Machine Learning Based mmWave Channel Tracking in Vehicular Scenario

Yiqun Guo†, Zihuan Wang†, Ming Li†*, and Qian Liu‡
†School of Information and Communication Engineering
Dalian University of Technology, Dalian, Liaoning 116024, China
E-mail: {guoyiqun, wangzihuan}@mail.dlut.edu.cn, mli@dlut.edu.cn
‡ School of Computer Science and Technology
Dalian University of Technology, Dalian, Liaoning 116024, China
E-mail: qianliu@dlut.edu.cn

***Abstract*—Millimeter wave (mmWave) communication has become a key enabling technology for 5G and beyond networks because of its large bandwidth and high transmission rate. In a vehicular mmWave system, beam tracking is a challenging task due to the user's fast mobility and narrow beam of mmWave transmission. In this paper, we study the intelligent beam tracking scheme with low training overhead for mmWave vehicular transmission. Specifically, we utilize the past channel state information (CSI) to efficiently predict the future channel by designing a machine learning prediction model. Using such predicted CSI, the base stations (BSs) reduce the number of channel estimations and save the overhead of pilots. We build the prediction model based on a long short term memory (LSTM) structure whose dataset is composed of the channel vectors of each coherence time duration. The experiments show that the proposed LSTM can accurately predict the channel of the vehicular user and achieve satisfactory transmission rate with less pilot overhead than that of traditional beam training scheme.**

***Index Terms*—Millimeter wave (mmWave) communications, channel tracking, machine learning, vehicular scenerio.**

I. Introduction

Millimeter wave (mmWave) communication, operating in frequency bands of 30-300 GHz, is a promising technology for 5G and beyond cellular systems because of its wide frequency bands [1]. Due to the serious path loss in the mmWave transmission process, large antenna arrays are usually used to transmit and/or receive the mmWave signals [2]. Thanks to the small wavelength of mmWave signals, a large antenna array of massive multiple-input multiple-output (MIMO) systems can be integrated in a small device [3]. A novel hardware-efficient hybrid precoding/combining architecture is introduced which only employs a limited number of simple phase over-samplers (POSs) and a switch (SW) network to achieve maximum hardware efficiency [4]. Thus, the large antenna array can provide sufficient beamforming gain with precoding and combining techniques to overcome severe free-space path loss of mmWave channels.

Generally, the beam generated by a large antenna array is very narrow, which means mmWave communications require much more precise beam alignments of users compared to the conventional sub-6 GHz communication systems [5]. In order to reduce the training overhead and establish stable link with mobile users, channel/beam tracking is proposed as an promising approach for mmWave vehicular systems.

Traditionally, the base station (BS) would implement channel tracking through prediction. Since the channel changing in adjacent time can be regarded as a first-order Markov chain, the Kalman filter algorithm can be utilized to track the time-varying channel [6]. In [7], channel tracking is realized by separately obtaining the information of angle of arrival (AoA) and the channel gain. The AoA information is acquired by a modified unscented Kalman filter and the gain information is estimated through beam training. In [8], a high-resolution angle tracking algorithm utilizing auxiliary beam pair is designed for mobile wideband mmWave systems with analog beamforming architecture. Experiments in [9] have shown that multiple BSs cooperation can overcome the problem of obstacle and establish stable link with mobile users in vehicular systems. Unfortunately, the aforementioned beam tracking schemes still lead to unaffordable pilot overhead due to the fast variation of mmWave channels in vehicular scenario.

In this paper, we investigate the intelligent channel trancking scheme with low training overhead in mmWave communication system. We consider the coordinated multiple BSs transmission, in which several BSs simultaneously serve one vehicular user to provide the stable communication link. Based on the cloud/centralized radio access network (C-RAN), we propose a machine learning model which learns how to predict the channel state information (CSI) based on the signals received at the distributed BSs in uplink training process. In particular, the proposed machine learning based channel tracking consists of online training and CSI prediction. A long short term memory (LSTM) model is built, which employs the past CSI to promote the prediction of the user's channel. The simulation results show that the proposed LSTM model can accurately predict the channel vector. Then, the beamformer can be obtained based on the predicted channel, which is testified to achieve satisfactory transmission quality with significantly reduced pilot overhead than that of the traditional

This paper is supported by the National Natural Science Foundation of China (Grant No. 61671101, 61601080, and 61761136019).
*Corresponding author.

978-1-7281-2373-8/19/$31.00 ©2019 IEEE

beam training scheme.

II. System Model and Problem Statement

A. System Model

We consider a mmWave multiple-input single-output (MISO) system, where a mobile user is served by N coordinated BSs, each of which is equipped with M antennas and a single radio frequency (RF) chain, as illustrated in Fig. 1. For simplicity, The mobile user is equipped with only one antenna.

In the uplink training stage, the BSs apply analog-only beamforming $\mathbf{f}_n, n = 1, \ldots, N$, via a network of phase shifters. Let $\mathbf{h}_n \in \mathbb{C}^{M\times 1}$ denote the uplink channel vector between the user and the n-th BS, the combined received signal at BS n can then be expressed as

$$y_n = \sqrt{P}\mathbf{f}_n^H \mathbf{h}_n s + \mathbf{f}_n^H \mathbf{v}_n, \tag{1}$$

where P represents the received power, s is the transmit signal, $\mathbb{E}\{|s|^2\}$=1, and $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_M)$ is the Gaussian noise corrupting the received signal.

A narrowband mmWave channel model with a line-of-sight (LoS) path and L non-line-of-signt (NLoS) paths is considered in our mmWave system. The channel between the BS and the user can be represented by the commonly adopted Saleh-Valenzuela channel model. When uniform planar array (UPA) is applied at the BSs, the channel vector $\mathbf{h}_n$ from the n-BS to the user is expressed as

$$\mathbf{h}_n = \sqrt{\frac{MK}{1+K}}\rho_0 \mathbf{a}(\theta_0, \phi_0) + \sqrt{\frac{M}{L(1+K)}}\sum_{l=1}^{L}\rho_l \mathbf{a}(\theta_l, \phi_l), \tag{2}$$

where K is the Ricean K-factor, $\rho_l \sim \mathcal{CN}(0,1)$, $l = 0, 1, \ldots, L$, is the complex channel gain of the LoS and NLoS paths, L is the number of NLoS paths. θ_l, ϕ_l, $l = 0, 1, \ldots, L$, represent the azimuth and elevation angles, respectively. Finally, $\mathbf{a}(\theta, \phi)$ is the array response vector, which is given by

$$\mathbf{a}(\theta, \phi) \triangleq \mathbf{a}_h(\theta, \phi) \otimes \mathbf{a}_v(\theta, \phi), \tag{3}$$

where $\otimes$ is Kronecker product, the array response vectors $\mathbf{a}_h(\theta, \phi)$ and $\mathbf{a}_v(\theta, \phi)$ are represented as

$$\begin{aligned}\mathbf{a}_h(\theta, \phi) &= \frac{1}{\sqrt{M}}[1, e^{j\frac{2\pi}{\lambda}d\sin(\theta)\cos(\phi)}, \ldots, e^{j(M-1)\frac{2\pi}{\lambda}d\sin(\theta)\cos(\phi)}]^T,\\ \mathbf{a}_v(\theta, \phi) &= \frac{1}{\sqrt{M}}[1, e^{j\frac{2\pi}{\lambda}d\sin(\phi)}, \ldots, e^{j(M-1)\frac{2\pi}{\lambda}d\sin(\phi)}]^T.\end{aligned} \tag{4}$$

According the duality of the uplink and downlink channels, the downlink data transmission can be represented as

$$\hat{s} = \sum_{n=1}^{N}\sqrt{P_n}\mathbf{h}_n^H \mathbf{f}_n s_n + z, \tag{5}$$

where P_n is the transmit power of the n-th BS, and s_n is the transmit signal, $\mathbb{E}\{|s_n|^2\} = 1$, $z \sim \mathcal{CN}(0, \sigma^2)$ represents the noise in downlink communication.

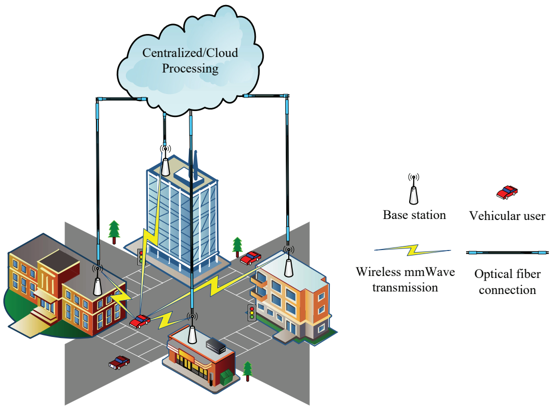

Fig. 1. A diagram of the proposed mmWave communication system where N BSs are installed in a street and simultaneously receive uplink training signals from one vehicular user. Each BS is equipped with M antennas and one RF chain, and is applying analog-only beamforming/combining strategy. The coordinated BSs provide sufficient channel gain for vehicle mobile users.

B. C-RAN Based Millimeter Wave System

Due to the large path loss in the mmWave communication, the service range of the mmWave BS is smaller than that of the 4G BS, which leads to the dense coverage of the mmWave BSs. In such cases, the multi-connection structure becomes the main way to establish links between the BSs and the mobile vehicular user. Specifically, when the vehicular user moves in the edge zone of the BSs service, the user is simultaneously served by the N BSs existing around. When the vehicular user moves in the center zone of the BS service, the user is served by a single BS. This structure not only satisfies the user's need for stable transmission rate during the movement, but also saves resources for BSs.

C-RAN is a green wireless access network architecture, which allows centralized processing, collaborative radio and real-time cloud calculation to realize a high reliability, low latency network. In this way, C-RAN enables multiple distributed BSs to share the resource processing with each other, and provides stable communication quality for mobile users. We mainly study the beam tracking when vehicular users move in the edge area of BSs service, and our designed C-RAN-based mmWave system is composed of one cloud and N coordinated BSs, as shown in Fig. 1.

Under the C-RAN structure, the received information at each BS is delivered to the central cloud processor through optical fiber, as illustrated in Fig. 1. In particular, each BS would obtain the CSI of vehicular user through traditional channel estimation approach [1], and the estimated channel vector is given by $\tilde{\mathbf{h}}_n$, $n = 1, \ldots, N$. All the BSs send the obtained channel vectors to the central cloud for further processing. The cloud would combine the channel vectors and obtain the integrated channel $\tilde{\mathbf{h}}$:

$$\tilde{\mathbf{h}} = \left[\tilde{\mathbf{h}}_1^T, \ldots, \tilde{\mathbf{h}}_N^T\right]^T. \tag{6}$$

Then, the machine learning model can be efficiently accomplished in the cloud based on these data. Note that the integrated channel vector $\tilde{\mathbf{h}}$ means that we only need to allocate one memory cell for a machine learning model instead of N memory cells for N BSs. In this way, the processing for one mobile user can be more concentrated, improving the efficiency of the mmWave communication.

C. Problem Statement

In this paper, we study the mmWave channel tracking in a vehicular scenario. We aim to efficiently predict the CSI of the mobile user based on the machine learning approach to reduce the training overhead while maintain the tracking accuracy. Once the machine learning model predicts the CSI, the cloud will feed back the predicted channel vectors to all BSs and the BSs calculate the optimal beamformer for downlink transmission. Specifically, the beamformer design or adjustment for the n-th BS, $n = 1, \ldots, N$, is accomplished through a codebook based approach. Let $\mathcal{F}$ denote the codebook consisting of candidate beamformers, which is defined as

$$\mathcal{F} = \left\{ \mathbf{a}\left(\frac{2\pi i}{2^B}, \frac{\pi j}{2^B}\right) : i, j = 1, \ldots, 2^B \right\}, \tag{7}$$

where B indicates the number of bits to quantize the AoAs. Then, the beamforming vector $\mathbf{f}_n$ for the n-th BS is chosen from $\mathcal{F}$ to maximize the downlink channel gain:

$$\begin{aligned} \mathbf{f}_n^\star = \arg \ \ &\max \ \left| \tilde{\mathbf{h}}_n^H \mathbf{f}_n \right|^2 \\ \text{s.t.} \ \ &\mathbf{f}_n \in \mathcal{F}, n = 1, \ldots, N, \end{aligned} \tag{8}$$

$\tilde{\mathbf{h}}_n$ is the predicted channel vector for the n-th BS. Finally, the achievable downlink sum-rate is

$$R = \log_2 \left(1 + \frac{\left| \sum_{n=1}^{N} \tilde{\mathbf{h}}_n^H \mathbf{f}_n \right|^2}{\sigma^2} \right). \tag{9}$$

The above procedure is implemented in the cloud. After the calculation is completed, the cloud transmits the index of the selected beamformer to each BS through the optical fiber. According to the beamformer index, each BS generates the beamforming vector $\mathbf{f}_n$, $n = 1, \ldots, N$, and adjusts the beam direction to the user's predicted position for downlink transmission.

III. Machine Learning Based Channel Tracking

Machine learning has attracted considerable attention in recent years due to its intelligent ability to accurately accomplish identification and prediction. Besides, machine learning is a key technique to deal with big data and thus can promote the channel estimation/tracking for mmWave communications.

In this section, we propose a machine learning based mmWave channel tracking scheme. In order to reduce the training overhead, we build a LSTM model, which utilize the past CSI to facilitate the channel prediction. In the following subsections, we first explain the main idea of the proposed machine learning based mmWave channel tracking. Then, we provide detailed operation of the learning and prediction phases followed by the LTSM modeling description.

A. The Main Idea

Note that the beam/channel tracking can be regarded as a function of surroundings and the channel variations may have a regular change with the mobility. In this paper, we propose to utilize machine learning to establish a connection between the surroundings and the channel tracking, and accurately predict the channel vector based on the past CSI. In our proposed approach, the BSs will rely on the predicted channel obtained from cloud and do not have to perform channel estimation. Therefore, the training overhead is reduced and the time spent on channel estimation can be saved to transfer information.

Specifically, the proposed machine learning based channel tracking contains two stages, i.e. online training stage and channel prediction stage. In the online training stage, the machine learning model studies the relationship between the moving trail and the surroundings based on the uplink training pilots. When the learning stage completes, the system moves into the prediction stage, and the prediction result derived from the machine learning model is used to design the beamforming vector for each BS. In order to reduce the training overhead, we adopt LSTM architecture which can learn from experience. The LSTM structure can precisely predict the user's channel based on the current input (current channel vector) and the hidden states of LSTM model, and thus the BSs only need to implement uplink training every other time slot. Next, we will present the detailed operation of the two-stage machine learning and the LSTM architecture.

B. Channel Tracking Operation

In this subsection, we introduce the two stages of the channel tracking operation in details, the procedure is shown in Fig. 2.

Stage 1. Online Training: In this stage, the vehicular user sends uplink pilot signals to the surrounding BSs, and each BS estimates the uplink channel vector according to the received signals by using traditional channel estimation methods [1]. Then, each BS will determine the optimal beamformer for downlink transmission based on criterion (8). Meanwhile, all the BSs will send the estimated results $\tilde{\mathbf{h}}_n$, $n = 1, \ldots, N$, to the center cloud for environment learning. In the cloud, the estimated channel vectors from different BSs are integrated to form the final estimated channel vector $\tilde{\mathbf{h}}$, as demonstrated in Sec. II. Finally, the neural network of LSTM model is trained based on the dataset $\tilde{\mathbf{h}}$. After the training of LSTM completes, the tracking system will move into the channel prediction stage.

Stage 2. Channel Prediction: In this stage, the trained LSTM model will predict the CSI based on the channel vector of the last coherence time, and there is no need to implement additional uplink training at the BSs. The block diagram of the channel prediction stage is depicted in Fig. 3, in which t indicates the time slot. Once the LSTM model predicts the next time slot channel vector $\hat{\mathbf{h}}^{(t+1)}$, the cloud will obtain

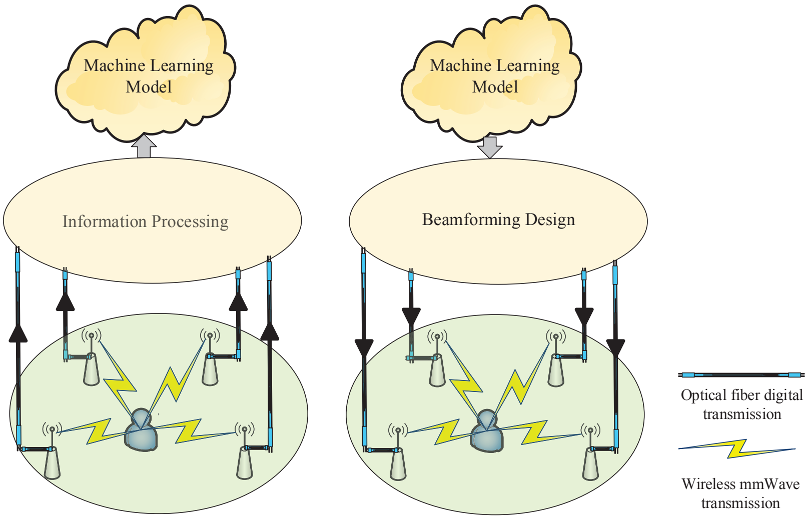

Fig. 2. The machine learning based mmWave channel tracking consists of two stages. During the online training, the learning model will learn the estimated channel vectors received from the N BSs through optical fiber. In the second stage, the cloud will calculate the predicted channels and feed them back to the BSs for downlink beamformer design.

the corresponding channel vectors for each BS and feed them back for downlink beamformer design. Note that we utilize $\hat{\mathbf{h}}$ and $\tilde{\mathbf{h}}$ to indicate the predicted channel by cloud and the estimated channel by BSs, respectively.

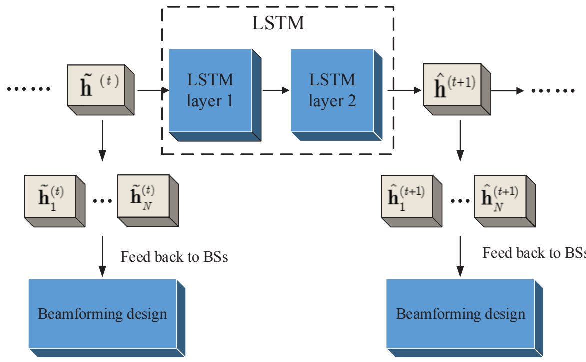

Fig. 3. The structure of data stream. The datasets $\tilde{\mathbf{h}}$ are well trained by a two-layer LSTM model and the output of the model is the predicted channel vector $\hat{\mathbf{h}}$. The predicted channel vector then is considered to design the predicted beamforming. The final output of the cloud is the predicted beamforming vectors for all the BSs.

C. LSTM Model

LSTM has the ability to handle the problem with long-term memory and it is an effective approach for mmWave channel prediction. In this subsection, we describe the training of the LSTM model.

The LSTM model introduces an intermediary sort of memory via the memory cell, and the memory cell includes a node with a self-connected recurrent edge of weight 1, ensuring that the gradient can pass without exploding and vanishing. Compared to recurrent neural network (RNN), which has only one transfer state $\mathbf{h}_t$, LSTM has two transfer states, one $\mathbf{c}_t$ (cell state), and one $\mathbf{h}_t$ (hidden state). Multiplicative gates are distinctive features of the LSTM model, which are an input gate $\mathbf{i}_t$, an output gate $\mathbf{o}_t$, and a forget gate $\mathbf{k}_t$. If the gate outputs 0, information through the gate is cut off. If the gate outputs 1, all messages are passed through the gate.

The block diagram of the LSTM model is illustrated in Fig. 4. During the time slot t, the inputs of the LSTM model are $\mathbf{x}_t$ and the hidden state of the last time slot $\mathbf{h}_{t-1}$. After multiplicative gates calculate all the outputs, $\mathbf{h}_t$ and $\mathbf{c}_t$ are updated according to the criterion (10)-(15). The hidden state $\mathbf{h}_t$ then feeds into the LSTM model at the next time step as well as the cell state $\mathbf{c}_t$. Learning is accomplished by iteratively updating each of the weights to minimize a loss function, $\mathcal{L}(\mathbf{y}_t, \bar{\mathbf{y}}_t)$, which penalizes the distance between the output $\mathbf{y}_t$ and the target $\bar{\mathbf{y}}_t$. It is worth noting that computation in the LSTM model proceeds according to the following calculations which must be evaluated at each time step, which are

$$\tilde{\mathbf{c}}_t = \phi(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1} + \mathbf{b}_c), \tag{10}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{11}$$

$$\mathbf{k}_t = \sigma(\mathbf{W}_{kx}\mathbf{x}_t + \mathbf{W}_{kh}\mathbf{h}_{t-1} + \mathbf{b}_k), \tag{12}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{13}$$

$$\mathbf{c}_t = \tilde{\mathbf{c}}_t \otimes \mathbf{i}_t + \mathbf{c}_{t-1} \otimes \mathbf{k}_t, \tag{14}$$

$$\mathbf{h}_t = \tanh(\mathbf{c}_t) \otimes \mathbf{o}_t, \tag{15}$$

$$\mathbf{y}_t = \sigma(\mathbf{W}_h\mathbf{h}_t), \tag{16}$$

where $\mathbf{W}_{cx}$, $\mathbf{W}_{ch}$, $\mathbf{W}_{ix}$, $\mathbf{W}_{ih}$, $\mathbf{W}_{kx}$, $\mathbf{W}_{kh}$, $\mathbf{W}_{ox}$, $\mathbf{W}_{oh}$ are weighted matrices, $\tilde{\mathbf{c}}_t$ is an intermediate vector for cell state $\mathbf{c}_t$, $\mathbf{b}_k, \mathbf{b}_i, \mathbf{b}_c, \mathbf{b}_o$ are biases of LSTM units, $\sigma(\cdot)$ and $\phi(\cdot)$ are the sigmoid functions for each gate, and finally, $\otimes$ represents the element-wise multiplication.

For our considered channel tracking system, the current channel estimation result at BSs $\tilde{\mathbf{h}}^{(t)}$ is the input of the LSTM model and the next time slot estimated channel $\tilde{\mathbf{h}}^{(t+1)}$ is the desired output of current time, which correspond to $\mathbf{x}_t$ and the desired output $\bar{\mathbf{y}}_t$ in the LSTM model, respectively. In this training procedure, some of the useless information in the past will be discarded and the prediction results will be continuously improved based on advanced memory. After the training, the output of the LSTM is the predicted channel vector $\hat{\mathbf{h}}^{(t+1)}$, and the difference between it and the actual channel vector at the next time $\mathbf{h}^{(t+1)}$ is negligible.

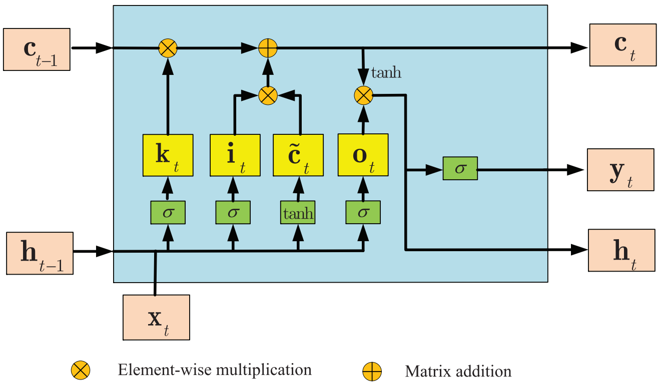

Fig. 4. The structure of LSTM. We adopt the most classic LSTM model, consisting of an input gate $\mathbf{i}_t$, an output gate $\mathbf{o}_t$, a forget gate $\mathbf{k}_t$ and a memory cell $\mathbf{c}_t$. $\mathbf{h}_{t-1}$ is the hidden state of the previous LSTM model, $\mathbf{x}_t$ is the input of the current moment, and $\mathbf{y}_t$ is the output of the current moment.

IV. Simulation Results

In this section, we first describe the process of building the simulation environment, including system and channel models, dataset generation, LSTM parameters, and simulation results. The simulations are based on the commercial simulator Wireless Insite [10], which is a widely used ray-tracing simulator. The system model and the channel model can be derived in Section II, and the channel vector can be constructed by using the parameters generated from Wireless Insite, such as AoA, AoD, path loss, etc. In Wireless Insite, we set the frequency of the mmWave at 60Ghz, while the 4 BSs are distributed on the top of the building with a height of 50 meters. Each BS is equipped with a UPA antenna with $M = 32$ antennas and the user is equipped with one single antenna. The BSs apply analog-only combining via a network of phase shifters. In order to predict the channel vector of vehicular mobile users, we construct some random routes with moving rates ranging from 10 m/s to 30 m/s.

The specific simulation environment is illustrated in Fig. 5. From Fig. 5, we can see that 4 BSs are placed on different buildings, and they can cover all the user's movements. The green dots represent the movement of the user, and two tracks can be seen in the figure. The training of the LSTM model requires a large amount of data, so there are still many random trajectories in the simulation of the experiment, and these trajectories are invisible in the figure.

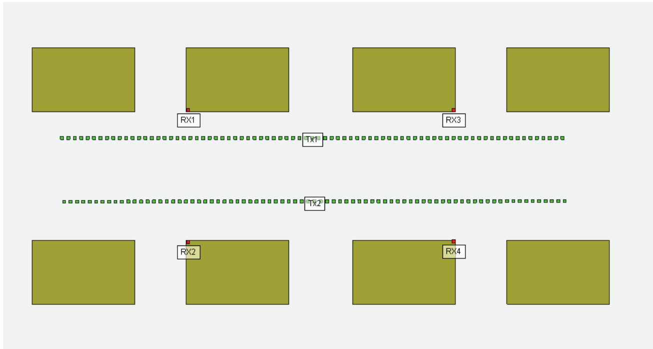

Fig. 5. The simulation environment in Wireless Insite. The transmitters are the green dots, which are randomly distributed to simulate the movement of vehicular user. The receivers are the red dots, which are coordinately designed and built on the top of the buildings. More routes of the transmitters are invisible and these data can be well trained in the LSTM model.

For each BS, we build up the uplink channel vector with the channel model in Section II, which will be sent to the same cloud as the dataset for machine learning model. In the cloud, the channel vectors of all the BSs are combined and the final output is $\tilde{\mathbf{h}}$. For the n-th BS, it is equipped with a UPA antenna array with $M = 32$ antennas. Therefore, with $N = 4$ BSs serving the same user at the same time, the dimension of the integrated channel vector $\tilde{\mathbf{h}}$ equals 128×1.

Before the training of neural network, $\tilde{\mathbf{h}}$ will be normalized by the maximum and the minimum value of the vector. In the LSTM network, the learning rate is set to 0.006, and the batch size is 20. We build our LSTM network in Tensorflow, and the rest of the simulation are implemented on MATLAB.

To evaluate the performance of our proposed machine learning system, we adopt the normalized mean square error (NMSE) to test the difference between the estimated channel vector $\tilde{\mathbf{h}}$ and the predicted channel vector $\hat{\mathbf{h}}$, which is defined as

$$\text{NMSE} = \frac{\|\hat{\mathbf{h}} - \tilde{\mathbf{h}}\|^2}{\|\tilde{\mathbf{h}}\|^2}. \tag{17}$$

In Fig. 6, we consider the NMSE of the channel vector predicted by the machine learning model with different size of UPA antennas. We can observe that our machine learning model will get better prediction results when the number of antennas decreases. The result of this figure shows that our machine learning model is well suited for small size UPA antennas. The reason is that the input of the machine learning model will become larger when the number of antennas getting larger, resulting in more difficulties in training the neural network.

In order to demonstrate that our algorithm can reduce the pilot overhead, we design the following experiment: We introduce the concept of the beam coherence time, which is a recent concept in mmWave communication to represent the average beam training time [9]. A beam coherence time consists of two parts: Uplink training and downlink data transmission. Our algorithm is applied after the BSs estimate the channel vector once. To be more specific, the BSs estimate

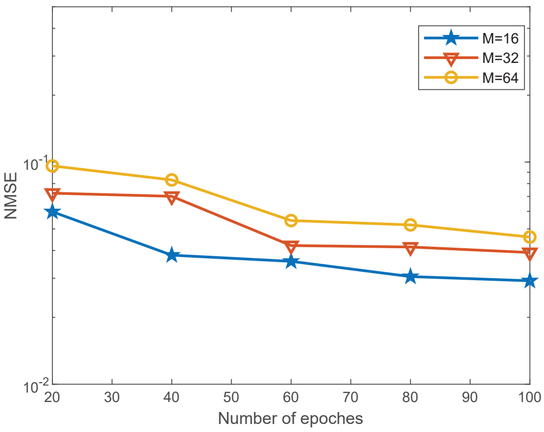

Fig. 6. The NMSE performance of the LSTM model.

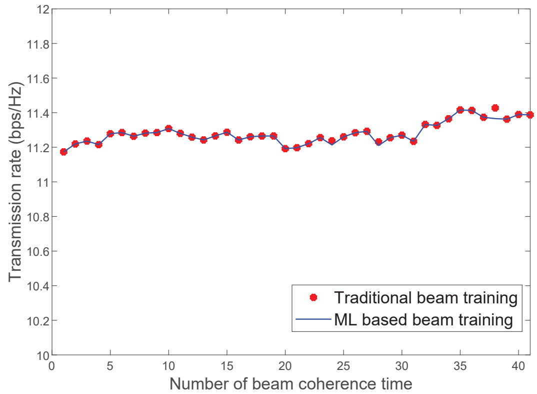

Fig. 7. Examples of how the proposed machine learning reduce the overhead. The red dots represent the transmission rate with traditional beam training algorithm. The blue line represents the transmission rate with machine learning based beam training algorithm. From the figure we can see that the traditional algorithm is slightly higher than our algorithm except for the individual points, and the transmission rates of the two are almost the same at most moments.

the channel vectors using traditional method and then design the beamformer in the first beam coherence time. Instead of estimating the channel vectors, the BSs design the beamformer using our proposed system in the second beam coherence time. Since long-term predictions may be inaccurate, we adopt our algorithm once after a traditional beam coherence time. In this way, we reduce half the overhead of two beam coherence time by making the second beam coherence time no overhead.

Fig. 7 shows that how our proposed system reduces the overhead of mmWave communication, once our machine learning model can accurately predict the channel vector. In Fig. 7, we can see that the transmission rates of the two algorithms are almost the same at most moments. Despite the transmission rates are the same, our algorithm can reduce the overhead by half. In the vehicle scenerio, the system we designed guarantees the same transmission rate as traditional algorithm, while reducing half the overhead of traditional algorithms.

V. Conclusions

In this paper, we investigated a intelligent channel tracking scheme with low training overhead for mmWave vehicular transmission. We proposed a machine learning based channel tracking algorithm which utilizes the past CSI to efficiently predict the future channel of the vehicular user. In particular, the proposed machine learning based channel tracking consisted of two stages, i.e. online training stage and CSI prediction stage. In order to accurately obtain the predicted channel, we built the prediction model based on a LSTM structure. The LSTM model was trained by the current estimated channel as well as the past experience to continuously improve the prediction results. The experiments demonstrated that the proposed LSTM can accurately predict the CSI of the vehicular user with less pilot overhead than that of traditional beam training scheme.

References

[1] Z. Pi and F. Khan, “An introduction to millimeter-wave mobile broadband systems,” *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101-107, June 2011.

[2] T. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi and F. Gutierrez, “Millimeter wave mobile communications for 5G cellular: It will work!” *IEEE Access*, vol. 1, pp. 335-349, May 2013.

[3] R. W. Heath Jr., N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave MIMO systems,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436-453, April 2016.

[4] M. Li, Z. Wang, H. Li, Q. Liu and L. Zhou, ”A hardware-efficient hybrid beamforming solution for mmWave MIMO systems,” *IEEE Trans. Wireless Commun.*, vol. 26, no. 1, pp. 137-143, Feb. 2019.

[5] X. Gao, L. Dai, S. Han, C.-L. I, and R. W. Heath Jr., “Energy-efficient hybrid analog and digital precoding for mmWave MIMO systems with large antenna arrays,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 998-1009, April 2016.

[6] O. Mehanna and N. Sidiropoulos, “Channel tracking and transmit beamforming with frugal feedback,” *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6402-6413, Dec. 2014.

[7] J. Zhao and F. Gao, “Angle domain hybrid precoding and channel tracking for millimeter wave massive MIMO systems,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6868-6880, Oct. 2017.

[8] D. Zhu, J. Choi, Q. Cheng, W. Xiao, and R. W. Heath Jr, “High-resolution angle tracking for mobile wideband millimeter-Wave systems with antenna array calibration,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7173-7189, Nov. 2018.

[9] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,” *IEEE Access.*, vol. 6, pp. 37328-37348, June 2018.

[10] Remcom, Wireless insite, http://www.remcom.com/wireless-insite.

[11] F. Sohrabi and W. Yu, “Hybrid digital and analog beamforming design for large-scale antenna arrays,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501-513, April 2016.

[12] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath Jr., “Spatially sparse precoding in millimeter wave MIMO systems,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499-1513, March 2014.

[13] A. Alkhateeb and R. W. Heath Jr., “Frequency selective hybrid precoding for limited feedback millimeter wave systems,” *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 1801-1818, May 2016.

[14] X. Gao, L. Dai, Y. Sun, S. Han, and C.-L. I, “Machine learning inspired energy-efficient hybrid precoding for mmWave massive MIMO systems,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017.

[15] R. Méndez-Rial, C. Rusu, N. González-Prelcic, and A. Alkhateeb, “Hybrid MIMO architectures for millimeter wave communications: Phase shifters fo switches?” *IEEE Access*, vol. 4, pp. 247-267, Jan. 2016.