Federated Echo State Learning for Minimizing Breaks in Presence in Wireless Virtual Reality Networks

Mingzhe Chen, Omid Semiari, *Member, IEEE*, Walid Saad, *Fellow, IEEE*, Xuanlin Liu, Changchuan Yin, *Senior Member, IEEE*,

***Abstract*—In this paper, the problem of enhancing the virtual reality (VR) experience for wireless users is investigated by minimizing the occurrence of breaks in presence (BIP) that can detach the users from their virtual world. To measure the BIP for wireless VR users, a novel model that jointly considers the VR application type, transmission delay, VR video quality, and users' awareness of the virtual environment is proposed. In the developed model, the base stations (BSs) transmit VR videos to the wireless VR users using directional transmission links so as to provide high data rates for the VR users, thus, reducing the number of BIP for each user. Since the body movements of a VR user may result in a blockage of its wireless link, the location and orientation of VR users must also be considered when minimizing BIP. The BIP minimization problem is formulated as an optimization problem which jointly considers the predictions of users' locations, orientations, and their BS association. To predict the orientation and locations of VR users, a distributed learning algorithm based on the machine learning framework of deep (ESNs) is proposed. The proposed algorithm uses concept from *federated learning* to enable multiple BSs to locally train their deep ESNs using their collected data and cooperatively build a learning model to predict the entire users' locations and orientations. Using these predictions, the user association policy that minimizes BIP is derived. Simulation results demonstrate that the developed algorithm reduces the users' BIP by up to 16% and 26%, respectively, compared to centralized ESN and deep learning algorithms.**

I. Introduction

Deploying virtual reality (VR) applications over wireless networks is an essential stepping stone towards flexible deployment of pervasive VR applications [2]. However, to enable a seamless and immersive wireless VR experience, it is necessary to introduce novel wireless networking solutions that can meet stringent quality-of-service (QoS) requirements of VR applications [3]. In wireless VR, any sudden drops in the data rate or increase in the delay can negatively impact the users' VR experience (e.g., due to interruptions in VR video streams). Due to such an interruption in the virtual world, VR users will experience *breaks in presence (BIP)* events that can be detrimental to their immersive VR experience. While the fifth-generation (5G) new radio supports operation at high frequency bands as well as flexible frame structure to minimize latency, the performance of communication links at high frequencies is highly prone to blockage. That is, if an object blocks the wireless link between the BS and a VR user, the data rate can drop significantly and lead to a BIP. In addition to wireless factors such as delay and data rate, behavioral metrics related to each VR user such as the user's *awareness* can also induce BIP. Awareness is defined as each wireless VR user's perceptions and actions in its individual VR environment. Therefore, to minimize the BIP of VR users, it is necessary to jointly consider all of the wireless environment and user-specific metrics that cause BIP, such as link blockage, user location, user orientation, user association, and user awareness.

Recently, several works have studied a number of problems related to wireless VR networks [4]–[11]. The work in [4] developed a multipath cooperative route scheme to enable VR wireless transmissions. In [5], the authors develop a framework for mobile VR delivery by leveraging the caching and computing capabilities of mobile VR devices. The authors in [6] study the problem of supporting visual and haptic perceptions over wireless cellular networks. A communications-constrained mobile edge computing framework is proposed in [7] to reduce wireless resource consumption. The work in [8] proposes a concrete measure for the delay perception of VR users. The authors in [9] present a scheme of proactive computing and high-frequency, millimeter wave (mmWave) [12] transmission for wireless VR networks. In [10], the authors design several experiments for quantifying the performance of tile-based 360° video streaming over a real cellular network. Our previous work in [11] studied the problems of resource allocation and 360° content transmission. However, most of these existing works do not provide a comprehensive BIP model that accounts for the transmission delay, the quality of VR videos, VR application type, and user awareness. Moreover, the prior art in [4]–[11] does not jointly consider the impact of the users' body movements when using mmWave communications.

To address this challenge, machine learning techniques can be used to predict the users' movements and proactively determine the user associations that can minimize BIP. However, in prior works on machine learning for user movement

M. Chen, X. Liu, and C. Yin are with the Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing, 100876, China, Emails: xuanlin.liu@bupt.edu.cn, ccyin@ieee.org.

M. Chen is also with the Department of Electrical Engineering, Princeton University, Princeton, NJ, 08544, USA and with the Chinese University of Hong Kong, Shenzhen, 518172, China,, Email: mingzhec@princeton.edu.

O. Semiari is with the Department of Electrical and Computer Engineering, University of Colorado Colorado Springs, Colorado Springs, CO, 80918, USA, Email: osemiari@uccs.edu.

W. Saad is with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24060, USA, Email: walids@vt.edu.

This work was supported in part by the National Natural Science Foundation of China under Grant 61629101, in part by the Beijing Natural Science Foundation and Municipal Education Committee Joint Funding Project under Grant KZ201911232046, in part by the 111 Project under Grant B17007, and in part by the U.S. National Science Foundation under Grant CNS-1836802 and Grant CNS-1941348.

A preliminary version of this work was published in the IEEE GLOBECOM conference [1].

predictions [13]–[17], the data for each user's movement must be collected by its associated BS. However, in real mobile VR scenarios, users will move and change their association and the data related to their movement is dispersed across multiple BSs. In such scenarios, the BSs may not be able to continuously share collected user data among each other, due to the high overhead of data transmission. Moreover, sending all the information to a centralized processing server will cause very large delays that cannot be tolerated by VR applications. Thus, centralized machine learning algorithms such as in [13]–[17] will not be useful to predict real-time movements of the VR users. To this end, *a distributed learning framework that can be trained by the collected data at each BS is needed.*

Recently, a number of existing works such as in [18]–[21] studied important problems related to the implementation of distributed learning over wireless networks. While interesting, these prior works [18]–[21] that focus on the optimization of the performance of distributed learning algorithms such as federated learning do not consider the use of distributed learning to optimize the performance of wireless networks. In particular, these existing works [18]–[21] do not consider the use of distributed learning algorithms to predict users' orientations and locations to reduce the BIP of wireless VR users. Note that, in [1], we have studied the use of a single-layer echo state network (ESN) model with federated learning for orientation and location predictions. However, the federated learning algorithm of [1] cannot be used to analyze a large dataset. Meanwhile, the work in [1] does not analyze the prediction accuracy or memory capacity of the introduced learning algorithm.

The key contribution of this work is to develop a novel framework for minimizing BIP within VR applications that operate over wireless networks. To our best knowledge, *this paper is the first to analyze how a wireless network with distributed learning can minimize BIP for VR users and enhance their virtual world experience*. The key contributions therefore include:

- For wireless VR users, we mathematically model a new BIP metric that jointly considers VR application type, the delay of VR video and tracking information transmission, VR video quality, and the users' awareness.
- To minimize the BIP of wireless VR users, we develop a federated ESN [22] learning algorithm that enables BSs to locally train their machine learning algorithms using the data collected from the users' locations and orientations. Then, the BSs can cooperatively build a learning model by sharing their trained models to predict the users' locations and orientations. Based on these predictions, we perform fundamental analysis to find an efficient user association for each VR user that minimizes the BIP.
- To analyze the prediction accuracy of the federated ESN learning algorithm, we study the memory capacity of federated ESNs. The memory capacity characterizes the ability of the ESN model to record historical locations and orientations of each VR user. As the memory capacity increases, the prediction accuracy will improve. Since the BSs determine the user association based on these predictions, better prediction accuracy will lead to a more effective user association scheme that will minimize the number of BIP. The analytical results show that the memory capacity of ESNs depends on the number of neurons in each ESN model and the values of matrices that are used to generate the ESN model.
- Simulation results demonstrate that our proposed algorithm can achieve significant improvements in the statistics of BIP that occur within a wireless VR network.

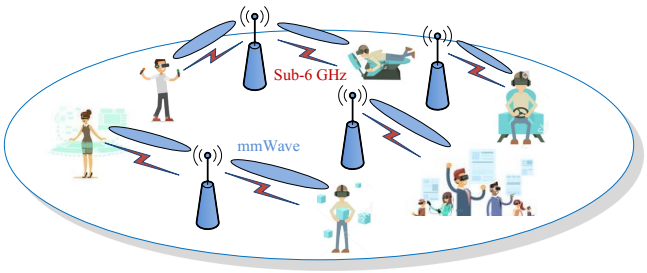


Fig. 1. The architecture of a wireless VR network. In this architecture, the Sub-6 GHz uplink is used to transmit tracking information and the mmWave downlink is used to transmit VR videos.

The rest of this paper is organized as follows. The problem formulation is presented in Section II. The federated ESN learning algorithm for the predictions is proposed in Section III. In Section IV, the memory capacity of various ESN models are analyzed. The user association is found in Section V. In Section VI, simulation results are presented and conclusions are drawn in Section VII.

II. System Model and Problem Formulation

Consider a cellular network that consists of a set $\mathcal{B}$ of B BSs that service a set $\mathcal{U}$ of U VR users. In this model, BSs act as VR controllers that can collect the tracking information related to the users' movements via VR sensors and use the collected data to generate the VR videos for their associated users, as shown in Fig. 1. In particular, the uplink is used to transmit tracking information such as users' locations and orientations from the VR devices to the BSs, while the downlink is used to transmit VR videos from BSs to VR users. For user association, the VR users can associate with different BSs for uplink and downlink data transmissions. Different from prior works such as in [3], [5]–[11] that assume the VR users to be static, we consider a practical scenario in which the locations and orientations of the VR users will impact the VR application performance.

A. Transmission Model

We consider both uplink and downlink transmission links between BSs and VR users. The VR users can operate at both mmWave and sub-6 GHz frequencies [23]–[25]. The VR videos are transmitted from BSs to VR users over the 28 GHz band. Meanwhile, the tracking information is transmitted from VR devices to their associated BSs over a sub-6 GHz frequency band. This is due to the fact that sub-6 GHz frequencies with limited bandwidth cannot support the large

data rates required for VR video transmissions. However, it can provide reliable communications for sending small data sized users' tracking information.

1) Uplink Transmissions of User Tracking Information: Let (x_{it}, y_{it}) be the Cartesian coordinates for the location of user i at time t and S be the data size of each user's tracking information, including location and orientation. S depends on the VR system (i.e., HTC Vive [26] or Oculus [27]). The data rate for transmitting the tracking information from VR user i to BS j is given by:

$$c_{ij}^{\mathrm{UL}}\left(x_{it}, y_{it}\right) = F^{\mathrm{UL}}\log_2\left(1+\frac{P_u g_{ij} d_{ij}^{-\beta}\left(x_{it}, y_{it}\right)}{\sum\limits_{k\in\mathcal{U}_i} P_u g_{kj} d_{kj}^{-\beta}(x_{kt}, y_{kt})+\rho^2}\right), \tag{1}$$

where F^{UL} is the bandwidth of each subcarrier, U_j^{UL} is the number of VR users associated with BS j over uplink, $\mathcal{U}_i$ is the set of VR users that use the same subcarriers with user i, P_u is the transmit power of each VR user (assumed equal for all users), g_{ij} is the Rayleigh channel gain, d_{ij} is the distance between VR user i and BS j, and ρ^2 is the noise power.

2) Downlink VR Video Transmission: In the downlink, antenna arrays are deployed at BSs to perform directional beamforming over the mmWave frequency band. For simplicity, a sectored antenna model [28] is used to approximate the actual array beam patterns. This simplified antenna model consists of four parameters: the half-power beamwidth ϕ, the boresight direction θ, the antenna gain of the mainlobe Q, and the antenna gain of the sidelobe q. Let φ_{ij} be the phase from BS j to VR user i. The antenna gain of the transmission link from BS j to user i is:

$$G_{ij} = \begin{cases} Q, \text{ if } |\varphi_{ij} - \theta_j| \leqslant \frac{\phi}{2}, \\ q, \text{ if } |\varphi_{ij} - \theta_j| > \frac{\phi}{2}. \end{cases} \tag{2}$$

Since the VR device is located in front of the VR user's head, the mmWave link will be blocked, if the user rotates. Let χ_{it} be the orientation of user i at time t and ϑ be the maximum angle using which BS j can directly transmit VR videos to a user without any human body blockage. ϕ'_{ij} denotes the phase from user i to BS j. For user i, the blockage effect, $b_i\left(\chi_{it}\right)$, caused by its own body can be given by:

$$b_i\left(\chi_{it}\right) = \begin{cases} 1, \text{ if } \left|\varphi'_{ij} - \chi_{it}\right| \leqslant \vartheta, \\ 0, \text{ if } \left|\varphi'_{ij} - \chi_{it}\right| > \vartheta. \end{cases} \tag{3}$$

We assume that each VR user's body constitutes a single blockage area and n_{ijt} represents the number of VR users located between user i and BS j at time t. If there are no users located between user i and BS j that block the mmWave link, i.e., $(b_i\left(\chi_{it}\right) + n_{ij} = 0$, then, as shown in Fig. 2(a)), the communication link between user i and BS j is line-of-sight (LoS). If the mmWave link between user i and BS j is blocked by the user i's own body (as shown in Fig. 2(b), $b_i\left(\chi_{it}\right) = 1$) or blocked by other users located between user i and BS j (as shown in Fig. 2(c), $+n_{ij} > 0$), then the communication link between user i and BS j is said to be non-line-of-sight (NLoS). From (3), we can see that $b_i\left(\chi_{it}\right)$ and n_{ij} can be directly determined by the users' orientations and locations.

Considering path loss and shadowing effects, the path loss for a LoS link and a NLoS link between VR user i and BS j in dB will be given by [28]:

$$\begin{aligned} h_{ij}^{\mathrm{LoS}}\left(x_{it}, y_{it}\right) =& 10\varpi_{\mathrm{LoS}}\log\left(d_{ij}\left(x_{it}, y_{it}\right)\right) \\ &+ 20\log\left(\frac{d^0 f_c 4\pi}{\nu}\right) + \mu_{\sigma_{\mathrm{LoS}}}, \end{aligned} \tag{4}$$

$$\begin{aligned} h_{ij}^{\mathrm{NLoS}}\left(x_{it}, y_{it}\right) =& 10\varpi_{\mathrm{NLoS}}\log\left(d_{ij}\left(x_{it}, y_{it}\right)\right) \\ &+ 20\log\left(\frac{d^0 f_c 4\pi}{\nu}\right) + \mu_{\sigma_{\mathrm{NLoS}}}, \end{aligned} \tag{5}$$

where $20\log\left(\frac{d_0 f_c 4\pi}{\nu}\right)$ is the free space path loss. Here, d^0 represents the reference distance, f_c is the carrier frequency and ν is the light speed. ϖ_{LoS} and ϖ_{NLoS} represent the path loss exponents for the LoS and NLoS links, respectively. $\mu_{\sigma_{\mathrm{LoS}}}$ and $\mu_{\sigma_{\mathrm{NLoS}}}$ represent Gaussian random variables with zero mean, respectively. σ_{LoS} and σ_{NLoS} represent the standard deviations for LoS and NLoS links in dB, respectively. The downlink data rate of VR video transmission from BS j to user i is given by:

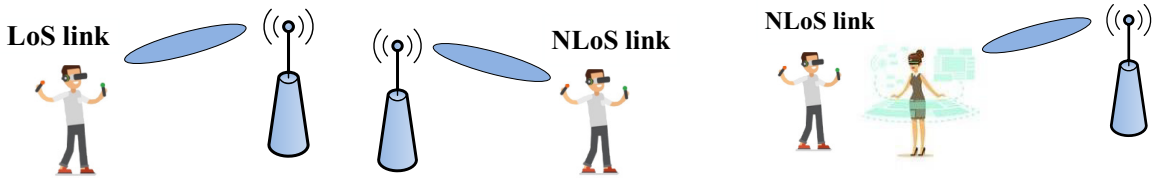
$$\begin{aligned} &c_{ij}^{\mathrm{DL}}\left(x_{it}, y_{it}, b_i\left(\chi_{it}\right), n_{ij}\right) \\ &= \begin{cases} F^{\mathrm{DL}}\log_2\left(1+\frac{P_{\mathrm{B}} G_{ij}}{10^{h_{ij}^{\mathrm{LoS}}/10}\rho^2}\right), \text{if } b_i\left(\chi_{it}\right) + n_{ij} = 0, \\ F^{\mathrm{DL}}\log_2\left(1+\frac{P_{\mathrm{B}} G_{ij}}{10^{h_{ij}^{\mathrm{NLoS}}/10}\rho^2}\right), \text{if } b_i\left(\chi_{it}\right) + n_{ij} > 0, \end{cases} \end{aligned} \tag{6}$$

where F^{DL} is the bandwidth allocated to each user and P_{B} is the transmit power of each BS j which is assumed to be equal for all BSs. Since the downlink uses mmWave links, we assume that, due to directivity, interference in (6) can be neglected, as done in [29].

B. Break in Presence Model

In a VR application, the notion of a BIP represents an event that leads the VR users to realize that they are in a fictitious, virtual environment, thus ruining their immersive experience. In other words, a BIP event transitions a user from the immersive virtual world to the real world [30]. For wired VR, BIP can be caused by various factors such as hitting the walls/ceiling, loss of tracking with the device, tripping on wire cords, or talking to another person from the real world [30]. For wireless VR, BIP can be also caused by the delay of VR video and tracking information transmission, the quality of the VR videos received by the VR users, and inaccurate tracking information received by BSs.

To model such BIP, we jointly consider the delay of VR video and tracking information transmission and the quality of the VR videos. We first define a vector $\boldsymbol{l}_{i,t}\left(c_{ij}^{\mathrm{DL}}\left(x_{it}, y_{it}, b_i\left(\chi_{it}\right), n_{ij}\right)\right) = [l_{i1,t}, \ldots, l_{iN_L,t}]$ that represents a VR video that user i received at time t with $l_{ik,t} \in \{0, 1\}$. $l_{ik,t} = 0$ indicates that pixel k is not successfully received by user i, and $l_{ik,t} = 1$, otherwise. We also define a vector $\boldsymbol{m}_{i,t}\left(G_A\right) = [m_{i1,t}, \ldots, m_{iN_L,t}]^{\mathrm{T}}$ that represents



(a) LoS links. (b) NLoS links caused by the user's own body. (c) NLoS links caused by other user's body.

Fig. 2. VR video transmission over LoS/NLoS links

the weight of the importance of each pixel constructing a VR video, where $m_{ik,t} \in [0,1]$ and G_A represents a VR application such as an immersive VR game or a VR video. $m_{ik,t} = 1$ indicates that pixel k is one of the most important elements for the generation of G_A. The value of $m_{ik,t}$ depends on the compression used for the VR video. In each VR application G_A, a number of pixels can be compressed at the BS and recovered by the user and, hence, these pixels are not important. However, the pixels that cannot be compressed by the BS are important and must be transmitted to the VR users. Therefore, each pixel will have different importance and $m_{ik,t} \in [0,1]$. Then, the BIP of VR user i caused by the wireless transmission will be given by:

$$\omega_{it}\left(x_{it}, y_{it}, \chi_{it}, \boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}\right) = \mathbb{1}_{\left\{\frac{A}{a_{ij,t}^{\mathrm{UL}} c_{ij}^{\mathrm{UL}}(x_{it}, y_{it})} + \frac{D(\boldsymbol{l}_{i,t})}{a_{ik,t}^{\mathrm{DL}} c_{ik}^{\mathrm{DL}}} > \gamma_{\mathrm{D}} \ \vee \ \boldsymbol{l}_{i,t}\boldsymbol{m}_{i,t}(G_A) < \gamma_{\mathrm{Q}}\right\}}. \quad (7)$$

where c_{ik}^{DL} and $\boldsymbol{l}_{i,t}$ are short for $c_{ik}^{\mathrm{DL}}\left(x_{it}, y_{it}, b_i\left(\chi_{it}\right), n_{ik}\right)$ and $\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}} c_{ik}^{\mathrm{DL}}\left(x_{it}, y_{it}, b_i\left(\chi_{it}\right), n_{ik}\right)\right)$, respectively, $\mathbb{1}_{\{x\}} = 1$ if x is true, and otherwise, we have $\mathbb{1}_{\{x\}} = 0$. $\mathbb{1}_{\{x\}} \vee \mathbb{1}_{\{y\}} = 1$ as y or x is true, $\mathbb{1}_{\{x\}} \vee \mathbb{1}_{\{y\}} = 0$, otherwise. $\boldsymbol{a}_{i,t}^{\mathrm{UL}} = \left[a_{i1,t}^{\mathrm{UL}}, \ldots, a_{iB,t}^{\mathrm{UL}}\right]$ is a vector that represents user i's uplink association with $a_{ik,t}^{\mathrm{UL}} \in \{0,1\}$ and $\sum_{k\in\mathcal{B}} a_{ik,t}^{\mathrm{UL}} = 1$. Similarly, $\boldsymbol{a}_{i,t}^{\mathrm{DL}} = \left[a_{i1,t}^{\mathrm{DL}}, \ldots, a_{iB,t}^{\mathrm{DL}}\right]$ is a vector that represents user i's downlink association with $a_{ik,t}^{\mathrm{DL}} \in \{0,1\}$ and $\sum_{k\in\mathcal{B}} a_{ik,t}^{\mathrm{DL}} = 1$. γ_D and γ_Q represent the target delay and video quality requirements, respectively. In (7), A represents the data size of the tracking information, $\frac{A}{c_{ij}^{\mathrm{UL}}(x_{it}, y_{it})}$ represents the time used for tracking information transmission from user i to BS j, $D\left(\boldsymbol{l}_{i,t}\left(c_{ik}^{\mathrm{DL}}\left(x_{it}, y_{it}, b_i\left(\chi_{it}\right), n_{ik}\right)\right)\right)$ represents the data size of a VR video, and $\frac{D\left(\boldsymbol{l}_{i,t}\left(c_{ik}^{\mathrm{DL}}(x_{it}, y_{it}, b_i(\chi_{it}), n_{ik})\right)\right)}{c_{ik}^{\mathrm{DL}}(x_{it}, y_{it}, b_i(\chi_{it}), n_{ik})}$ represents the transmission latency for sending the tracking information from BS k to user i. For simplicity, hereinafter, ω_{it} is referred as $\omega_{it}\left(x_{it}, y_{it}, \chi_{it}, \boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}\right)$. (7) shows that if the delay of VR video and tracking information transmission exceeds the target delay threshold allowed by VR systems or the quality of the VR video cannot meet the video requirement, users will experience a BIP (ω_{it}=1). From (7), we can also see that, the BIP of user i caused by wireless transmission depends on user i's location, orientation, VR applications, and user association. (7) captures the BIP caused by wireless networking factors such as transmission delay and video quality. Next, we define a BIP model that jointly considers wireless transmission, the VR application type, and the users' awareness. The BIP of user i can be given by [31]:

$$P_i\left(x_{it}, y_{it}, G_A, \chi_{it}, \boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}\right) = \frac{1}{T}\sum_{t=1}^{T}\left(G_A + \omega_{it} + G_A\omega_{it} + \epsilon_i + \epsilon_{G_A|i} + \epsilon_B\right), \quad (8)$$

where ϵ_i is user i's awareness, $\epsilon_{G_A|i}$ is the joint effect caused by user i's awareness and VR application G_A, and ϵ_B is a random effect. ϵ_i, $\epsilon_{G_A|i}$, and ϵ_B follow a Gaussian distribution [31] with zero mean and variances σ_i^2, $\sigma_{G_A|i}^2$, and σ_B^2, respectively. In (8), the value of $P_i\left(x_{it}, y_{it}, G_A, \chi_{it}, \boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}\right)$ quantifies the average number of BIP that user i can identify during a period. From (8), we can see that, as the VR application for user i changes, the BIP value will change. For example, a given user watching VR videos will experience fewer BIP compared to a user engaged in an immersive first-person shooting game. This is due to the fact that in an immersive game environment, users are fully engaged with the virtual environment, as opposed to some VR applications that require the user to only watch VR videos. In (8), we can also see that the BIP depend on the users' awareness. This means that different users will have different actions and perceptions when they interact with the virtual environment and, hence, different VR users may experience different levels of BIP.

C. Problem Formulation

From (8), we can see that the BIP of each user depends on this user's location, orientation, and selected BSs. By using an effective learning algorithm to predict the users' locations and orientations, the BSs can proactively determine the users' association to improve the downlink and uplink data rates and minimize BIP for each VR user. The BIP minimization problem is:

$$\min_{\boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}} \sum_{i\in\mathcal{U}} P_i\left(\hat{x}_{it}, \hat{y}_{it}, G_A, \hat{\chi}_{it}, \boldsymbol{a}_{i,t}^{\mathrm{UL}}, \boldsymbol{a}_{i,t}^{\mathrm{DL}}\right) \quad (9)$$

$$\text{s. t.} \quad U_j \leqslant V, \quad \forall j \in \mathcal{B}, \quad (9a)$$

$$a_{ij,t}^{\mathrm{UL}}, a_{ij,t}^{\mathrm{DL}} \in \{0,1\}, \quad \forall i \in \mathcal{U}, \forall j \in \mathcal{B}, \quad (9b)$$

$$\sum_{j\in\mathcal{B}} a_{ij,t}^{\mathrm{UL}} = 1, \sum_{j\in\mathcal{B}} a_{ij,t}^{\mathrm{DL}} = 1, \quad \forall i \in \mathcal{U}, \quad (9c)$$

where $\hat{x}_{it}$, $\hat{y}_{it}$, and $\hat{\chi}_{it}$ are the predicted locations and orientation of user i at time t, which depend on the actual historical locations and orientation of user i. U_j is the number of VR users associated with BS j over downlink and V is the maximum number of users that can be associated with each

BS. (9b) and (9c) show that each user can associate with only one uplink BS and one downlink BS. From (9), we can see that the BIP of each user will depend on the user association as well as the users' locations and orientations. Meanwhile, the user association depends on the locations and orientations of the VR users. If the BSs perform the user association without knowledge of the locations and orientations of the users, the body blockage between the user-BS transmission links can potentially be significant, thus increasing the BIP of each user. Therefore, the BSs must use historical information related to the users' locations and orientations to determine the user association. As the users' locations and orientations will continuously change as time elapses, BSs must proactively determine the user association to reduce the BIP of VR users. In consequence, it is necessary to introduce a machine learning algorithm to predict the users' locations and orientations in order to determine the user association and minimize BIP of VR users. In the model defined in Section II, the user association changes as the users' location and orientation vary with time. Consequently, each BS that connects to a given VR user can only collect partial information about this user's locations and orientation. However, a BS cannot rely on partial information to predict each user's location and orientation. Moreover, since a given VR user will change its association, the data pertaining to this VR user's movement will be located at multiple BSs. Hence, traditional centralized learning algorithms that are implemented by a given BS cannot predict the entire VR user's locations and orientations without knowing the user's data collected by other BSs. To overcome the challenges mentioned previously, we introduce a distributed federated learning framework that can predict the location and orientation of each VR user as the training data related to each user's locations and orientations is located at multiple BSs.

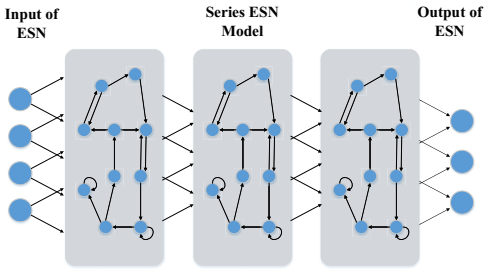
III. Federated Echo State Learning for Predictions of the Users' location and Orientation

Federated learning is a decentralized learning algorithm [32] that can operate by using training datasets that are distributed across multiple devices (e.g., BSs). For our system, one key advantage of federated learning is that it can allow multiple BSs to locally train their local learning model using their collected data and cooperatively build a learning model by sharing their locally trained models. Compared to existing federated learning algorithms [33] that use matrices to record the users' behavior and cannot analyze the correlation of the users' behavior data, we propose an ESN-based federated learning algorithm that can use an ESN to efficiently analyze the data related to the users' location and orientation. The proposed algorithm enables the BSs to collaboratively generate a global ESN model to predict the whole set of locations and orientations for each user without transmitting the collected data to other BSs. However, if the BSs use the centralized learning algorithms for the orientation and location predictions, the BSs must use the data collected from all BSs to train the algorithm. ESNs have two unique advantages: simple training process and the ability to analyze time-dependent data [22]. Since the data that is related to the orientation and locations of the users is time-dependent and the users' orientation and locations will change frequently, we must use ESNs that can efficiently analyze time-dependent data and converge quickly to obtain the prediction results on time and determine the user association. Next, we first introduce the components of the federated ESN learning model. Then, we explain the entire procedure of using our federated ESN learning algorithm to predict the users' locations and orientation.

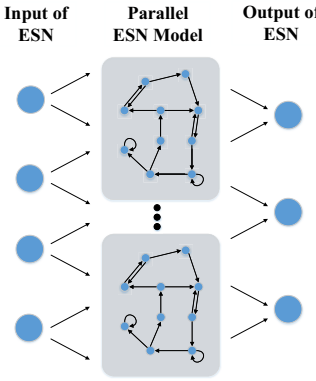
A. Components of Federated ESN Learning Algorithm

A federated ESN learning algorithm consists of five components: a) agents, b) input, c) output, and d) local ESN model, which are specified as follows:

- *Agent*: In our system, we need to define an individual federated ESN learning algorithm to predict the location and orientation of each VR user. Meanwhile, each user's individual federated ESN learning algorithm must be implemented by all BSs that have been associated with this user. Each BS j must implement U learning algorithms to predict the locations and orientations of all users.
- *Input:* The input of the federated ESN learning algorithm that is implemented by BS j for the predictions of each VR user i is defined by a vector $\boldsymbol{v}_{ij} = [\boldsymbol{v}_{ij,1}, \cdots, \boldsymbol{v}_{ij,T}]^{\mathrm{T}}$ that represents the information related to user i's location and orientation where $\boldsymbol{v}_{ij,t} = [\xi_{ij1,t}, \ldots, \xi_{ijN_x,t}]$ represents user i's information related to location and orientation at time t. This information includes user i's locations, orientations, and VR applications. N_x is the number of properties that constitute a vector $\boldsymbol{v}_{ij,t}$. The input of the proposed algorithm will be combined with the ESN model to predict users' orientation and locations. BSs will use these predictions to determine user associations.
- *Output:* For each user i, the output of the federated ESN learning algorithm at BS j is a vector $\boldsymbol{y}_{ij,t} = \left[\hat{\boldsymbol{y}}_{ijt+1}, \ldots, \hat{\boldsymbol{y}}_{ijt+Y}\right]$ of user i's locations and orientations where $\hat{\boldsymbol{y}}_{ijt+k} = [\hat{x}_{it+k}, \hat{y}_{it+k}, \hat{\chi}_{it+k}]$ with $\hat{x}_{it+k}$ and $\hat{y}_{it+k}$ being the predicted location coordinates of user i at time $t+k$ and $\hat{\chi}_{it+k}$ being the estimated orientation of user i at $t+k$. Y is the number of future time slots that a federated ESN learning algorithm can predict. The predictions of the locations and orientations can be used to determine the user's association.
- *Local ESN model:* For each BS j, a local ESN model is used to build the relationship between the input of all BSs and the predictions of the users' location and orientation, as shown in Fig. 4. The local ESN model consists of the input weight matrix $\boldsymbol{W}_j^{\text{in}} \in \mathbb{R}^{N_W \times T}$, recurrent matrix $\boldsymbol{W}_j \in \mathbb{R}^{N_W \times N_W}$, and the output weight matrix $\boldsymbol{W}_j^{\text{out}} \in \mathbb{R}^{Y \times (N_W + T)}$. The values of $\boldsymbol{W}_j^{\text{in}}$ and $\boldsymbol{W}_j$ are generated randomly. However, the output weight matrix $\boldsymbol{W}_j^{\text{out}}$ need to be trained according to the inputs of all BSs.



(a) a series ESN model



(b) a parallel ESN model

Fig. 3. Architectures of deep ESN models.

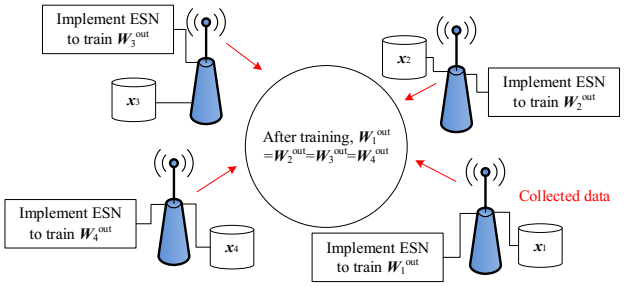


Fig. 4. The implementation of the ESN based federated learning. Here, the data is located at the BSs and the learning model $\boldsymbol{W}_j^{\text{out}}$ that is trained by each BS's collected data is the local model.

We introduce three ESN models: *single ESN model, series ESN model, and parallel ESN model*. In the single ESN model, an ESN is directly connected to the input and output. Moreover, as shown in Fig. 3, series and parallel ESN models connect single ESN models in series and parallel, respectively. Each ESN model has its own advantage for our problem. In particular, a single ESN model can converge faster than a series ESN model and a parallel ESN model. A parallel ESN model has a larger memory capacity than a series ESN model. A series ESN model can decrease the prediction errors in the training process.

B. ESN Based Federated Learning Algorithm for Users' Location and Orientation Predictions

Next, we explain the entire procedure of training the proposed ESN-based federated learning algorithm. Our purpose of training ESN is to find an optimal output weight matrix in order to accurately predict the users' locations and orientations, as shown in Fig. 4.

To introduce the training process, we first explain the state of the neurons in ESN. The neuron states of the proposed algorithm implemented by BS j for the predictions of user i are:

$$\boldsymbol{\mu}_{j,t} = \boldsymbol{W}_j \boldsymbol{\mu}_{j,t-1} + \boldsymbol{W}_j^{\text{in}} \boldsymbol{\upsilon}_{ij,t}. \tag{10}$$

Based on the states of neurons and the inputs, the ESN can estimate the output, which is:

$$\hat{\boldsymbol{y}}_{ij,t} = \boldsymbol{W}_{j,t}^{\text{out}} \begin{bmatrix} \boldsymbol{\upsilon}_{ij,t} \\ \boldsymbol{\mu}_{j,t} \end{bmatrix}. \tag{11}$$

From (11), we can see that, in order to enable an ESN to predict the users' locations and orientations, we only need to adjust the value of the output weight matrix. However, each BS can collect only partial data for each user and, hence, we need to use a distributed learning algorithm to train the ESNs. To introduce the distributed learning algorithm, we first define two matrices which are given by:

$$\boldsymbol{H}_j = \begin{bmatrix} \boldsymbol{\upsilon}_{ij,1} & \boldsymbol{\mu}_{j,1} \\ \vdots & \\ \boldsymbol{\upsilon}_{ij,T} & \boldsymbol{\mu}_{j,T} \end{bmatrix} \text{ and } \boldsymbol{E}_j = [\boldsymbol{e}_{ij,1}, \ldots, \boldsymbol{e}_{ij,T}], \tag{12}$$

where $\boldsymbol{e}_{ij,t}$ is the desired locations and orientations of each VR user, given the ESN input $\boldsymbol{\upsilon}_{ij,t}$. Then, the training purpose can be given as follows:

$$\min_{\boldsymbol{W}^{\text{out}}} \frac{1}{2} \left(\sum_{j=1}^{B} \left\| \boldsymbol{W}^{\text{out}} \boldsymbol{H}_j^{\text{T}} - \boldsymbol{E}_j \right\|^2 \right) + \frac{\lambda}{2} \left\| \boldsymbol{W}^{\text{out}} \right\|. \tag{13}$$

(13) is used to find the optimal global output weight matrix $\boldsymbol{W}^{\text{out}}$ according to which the BSs can predict the entire users' locations and orientations without the knowledge of the users' data collected by other BSs. From (13), we can see that, each BS j needs to adjust its output weight matrix $\boldsymbol{W}_j^{\text{out}}$ and find the optimal output weight matrix $\boldsymbol{W}^{\text{out}}$. After the learning step, we have $\boldsymbol{W}_j^{\text{out}} = \boldsymbol{W}^{\text{out}}$, which means that when the learning algorithm converges, the local model of each BS will converge to the global model.. A standard update policy of $\boldsymbol{W}_j^{\text{out}}$ for the augmented Lagrangian problem in (13) is given by [34]:

$$\begin{aligned} \boldsymbol{W}_{j,t+1}^{\text{out}} =& \varsigma^{-1} \left[\boldsymbol{I} - \boldsymbol{H}_j^{\text{T}} \left(\varsigma \boldsymbol{I} + \boldsymbol{H}_j \boldsymbol{H}_j^{\text{T}} \right) \boldsymbol{H}_j^{\text{T}} \right] \\ & \times \left(\boldsymbol{H}_j^{\text{T}} \boldsymbol{E}_j - \boldsymbol{n}_{j,t} + \varsigma \boldsymbol{W}_t^{\text{out}} \right), \end{aligned} \tag{14}$$

where ς is the learning rate and $\boldsymbol{W}_t^{\text{out}}$ is the optimal output weight matrix that the ESN model of each BS needs to find. From (14), we can see that $\boldsymbol{W}_{j,t+1}^{\text{out}}$ is the output weight matrix that is generated at BS j. $\boldsymbol{W}_{j,t+1}^{\text{out}}$ can only be used to predict partial locations and orientations given the users' data collected by BS j. $\boldsymbol{W}_{j,t+1}^{\text{out}}$ is different from the output weight matrices of other BSs. The optimal output weight matrix is given by:

$$\boldsymbol{W}_{t+1}^{\text{out}} = \frac{B\varsigma \hat{\boldsymbol{W}}_{t+1}^{\text{out}} + B\hat{\boldsymbol{n}}_t}{\lambda + \varsigma B}, \tag{15}$$

Algorithm 1 Federated ESN learning algorithm for location and orientation predictions

Input: Training data set (local), $\boldsymbol{v}_{ij}$.
Initialization: Each BS j generates the ESN model for each user including $\boldsymbol{W}_j^{\text{in}}$ (local), $\boldsymbol{W}_j$ (global), and $\boldsymbol{W}_j^{\text{out}}$ (local).
1: Obtain the matrices $\boldsymbol{H}_j$ and $\boldsymbol{E}_j$ based on (10).
2: **for** time t **do**
3: Compute $\boldsymbol{W}_{j,t+1}^{\text{out}}$ using (14).
4: Calculate $\hat{\boldsymbol{W}}_{t+1}^{\text{out}}$ and $\hat{\boldsymbol{n}}_t^{\text{out}}$ based on (16).
5: Calculate $\boldsymbol{W}_{t+1}^{\text{out}}$ based on (15).
6: Compute $\boldsymbol{n}_{j,t+1}$ based on (17).
7: Compute $\|\boldsymbol{r}_{j,t+1}\|$ and $\|\boldsymbol{s}_{j,t}\|$.
8: If $\|\boldsymbol{r}_{j,t+1}\| \leqslant \gamma_A$ or $\|\boldsymbol{s}_{j,t}\| \leqslant \gamma_A$, the algorithm converges.
9: **end for**

where $\hat{\boldsymbol{W}}_{t+1}^{\text{out}}$ and $\hat{\boldsymbol{n}}_{t+1}^{\text{out}}$ can be calculated as follows:

$$\hat{\boldsymbol{W}}_{t+1}^{\text{out}} = \frac{1}{B}\sum_{j=1}^{B} \boldsymbol{W}_{j,t+1}^{\text{out}}, \ \hat{\boldsymbol{n}}_t = \frac{1}{B}\sum_{j=1}^{B} \boldsymbol{n}_{j,t}. \tag{16}$$

From (14) to (16), we can see that the global output weight matrix $\boldsymbol{W}^{\text{out}}$ is based on (15) and (16) while the local output weight matrix $\boldsymbol{W}_j^{\text{out}}$ is based on (14). In (14), $\boldsymbol{n}_{j,t}$ is the deviation between the output weight matrix $\boldsymbol{W}_{j,t+1}^{\text{out}}$ of each BS j and the optimal output weight matrix $\hat{\boldsymbol{W}}_{t+1}^{\text{out}}$ that the ESN model of each BS needs to converge, which is given by:

$$\boldsymbol{n}_{j,t+1} = \boldsymbol{n}_{j,t} + \gamma\left(\boldsymbol{W}_{j,t+1}^{\text{out}} - \boldsymbol{W}_{t+1}^{\text{out}}\right). \tag{17}$$

$\boldsymbol{W}_{t+1}^{\text{out}}$ is the global optimal output weight matrix that can be used to predict the entire locations and orientations of a given user. This means that using $\boldsymbol{W}_{t+1}^{\text{out}}$, each BS can predict the entire user's locations and orientations as the BS only collects partial data related to the user's locations and orientations. As time elapses, $\boldsymbol{W}_{j,t+1}^{\text{out}}$ will finally converge to $\boldsymbol{W}_{t+1}^{\text{out}}$. In consequence, all of BSs can predict the entire locations and orientations of each user. To measure the convergence, we define two vectors which can be given by $\boldsymbol{r}_{j,t} = \boldsymbol{W}_{j,t}^{\text{out}} - \boldsymbol{W}_t^{\text{out}}$ and $\boldsymbol{s}_{j,t} = \boldsymbol{W}_t^{\text{out}} - \boldsymbol{W}_{t-1}^{\text{out}}$. As $\|\boldsymbol{r}_{j,t+1}\| \leqslant \gamma_A$ or $\|\boldsymbol{s}_{j,t}\| \leqslant \gamma_A$, the proposed algorithm converges. γ_A is determined by the BSs. Since the minimization function in (13) is a convex function, the BSs are guaranteed to find an optimal output weight matrix that satisfy $\|\boldsymbol{r}_{j,t+1}\| \leqslant \gamma_A$ or $\|\boldsymbol{s}_{j,t}\| \leqslant \gamma_A$. As γ_A increases, the accuracy of the predictions and the number of iterations decrease. Therefore, BSs need to jointly account for the time used for training ESN and the prediction accuracy to determine the value of γ_A. In fact, the ESN As the learning algorithm converges, each BS can use its own ESN to predict the entire location and orientation of each VR user. According to these predictions, BSs can determine the user association to minimize the BIP of VR users. Algorithm 1 summarizes the entire process of using ESN based federated learning algorithm for the predictions of the users' locations and orientations. From Algorithm 1, we can see that $\boldsymbol{W}_j^{\text{in}}$ and $\boldsymbol{W}_j^{\text{out}}$ are local parameters which means that each BS j will generate its own $\boldsymbol{W}_j^{\text{in}}$ and $\boldsymbol{W}_j^{\text{out}}$. However, $\boldsymbol{W}_j$ is a global parameter which means that all of the BSs will have the same $\boldsymbol{W}_j$.

IV. Memory Capacity Analysis

To improve the prediction accuracy of the proposed algorithm, we analyze the memory capacity of the proposed ESN model. The memory capacity quantifies the ability of each ESN to record the historical locations and orientations of each VR user. As the memory capacity of the ESNs increases, the ESNs can record more historical data related to users' locations and use this information to achieve better prediction[1] for the users' locations and orientations. The analysis of the ESN memory capacity will be used for the choice of the ESN models for the predictions of the users' locations and orientations. Next, we derive closed-form expressions of the memory capacity of the three ESN models that we described in Section III, namely, the single ESN model, the parallel ESN model, and the series ESN model. Note that, our previous work [35] analyzed the memory capacity for a centralized parallel ESN model. In contrast, here, we analyze the memory capacity for three ESN models used for federated learning.

We assume that the input of each ESN model at time t is m_t and the output of each ESN model is z_t. Then, the memory capacity of each ESN model is given by [36]:

$$M = \sum_{k=1}^{\infty} \frac{\text{Cov}^2(m_{t-k}, z_t)}{\text{Var}(m_t)\text{Var}(z_t)}, \tag{18}$$

where Cov and Var represent the covariance and variance operators, respectively. In (18), $\frac{\text{Cov}^2(m_{t-k},z_t)}{\text{Var}(m_{t-k})\text{Var}(z_t)}$ captures the correlation between the ESN input m_{t-k} at time $t-k$ and the ESN output z_t at time t. $\frac{\text{Cov}^2(m_{t-k},z_t)}{\text{Var}(m_{t-k})\text{Var}(z_t)} = 1$ indicates that m_{t-k} and z_t are related which means that the output z_t includes the information of m_{t-k} and, hence, the ESN can record input m_{t-k} at time t. $\frac{\text{Cov}^2(m_{t-k},z_t)}{\text{Var}(m_t)\text{Var}(z_t)} = 0$ indicates that m_{t-k} and z_t are unrelated, which means that z_t does not include any information related to m_{t-k} and, hence, the ESN cannot record m_{t-k}. In consequence, M represents the total number of historical input data that each ESN can record. The recurrent matrix $\boldsymbol{W}$ in each ESN model is given by:

$$\boldsymbol{W}_l = \begin{bmatrix} 0 & 0 & \cdots & w \\ w & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & w & 0 \end{bmatrix}, \tag{19}$$

and the input weight matrix is given by $\boldsymbol{W}^{\text{in}} = \left[w_1^{\text{in}}, \ldots, w_{N_W}^{\text{in}}\right]^{\text{T}}$. We also define a matrix that will be used to derive the memory capacity of the ESNs, which can be given by:

$$\boldsymbol{V} = \begin{bmatrix} w_1^{\text{in}} & w_{N_W}^{\text{in}} & \cdots & w_2^{\text{in}} \\ w_2^{\text{in}} & w_1^{\text{in}} & \cdots & w_3^{\text{in}} \\ \vdots & \vdots & \cdots & \vdots \\ w_{N_W}^{\text{in}} & w_{N_W-1}^{\text{in}} & \cdots & w_1^{\text{in}} \end{bmatrix}. \tag{20}$$

[1]Here, as the size of the recorded data increases, the ESNs can use more historical data to build a relationship between historical orientations and locations, and future orientations and locations. Hence, the ESN prediction accuracy improves.

Based on the above definitions, we can invoke our result from [36, Theorem 2] to derive the memory capacity of single ESN model, which can be given as follows.

Corollary 1 (Single ESN model)**.** Given the recurrent matrix $\boldsymbol{W}$ and the input matrix $\boldsymbol{W}^{\text{in}}$ that guarantees the matrix $\boldsymbol{V}$ regular, the memory capacity of the single ESN model is:

$$M = N_W - 1 + w^{2N_W}. \tag{21}$$

Proof. Given the input stream vector $\boldsymbol{m}_{\dots t} = [m_1, \dots, m_{t-1}, m_t]$, we can calculate the activations $\boldsymbol{\mu}_{j,t}$ using (10). The output weight matrix of the ESN model can be given by $\boldsymbol{W}^{\text{out}} = \boldsymbol{R}^{-1}\boldsymbol{p}_k$, where $\boldsymbol{R} = \mathbb{E}\left[\boldsymbol{\mu}_t \left(\boldsymbol{\mu}_t\right)^{\text{T}}\right]$ represents the covariance matrix with $\boldsymbol{\mu}_t = [\mu_{1,t}, \dots, \mu_{N_W,t}]$ and $\boldsymbol{p}_k = \mathbb{E}\left[\boldsymbol{\mu}_t m_{t-k}\right]$. Assume that $\boldsymbol{w}^{\text{in}}_{N_W \dots 1} = \left[w^{\text{in}}_{N_W}, w^{\text{in}}_{N_W-1}, \dots, w^{\text{in}}_1\right]$ and $\text{rot}_k\left(\boldsymbol{w}^{\text{in}}_{N_W \dots 1}\right)$ is an operator that rotates vector $\boldsymbol{w}^{\text{in}}_{N_W \dots 1}$ by k positions to the right. We have $\boldsymbol{W}^{\text{out}} = (1 - w^{2N_W}) w^k \boldsymbol{A}^{-1} \text{rot}_k(\boldsymbol{w}^{\text{in}}_{1 \dots N_W})$, where $\boldsymbol{A} = \boldsymbol{V}^{\text{T}} \Gamma^2 \boldsymbol{V}$ with $\Gamma = \text{diag}\left(1, w, \dots, w^{N_W - 1}\right)$. Based on $\boldsymbol{W}^{\text{out}}$, we can the covariance of the output with the k-slot delayed input, which is given by $\text{Cov}(z_t, m_{t-k}) = (1 - w^{2N_W}) w^{2k} \sigma^2 \zeta_k$. We can also obtain $\text{Var}(z_t) = \mathbb{E}\left[z_t z_t\right] = (1 - w^{2N_W}) w^{2k} \sigma^2 \zeta_k$. Since $\text{Var}(m_t) = \sigma^2$, we have $M = \sum_{k=1}^{\infty} \frac{\text{Cov}^2(m_{t-k}, z_t)}{\text{Var}(m_t)\text{Var}(z_t)} = N_W - 1 + w^{2N_W}$. □

From Corollary 1, we can see that the memory capacity of the single ESN model depends on the number of neurons and values of the recurrent matrix. Corollary 1 also shows that the memory capacity of the single ESN model will not exceed N_W. That means the single ESN model based federated learning algorithm can only record N_W locations or orientations.

Next, we derive the memory capacity of the parallel ESN model, which can be given by the following theorem.

Theorem 1 (Parallel ESN)**.** Given a parallel ESN model during which L ESN models are parallel connected with each other, each ESN model's input weight matrix $\boldsymbol{W}^{\text{in}}$ that guarantees the matrix $\boldsymbol{V}$ regular and recurrent matrix $\boldsymbol{W}$, then the memory capacity of each parallel ESN can be given by:

$$M = N_W - 1 + w^{2N_W}. \tag{22}$$

Proof. See Appendix A. □

Theorem 1 shows that the memory capacity of a parallel ESN model is similar to the memory capacity of a single ESN. Hence, adding multiple ESN models will not increase the memory capacity. This is due to the fact that, in a parallel ESN model, there is no connection among the ESNs, as shown in Fig. 3(b). Therefore, the input of the parallel ESN model will separately connect to each single ESN and, hence, the parallel ESN models do not need to use more neurons to record the input data compared to the single ESN model. Theorem 1 also shows that the memory capacity of a parallel ESN depends on the number of neurons in each ESN model and the values of the recurrent weight matrix of each ESN model. Accordingly, we can increase the value of output weight matrix and the number of neurons in each ESN model to increase the memory capacity of the parallel ESN models. As the memory capacity of the parallel ESN models increases, BSs can record more users' data to predict the users' locations and orientations accurately. Next, we derive the memory capacity of the series ESN model.

Theorem 2 (Series ESN model)**.** Given a series ESN model during which L ESN models are series connected with each other, a recurrent matrix $\boldsymbol{W}$ of each ESN model, and each ESN model's input weight matrix $\boldsymbol{W}^{\text{in}}$ that guarantees the matrix $\boldsymbol{V}$ regular, the memory capacity of each series ESN model is:

$$M = \left(1 - w^{2N_W}\right)^{L-1} \left(N_W - 1 + w^{2N_W}\right). \tag{23}$$

Proof. See Appendix B. □

From Theorem 2, we can see that the memory capacity of each series ESN model is smaller than the memory capacity of a single ESN or a series ESN. Theorem 2 also shows that the memory capacity of each series ESN model decreases as the number of ESN models L increases. Thus, it would be better to use a single ESN model or a parallel ESN model to predict the users' locations and orientations.

Theorems 1 and 2 derive the memory capacities of the parallel ESN model and the series ESN model with single input. Next, we formulate the memory capacity of a single ESN model given multiple inputs, which is given by the following theorem.

Theorem 3 (Multi-input single ESN)**.** Consider a single ESN with a recurrent matrix $\boldsymbol{W}$, input vector $\boldsymbol{m}_t = [m_{1t}, \dots, m_{Kt}]$, the input weight matrix $\boldsymbol{W}^{\text{in}}$ that guarantees the matrix $\boldsymbol{V}$ regular, the memory capacity of each single ESN is

$$M = \left(\frac{\sum_{l=1}^{K} \sigma_l^2}{\sum_{k=1}^{K} \sum_{n=1}^{K} \rho_{kn} \sigma_k \sigma_n}\right)^2 \left(N_W - 1 + w^{2N_W}\right), \tag{24}$$

where ρ_{kn} represents the correlation coefficient between input m_{kt} and m_{nt}.

Proof. See Appendix C. □

From Theorem 3, we can observe that the correlation among input elements in vector $\boldsymbol{m}_t$ will affect the memory capacity of each ESN model. In particular, as the correlation of the input data increases, the memory capacity of the ESN model increases. This is because the ESN can use more input data to predict the users' locations and orientations, hence improving the predictions accuracy. Therefore, it would be better to jointly predict the users' locations and orientations.

Theorems 1-3 allow each BS to determine its ESN model, the number of neurons N_W in each ESN model, and the values of the recurrent matrix $\boldsymbol{W}$ as the size of the data collected by each BS changes. A parallel ESN model has a larger memory capacity compared with the series ESN model and is more

stable than the single ESN model, and, hence, a parallel ESN model can record more historical data to predict the users' orientations and locations so as to improve the prediction accuracy. As the prediction accuracy is improved, the BSs can determine the user association more accurately. Hence, the BIP of the users can be minimized. Therefore, we use the parallel ESN model in our proposed algorithm.

V. User Association for VR Users

Based on the analysis presented in Sections III and IV, each BS can predict the users' locations and orientations. Next, we explain how to use these predictions to find the user association for each VR user. Given the predictions of the locations and orientations, the BIP minimization problem in (9) can be rewritten as follows:

$$\min_{\boldsymbol{a}_{i,t}^{\mathrm{UL}},\boldsymbol{a}_{i,t}^{\mathrm{DL}}} \sum_{i\in\mathcal{U}} P_i\left(\hat{x}_{it},\hat{y}_{it},G_A,\hat{\chi}_{it},\boldsymbol{a}_{i,t}^{\mathrm{UL}},\boldsymbol{a}_{i,t}^{\mathrm{DL}}\right). \tag{25}$$

We use the reinforcement learning algorithm given in [37] to find a sub-optimal solution of the problem in (25). In the reinforcement learning algorithm given in [37], the actions are the user association schemes, the states are the strategies of other BSs, and the output is the estimated BIP. Hence, this reinforcement learning algorithm can learn the VR users state and exploit different actions to adapt the user association according to the predictions of the users' locations and orientations. After the learning step, each BS will find a sub-optimal user association to service the VR users. To simplify the learning process and improve the convergence speed, we first select the uplink user association scheme. This is because as the uplink user association is determined, the BSs that the users can associate in downlink will be determined, as follows:

Proposition 1. Given the predicted location and orientation of user i at time t as well as the uplink user association $a_{i*,t}^{\mathrm{UL}}$, the downlink cell association for a VR user i is:

$$a_{ik,t}^{\mathrm{DL}} = \mathbb{1}_{\left\{\frac{D\left(\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}\right)\right)}{a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}}\leqslant\gamma_{\mathrm{D}}-\frac{A}{a_{i*,t}^{\mathrm{UL}}c_{i*}^{\mathrm{UL}}(\hat{x}_{it},\hat{y}_{it})}\right\}} \wedge \mathbb{1}_{\left\{\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}\right)\boldsymbol{m}_{i,t}(G_A)\geqslant\gamma_{\mathrm{Q}}\right\}}, \tag{26}$$

where $c_{ik}^{\mathrm{DL}}\left(\hat{x}_{it},\hat{y}_{it},b_i\left(\hat{\chi}_{it}\right),n_{ik}\right)$ is short for c_{ik}^{DL} and $a_{ik,t}^{\mathrm{DL}}$ is the downlink user association obtained in (26). $c_{i*}^{\mathrm{UL}}\left(x_{it},y_{it}\right)$ is the uplink data rate of user i.

Proof. For downlink user association, each VR user i needs to find a BS that can guarantee the transmission delay and VR video quality. Since we have determined the user association over uplink, the maximum time used for VR video transmission can be given by $\gamma_{\mathrm{D}}-\frac{A}{a_{i*,t}^{\mathrm{UL}}c_{i*}^{\mathrm{UL}}(\hat{x}_{it},\hat{y}_{it})}$. Consequently, user i needs to connect with a BS that can satisfy the transmission delay requirement of user i, i.e., $\frac{D\left(\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}(\hat{x}_{it},\hat{y}_{it},b_i(\hat{\chi}_{it}),n_{ik})\right)\right)}{a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}(\hat{x}_{it},\hat{y}_{it},b_i(\hat{\chi}_{it}),n_{ik})} \leqslant \gamma_{\mathrm{D}}-\frac{A}{a_{i*,t}^{\mathrm{UL}}c_{i*}^{\mathrm{UL}}(\hat{x}_{it},\hat{y}_{it})}$. Moreover, user i needs to associate with a BS that can meet the requirement of VR video quality, i.e., $\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}\left(\hat{x}_{it},\hat{y}_{it},b_i\left(\hat{\chi}_{it}\right),n_{ik}\right)\right)\boldsymbol{m}_{i,t}\left(G_A\right) \geqslant \gamma_{\mathrm{Q}}$. Thus, if BS k can satisfy the conditions: $\frac{D\left(\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}(\hat{x}_{it},\hat{y}_{it},b_i(\hat{\chi}_{it}),n_{ik})\right)\right)}{a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}(\hat{x}_{it},\hat{y}_{it},b_i(\hat{\chi}_{it}),n_{ik})} \leqslant \gamma_{\mathrm{D}}-\frac{A}{a_{i*,t}^{\mathrm{UL}}c_{i*}^{\mathrm{UL}}(\hat{x}_{it},\hat{y}_{it})}$ and $\boldsymbol{l}_{i,t}\left(a_{ik,t}^{\mathrm{DL}}c_{ik}^{\mathrm{DL}}\left(\hat{x}_{it},\hat{y}_{it},b_i\left(\hat{\chi}_{it}\right),n_{ik}\right)\right)\boldsymbol{m}_{i,t}\left(G_A\right) \geqslant \gamma_{\mathrm{Q}}$, user i can associate with it. This completes the proof. □

TABLE I
SYSTEM PARAMETERS

Parameters	Values	Parameters	Values
P_B	30 dBm	d_0	5 m
P_U	10 dBm	f_c	28 GHz
σ	-94 dBm	c	3×10^8 m/s
F^{UL}	10 Mbit	$\varpi_{\mathrm{LoS}}, \varpi_{\mathrm{NLoS}}$	2, 2.4
F^{DL}	10 Mbit	$\mu_{\sigma_{\mathrm{LoS}}}, \mu_{\sigma_{\mathrm{NLoS}}}$	5.3, 5.27
N_W	30	G_A	11
β	2	γ_D	10 ms
M	15 dB	γ_Q	0.8
m	0.7 dB	${\sigma_i}^2$	0.193
ϕ	30°	A	50 kbits
Y	10	T	5
γ	0.5	λ	0.005
w	0.98	L	3
V	10	ϑ	2
$\sigma^2_{G_A\|i}$	0.151	${\sigma_B}^2$	0.05

From Proposition 1, we can see that the user association of each user i depends on user i's location and orientation. Proposition 1 shows that, for each user i, the uplink user association will affect the downlink user association. This is due to the fact that the VR system has determined the total transmission delay of each user. As a result, when the uplink user association is determined, the uplink transmission delay and the requirement of the downlink transmission delay will be determined.

VI. Simulation Results and Analysis

For our simulations, we consider a circular area with radius $r = 500$ m, $U = 20$ wireless VR users, and $B = 5$ BSs distributed uniformly. To simulate blockage, each user is considered as a two-dimensional point. For simplicity, we ignore the altitudes of the BSs and the height of the users. If blockage points are located between a user and its BS, the communication link will be considered to be NLoS. Real data traces for locations are collected from 50 students at the Beijing University of Posts and Telecommunications. The locations of each student is collected every hour during 9:00 am – 9:00 pm. For orientation data collection, we searched 25 videos related to a first-person shooter game from youTube. Then, we input these VR videos to HTC Vive devices. The HTC Vive deveploper system can directly measure the movement of the VR videos using HTC Vive devices. We arbitrarily combine one user's locations with one orientation for each VR user. In simulations, a parallel ESN model[2] is used for the proposed algorithm due to its stability and large memory capacity. The other system parameters are listed in Table I. For comparison purposes, we consider the deep learning algorithm in [15] and the ESN algorithm in [16], as two baseline schemes. The deep learning algorithm in [15] is a deep autoencoder that consists of multiple layers of restricted Boltznann machines. The centralized ESN-based learning algorithm in [16] is essentially a single layer ESN algorithm. The input and output of the centralized ESN and

[2]The code can be found in https://github.com/lasisal/deepESN.

deep learning algorithms are similar to the proposed algorithm. However, for the deep learning algorithm and the centralized ESN algorithm, each BS can use only its collected data to train the learning model. Both the centralized and deep learning algorithms are trained in an offline manner. All statistical results are averaged over a large number of independent runs.

Figs. 5 and 6 show the predictions of the VR users' locations and orientations as time elapses. To simplify the model training, the collected data related to locations and orientations are mapped to $[-0.5, 0.5]$. The orientation and location of each user are, respectively, mapped by the function $\frac{\chi_{it}}{360^\circ} - 0.5$ and $\frac{z}{z_{\max}} - 0.5$ where $z = \left(\sum_{n=1}^{\hat{x}_{it}+\hat{y}_{it}} n \right) \times \hat{y}_{it}$. From Figs. 5 and 6, we observe that the proposed algorithm can predict the users' locations and orientations more accurately than the centralized ESN and deep learning algorithms. Figs. 6(b) and 6(c) also show that the prediction error mainly occurs at time slot 8 to 12. This is due to the fact that the proposed algorithm can build a learning model that predicts the entire locations and orientations of each user. In particular, the output weight matrices of all ESN algorithms implemented by each BS will converge to a common matrix. Hence, BSs can predict the entire locations and orientations of each VR user.

Fig. 7 shows how the total BIP of all VR users changes as the number of BSs varies. From Fig. 7, we can see that, as the number of BSs increases, the total BIP of all VR users decreases. That is because as the number of BSs increases, the VR users have more connection options. Hence, the blockage caused by human bodies will be less severe, thereby improving the data rates of VR users. Fig. 7(a) also shows that the proposed algorithm can achieve up to 16% and 26% reduction in the number of BIP, respectively, compared to centralized ESN algorithm and deep learning algorithm for a network with 9 BSs. These gains stem from the fact that the centralized ESN and deep learning algorithms can partially predict the locations and orientation of each VR user as they rely only on the local data collected by a BS. In contrast, the proposed algorithm facilitates cooperation among BSs to build a learning model that can predict the entire users' locations and orientations. Fig. 7(b) shows that the proposed algorithm using a parallel ESN model can achieve up to 8% and 14% gains in terms of the total BIP of all users compared to the proposed algorithm with a single ESN model and with a series model. Clearly, compared to a single ESN, using a parallel ESN model can increase the stability of the proposed algorithm. Meanwhile, the memory capacity of a parallel model is larger than a series ESN model thus improving the prediction accuracy and reducing BIP for users.

In Fig. 8, we show how the total BIP of all VR users changes with the number of VR users. This figure shows that, with more VR users, the total BIP of all VR users increases rapidly due to an increase in the uplink delay, as the sub-6 GHz bandwidth is shared by more users. Fig. 8 also shows that the gap between the proposed algorithm and the centralized ESN algorithm decreases as more VR users are present in the network.. Clearly, with more VR users, it becomes more probable that a user located between a given VR user and its associated BS blocks the mmWave link. Thus, as the number of users increases, more VR users will receive their VR videos over NLoS links and, the total BIP significantly increases.

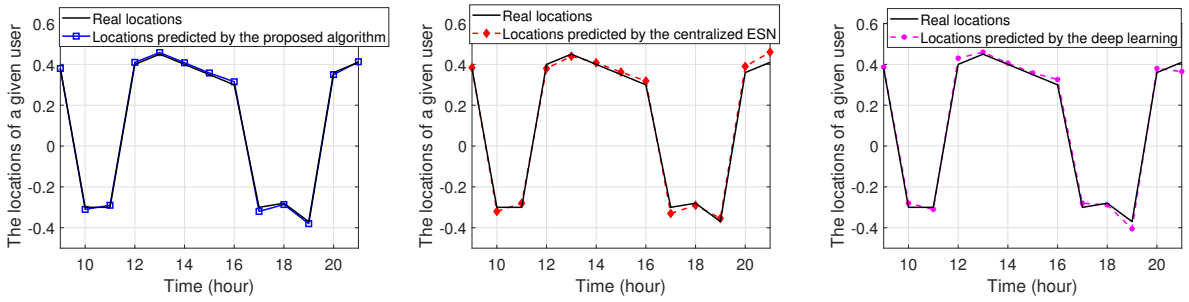
In Fig. 9, we show the CDF for the VR users' BIP for all three algorithms. Fig. 9 shows that the BIP of almost 98% of users resulting from the considered algorithms will be larger than 10. This is due to the fact that the BIP will also be caused by other factors such as VR applications and user's awareness. In Fig. 9, we can also see that the proposed algorithm improves the CDF of up to 38% and 71% gains at a BIP of 25 compared to the centralized ESN and deep learning algorithms, respectively. These gains stem from the fact the ESNs are effective at analyzing the time related location and orientation data and, hence, they can accurately predict the users' locations and orientations.

Fig. 10 shows how the normalized root mean square error (NRMSE) of the predictions changes as the number of neurons in each ESN model varies. In this figure, the NRMSE of the predictions is given by $\left\| \hat{\boldsymbol{y}}_{ij,t} - \boldsymbol{e}_{ij,t} \right\|$. From Fig. 10, we can see that, with more neurons, the NRMSE of the predictions resulting from all of the considered ESN models decreases. This is because, as the number of neurons increases, each ESN model can record more historical data related to the users' locations and orientations. Fig. 10 also shows that the parallel model can achieve up to 37.5% and 90% gains in terms of NRMSE compared to the series model for the ESN models have 30 neurons. This is due to the fact that the prediction errors of a parallel ESN model is averaged over multiple outputs, thus, improving the prediction accuracy.

Fig. 11 shows how the total BIP of all VR users changes as the number of BSs varies. In this figure, all of the considered algorithms used the reinforcement learning algorithm given in [31] to solve the problem in (25). In the algorithm without the knowledge of the locations and orientations, the users are randomly associated with the BSs. From Fig. 11, we can see that the proposed algorithm yields 23% fewer BIP compared to the algorithm without the knowledge of the locations and orientations. This is due to the fact that the proposed algorithm uses federated ESN algorithm to predict the users' orientations and locations to optimize the user association and reduce the BIP of the users. From Fig. 11, we can also see that, a gap exists between the proposed algorithm and the algorithm with the perfect knowledge of the users' orientations and locations. This gap stems from the prediction inaccuracy caused by the proposed algorithm.

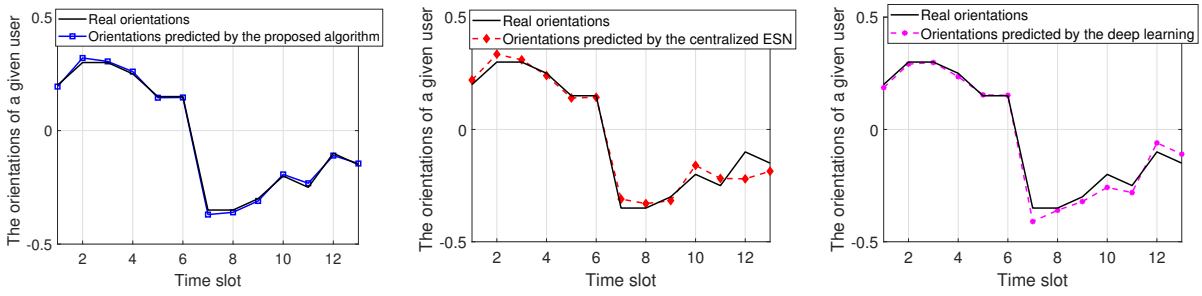
VII. Conclusion

In this paper, we have developed a novel framework for minimizing BIP within VR applications that operate over wireless networks. To this end, we have developed a BIP model that jointly considers the VR applications, transmission delay, VR video quality, and the user's awareness. We have then formulated an optimization problem that seeks to minimize the BIP of VR users by predicting users' locations



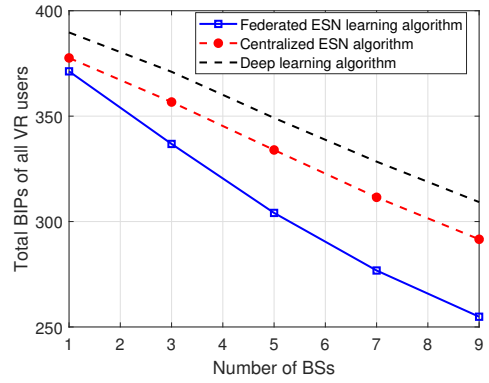
(a) (b) (c)

Fig. 5. Predictions of the VR users' orientations and locations as time elapses.

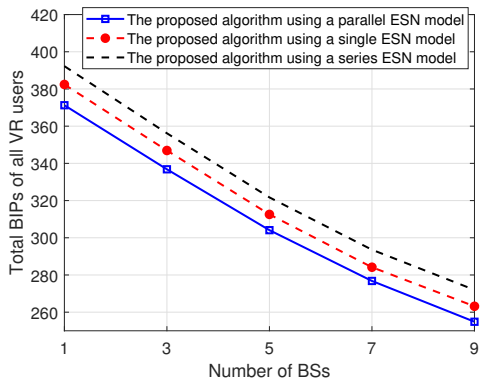


(a) (b) (c)

Fig. 6. Predictions of the VR users' orientations and locations as time elapses.



(a)



(b)

Fig. 7. Total BIP experienced by VR users as the number of BSs varies.

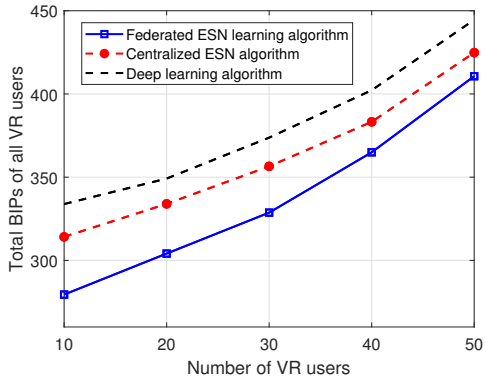


Fig. 8. Total BIP of all VR users as the number of VR users varies.

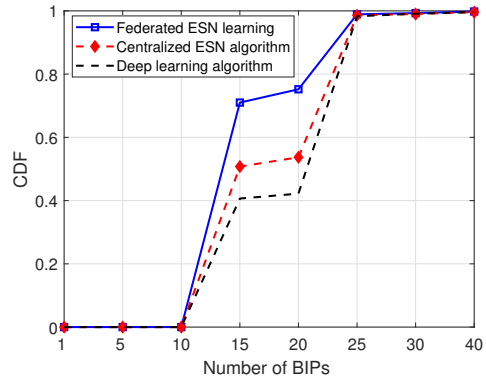


Fig. 9. CDFs of the BIP resulting from the different algorithms.

and orientations, as well as determining the user association. To solve this problem, we have developed a novel federated learning algorithm based on echo state networks. The proposed federated ESN algorithm enables the BSs to train their ESN with their locally collected data and share these models to build a global learning model that can predict the entire locations and orientations of each VR user. To improve the prediction accuracy of the proposed algorithm, we derive a closed-form expression of the memory capacity for ESNs to determine the number of neurons in each ESN model and the values of the recurrent weight matrix. Using these predictions, each BS can determine the user association in both uplink and

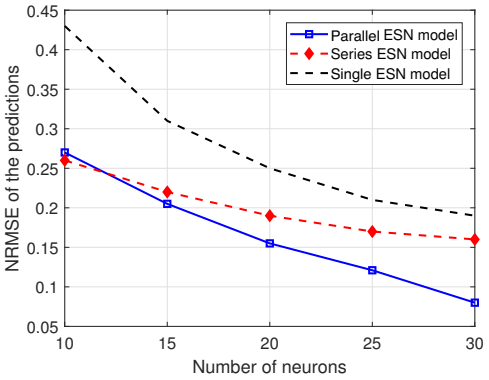


Fig. 10. Normalized root mean square error of the predictions as the number of neur

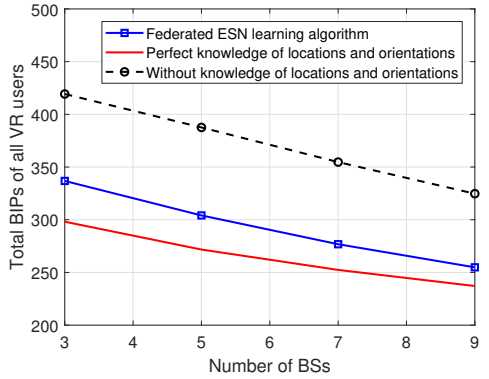


Fig. 11. Total BIP of all VR users as the number of BSs varies.

downlink. Simulation results have shown that, when compared to the centralized ESN and deep learning algorithms, the proposed approach achieves significant performance gains of BIP.

APPENDIX

A. *Proof of Theorem* 1

Given the input stream vector $\boldsymbol{m}_{\dots t} = [m_1, \dots, m_{t-1}, m_t]$, the activations $\boldsymbol{\mu}_{j,t}$ in (10) of the reservoir neuron in ESN model l at time t can be given by:

$$\begin{aligned}\mu_{j1,t}^{(l)} =& w_{1,l}^{\text{in}} m_t + w w_{N_W,l}^{\text{in}} m_{t-1} + w^2 w_{N_W-1,l}^{\text{in}} m_{t-2} \\ &+ \cdots + w^{N_W-1} w_{2,l}^{\text{in}} m_{t-(N_W-1)} + w^{N_W} w_{1,l}^{\text{in}} m_{t-N_W} \\ &+ w^{N_W+1} w_{N_W,l}^{\text{in}} m_{t-(N_W+1)} + \cdots \\ &+ w^{2N_W-1} w_{2,l}^{\text{in}} m_{t-(2N_W-1)} + w^{2N_W} w_{1,l}^{\text{in}} m_{t-2N_W} \\ &+ w^{2N_W+1} w_{N_W,l}^{\text{in}} m_{t-(2N_W+1)} + \cdots, \\ &\vdots \\ \mu_{jN_W,t}^{(l)} =& w_{N_W,l}^{\text{in}} m_t + w w_{N_W-1,l}^{\text{in}} m_{t-1} + w^2 w_{N_W-2,l}^{\text{in}} m_{t-2} \\ &+ \cdots + w^{N_W-1} w_{1,l}^{\text{in}} m_{t-(N_W-1)} + w^{N_W} w_{N_W,l}^{\text{in}} m_{t-N_W} \\ &+ w^{N_W+1} w_{N_W-1,l}^{\text{in}} m_{t-(N_W+1)} \\ &+ \cdots + w^{2N_W} w_{N_W,l}^{\text{in}} m_{t-2N_W} \\ &+ w^{2N_W+1} w_{N_W-1,l}^{\text{in}} m_{t-(2N_W+1)} + \cdots, \end{aligned} \tag{27}$$

where w is an element of the recurrent matrix $\boldsymbol{W}$ which is assumed to be equal for all of the ESN models. The output weight matrix of each ESN model l can be given by

$$\boldsymbol{W}_l^{\text{out}} = {\boldsymbol{R}_l}^{-1} \boldsymbol{p}_{k,l},$$

where $\boldsymbol{R}_l = \mathbb{E}\left[\boldsymbol{\mu}_t^{(l)} \left(\boldsymbol{\mu}_t^{(l)}\right)^{\text{T}}\right]$ represents the covariance matrix with $\boldsymbol{\mu}_t^{(l)} = \left[\mu_{1,t}^{(l)}, \dots, \mu_{N_W,t}^{(l)}\right]$ and $\boldsymbol{p}_{k,l} = \mathbb{E}\left[\boldsymbol{\mu}_t^{(l)} m_{t-k}\right]$. The element $\boldsymbol{R}_{l,12}$ in $\boldsymbol{R}_l$ can be calculated as follows

$$\begin{aligned}\boldsymbol{R}_{l,12} =& \mathbb{E}\left[\mu_{2,t}^{(l)} \mu_{1,t}^{(l)}\right] \\ =& \mathbb{E}[w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} m_t^2 + w^2 w_{N_W,l}^{\text{in}} w_{1,l}^{\text{in}} m_{t-1}^2 + \cdots \\ &+ w^{2(N_W-1)} w_{2,l}^{\text{in}} w_{3,l}^{\text{in}} m_{t-(N_W-1)}^2 + w^{2N_W} w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} m_{t-N_W}^2 \\ &+ \cdots + w^{2(2N_W-1)} w_{2,l}^{\text{in}} w_{3,l}^{\text{in}} m_{t-(2N_W-1)}^2 \\ &+ w^{4N_W} w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} m_{t-2N_W}^2 + \cdots] \\ =& \sigma^2 (w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} + w^2 w_{N_W,l}^{\text{in}} w_{1,l}^{\text{in}} + \cdots + w^{2(N_W-1)} w_{2,l}^{\text{in}} w_{3,l}^{\text{in}} \\ &+ w^{2N_W} w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} + w^{2(N_W+1)} w_{N_W,l}^{\text{in}} w_{1,l}^{\text{in}} + \cdots \\ &+ w^{2(2N_W-1)} w_{2,l}^{\text{in}} w_{3,l}^{\text{in}} + w^{4N_W} w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} + \cdots) \\ =& \sigma^2 \sum_{j=0}^{\infty} w^{2N_W j} (w_{1,l}^{\text{in}} w_{2,l}^{\text{in}} + w^2 w_{N_W,l}^{\text{in}} w_{1,l}^{\text{in}} + \cdots \\ &+ w^{2(N_W-1)} w_{2,l}^{\text{in}} w_{3,l}^{\text{in}}) \\ =& \frac{\sigma^2}{1 - w^{2N_W}} \left(\text{rot}_1\left(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}\right)\right)^{\text{T}} \Gamma^2 \text{rot}_2(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}), \end{aligned} \tag{28}$$

where σ^2 is the variance of the input signal m_t. $\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}} = \left[w_{N_W,l}^{\text{in}}, w_{N_W-1,l}^{\text{in}}, \dots, w_{1,l}^{\text{in}}\right]$ and $\text{rot}_k\left(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}\right)$ denote an operator that rotates vector $\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}$ by k place to the right. For example, $\text{rot}_1\left(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}\right) = \left[w_{1,l}^{\text{in}}, w_{N_W,l}^{\text{in}}, \dots, w_{2,l}^{\text{in}}\right]$. $\Gamma = \text{diag}\left(1, w, \dots, w^{N_W-1}\right)$. The element $\boldsymbol{R}_{l,ij}$ is given by:

$$\boldsymbol{R}_{l,ij} = \frac{\sigma^2}{1 - w^{2N_W}} (\text{rot}_i(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}))^{\text{T}} \Gamma^2 \text{rot}_j(\boldsymbol{w}_{N_W \dots 1,l}^{\text{in}}). \tag{29}$$

Thus, $\boldsymbol{R}_l = \frac{\sigma^2}{1-w^{2N_W}} \boldsymbol{V}_l^{\text{T}} \Gamma^2 \boldsymbol{V}_l = \frac{\sigma^2}{1-w^{2N_W}} \boldsymbol{A}_l$ where $\boldsymbol{A}_l = \boldsymbol{V}_l^{\text{T}} \Gamma^2 \boldsymbol{V}_l$. Similarly, based on (28), element $p_{k1,l}$ is given by:

$$p_{k1,l} = \mathbb{E}\left[\mu_{1,t}^{(l)} m_{t-k}\right] = \sigma^2 w^k w_{(N_W-k+1) \bmod N_W,l}^{\text{in}}. \tag{30}$$

We assume that $w_{0,l}^{\text{in}} = w_{N_W,l}^{\text{in}}$. Hence, $\boldsymbol{p}_{k,l} = \sigma^2 w^k \text{rot}_k\left(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}}\right)$. Then, the output weight matrix of each ESN model l is $\boldsymbol{W}_l^{\text{out}} = (1 - w^{2N_W}) w^k \boldsymbol{A}_l^{-1} \text{rot}_k(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}})$.

The output of each ESN model l at time t can be given by $z_{l,t} = (\boldsymbol{\mu}_t^{(l)})^{\text{T}} \boldsymbol{W}_l^{\text{out}} = (1 - w^{2N_W}) w^k (\boldsymbol{\mu}_t^{(l)})^{\text{T}} \boldsymbol{A}_l^{-1} \text{rot}_k(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}})$ and $z_t = \sum_{l=1}^{L} z_{l,t}$. Then, the covariance of the output with the k-slot delayed input can be calculated by:

$$\begin{aligned}&\text{Cov}(z_t, m_{t-k}) \\ =& \sum_{l=1}^{L} (1 - w^{2N_W}) w^k \text{Cov}\left(\left(\boldsymbol{\mu}_t^{(l)}\right)^{\text{T}}, m_{t-k}\right) \boldsymbol{A}_l^{-1} \text{rot}_k(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}}) \\ =& L(1 - w^{2N_W}) w^{2k} \sigma^2 \left(\text{rot}_k(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}})\right)^{\text{T}} \boldsymbol{A}_l^{-1} \text{rot}_k(\boldsymbol{w}_{1 \dots N_W,l}^{\text{in}}) \\ \overset{(a)}{=}& L(1 - w^{2N_W}) w^{2k} \sigma^2 \zeta_k, \end{aligned} \tag{31}$$

where (a) is obtained from the fact that $\zeta_k = \left(\mathrm{rot}_k(\boldsymbol{w}^{\text{in}}_{1\ldots N_W,l})\right)^{\mathrm{T}} \boldsymbol{A}_l^{-1}\mathrm{rot}_k(\boldsymbol{w}^{\text{in}}_{1\ldots N_W,l})$. The variance of the output can be given by:

$$\begin{aligned}\mathrm{Var}(z_t) &= \mathbb{E}\left[\sum_{l=1}^{L} z_{l,t}\sum_{p=1}^{L} z_{p,t}\right] - \left(\mathbb{E}\left[\sum_{l=1}^{L} z_{l,t}\right]\right)^2 \\ &= \sum_{l=1}^{L}\sum_{p=1}^{L}\mathbb{E}\left[z_{l,t}z_{p,t}\right].\end{aligned} \tag{32}$$

Since $\mathbb{E}\left[z_{p,t}z_{l,t}\right] = (1-w^{2N_W})w^{2k}\sigma^2\zeta_k$, we have $\mathrm{Var}(z_t) = L^2(1-w^{2N_W})w^{2k}\sigma^2\zeta_k$.

The memory capacity of the parallel ESN model can be given by

$$\begin{aligned}M =& \sum_{k=1}^{\infty}\frac{\mathrm{Cov}^2(m_{t-k},z_t)}{\mathrm{Var}(m_t)Var(z_t)} = \frac{1}{L}\sum_{k=0}^{\infty}\frac{\mathrm{Cov}(m_{t-k},z_t)}{\sigma^2} - \frac{\mathrm{Cov}(m_t,z_t)}{L\sigma^2} \\ =& \frac{1}{L}\sum_{k=0}^{\infty}\sum_{l=1}^{L}(1-w^{2N_W})w^{2k}\zeta_k - \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\zeta_0 \\ =& \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\sum_{k=0}^{\infty}w^{2k}\zeta_k - \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\zeta_0 \\ =& \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\left[\sum_{k=0}^{N_W-1}w^{2k}\zeta_k + \sum_{k=N_W}^{2N_W-1}w^{2k}\zeta_k + \ldots\right] \\ & - \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\zeta_0 \\ =& \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\left(\sum_{k=0}^{N_W-1}w^{2k}\zeta_k\right)\left(\sum_{k=0}^{\infty}w^{2N_W k}\right) \\ & - \frac{1}{L}\sum_{l=1}^{L}(1-w^{2N_W})\zeta_0 \\ =& \frac{1}{L}\sum_{l=1}^{L}\left(\sum_{k=1}^{N_W-1}w^{2k}\zeta_k + w^{2N_W}\zeta_{N_W}\right) \\ =& \frac{1}{L}\sum_{l=1}^{L}\left(\sum_{k=1}^{N_W}w^{2k}\zeta_k\right) \overset{(b)}{=} N_W - 1 + w^{2N_W},\end{aligned} \tag{33}$$

where (a) is obtained from the fact that $\zeta_0 = \zeta_{N_W}$ and (b) stems from the fact that $w_l^{2k}\zeta_k = 1$ as $k = 1,\ldots,N_W-1$ and $w^{2N_W}\zeta_{N_W} = w^{2N_W}$. This completes the proof.

B. Proof of Theorem 2

Let $\boldsymbol{m}_{\ldots t} = [m_1,\ldots,m_{t-1},m_t]$ be the input steam vector and $z_t^{(l)}$ be the output of ESN model l. Next, we derive the memory capacity of a series ESN model using an enumeration method. First, according to (27)-(32), we have $\mathrm{Cov}(z_t^{(1)},m_{t-k}) = (1-w^{2N_W})w^{2k}\sigma^2\zeta_k$. Given the output of the first ESN model, $z_t^{(1)}$, which is the input of the second ESN model, then $\mathrm{Cov}(z_t^{(2)},m_{t-k}) = \left(1-w^{2N_W}\right)^2 w^{2k}\zeta_k\sigma^2$. Similarly, we can obtain that $\mathrm{Cov}(z_t^{(3)},m_{t-k}) = \left(1-w^{2N_W}\right)^3 w^{2k}\zeta_k\sigma^2$. Therefore, we can conclude that $\mathrm{Cov}(z_t^{(L)},m_{t-k}) = \left(1-w^{2N_W}\right)^L w^{2k}\zeta_k$. Based on (34), the memory capacity of a series ESN can be given by $\sum_{k=0}^{\infty}\left(1-w^{2N_W}\right)^L w^{2k}\zeta_k - \left(1-w^{2N_W}\right)^L\zeta_0$. Then, the memory capacity of a series ESN is

$$\begin{aligned}M =& \sum_{k=0}^{\infty}\left(1-w^{2N_W}\right)^L w^{2k}\zeta_k - \left(1-w^{2N_W}\right)^L\zeta_0 \\ =& \left(1-w^{2N_W}\right)^L\sum_{k=0}^{\infty}w^{2k}\zeta_k - \left(1-w^{2N_W}\right)^L\zeta_0 \\ =& \left(1-w^{2N_W}\right)^L\left[\sum_{k=0}^{N_W-1}w^{2k}\zeta_k + \sum_{k=N_W}^{2N_W-1}w^{2k}\zeta_k + \ldots\right] \\ & - \left(1-w^{2N_W}\right)^L\zeta_0 \\ =& \left(1-w^{2N_W}\right)^L\left(\sum_{k=0}^{N_W-1}w^{2k}\zeta_k\right)\left(\sum_{k=0}^{\infty}w^{2N_W k}\right) \\ & - \left(1-w^{2N_W}\right)^L\zeta_0 \\ =& \left(1-w^{2N_W}\right)^L\left(\frac{\sum_{k=0}^{N_W-1}\left(w^{2k}\zeta_k\right)^L}{1-w^{2N_W}} - \frac{\zeta_0-\zeta_0 w^{2N_W}}{1-w^{2N_W}}\right) \\ =& \left(1-w_l^{2N_W}\right)^{L-1}\left(N_W - 1 + w^{2N_W}\right).\end{aligned} \tag{34}$$

This completes the proof.

C. Proof of Theorem 3

The memory capacity of a single ESN with multiple inputs is derived using an enumeration method. Consider $K=2$, then the input stream will be $\boldsymbol{m}_{\ldots t} = [\boldsymbol{m}_{\ldots 1t},\boldsymbol{m}_{\ldots 2t}]$ where $\boldsymbol{m}_{\ldots kt} = m_{k1}\ldots m_{kt-2}m_{kt-1}m_{kt}$. Based on the proof of Theorems 1 and 2, $\boldsymbol{R}$ can be given by:

$$\begin{aligned}\boldsymbol{R} &= \frac{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2}{1-w^{2N_W}}\boldsymbol{V}^{\mathrm{T}}\Gamma^2\boldsymbol{V} \\ &= \frac{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2}{1-w^{2N_W}}\boldsymbol{A},\end{aligned} \tag{35}$$

and $\boldsymbol{p}_k = w^k\left(\sigma_1^2+\sigma_2^2\right)\mathrm{rot}_k\left(\boldsymbol{V}_{1\ldots N}\right)$. Then the output weight matrix can be given by:

$$\boldsymbol{W}^{\text{out}} = \frac{\left(1-w^{2N_W}\right)w^k\left(\sigma_1^2+\sigma_2^2\right)}{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2}\boldsymbol{A}^{-1}\mathrm{rot}_k\left(\boldsymbol{V}_{1\ldots N}\right). \tag{36}$$

The output at time t is

$$\begin{aligned}z_t &= \left(\boldsymbol{m}_t\right)^{\mathrm{T}}\boldsymbol{W}^{\text{out}} \\ &= \frac{\left(1-w^{2N_W}\right)w^k\left(\sigma_1^2+\sigma_2^2\right)}{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2}\left(\boldsymbol{m}_t\right)^{\mathrm{T}}\boldsymbol{A}^{-1}\mathrm{rot}_k\left(\boldsymbol{V}_{1\ldots N}\right).\end{aligned} \tag{37}$$

The covariance of the output at time t and $t-k$ is given by:

$$\begin{aligned}\mathrm{Cov}\left(z_t,\boldsymbol{m}_{t-k}\right) &= \left(1-w^{2N_W}\right)w^k\frac{\sigma_1^2+\sigma_2^2}{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2} \\ &\quad\times\mathrm{Cov}\left(\left(\boldsymbol{\mu}_t\right)^{\mathrm{T}},\boldsymbol{m}_{t-k}\right)\times\boldsymbol{A}^{-1}\mathrm{rot}_k\left(\boldsymbol{V}_{1\ldots N}\right) \\ &= \left(1-w^{2N_W}\right)w^{2k}\frac{\left(\sigma_1^2+\sigma_2^2\right)^2}{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+\sigma_2^2}\zeta_k.\end{aligned}$$

Based on (32), $\mathrm{Var}\left(z_t\right) = \mathrm{Cov}\left(z_t,\boldsymbol{m}_{t-k}\right)$. Then, the memory capacity can be given by

$$
\begin{aligned}
M =& \sum_{k=0}^{\infty} \frac{\mathrm{Cov}^2\left(z_t, \boldsymbol{m}_{t-k}\right)}{\mathrm{Var}\left(\boldsymbol{m}_{t-k}\right)\mathrm{Var}\left(z_t\right)} - \frac{\mathrm{Cov}^2\left(z_t, \boldsymbol{m}_{t}\right)}{\mathrm{Var}\left(\boldsymbol{m}_{t}\right)\mathrm{Var}\left(z_t\right)} \\
=& \sum_{k=0}^{\infty} \frac{\mathrm{Cov}\left(z_t, \boldsymbol{m}_{t-k}\right)}{\sigma_1^2 + 2\rho_{12}\sigma_1\sigma_2 + \sigma_2^2} - \frac{\mathrm{Cov}\left(z_t, \boldsymbol{m}_{t}\right)}{\sigma_1^2 + 2\rho_{12}\sigma_1\sigma_2 + \sigma_2^2} \\
=& \left(\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + 2\rho_{12}\sigma_1\sigma_2 + \sigma_2^2}\right)^2 \times \left(1 - w^{2N_W}\right) \\
& \times \sum_{k=0}^{N_W-1} w^{2k}\zeta_k \sum_{j=0}^{\infty} r^{2N_W j} - \left(1 - w^{2N_W}\right) \\
=& \left(\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + 2\rho_{12}\sigma_1\sigma_2 + \sigma_2^2}\right)^2 \sum_{k=0}^{N_W-1} w^{2k}\zeta_k - (1 - w^{2N_W}) \\
=& \left(\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + 2\rho_{12}\sigma_1\sigma_2 + \sigma_2^2}\right)^2 \left(N_W - 1 + w^{2N_W}\right).
\end{aligned} \tag{38}
$$

Similarly, we can formulate the memory capacity of the single ESN with input vector $\boldsymbol{m}_t = [\boldsymbol{m}_{1t}, \boldsymbol{m}_{2t}, \boldsymbol{m}_{3t}]$, which is given by $\left(\frac{\sigma_1^2+\sigma_2^2+\sigma_3^2}{\sigma_1^2+2\rho_{12}\sigma_1\sigma_2+2\rho_{13}\sigma_1\sigma_3+2\rho_{23}\sigma_2\sigma_3+\sigma_2^2+\sigma_3^2}\right)^2 \left(N_W - 1 + w^{2N_W}\right)$. In consequence, the memory capacity of a single ESN with input vector $\boldsymbol{m}_t = [\boldsymbol{m}_{1t}, \ldots, \boldsymbol{m}_{Kt}]$ can be given by $\left(\frac{\sum_{l=1}^{K}\sigma_l^2}{\sum_{k=1}^{K}\sum_{n=1}^{K}\rho_{kn}\sigma_k\sigma_n}\right)^2 \left(N_W - 1 + w^{2N_W}\right)$. This completes the proof.

REFERENCES

[1] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Minimization of breaks in presence in wireless VR networks: An ESN-based federated learning approach," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, December 2019.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, to appear, 2019.

[3] E. Baştuğ, M. Bennis, M. Médard, and M. Debbah, "Towards interconnected virtual reality: Opportunities, challenges and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, Jan. 2017.

[4] X. Ge, L. Pan, Q. Li, G. Mao, and S. Tu, "Multipath cooperative communications networks for augmented and virtual reality transmission," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2345–2358, Oct 2017.

[5] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communication, computing and caching for mobile VR delivery: Modeling and trade-off," *arXiv preprint arXiv:1804.10335*, April 2018.

[6] J. Park and M. Bennis, "URLLC-eMBB slicing to support VR multimodal perceptions over wireless cellular systems," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec 2018.

[7] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, and Y. Zhao, "Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff," *IEEE Access*, vol. 6, pp. 16665–16677, March 2018.

[8] A. Taleb Zadeh Kasgari, W. Saad, and M. Debbah, "Human-in-the-loop wireless communications: Machine learning and brain-aware resource management," *IEEE Transactions on Communications*, to appear, 2019.

[9] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord," in *Proc. of IEEE Wireless Communications and Networking Conference*, Barcelona, Spain, April 2018.

[10] W. C. Lo, C. L. Fan, S. C. Yen, and C. H. Hsu, "Performance measurements of 360 video streaming to head-mounted displays over live 4G cellular networks," in *Proc. of Asia-Pacific Network Operations and Management Symposium*, Seoul, South Korea, Sept 2017.

[11] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[12] O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Integrated millimeter wave and sub-6 GHz wireless networks: A roadmap for joint mobile broadband and ultra-reliable low-latency communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 109–115, April 2019.

[13] J. Yin, L. Li, H. Zhang, X. Li, A. Gao, and Z. Han, "A prediction-based coordination caching scheme for content centric networking," in *Proc. of Wireless and Optical Communication Conference*, Hualien, Taiwan, April 2018.

[14] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5435–5444, June 2018.

[15] N. T. Nguyen, Y. Wang, H. Li, X. Liu, and Z. Han, "Extracting typical users' moving patterns using deep learning," in *Proc. of IEEE Global Communications Conference*, Anaheim, CA, USA, Dec 2012.

[16] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas on Communications (JSAC)*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[17] O. Esrafilian, R. Gangula, and D. Gesbert, "Learning to communicate in UAV-aided wireless networks: Map-based approaches," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1791–1802, April 2019.

[18] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[19] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[20] J. Konečnỳ, H. B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[21] S. Samarakoon, M. Bennis, W. Saady, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *arXiv preprint arXiv:1807.08127*, 2018.

[22] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys Tutorials*, to appear, 2019.

[23] O. Semiari, W. Saad, and M. Bennis, "Joint millimeter wave and microwave resources allocation in cellular networks with dual-mode base stations," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4802–4816, July 2017.

[24] Q. Li, M. Yu, A. Pandharipande, X. Ge, J. Zhang, and J. Zhang, "Performance of virtual full-duplex relaying on cooperative multi-path relay channels," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3628–3642, May 2016.

[25] Q. Li, M. Yu, A. Pandharipande, and X. Ge, "Outage analysis of co-operative two-path relay channels," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3157–3169, May 2016.

[26] HTC, "HTC vive," https://www.vive.com/us/.

[27] Oculus, "Mobile VR media overview," https://www.oculus.com/.

[28] O. Semiari, W. Saad, M. Bennis, and Z. Dawy, "Inter-operator resource management for millimeter wave multi-hop backhaul networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5258–5272, Aug 2017.

[29] K. Venugopal, M. C. Valenti, and R. W. Heath, "Device-to-device millimeter wave communications: Interference, coverage, rate, and finite topologies," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6175–6188, Sep. 2016.

[30] J. Jerald, *The VR book: Human-centered design for virtual reality*, Morgan & Claypool, Sept. 2015.

[31] J. Chung, H. J. Yoon, and H. J. Gardner, "Analysis of break in presence during game play using a linear mixed model," *ETRI journal*, vol. 32, no. 5, pp. 687–694, Oct. 2010.

[32] M. M. Amiri and D. Gunduz, "Computation scheduling for distributed machine learning with straggling workers," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

[33] V. Smith, C. K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. of Advances in Neural Information Processing Systems*, Long beach, CA,USA, Dec. 2017.

[34] S. Scardapane, D. Wang, and M. Panella, "A decentralized training algorithm for echo state networks in distributed big data applications," *Neural Networks*, vol. 78, pp. 65–74, June 2016.

[35] X. Liu, M. Chen, C. Yin, and W. Saad, "Analysis of memory capacity for deep echo state networks," in *Proc. of IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, USA, Dec 2018.

[36] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3520–3535, June 2017.

[37] M. Bennis and D. Niyato, "A Q-learning based approach to interference avoidance in self-organized femtocell networks," in *Proc. of IEEE Global Communications Conference Workshops*, Miami, FL, USA, Dec 2010.