). Proactive learning is necessary because many wireless system applications (e.g., digital healthcare) have strict latency constraints. A typical wireless system has a wide variety of players, such as edge/cloud computing resources, wireless resources, device computing resources, and core network resources. There is a need for seamless interaction among these players to enable various applications. Therefore, proactive learning prior to a user request can enable us to train machine learning (ML) models that can be used for future user requests by efficiently optimizing the application resources.

There are two main aspects, wireless for metaverse and metaverse for wireless [1], [3]. The use of wireless technologies to enable the metaverse is referred to as wireless for the metaverse, whereas metaverse for wireless is the use of the metaverse to enable wireless applications. A metaverse represents a virtual model of the real world system and comprises of digital twins (e.g., hospitals and base stations) and digital avatars (e.g., mobile users and moving autonomous cars). There is a need for effective modeling techniques to create avatars and digital twins. One can use a mathematical modeling; however, it suffers from the limitation of inaccurate results due to various assumptions, especially complete information about the environment. We can also use experimental modeling to model avatars and twins. However, it also limited of experimental equipment and human errors. To address these limitations, one can use ML for modeling (i.e., data-driven modeling) of meta space entities. Various works in literature considered metaverse [1], [3]–[5]. The work conducted in [4] presents the concept, recent advances, and open challenges. Huynh-The et al. in [5] discussed artificial intelligence (AI) towards enabling the metaverse. Specifically, the authors presented an overview of various AI techniques as well as other key enablers (e.g., computer vision and natural language processing) of the metaverse. Another work [3] discussed the fusion of AI and blockchain towards realizing a metaverse. The work in [1] presented the vision as well as the architecture of using wireless systems by metaverse. In contrast to the studies conducted in [1], [3]–[5], our work focuses on the key role, requirements, and challenges in advancing ML for the metaverse-based wireless networks. Our key contributions are as follows.

- We present an overview of the metaverse and its architecture. The high-level architecture consists of meta space and physical space, as well as interfaces for communication. We discuss the role of ML in enabling metaverse applications and functions (e.g., avatar modeling and air interface modeling).
- We identify and discuss a set of key requirements for enabling efficient and effective ML for the metaverse. Also, we present the causes and importance of the key requirements for ML-based metaverse-enabled wireless

L. U. Khan and M. Guizani are with the Department of Machine Learning, Mohamed Bin Zayed University of Artificial Intelligence, United Arab Emirates.
I. Yaqoob and K. Salah are with the Department of Electrical Engineering and Computer Science, Khalifa University, United Arab Emirates.
C. S. Hong is with the Department of Computer Science and Engineering, Kyung Hee University, South Korea.
D. Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.
Zhu Han is with the Electrical and Computer Engineering Department, University of Houston, Houston, TX 77004 USA.

Mohamed Bin Zayed University of Artificial Intelligence, United Arab Emirates.
Senior Member, IEEE
Fellow, IEEE
Senior Member, IEEE
systems. Additionally, we present a case study of distributed split federated learning for training meta space models.

- We present open challenges that exist in advancing ML for the metaverse.

II. OVERVIEW OF METAVERSE-ENABLED WIRELESS SYSTEMS

Emerging wireless systems are characterized by a wide variety of requirements for quality of physical experience (QoPE), quality of experience (QoE), and quality of service (QoS) [6]. QoS deals with the traditional network requirements (e.g., latency and reliability), whereas QoE refers to users’ experiences (personal user satisfaction). However, QoPE merges both QoS and QoE to combine human factors with physical factors [7]. To meet the aforementioned requirements, there is an immediate need for a novel design of wireless systems. Such a novel design can be based on metaverse [8]. A typical metaverse-based wireless system consists of two spaces (i.e., shown in Fig. 1): meta space and physical interaction space [1]. The physical interaction space consists of all physical devices, edge/cloud servers, and other network switches, and is responsible for actual communication and computation. On the other hand, the logical space (i.e., meta space) consists of digital twins and digital avatars. Additionally, meta space will consider interactive experience technologies (i.e., virtual reality (VR), augmented reality (AR), mixed reality (MR), and extended reality (XR)), mathematical optimization, game theory, and machine learning for performance enhancement. Note that the role of a digital avatar in the metaverse is to address the wireless system uncertainties due to mobile devices/users [1], [3], [8]. A typical digital twin can represent static entities (e.g., smart homes and base stations) of the wireless system. However, wireless systems have mobile users and devices that will significantly affect the performance. Therefore, there is a need to tackle this issue. To account for statistical uncertainties in the wireless systems due to mobile devices, there is a need for digital avatars.

To model twins and avatars, mathematical modeling can be a preferable solution that is based on various assumptions (e.g., linear assumptions in modeling of actual non-linear robotic system functions). To tackle this challenge, experimental modeling using various experiments can be used. Similar to mathematical modeling, experimental modeling also has limitations. Experimental modeling can suffer from human experimental errors and equipment errors. To address these limitations, one can prefer to use a machine learning (ML)-based model for twins and avatars. From the perspective of the metaverse for wireless systems, one can use ML mainly for various functions/applications. These functions are twins/avatar modeling, interface management, meta space deployment, intelligent media rate control, intelligent scheduling, and intelligent devices, among others. Due to the significant importance of the ML in enabling the metaverse, there is a need for proper design. We can divide the metaverse operation into two phases: online control and offline training. Offline training deals with training meta models for meta space using distributed ML that has the capability of better privacy preservation than centralized ML. In distributed ML, one can train local models at the end-devices in a physical space. After local models are trained, a global model is obtained using a blockchain network. After getting a global model, a blockchain network is used to store pre-trained models for serving future end-user requests. Such a pre-training of ML enables serving users with low latency because many applications (e.g., healthcare, industry 4.0, and intelligent transportation systems) have strict latency requirements. Also, while serving the end-users, one can further train the pre-trained models for getting more accurate results.
models. On the other hand, upon request from a user, there is a need for online control of end-devices whose overview is given in Fig. [1]. Next, we present the key requirements for efficiently and effectively realizing ML for the metaverse.

III. KEY REQUIREMENTS

A. Requirements

1) Scalability: A scalable ML scheme refers to an algorithm that can handle large datasets or data from more users in the case of distributed learning without adding significant complexity (i.e., computational for centralized ML and both computational as well as communication for distributed ML) to the system. For instance, XR (i.e., visually) along with haptics (i.e., dimensional touch using haptics gloves [9]) can be used to realize the metaverse. One can use such metaverse for various applications, such as healthcare and autonomous cars. For healthcare applications (e.g., breast cancer detection), XR and haptics gloves will use ML for object/disease recognition [10]. Scalability in terms of ML for the metaverse has two main aspects: scalable algorithms and resource optimization (i.e., computational and communication). Devices in metaverse are expected to generate an enormous amount of data that can be used in the training of ML schemes. However, considering the data of a large number of devices for ML is challenging. Typically, for large training datasets in a centralized ML, the training time increases proportionally with the dataset size, and vice versa. On the other hand, there will be a large communication overhead for enabling communication between a massive number of devices and a centralized aggregator for distributed learning. For instance, consider an intelligent resource scheduling for training and online operation as shown in Fig. [1] For such resource scheduling for large number of devices (i.e., training devices and online operation devices), there is a need for efficient and scalable ML algorithm. On the other hand, distributed learning uses on-device training. Therefore, there is a need for low-complexity learning schemes to enable efficient training for both distributed and centralized ML. One way to propose a low-complexity design for large datasets is the Neural Architecture Search (NAS) [11]. NAS automates the design of ML by enabling us to search for different architectures based on our desired strategy and choose the best one. The three main parts of NAS are search strategy, performance estimation strategy, and search space. The search space allows one to search for different architectures based on a certain search strategy. The performance strategy allows us to evaluate different architectures. Therefore, using NAS, we can propose scalable machine learning schemes for the metaverse.

2) Context-awareness: Context-awareness in an ML-enabled metaverse refers to the knowledge about IoT devices’ locations and the network. Context-awareness in metaverse is of significant importance. For instance, consider healthcare system using haptics gloves along with XR. For such applications, context-awareness can help in getting more getting accurate (i.e., using local context-aware ML model) results due to that fact that patients may have significant variations in diseases among different regions. Context-awareness can be of two types, such as local context-awareness and global context-awareness. Local context-aware applications based on ML deal with the IoT devices and their data located in close proximity, whereas global context-aware applications deal with the IoT devices and their data located in a large geographical area. Depending on the application of nature, we can choose a local context-aware ML model or a global context-aware ML model. For instance, if we want to train a keyword suggestion model for an entertainment system using metaverse, we should use the local context-aware model. Training such a model for a specialized data (i.e., single user or a set of similar users) might generally give better results than considering a generalized (global context-aware) model for many different people [12]. Similarly, if we consider the healthcare data of a particular region, we should generally choose to use the local context-aware model if the healthcare issues of the region under consideration is significantly different than other regions.

3) Noiseless Data: Smart applications based on metaverse generate an enormous amount of data that might be corrupted by noise (e.g., noise in AR sensors [13] and haptics noise [14]). For instance The main causes of noise are errors in processing, storage, and data collection. Artificial intelligence techniques are significantly degraded by the presence of noise. For instance, consider a predictive machine learning scheme. The noise in the dataset hampers the induction of the machine learning model. Furthermore, noise may prolong the convergence of the machine learning models. Therefore, it is necessary to enable clean datasets for an ML-enabled metaverse. Several different approaches can be used to minimize the impact of noise in datasets. One possible way could be to introduce a preprocessing step that is based on cleaning the data. The preprocessing step can be implemented at the data collection device. For instance, the latest smartphones have multiple microphones that are placed at different locations. The sound from the multiple microphones can be further used to yield a clean voice.

4) Forensics: An ML-enabled metaverse can be offered with a variety of security attacks. These security attacks are causative attacks (i.e., altering the training data at devices or cloud), exploratory attacks (i.e., attacks after meta space model training and exploiting misclassifications), and evasion attacks (i.e., attacks on the meta space model training process), among others. Other categorizations of attacks are white-box attacks (i.e., an attacker has complete knowledge of a meta space ML model), gray-box attacks (i.e., an attacker has some knowledge of a meta space ML model), and black-box attacks (i.e., an attacker has no knowledge of an ML model). These wide variety of attacks in ML-enabled metaverse must be handled properly. Therefore, it is necessary to properly investigate these metaverse applications based on ML for attacks. Forensic refers to investigating these kinds of attacks in a metaverse. Although the process of forensics is indispensable, designing it for a variety of applications is challenging.
B. Case Study

To train meta space models, one can use various techniques (i.e., centralized ML and FL). Although FL can offer various benefits (e.g., better privacy preservation), it has a few limitations as well. These limitations are scalability, slow convergence, and end-devices computing resource constraints. To enable FL by addressing the aforementioned limitations, we propose the use of distributed split FL (DSFL) that combines dispersed FL with split learning, and thus enables FL for resource-constrained devices with fast convergence.

1) System Model: We consider a set of resource-constrained devices (i.e., mostly devices have resource limitations) that want to train a meta space model. For training, one can preferably use federated learning (FL) because of its privacy preserving nature. However, devices need significant computing power for FL that is generally unavailable. Therefore, we consider split FL with modifications, namely, DSFL. DSFL combines both hierarchical fashion of learning with split FL (SFL), and thus can achieve faster convergence and more scalability compared to traditional SFL. Therefore, DSFL can be preferably used for training meta space models. The set of devices will be served by a set of edge servers. The resource-constrained devices train partial local models and send trained layers to nearby edge servers for training the remaining part of their local models. The local model
computing time for partial models depends on local devices’ computing power (i.e., CPU-cycles/sec) and sizes of local data sets. After computing the partial local models for devices, to send the computed partial model data to the edge server, we consider a set of resource blocks (i.e., occupied by cellular users) with a restriction that every device will not get more than one resource block. Meanwhile, every edge server has a certain maximum capacity to serve end-devices. Also, a single device can offload its task to only one edge server. We define a cost function (i.e., $(1 + \text{relative local accuracy})(0.5 \times \text{Latency}_{\text{trans}} + 0.5 \times \text{Energy}_{\text{trans}})$) that considers transmission energy, transmission latency, and relative local accuracy. Note that low values of the relative local accuracy are desirable. The relative local accuracy depends mainly on local learning model performance that is determined by various factors: (a) local data set quality (i.e., noiseless or noisy data set), (b) local iterations, and local model architecture. The relative local accuracy generally depends on number of local iterations using fixed model architecture and local dataset. The local model performance enhances with local iterations, but at the cost of energy consumption. Our goal is to jointly minimize transmission energy and transmission latency by optimizing relative local accuracy, transmission power, resource allocation, and association under various constraints. Every device must not get more than a resource block. Moreover, it must be connected to a maximum of one SBS. The association between devices and SBSs should be within maximum limit. Additionally, the assignment of resource blocks for all devices must be within limits. The formulated problem has a mixed-integer non-linear programming problem (MINLP) nature. The problem is non-convex even after relaxing the association and resource allocation variables. Therefore, we use a block successive upper-bound minimization (BSUM) technique. Also, the proposed BSUM-based solution has a sub-linear iteration complexity $O(1/i)$, where $i$ denotes the iteration. Additionally, BSUM can be readily used for solving MINLP. Therefore, the proposed BSUM-based scheme will converge within finite iterations.

2) Performance Evaluation: The number of users and edge servers are assumed to be 48 and 6, respectively. A set of 48 resource blocks are considered for communication between the devices and the edge servers. An area of $1000 \times 1000$ is considered. We consider random fixed positions for the edge servers (e.g., co-located with the base stations), whereas the positions of devices are randomly generated for each run. The resource blocks used in our model for communication between edge servers and devices are reused from cellular users. The number of sub-carriers per resource block and the carrier frequency are assumed to be 12 and 2 GHz, respectively. A free-space path loss model is considered and a thermal noise for 1 Hz at $20^\circ\text{C}$ of $-174 \ \text{dBm}$ is used in simulations. The cellular users transmit power is 46 dBm. The devices involved in learning will incur interference from the cellular users. However, there will be no interference between the devices involved in learning. Two baselines: baseline-A and baseline-R are considered. Baseline-A considers random resource allocation with BSUM-based power allocation, association, and relative local accuracy minimization. Baseline-R considers random association and BSUM-based resource, power allocation, and relative local accuracy minimization. Fig. 4 shows the cost function for various values of relative local accuracy and signal-to-interference-plus-noise ratio (SINR). It is clear from Fig. 4 that cost is lowest for least value of relative local accuracy and highest SINR. However, achieving the least value of relative local accuracy will be at the cost of local computing resources. Performance of DSFL in terms of cost vs. iterations is shown in Fig. 4. The proposed scheme outperforms other baselines due to the fact that it jointly considers relative local accuracy minimization, transmit power allocation, resource allocation. Among the baselines, baseline-A outperforms baseline-R because, for the assumed cost function, association with power allocation and relative local accuracy minimization has a more prominent effect on the cost for DSFL compared to resource allocation with power allocation and relative local accuracy minimization. In Fig. 4, we study accuracy vs. communication round for DSFL and SFL. We use a non-IID distribution of data based on sorting the images in the MNIST dataset. Next to sorting, the entire data set is divided into 200 shards, each shard of 300 images. For non-IID, every device is given 1 shard. It is clear from Fig. 4 that DSFL outperforms SFL for all cases (i.e., different number of edge aggregations) and achieves faster convergence.
IV. CHALLENGES IN ADVANCING ML FOR METAVERSE

A. Training Fashion

How do we efficiently train (i.e., centralized training or distributed training) ML models for various emerging applications/functions? Recently, we observed various ML models based on both centralized (e.g., Deep Reinforcement Learning (DRL) agents deployed at a centralized cloud/edge for making caching decisions) and distributed training (e.g., DRL agents deployed at end-devices to perform distributed resource management). Fig. 2 shows training for both centralized and distributed ML models for the metaverse. Additionally, the pros and cons of both schemes are shown in Fig. 2. Meta space based on centralized ML (i.e., deployed at the edge or cloud) has low management complexity compared to meta space based on distributed ML (i.e., deployed at multiple edge servers/nodes (e.g., UAVs)). The reason for this is the fact that multiple entities implementing meta space for a certain function/application requires more signaling overhead for management compared to a centralized ML-based implementation of meta space. However, this ease in management comes at the cost of a loss in scalability, which is one of the key design requirements of metaverse-based wireless systems. Although distributed ML-based can offer benefits, it has a few challenges: inaccurate estimation in highly dynamic scenarios, data heterogeneity, and slow training convergence. Also, for distributed learning schemes, the learning model might not be very accurate for highly dynamic scenarios. The reason can be the dynamics of the local learning models due to changes in their sensory data. There might be significant changes in the local model due to the highly dynamic changes in the environment that may cause the global model to converge slowly as well as provide less accurate results. However, if the data were made available at a remote cloud for training a centralized model, the model could adapt to dynamic changes in the environment. However, migrating the devices’ data to the centralized local for training is a challenge and will require a significant amount of communication resources. Therefore, we must make a tradeoff between the meta model accuracy and communication overhead while choosing between centralized and distributed learning.

B. Standardization

How does one propose an efficient standard for ML-enabled metaverse for wireless systems? Existing standards (e.g., ISO/IEC 23005 and IEEE 2888) of the metaverse mainly focus on interfaces for seamless connectivity between the real world and the virtual world [8]. ISO/IEC 23005 standards enable various metaverse business services. In these services, the virtual objects (e.g., virtual items and avatars) characteristics, association of rendered sensory effects, and audiovisual information leverages interactions between real worlds and virtual worlds. Specifically, ISO/IEC 23005 standards focus on sensory effects. On the other hand, IEEE 2888 standards enable the foundations of metaverse systems. IEEE 2888.1 and IEEE 2888.2 standards are used for exchange of actuator-related information and sensory information between the physical and virtual worlds, respectively. The ISO/IEC 23005 standards lack general-purpose interfaces for communication between the virtual and physical worlds to enable various emerging applications. Furthermore, while both the ISO/IEC 23005 and IEEE 2888 standards can provide numerous benefits, they do not explicitly address the role of machine learning in enabling the metaverse.

One can use ML to effectively enable metaverse-based wireless systems. For instance, modeling of meta objects for a particular application/function can be performed using supervised learning-based schemes, reinforcement learning-based schemes, and transfer learning-based schemes. For power control in an access network, one can use Q-learning. Therefore, it is clear that ML is inevitable to enable metaverse-based wireless systems. However, the existing architecture proposed by many metaverse works may not provide enough flexibility to support ML effectively. Therefore, there is a need to propose novel standardization schemes for the practical implementation of ML-based metaverse, as shown in Fig. 3. Fig. 3 shows the general overview of using ML for enabling the metaverse. There is a need for an ML controller that will use ML to control the activities in the physical space, meta space, and interfaces. Note that existing standards should be used in addition to existing standards, such as IEEE 288.1, IEEE 288.2, and ISO/IEC 23005. There is a need to propose a novel standard for data homogenization in a metaverse. The need for data homogeneity arises due to the fact that different data-generating sources in the metaverse will have different forms that need to be transformed into a general form acceptable by the meta space [1]. This approach will enable a general design of meta space for various applications and will also enable a less time-consuming solution. Additionally, there is a need to standardize the interfaces, such as interface A, interface B, and interface C, as shown in Fig. 3. The interfaces (i.e., interfaces A and B) between the ML controller and physical space must be real-time, whereas the interface between the ML controller and meta space can be real-time or non-real-time. For offline training of meta models in meta space, the interface C can be non-real-time, whereas the management of meta objects can be controlled by a real-time interface C. Due to the important role of ML in enabling the metaverse (as shown in Fig. 3), there is a need to propose novel standards.
C. Blockchain for Secure ML-Enabled Metaverse-based Wireless Systems

How does one use blockchain to enable a secure and immutable ML-based metaverse? A typical metaverse has a variety of data sources, such as (a) data from the physical space devices that can be used for training; (b) pre-trained meta models; and (c) metaverse signaling data. To handle such data in a transparent and immutable manner, one can use a blockchain. The role of blockchain in enabling an ML-based metaverse is shown in Fig. 5. The purpose of data homogenization and dehomogenization is to enable a common meta space for various applications to minimize the implementation complexity. Various applications generate data in different formats. Therefore, there is a need for data homogenization based on ML schemes to transform the sensory data to a single form. Additionally, there is a need to perform some operations to recover the missing data as well in homogenization. The data fusion using ML will provide a collection of data from various sources to enable consistent interpretation of a certain process [15]. For instance, consider the production of autonomous vehicles. Such a process will require data from various sources (e.g., traffic accident data, traffic congestion control data, and real-time data about the types of vehicles) before starting design and production. To handle such data in a transparent and immutable manner, one can use blockchain. In connection to the aforementioned role of blockchain in enabling ML for the metaverse, there is a need for an efficient design of blockchain. Note that a blockchain has privacy concerns as well due to its distributed design. Every blockchain node has access to the data, and thus there might be a privacy leakage issue in the presence of a malicious user. Typically, a blockchain consensus algorithm (e.g., PoW) consumes a significant amount of energy and latency. For instance, some of the blockchain consensus mechanisms are aimed at low energy consumption (e.g., delegated proof of stake), whereas some are for low latency (e.g., Byzantine fault tolerance). Therefore, there is a need to design a novel blockchain consensus algorithm that can offer a tradeoff between latency and energy consumption.

V. Conclusions

In this paper, we have discussed the role of ML in enabling metaverse-based wireless systems. The role and a set of key requirements for ML towards enabling metaverse-based wireless systems are presented. We outlined and discussed in detail the key challenges that exist in advancing ML for metaverse-based wireless systems. Furthermore, causes and possible solutions to these challenges are presented. We conclude that ML will play a key role in enabling metaverse-based wireless systems.

REFERENCES

[1] L. U. Khan, Z. Han, D. Niyato, E. Hossain, and C. S. Hong, “Metaverse for wireless systems: Vision, enablers, architecture, and future directions,” arXiv preprint arXiv:2007.00413, 2022.
[2] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong, “Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities,” arXiv preprint arXiv:2202.02559, 2022.
[3] Q. Ynag, Y. Zhao, H. Huang, and Z. Zheng, “Fusing blockchain and ai with metaverse: A survey,” arXiv preprint arXiv:2201.03201, 2022.
[4] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, “A survey on metaverse: the state-of-the-art, technologies, applications, and challenges,” arXiv preprint arXiv:2111.09673, 2021.
[5] T. Huynh-The, Q.-V. Pham, X.-Q. Pham, T. T. Nguyen, Z. Han, and D.-S. Kim, “Artificial intelligence for the metaverse: A survey,” arXiv preprint arXiv:2202.10336, 2022.
[6] N. Saxena, E. Rastogi, and A. Rastogi, “6g use cases, requirements, and metrics,” in 6G Mobile Wireless Networks. Springer, 2021, pp. 7–24.
[7] 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, pp. 134–142, 2020.
[8] Y. Wang, Z. Su, N. Zhang, D. Liu, R. Xing, T. H. Luan, and X. Shen, “A survey on metaverse: Fundamentals, security, and privacy,” IEEE Communications Surveys & Tutorials, 2022.
[9] W. Y. B. Lim, Z. Xiong, D. Niyato, X. Cao, C. Miao, S. Sun, and Q. Yang, “Realizing the metaverse with edge intelligence: A match made in heaven,” arXiv preprint arXiv:2201.01634, 2022.
[10] M. Zhu, Z. Sun, Z. Zhang, Q. Shi, T. He, H. Liu, T. Chen, and C. Lee, “Haptic-feedback smart glove as a creative human-machine interface (hmi) for virtual/augmented reality applications,” Science Advances, vol. 6, no. 19, p. eaaz8693, 2020.
[11] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for internet of things: Recent advances, taxonomy, and open challenges,” IEEE Communications Surveys & Tutorials, 2021.
[12] L. U. Khan, W. Saad, Z. Han, and C. S. Hong, “Dispersed federated learning: Vision, taxonomy, and future directions,” IEEE Wireless Communications, vol. 28, no. 3, pp. 192–198, 2021.
[13] F. Z. Eishita and K. G. Stanley, “The impact of sensor noise on player experience in magic window augmented reality aiming games,” in International Conference on Entertainment Computing. Springer, 2015, pp. 502–507.
[14] J. M. Cruz-Hernandez, V. Levesque, J. Saboune, D. Grant, R. Lacroix, and A. Hamam, “Haptic feedback in a haptically noisy environment,” May 23 2017, uS Patent 9,659,468.
[15] S. Suhail, R. Hussain, R. Jurdak, A. Oracevic, K. Salah, C. S. Hong, and R. Matulevičius, “Blockchain-based digital twins: research trends, issues, and future challenges,” ACM Computing Surveys (CSUR), vol. 54, no. 11s, pp. 1–34, 2022.