Abstract—We investigate the viability of using machine-learning techniques for estimating user-channel features at a large-array base station (BS). In the scenario we consider, user-pilot broadcasts are observed and processed by the BS to extract angle-of-arrival (AoA) specific information about propagation-channel features, such as received signal strength and relative path delay. The problem of interest involves using this information to predict the angle-of-departure (AoD) of the dominant propagation paths in the user channels, i.e., channel features not directly observable at the BS. To accomplish this task, the data collected in the same propagation environment are used to train neural networks. Our studies rely on ray-tracing channel data that have been calibrated against measurements from Shinjuku Square, a famous hotspot in Tokyo, Japan. We demonstrate that the observed features at the BS side are correlated with the angular features at the user side. We train neural networks that exploit different combinations of measured features at the BS to infer the unknown parameters at the users. The evaluation based on standard statistical performance metrics suggests that such data-driven methods have the potential to predict unobserved channel features from observed ones.

I. INTRODUCTION

Cellular standardization efforts have mainly focused on enabling wireless communication in the sub-6GHz spectrum. Given the scarcity of spectrum below 6GHz, new efforts in 3GPP are expanding their scope to include spectrum above 6GHz, in particular the millimeter wave (mmWave) frequencies. To enable the deployment of reliable cost-efficient wireless networks operating at mmWave frequencies, however, a number of serious challenges must be addressed [1], [2]. This is because compared to its sub-6GHz counterpart, communication at mmWave is impacted by much larger propagation pathloss, more rapidly changing channels, severe penetration loss, dynamic shadowing, etc. Fortunately, due to shorter wavelengths, much larger arrays can be packed into a small footprint, which enables the use of massive arrays in small-cell BSs and moderately large arrays at the user terminals. Such BS and user-terminal arrays are essential at mmWave as they can be exploited to provide combined transmit-receive (TX-RX) beamforming (BF) gains. Given the harsh propagation conditions at mmWave, such BF gains play a vital role as they can greatly extend the range of mmWave communication, thereby greatly improving coverage.

Learning the user-channels to create beams, however, is more challenging at mmWave. The larger the arrays, the larger the set of beams and TX-RX beam combinations that need to be searched. As channels decorrelate much faster than at sub-6 GHz bands, high-gain TX-RX beam combinations must be learned much faster. This results in substantially larger training overheads at mmWave due to the need for more frequent searches and within a larger space of beam pairs. In real-world cellular environments where most user terminals communicate to BSs through channels with no line-of-sight path, having access to precise geometric information regarding the relative locations of the BS and TX arrays (and their relative orientation) is not sufficient to identify TX-RX beam pairs that yield high BF gains.

There is a large body of works on finding the optimal TX-RX beam pair. In [3], Abari et al. proposed “Agile-Link” to find the optimal beam alignment through employment of carefully concocted hash functions that can quickly identify and remove the direction bins with no energy. In [4], Hur et al. proposed an adaptive beam alignment technique where a hierarchical BF codebook is leveraged in lieu of the exhaustive search, but the required feedback from the receiver to the transmitter incurs additional overhead and latency. An attractive alternative is to estimate the multipath channel parameters [5]–[10]. Due to the sparse structure of mmWave scattering, formulated channel estimation as a sparse signal recovery problem, significantly reducing the training overhead in the process. In addition, low-rank tensor factorization has been exploited in [11]–[12] to further improve estimation accuracy and to reduce the computational cost. Based on several recent empirical studies, Li et al. showed the existence of a joint sparse and low-rank structure in mmWave channels in presence of angular spreads [13], and they leveraged this structure to further reduce the computational cost.

While the aforementioned prior works targeted the channel-parameter prediction problem from a system modeling/signal processing perspective, in this paper we leverage the use of machine learning towards this task, so as to improve the user experience and/or the radio-resource utilization in wireless networks. In particular, we exploit machine learning at the BS to predict user-channel features that are not observable at the BS. Specifically, we consider using AoA dependent propagation-channel features at the BS (extracted using a large array) to predict the AoD of the dominant propagation paths in the user channels. The dominant-path AoD prediction is
performed by neural networks, which are trained by using the
data collected from the same propagation environment. The
data of propagation channels that we use were generated by
a channel tracer (and also calibrated against measurements)
over Shinjuku square, a typical hotspot in Tokyo, Japan.

We recast channel-parameter prediction as a learning-based
optimization problem by first constructing appropriate feature
representations. Then we conduct correlation analysis between
the observed and the unknown features in the model. Next, we
train several neural networks, each using different combina-
tions of observed features to infer the unknown parameters
at the user side. Our preliminary results suggest that machine
learning could prove a valuable tool in allowing big data to
be used for predicting unobserved channel parameters, and
improving network resource utilization and user experience.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section we describe how the channel features that are
provided by the ray-tracing data, are turned into observable
and unobservable channel features at the BS side. In the
process, we will describe the channel models giving rise to
these representations.

We assume half-wavelength spaced uniform linear arrays
(ULAs) at the BS and the mobile station (MS) or the user,
equipped with \( N_{\text{BS}} \) and \( N_{\text{MS}} \) antenna elements, respectively. As introduced in [14], a multipath wireless channel can be
modeled as a linear system with the following \( N_{\text{BS}} \times N_{\text{MS}} \)
time-frequency response matrix:

\[
H(t, f) = \sum_{l=1}^{L} \alpha_l a_{\text{BS}}(\theta_l) a_{\text{MS}}^H(\phi_l) e^{j2\pi f \tau_l},
\]

where \( L \) is the number of multipaths, and for the \( l^{th} \) path, \( \alpha_l \)
denotes the complex channel coefficient, \( \theta_l \in [0, 2\pi) \) and \( \phi_l \in
[0, 2\pi) \) represent the associated AoA and AoD, respectively,
\( \tau_l \) is path delay, and \( a_{\text{BS}} \) and \( a_{\text{MS}} \) are the associated array
steering vectors at the BS and the MS, respectively, which can
be written as

\[
a_{\text{BS}}(\theta_l) = \frac{1}{\sqrt{N_{\text{BS}}}} \begin{bmatrix} 1, e^{j \frac{2\pi}{\lambda} \sin(\theta_l)}, \ldots, e^{j(N_{\text{MS}}-1) \frac{2\pi}{\lambda} \sin(\theta_l)} \end{bmatrix}^T,
\]

\[
a_{\text{MS}}(\phi_l) = \frac{1}{\sqrt{N_{\text{MS}}}} \begin{bmatrix} 1, e^{j \frac{2\pi}{\lambda} \sin(\phi_l)}, \ldots, e^{j(N_{\text{MS}}-1) \frac{2\pi}{\lambda} \sin(\phi_l)} \end{bmatrix}^T,
\]

where \( \lambda \) is the carrier wavelength. Assuming the channel
follows block fading and channeling in a block within coherence time, we can drop the time index \( t \). In this
paper, for simplicity, we only consider the 2-D system where
AoA and AoD are captured by the azimuth components only.

According to (1), \( H(f) \) is determined by the parameters
\( \{\alpha_l, \tau_l, \theta_l, \phi_l\}_{l=1}^{L} \), where each is continuous-valued
in the corresponding field. However, the observation precision
of these parameters is subject to the observation resolution.
Specifically, the observation resolution of \( \theta_l \) and \( \phi_l \) is limited
by the size of the antenna arrays at the BS and the user,
respectively, and that of \( \tau_l \) is determined by the system
bandwidth\(^1\). Hence, after sampling over AoA/AoD domains,
the virtual representation of the channel in [1] is given by [14]:

\[
H(f) = \sum_{l=1}^{L} W_{\text{BS}} H_l^T(l) W_{\text{MS}}^T e^{-j2\pi f \tau_l}, \tag{4}
\]

where \( H_l(l) \) is the \( N_{\text{BS}} \times N_{\text{MS}} \) matrix associated with the \( n_{l}^{th} \)
revolvable AoA and \( n_{l}^{th} \) revolvable AoD of the \( l^{th} \) path
(see equation (4) in [14]); \( W_{\text{BS}} \) and \( W_{\text{MS}} \) are \( N_{\text{BS}} \times N_{\text{BS}} \)
and \( N_{\text{MS}} \times N_{\text{MS}} \) unitary matrices, respectively, which comprise
\( a_{\text{BS}}(n_{l}/N_{\text{BS}}) \) and \( a_{\text{MS}}(n_{l}/N_{\text{MS}}) \) as their \( n_{l}^{th} \) and \( n_{l}^{th} \) columns,
respectively. Both \( W_{\text{BS}} \) and \( W_{\text{MS}} \) turn out to be DFT matrices
with \( N_{\text{BS}} \), \( N_{\text{MS}} \) columns, respectively. Indeed as was shown
in [15] for large ULAs projecting the DFT matrix acts like a
Karhunen-Loeve expansion, as projecting onto the DFT matrix
effectively whitens and sparsifies the channel. Thus, if the BS
and the user both apply their respective \( W_{\text{BS}} \) and \( W_{\text{MS}} \) as the
BF matrices, then we can obtain the effective channel:

\[
H_{\text{eff}}(f) = W_{\text{BS}}^H H(f) W_{\text{MS}} = \sum_{l=1}^{L} H_l^T(l) e^{-j2\pi f \tau_l}. \tag{5}
\]

Inspection of (5) reveals that selecting the optimal beams, i.e.,
the DFT columns that yield largest-power projections into the
AoA/AoD angular bins, is equivalent to seeking the entries of
\( H_{\text{eff}}(f) \) with the highest power.

Notations: In this paper, we use \(|\cdot|\), \(||\cdot||\) and \(\|\cdot\|_F\) to denote
the amplitude of a scalar, the \(\ell_2\)-norm of a vector, and the
Frobenius norm of a matrix, respectively. Also, given a set \(A\),
we use \(|A|\) to denote its cardinality.

A. Problem Formulation

We refer to the received signal strength (RSS), multipath
delays, (azimuth) AoA bins and AoD bins, corresponding to
\( r_l = |\alpha_l|^2, \tau_l, \theta_l \)’s and \( \phi_l \)’s respectively, as features. Note that
the BS can observe parameters \( \{r_l, \tau_l, \theta_l\}_{l=1}^L \) subject to its
observation resolution but not their \( \phi_l \)’s since these can only
be observed at the user side. Thus, an interesting question
arises: Can the BS extract features that are only available at
the user side, based on what the BS observes?

While the answer to the question might not be straightforward,
let us rethink how the parameters \( \{r_l, \tau_l, \theta_l, \phi_l\}_{l=1}^L \)
are generated. As introduced in Sec. I, given a specific area,
they are obtained via measuring the signals that travel
through the environment where the locations of buildings and
constructions are fixed\(^3\). Hence, they are actually correlated
among themselves.

In this paper, we explore a machine learning approach
to resolve the problem above. In particular, we design and
evaluate four distinct neural network models. For each model,
we design its corresponding input using the RSS, multipath
delays, or their combined use. To enable the machine learning
approach, we train and test the models with ray-tracing chan-
del data collected by NTT DOCOMO Research Labs from

\(^1\)Analysis for the 3-D system can be easily generalized by incorporating
elevation components, which would yield channels that are even sparser.

\(^2\)Without specifying the bandwidth, we consider the observed delay at the
BS can take continuous values by assuming the resolution is infinitely large.

\(^3\)The blocking issue is important especially on higher frequency bands, but
for simplicity it is not considered in this paper.
For the multipath channel between each BS and each user, the distance between any two adjacent user spots on the grid is one meter. BSs, 13609 user spots located on a grid, and the distance

III. MACHINE LEARNING PRELIMINARIES

In this section we introduce several basic concepts for later use, and then describe how to construct the neural network inputs based on the available data introduced in Sec. II-A.

A. Function Approximation with Neural Networks

The problem above can be recast as function approximation where the goal is to approximate an unknown mapping \( f : X \rightarrow Y \) from a set of parameters \( X \) to another set \( Y \). In fact, function approximation by employing neural networks has been used extensively in various contexts for learning complicated and non-linear functions such as 16. In particular, as established in 16, any function can be approximated up to arbitrary accuracy by a neural network with two hidden layers, as shown in Fig. 2. Note that each neuron in the hidden and output layers receives an affine transformation (linear combination) of the neurons' values in the preceding layer, and passes it through a nonlinear activation function such as ReLu, sigmoid and tanh functions. Given enough amount of data, the network can be trained by using the well-known backpropagation algorithm 17.

B. Features Representation

An important aspect of formulating the problem of interest as a supervised learning problem involves constructing appropriate feature representations for the input and output of the neural network.

Let \((x, y)\) denote the input-output pair of a data sample for training the neural network, where \(x\) is an \(N_{\text{BS}} \times 1\) vector, \(y\) is an \(N_{\text{MS}} \times 1\) vector, and each entry corresponds to one of their DFT columns. Note that all the entries of \(x\) and \(y\) should be very close to zero, except for those corresponding to the multipath components, i.e., the selected DFT columns indices for beam selection, with significant power. Consider for instance the example in Fig. 3. Suppose that with the angular resolution offered by its \(N_{\text{BS}} = 6\) ULA the BS can distinguish 3 paths, falling into the 2\(^{nd}\), 4\(^{th}\), and 5\(^{th}\) sectors/bins. A \(6 \times 1\) feature vector can be created where the corresponding 2\(^{nd}\), 4\(^{th}\), 5\(^{th}\) entries are features of the associated paths, such as delays or RSSs, appropriately renormalized. In particular, in the case where RSS values are used as features, first the minimum and the maximum RSS values (in dB) in the data set, \(r_{\text{min}}\) and \(r_{\text{max}}\), are obtained. For an RSS value of \(r\) dB, the entered value in the feature vector is \((r - r_{\text{min}})/(r_{\text{max}} - r_{\text{min}})\), while zeros are entered for all empty bins. In the case where path delay values are used as features, again the minimum and the maximum path delays in the data set, \(\tau_{\text{min}}\) and \(\tau_{\text{max}}\), are obtained. For a path delay value of \(\tau\), the entered value in the feature vector is \((\tau - \tau_{\text{min}})/(2\tau_{\text{max}} - \tau_{\text{min}})\). In addition, the value 1 (corresponding to a fictitious delay of \(2\tau_{\text{max}}\) indicating non-existent paths) is entered for all empty bins.

At the user, the angular resolution is lower, since it has a smaller array than the BS. In the example depicted in Fig. 3, all the paths fall in two angular bins at the user, and in particular two paths fall into the 2\(^{nd}\) bin, and the other path falls into the 4\(^{th}\) bin. Thus, we obtain a \(4 \times 1\) column vector first where the corresponding 2\(^{nd}\) and 4\(^{th}\) entries are the feature of delay or RSS. Next, we convert this \(4 \times 1\) vector into another indication vector, where the 2\(^{nd}\) and the 4\(^{th}\) entries are replaced by 1, indicating the beam index at the user having that AoD component. During the training phase, this information is extracted from the training data set in the form of input-output vector pairs and is used to train the parameters of the neural network. During the testing phase, the trained neural network is fed an input \(x\) to obtain the estimated output \(f(x)\). We can then evaluate the performance of the trained neural network by comparing the estimate \(f(x)\) to the actual output \(y\) across the test data set using appropriately chosen performance metrics.

IV. ANALYSIS OF THE NEURAL NETWORK MODEL

In this section, we first conduct correlation analysis on the features representation that we constructed in Sec. III-B for our ray-tracing data by borrowing the concepts of entropy and mutual information from information theory, so that we


![Fig. 3: Input and output features representation.](image)

![Fig. 4: A graphical illustration of the coupling between input and output feature representations.](image)

![Fig. 5: Correlation between the input X and the output Y: (left) \(H(Y|X)\), (middle) \(I(Y; X)\), and (right) \(I(Y; X)/H(Y)\), as functions of \(N_{BS}\) (the dimension of the DFT codebook).](image)

**TABLE I: Neural network architecture.**

| Layer      | Layer Size | Activation Function |
|------------|------------|---------------------|
| Input layer| 100 nodes  |                     |
| Hidden layer 1 | 50 neurons | Sigmoid             |
| Hidden layer 2 | 20 neurons | Sigmoid             |
| Output layer | 10 neurons | Linear              |

decreases as \(N_{BS}\) increases. Intuitively, this is because when the antenna resolution at the BS is higher, the BS ability to resolve multipaths improves. Next, in the middle one, \(I(Y; X)\) grows larger when \(N_{BS}\) increases, since \(H(Y|X)\) decreases. Finally, in the rightmost one, \(I(Y; X)/H(Y)\) increases and is very close to 1 (≈ 0.97) when \(N_{BS} = 120\). Intuitively, it implies that when \(N_{BS}\) increases, \(Y\) becomes more predictable from \(X\) due to the higher antenna resolution.

### B. Neural Network Optimization

Having verified the correlation between the input and the output of the model, we proceed to formulating the estimation problem as an optimization problem and subsequently building a neural network solver. In this paper we restrict our attention to neural networks with only two hidden layers. In addition, the other parameters of the neural network are summarized in Table I. Note that the activation function of the output layer is chosen to be linear, because we are solving a regression problem (to estimate the RSS or the delay over AoD bins), and linear transformation is more capable of capturing a wide range of the output values.

As shown in Table I the neural network consists of the input layer \(l = 0\), two hidden layers \(l = 1, 2\) and the output layer \(l = 3\). Let \(W^{[l,l-1]}\) denote the weight matrix from layer \(l - 1\) to layer \(l\) and \(b^l\) denote the bias at the neurons in layer \(l\). In addition, we use \(D_t = \{(x^{(n)}_t,y^{(n)}_t)\}_{n=1}^{N_t}\) and \(D_v = \{(x^{(n)}_v,y^{(n)}_v)\}_{n=1}^{N_v}\) to denote the training and the test sets, respectively, where \(N_t = |D_t|\) and \(N_v = |D_v|\) and \(D_t \cap D_v = \emptyset\). Moreover, let \(h_t(x^{(n)}_t)\) denote the output of the neural network in response to the input \(x^{(n)}_t\). Hence, the cost function during the training phase can be written as

\[
J(W, b) = \frac{1}{N_t} \sum_{n=1}^{N_t} \left\| h_t(x^{(n)}_t) - y^{(n)}_t \right\|^2 + \frac{\lambda}{2} \sum_{l=1}^{3} \left\| W^{[l,l-1]} \right\|^2_F \tag{6}
\]

where \(W = \{W^{[l,l-1]}\}_{l=1}^3\), \(b = \{b^l\}_{l=1}^3\), \(\lambda\) is the regularization factor, and \(\| \cdot \|_F\) is the Frobenius norm. Training the model involves optimizing the weights \(W\) and the bias \(b\) through iterations of the backpropagation algorithm. In this paper, the backpropagation algorithm is implemented using the iterative batch gradient descent optimization algorithm.

### C. Statistical Performance Metrics

To assess the estimation performance of the optimized neural network, we define the following statistical metrics:

4 Although in principle neural networks with three or more layers are surely worth investigating as solvers for the types of problems we consider in this paper, in the context of the ray tracing data set upon which we based our study two-hidden layer neural networks performed sufficiently well.

5 If we model the output as a multi-task classification problem where for each task (for each AoD bin), we determine if there exists a path, then the sigmoid function can be chosen as the activation function of the output layer.
It can be seen that the mean squared error of the trained neural network estimator $\rho_v$ is expected, since the training set sample mean $\bar{y}_t$ is not identical to the sample average of the test data set, which minimizes the mean squared error. Moreover, a $\rho_v$ value close to 1 is an indication of the “closeness” of the statistical distributions of the output vectors in the training and test sets. Thus, if $\rho_v$ is close to 1, the model that learns the desired mapping based on the samples in $D_t$ would likely be a good estimator of the outputs in $D_v$, despite not being exposed to the samples therein during training.

In addition, based on (2) and (7), we define the last metric

$$\rho_{NN} \triangleq \eta_{NN}/S_t.$$  

If the statistical similarity between the data in the training and test sets is sufficiently large (i.e., $\rho_v \to 1$), then $\rho_{NN}$ is expected to be positive and less than 1. The smaller $\rho_{NN}$, the more predictable and reliable the trained estimator is likely to be when applied to the new data samples that are not used to train the model. Thus, the combined use of $\rho_v$ and $\rho_{NN}$ helps assess the estimation performance of our model.

Finally, we also employ the widely used receiver operating characteristic (ROC) curves to statistically evaluate the performance of the optimized neural network based estimator. This is because correct prediction of a dominant AoD bin on the user side can be viewed as correct detection whereas incorrect identification of an unfavorable AoD bin as a dominant one can be declared as false alarm. For brevity, we use $P_D$ and $P_F$ to denote their probabilities, respectively.

### V. Simulations

In this section, we present a statistical performance based evaluation of the neural network techniques we developed. Each neural network was trained and tested using the data collected at BS 8 only (represented by the red dot in Fig. 1), which is roughly located at the center of the network and thus can observe a wider angular range. The data set was randomly partitioned into the training set (10% of total samples) and the test set (90% of total samples). Based on the correlation analysis in Sec. III-B, we train and evaluate the four distinct neural network schemes, that differ in terms of their inputs:

1) Each entry of the input vector contains the (normalized) RSS value of the associated angular path. Let $NN_r$ refer to the neural networks trained as such.
2) Each entry of the input vector is given by the delay of the associated angular path. Let $NN_r$ refer to the neural networks trained as such.
3) We sequentially train multiple neural networks. Specifically, the outputs of $NN_r$ and $NN_r$ are fed to another neural network as inputs. Let us refer to this concatenated neural network architecture as $NN_{seq}$.
4) We concatenate the two $100 \times 1$ input vectors to $NN_r$ and $NN_r$ to form a $200 \times 1$ vector, which are then fed into a neural network to estimate the outputs. We denote the neural network trained as such by $NN_{r-r}$.

The results of the four schemes above are presented in Table II. First, it can be seen that $\rho_v = 1.0002$ is promising in the sense that learning the desired mapping from $D_t$ could potentially lead to accurate estimations. Next, the comparison between the $\rho_{NN}$ values for $NN_r$ and $NN_r$ demonstrates that for the specific ray-tracing data we have, the RSS, at the BS, i.e., the $r_i$'s, turn out to be the more reliable predictors of the AoD than the delay, i.e., the $\tau_i$'s. Our finding differs from what is observed in location estimation problems, where delay based method outperform RSS-based ones, especially in the presence of large bandwidths. It would be worth evaluating and comparing these networks with real-world measurements, as opposed to ray-tracking data which are systematically generated by the ray-tracing simulator.

| Optimized NN | $\rho_v$ | $\rho_{NN}$ |
|-------------|----------|-------------|
| $NN_r$      | 1.0002   | 0.5735      |
| $NN_r$      | 1.0002   | 0.6579      |
| $NN_{seq}$  | 1.0002   | 0.5246      |
| $NN_{r-r}$  | 1.0002   | 0.5297      |

---

Such a partition approach is used for showing the simulation curves only. In the experiments, we partitioned the collected data into 95% of total samples for training 5% for testing to obtain better results.
Furthermore, as shown in Table I when we combine both the RSSs and the path delays as the input to both NN_seq and NN_r,τ, the estimation accuracy improves with respect to both NN_r and NN_r,τ. This is quite intuitive as incorporating more information into the estimation model should, in principle, lead to better estimation performance. Also, note that the structures of the last two neural networks where the RSS and delay features are in the combined use as the input feature are not exactly identical to the first two neural networks in terms of the number of neurons or layers. This motivates us to explore better neural networks with the input features in future.

Fig. 6 shows the ROC performance of all the schemes under investigation. In addition, the curve marked “SA” in the figure shows the ROC performance of the sample average of the output vectors (i.e., \( \mathbf{y}_t \)) in the training data set. The sample average can be viewed as a baseline benchmark for the 4 schemes of interest, as it is obtained by using the marginal distribution of the output in case that we do not have any access to the input features.

As shown in Fig. 6, the ROC curves of the other 4 schemes all lie above the sample average curve. Subject to any fixed and small false alarm probability (e.g., \( P_F = 0.1 \)), they all have higher detection probability than the benchmark, which means that they all outperform the sample average (i.e., black-squared “SA”) benchmark estimator. In addition, as shown in the embedded sub-figure, the networks of NN_r (i.e., red-colored “RSS”), NN_seq (i.e., blue-colored “Seq”) and NN_r,τ (i.e., green-colored “both”) are all capable of detecting the target beams with at least 90% probability (\( P_D \geq 0.9 \)) while producing occasional false alarm probability no more than 10% (\( P_F \leq 0.1 \)). Finally, the ROC curve of NN_r,τ (i.e., black-colored “Delay”) lies beneath the other three curves. This confirms our earlier observation from Table I that solely relying on path delay information leads to less accurate estimation.

VI. Conclusions

In this paper, we investigated the problem of predicting wireless channel features that are not directly observable at a BS, based on directly observable features and machine learning driven by large amounts of channel data from the BS’s geographical area. In particular, we focused on estimating the dominant virtual angular beams at the user side based on features directly observed at the BS side, such as AoA-specific path RSSs and delays. We developed four learning-based schemes and demonstrated that they can resolve the features of interest with reasonable accuracy by using ray tracing data from Shinjuku Square. Several interesting directions are worthy of further investigation. Training and testing the proposed learning-based estimation schemes with real-world measurement data is necessary to assess their viability in practice. In addition, exploring other machine-learning architectures and then developing beam-selection algorithms are also goals worth pursuing.

REFERENCES

[1] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-wave cellular wireless networks: Potentials and challenges,” Proceedings of the IEEE, vol. 102, no. 3, pp. 366–385, 2014.
[2] Ghosh et al., “Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks,” IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1152–1163, 2014.
[3] O. Abari, H. Hassanieh, M. Rodriguez, and D. Katabi, “Millimeter wave communications: From point-to-point links to agile network connections.” in HotNets, 2016, pp. 169–175.
[4] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, “Millimeter wave beamforming for wireless backhaul and access in small cell networks.” IEEE Transactions on Communications, vol. 61, no. 10, pp. 4391–4403, 2013.
[5] A. Alkhateeb, Ō. El Ayach, G. Leus, and R. W. Heath, “Channel estimation and hybrid precoding for millimeter wave cellular systems,” IEEE J. Sel. Topics in Signal Proc., vol. 8, no. 5, pp. 831–846, 2014.
[6] P. Schniter and A. Sayeed, “Channel estimation and precoder design for millimeter-wave communications: The sparse way,” in 48th Asilomar Conf. on Signals, Systems and Computers., IEEE, 2014, pp. 273–277.
[7] T. Kim and D. J. Love, “Virtual aoa and aod estimation for sparse millimeter wave mimo channels,” in 16th Int. Workshop on Signal Processing Advances in Wireless Comm. IEEE, 2015, pp. 146–150.
[8] Z. Marzi, D. Ramasamy, and U. Madhow, “Compressive channel estimation and tracking for large arrays in mm-wave picocells,” IEEE Journal of Selected Topics in Signal Proc., vol. 10, no. 3, pp. 514–527, 2016.
[9] Z. Gao, L. Dai, Z. Wang, and S. Chen, “Spatially common sparsity based adaptive channel estimation and feedback for fdd massive mimo,” IEEE transactions on signal proc., vol. 63, no. 23, pp. 6169–6183, 2015.
[10] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and S. Li, “Channel estimation for millimeter-wave multiuser mimo systems via parafac decompostion,” IEEE Transactions on Wireless Communications, vol. 15, no. 11, pp. 7501–7516, 2016.
[11] A. Alkhateeb, G. Leus, and R. W. Heath, “Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?” in IEEE Int. Conf. on Acoustics, Speech and Signal Processing. IEEE, 2015, pp. 2909–2913.
[12] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. Blum, “Low-rank tensor decomposition-aided channel estimation for millimeter wave mimo-ofdm systems,” IEEE Journal on Selected Areas in Comm., 2017.
[13] X. Li, J. Fang, H. Li, and P. Wang, “Millimeter wave channel estimation via exploiting joint sparse and low-rank structures,” arXiv preprint arXiv:1705.02455, 2017.
[14] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1058–1076, 2010.
[15] A. Adhikary, J. Nam, J. Ahn, and G. Caire, “Joint spatial division and multiplexing – the large-scale array regime,” IEEE Transactions on Information Theory, vol. 59, no. 10, pp. 6441–6463, 2013.
[16] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Math. of Control, Signals, and Systems, vol. 2, no. 4, pp. 303–314, 1989.
[17] T. M. Mitchell, Machine learning. McGraw-Hill, Boston, MA, 1997.
[18] M. Schmidt. (2005) minfunc: unconstrained differentiable multivariate optimization in matlab. [Online]. Available: http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html.