

A Machine Learning Solution for Beam Tracking in mmWave Systems

Daoud Burghal, *Student Member, IEEE*, Naveed A. Abbasi, *Member, IEEE*, and Andreas F. Molisch, *Fellow, IEEE*

Abstract—Utilizing millimeter-wave (mmWave) frequencies for wireless communication in *mobile* systems is challenging since it requires continuous tracking of the beam direction. Recently, beam tracking techniques based on channel sparsity and/or Kalman filter-based techniques were proposed where the solutions use assumptions regarding the environment and device mobility that may not hold in practical scenarios. In this paper, we explore a machine learning-based approach to track the angle of arrival (AoA) for specific paths in realistic scenarios. In particular, we use a recurrent neural network (R-NN) structure with a modified cost function to track the AoA. We propose methods to train the network in sequential data, and study the performance of our proposed solution in comparison to an extended Kalman filter based solution in a realistic mmWave scenario based on stochastic channel model from the QuaDRiGa framework. Results show that our proposed solution outperforms an extended Kalman filter-based method by reducing the AoA outage probability, and thus reducing the need for frequent beam search.

Index Terms—millimeter-wave, Beamforming, Beam Tracking, Machine Learning.

I. INTRODUCTION

Next-generation wireless standards aim to utilize millimeter-wave (mmWave) frequencies as one of the key avenues to meet the ever-increasing demands for bandwidth. Due to channel impairments at high frequencies such as high path loss, diffraction loss etc. [1], using multi-antenna systems is necessary. This in turn requires proper beam management techniques to harness the advantages of mmWave band. More specifically, beam management techniques can be grouped in two main categories: beam-search and beam-steering, where the former refers to the procedures needed during the initial communication link establishment or after a link failure, while the latter refers to the techniques used to steer a beam to different directions. Beam management usually consumes considerable overhead as both ends may need to exchange pilot signals to steer the beams in the right direction.

The overhead problem is exacerbated in mobile systems where the optimal beam direction changes over time due to the mobility of the user equipment (UE). Several recent works propose techniques to track the beam directions, [2]–[10]. In [9] and [7], the authors make use of channel sparsity

in mmWave band to simplify the beam tracking problem. In [3], [4], [8], [10], the authors use Kalman filter (KF)-based methods to track the beam directions. However, these methods have several issues, such as the intensive calculations required by sparse optimization problems that might not be suitable for implementation especially at the UE side, or the simplified channel models that are used in KF-based methods. The authors of [2], [5], [6] use prior beam direction to narrow down the search space for efficient tracking; however, these methods do not fully capture the true channel dynamics.

In this paper, we explore a machine learning (ML) approach for the beam tracking problem. In particular, we use a recurrent neural network (R-NN) to track the angle of arrival (AoA) at the UE side when only the channel coefficient and previous estimates of AoAs are available. The main advantages of using ML over other techniques are twofold: (i) based on the Universal Approximation property of NNs, we anticipate that the proposed solution can capture the channel and environment dynamics well, and (ii) low complexity of mathematical operations. For instance, the complex mathematical operations needed by "ML training", can be done offline at the base-station (BS) side, and the trained solution can be relayed to UEs as required. Once the training has been completed, ML solutions typically require basic mathematical operations that can be performed easily in real-time.

Applying ML to wireless systems has gained considerable attention lately, where applications span different layers of communication systems [11]–[14], such as scheduling and resource allocation [15], [16], detection, decoding [17], and estimation [11]. There have been recent efforts to apply ML to estimate the AoA. In [18], the authors use Deep NNs to provide a candidate set of angle estimation that can be used as an input to a maximum likelihood estimator for optimal angle search and selection. The authors of [19] use observable features at the BS side to predict the AoA at the UE side. In [20], the authors take sparsity of the incident signals into consideration to propose a ML solution that solves the transformation from array outputs to AoA spectrum. Moreover, in [21], the authors propose a ML solution based on autoencoders and followed NN layers for robust AoA estimation.

In contrast to prior work, we tackle the problem of AoA *tracking* for mobile user over a path. In such cases, the UE parameters are in the form of sequences that evolve over time, which calls for the selection of suitable recurrent ML

This work was supported financially in part by the National Science Foundation (NSF CIF-1618078).

The authors are at the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA (email: burghal,nabbasi,molisch@usc.edu).

solutions. We, therefore, use the long short term memory (LSTM) network architecture for the AoA tracking problem. Moreover, due to the periodicity of the angles, we propose a cosine-based loss function. We study the performance of the network in a stochastic channel model that we generate using the QuaDRiGa framework [22], and we also compare the performance of the ML solution against the KF-based solution proposed in [4].

The structure of the paper is as follows. In Sec. II, we provide the problem setup and summarize the KF based solution of [4]. We introduce the ML solution and the cost function in III. In Sec. IV, we discuss the method used to generate the data and train the ML solution. We analyze the performance of our solution in Sec V, and finally, provide the concluding remarks in Sec. VI.

II. SYSTEM MODEL

In this section, we firstly describe our system model for the beam tracking problem and briefly summarize the Extended Kalman filter (EKF) based solution [3], [4].

A. System Model

We consider a scenario where an omni-directional BS is communicating with a UE that is equipped with a uniform linear array (ULA) with N_r antenna elements. The ULA's response (steering vector) is given as

$$a(\theta) = \frac{1}{\sqrt{N_r}} [1, e^{j\frac{2\pi}{\lambda}d\cos(\theta)}, \dots, e^{j\frac{2\pi}{\lambda}d(N_r-1)\cos(\theta)}], \quad (1)$$

where λ is the wavelength of the carrier frequency, and d is the distance between array elements. Considering a sparse system where only a single multipath component (MPC) falls into the antenna's main beam direction, the channel observation at a particular time n is given as (similar to [4])

$$y[n] = \frac{\alpha[n]}{N_r} \frac{1 - e^{jN_rnd(\cos(\theta) - \cos(\theta_B))}}{1 - e^{jnd(\cos(\theta) - \cos(\theta_B))}} + z[n], \quad (2)$$

where $\alpha[n] = \alpha_R[n] + j\alpha_I[n]$ is the complex path gain, θ is the AoA of the most dominant MPC, θ_B is the direction of the main beam of the ULA, and $z[n]$ is the noise and interference from other MPC components.

In addition to the observation model provided above, we assume that an initial connection between the BS and the UE is successfully established in the current scenario, and the fundamental problem at hand is that of maintaining this link by means of beam tracking techniques. This assumption implies the complete knowledge of initial conditions for complex gain and AoA at $n = 0$. Finally, please note that the models, proposed solutions, and the results for the current paper can be easily extended to the case where BS also has ULAs, however, the current formulation was kept for simplicity.

B. Overview of EKF-based Solutions

KF-based solutions for beam tracking systems have been explored in the literature earlier, therefore, it is important that any newly proposed solution is compared to the same. In this subsection, we briefly describe an EKF-based solution for the beam tracking problem described above that is similar in approach to [4] and [3]. To avoid implementation issues in dealing with the complex path gain the state vector for the system is formulated as

$$x[n] = [\alpha_R[n], \alpha_I[n], \theta]^T. \quad (3)$$

The evolution processes for both the real and imaginary part of the path gain are assumed to be described as

$$\alpha_{R/I}[n+1] = \rho\alpha_{R/I}[n] - \zeta[n], \quad (4)$$

where ρ is the correlation coefficient, and the ζ is a random variable distributed according to $\mathcal{N}(0, (1 - \rho^2)/2)$. Since the basic recursion for EKF is fundamentally similar to the one described in [4], we do not go into its detail here, however, we note that the correlation coefficient and other statistics used for the EKF method are specifically calculated from the data generated for the current problem.

III. PROPOSED ML SOLUTION

As discussed earlier, we use a recurrent NN architecture for the current beam tracking problem. The input to such a network is the observed noisy signal, $y[n]$ and the previous estimate of the AoA, $\hat{\theta}[n-1]$. The output of the network is an estimate of the AoA at the current time $\hat{\theta}[n]$.¹ While the input is restricted to $y[n]$ and $\hat{\theta}[n-1]$, the recurrent structure of the NN implicitly uses previous observations and predictions as well that make it suitable for learning over sequential data. Although the explicit input $\hat{\theta}[n-1]$ might seem redundant, it is proved to be necessary as we discuss later.

The proposed recurrent NN in this paper uses an LSTM architecture. LSTMs have been implemented successfully in several challenging tasks such as speech recognition and translation. At each time instant, the inputs to the LSTM are the observed features, the previous outputs, and the previous "cell state". The cell state is a memory that is controlled by three gates, which control when to read, write, and erase the value of the cell. The decisions of the gates are controlled by NNs that provide nonlinear transformations of the input values.

A. Custom Cost Function

The weights of the above solution are determined during the training phase over a training dataset where the goal is to minimize the prediction error of the label values at the output over the observed data points. Mean Square Error (MSE) is usually used to quantify the error, where the square error

¹Note that in the actual implementation we track both estimates of coefficient $\hat{\alpha}[n]$ and $\hat{\theta}[n]$, however, we report only the results of AoA tracking for clarity and due to space limitation. The elaboration and analysis for the joint tracking of both the complex path gain and the AoA is left for a future work.

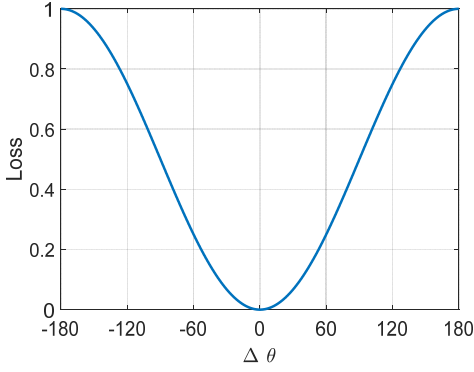


Fig. 1. The loss function \mathcal{E}_θ as function of $\Delta\theta$

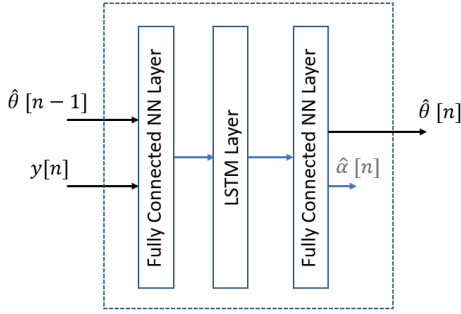


Fig. 2. The ML network structure.

$\mathcal{E}_{SE} = |\hat{\theta} - \theta|^2 = |\Delta\theta|^2$. However, due to the periodicity of θ , large $\Delta\theta$ values might not necessary bad, for instance, $\Delta\theta = 358$ when $\theta = 1^\circ$ and $\hat{\theta} = 359^\circ$, a cost function should be able to take this into account. In this paper, we define the error function \mathcal{E}_θ as follows

$$\mathcal{E}_\theta(\Delta\theta) = \frac{1}{2}(1 - \cos(\Delta\theta)). \quad (5)$$

This cost function solves the problem of $\Delta\theta$ ambiguity that is otherwise present in MSE. For small errors, it is identical to the square error. Fig. 1 shows a plot of this cost function. Another interesting property of the subject cost function is that it is restricted to $[0, 1]$ range, where 1 represents the highest error value. This adds more stability to the training process our error is bounded for the current problem.

B. The Network

Fig. 2 shows the used ML solution, which consists of one fully connected layer with 20 neurons, one LSTM layer with 40 hidden units, followed by another fully connected layer with 20 neurons. We do not use activation functions after the fully connected layer. Further details of the input and output of the proposed network are discussed in subsection IV-A.

IV. DATA GENERATION

The QuaDRiGa framework implements several channel models and allows for spatial consistency of generated sequences, which is necessary for such sequential data [22].

Therefore, in this paper, we consider a Non-Line Of Sight (NLOS) urban micro-cell environment within this framework where the channel realizations are generated based on the mmMAGIC model [23], this propagation model is supported by a number of channel measurement campaigns over several frequencies [23].

To generate the data for the current study, we consider the following setup. A cell has a single BS equipped with a single omnidirectional antenna located in the center of a cell with 60 m radius and the carrier frequency is assumed to be 28 GHz. We generate 2500 realization of UE mobility where each of these are around 20 m long, in a straight line, with a random initial starting location and orientation. We record the AoA and the channel coefficients of the MPCs every 0.1 m of each path realization.

A. Training and Testing

For this supervised learning problem, we need to create features and labels sets before proceeding with the training and testing steps. Ideally, we would like the features $\mathcal{F}_{i,n}$ be $[\text{real}(y_i[n]), \text{imag}(y_i[n]), \theta_i[n-1]]$ and the label $\mathcal{L}_{i,n}$ be $[\theta_i[n]]$,² for the training sample at time n of the i^{th} training sequence. However, for a beam tracking problem, the definition of the labels is challenging, as the true label depends on the beam-steering decision of the UE in a previous time-step. To see this, note that $y[n]$ depends on the true $\theta[n-1]$ by setting $\theta_B = \theta[n-1]$ in (2), however, since in reality the UE has access to $\hat{\theta}[n-1]$, the observed feature $[y[n], \hat{\theta}[n-1]]$ will be different for different $\theta[n-1]$. To alleviate this problem, we consider the following. We add noise to the true $\theta[n-1]$ and generate $y[n]$ accordingly. In other words, we let the training features be:

$$\mathcal{F}_{i,n} = [\text{real}(y_{\text{trn},i}[n]), \text{imag}(y_{\text{trn},i}[n]), \theta_{\text{trn},i}[n-1]],$$

we here introduced the subscript \cdot_{trn} to emphasize that these values are different from the one we generated above, in particular, we use $\theta_{\text{trn},i}[n-1] = \theta_i[n-1] + \psi[n-1]$, where $\psi[n-1] \sim \mathcal{N}(0, \sigma_\psi^2)$. We here choose $\sigma_\psi = 5^\circ$, we assume that $\psi[n-1]$ is generated independently for all n .

During the testing, the observed features are $[\text{real}(y_i[n]), \text{imag}(y_i[n]), \hat{\theta}_i[n-1]]$, throughout the testing phase we assume that, for a given threshold value θ_{Th} , when $|\hat{\theta}_i[n] - \theta_i[n]| > \theta_{\text{Th}}$, the UE announces an outage and initializes a conventional AoA search/estimation procedure through a full search. In such cases, the input to the ML solution (and comparable EKF estimation) at the next time-frame will be based on the true angle once again.

Finally, please note that we split our dataset by selecting 1750 sequences for training and 750 for testing and used the Adam optimizer for the network training [24].

²As highlight above, the actual implementation uses predicts $\alpha[n]$ (as can be seen in Fig. 2), but we drop that from our discussion here.

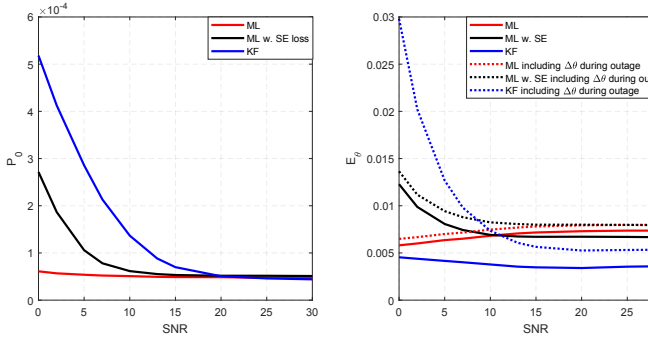


Fig. 3. The performance vs test SNR: (a) Outage Probability P_0 , (b) Angle Error \bar{E}_θ . Results are for training SNR = 7 dB.

V. RESULTS

In this section, we report the performance of the proposed ML solution by analyzing the outage probability and the loss values. Outage probability for the network is given as

$$P_0 = \frac{1}{\sum_i N_{\text{tst},i}} \sum_i \sum_j \mathbf{I}(|\Delta\theta_{i,j}| > \theta_{\text{Th}}), \quad (6)$$

where $N_{\text{tst},i}$ is the number of training samples for sequence i , \mathbf{I} is an indicator function, and $\Delta\theta_{i,j} = \theta_{i,j} - \hat{\theta}_{i,j}$, where $\theta_{i,j}$ is the AoA at j^{th} instant of the i^{th} training sequence. We choose the threshold angle θ_{Th} as a function of number of UE antennas N_r as $\theta_{\text{Th}} = \frac{2}{3} \frac{360}{N_r}$. The second quantity of interest is the average error, is it simply defined as

$$\bar{E}_\theta = \frac{1}{\sum_i N_{\text{tst},i}} \sum_i \sum_j \mathcal{E}_\theta(\Delta\theta_{i,j}). \quad (7)$$

Fig. 3, shows the performance of outage probability and average error for different values of the test signal to noise ratio (SNR). where the training is done at SNR = 7 dB. We notice that the proposed ML solution shows higher robustness for smaller SNR values, while it converges to similar outage probability as EKF solution for other cases. This shows a clear advantage of the proposed ML scheme. Note that in Fig. 3-(b), we plot \bar{E}_θ for two cases depending on whether we consider the angles during the outage. Since EKF suffers from more outages, when we exclude the angle estimates during the outage, the loss value drops significantly, however, this is not an accurate measure for the cases where the outage probability is different. For large SNR, we notice that EKF shows smaller loss value, which could indicate that EKF shows better tracking of the angles when it is not in an outage. Nevertheless, in reality, a combination of P_0 and \bar{E} is important to properly analyze the performance.

In Fig. 4 we consider the performance of the solution against the number of receiver antenna, N_r . As discussed above, θ_{Th} is defined as function of N_r . The testing SNR here is 10 dB for this case. We note that the proposed ML solution shows a lower outage probability as a function of N_r . The values of the loss function are comparable at this SNR

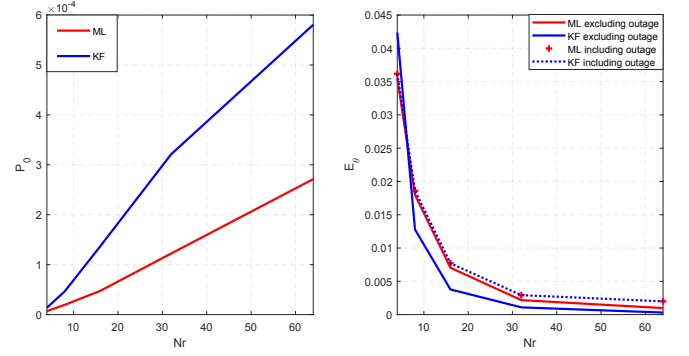


Fig. 4. The impact of number of UE antenna N_r on the performance (a) Outage Probability P_0 , (b) Angle Error \bar{E}_θ . Results are for test SNR = 10 dB.

value, as it also can be seen from Fig. 3. Note that the decay in the loss value is expected as (i) we have small beam-width (ii) frequent outages result in more angle corrections.

In Fig. 5, we plot the performance when we vary the number of samples per meter at test and observe that the ML solution shows smaller outage probability even when smaller number of samples are used. We notice that the loss values slightly increase for ML solution as we increase the number of samples, while they decrease for the EKF solution. This could be attributed to our previous observation; the EKF shows better tracking of the AoA angle when not it is not in an outage.

Finally, we notice that the performance of the ML solution will be impacted by the training SNR. Fig. 6 shows P_0 as a function of the training SNR. We notice that moderate noise levels in training signals improve the performance, this has been noticed in the ML literature as it guards against overfitting and improves generalization.

VI. CONCLUSIONS

In this paper, we propose a ML-based beam tracking solution that tracks the AoA for a mobile user. The proposed solution is based a recurrent NN and two fully connected NNs.

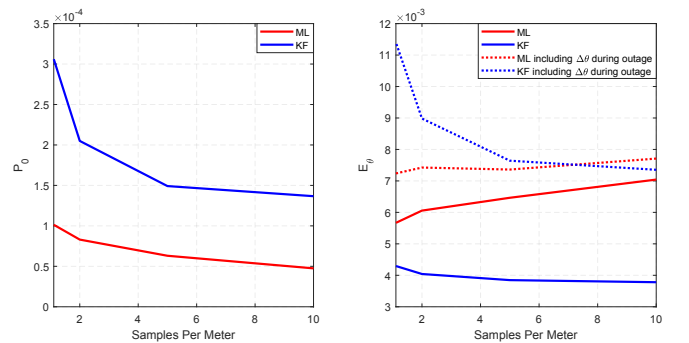


Fig. 5. The impact of observation density on the performance (a) Outage Probability P_0 , (b) Angle Error \bar{E}_θ . Results are for test SNR = 10 dB.

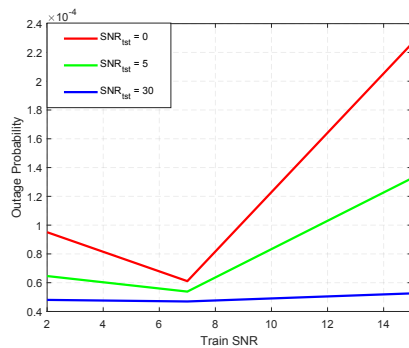


Fig. 6. Outage probability P_0 over different training SNR values for several test SNR.

Due to the periodicity of the AoA, we propose a customized loss function based on cosine function. The performance of the solution is studied in a standards-based stochastic environment that we generated at mmWave frequency using the QuaDRiGa framework. The results show a promising performance of the ML solution when compared to an EKF-based solution. In particular, we observe smaller outage probability, especially at low to moderate SNR.

REFERENCES

- [1] A. F. Molisch, *Wireless Communications*. Wiley; 2 edition, 2010.
- [2] J. Palacios, D. De Donno, and J. Widmer, "Tracking mm-wave channel dynamics: Fast beam training strategies under mobility," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.
- [3] S. Jayaprakasam, X. Ma, J. W. Choi, and S. Kim, "Robust beam-tracking for mmwave mobile communications," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2654–2657, 2017.
- [4] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2016, pp. 743–747.
- [5] J. He, T. Kim, H. Ghauch, K. Liu, and G. Wang, "Millimeter wave mimo channel tracking systems," in *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2014, pp. 416–421.
- [6] K. Gao, M. Cai, D. Nie, B. Hochwald, J. N. Laneman, H. Huang, and K. Liu, "Beampattern-based tracking for millimeter wave communication systems," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.
- [7] D. Ramasamy, S. Venkateswaran, and U. Madhow, "Compressive tracking with 1000-element arrays: A framework for multi-gbps mm wave cellular downlinks," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 690–697.
- [8] C. Zhang, D. Guo, and P. Fan, "Tracking angles of departure and arrival in a mobile millimeter wave channel," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.
- [9] Q. Duan, T. Kim, H. Huang, K. Liu, and G. Wang, "Aod and aoa tracking with directional sounding beam design for millimeter wave mimo systems," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 2271–2276.
- [10] S. G. Larew and D. J. Love, "Adaptive beam tracking with the unscented kalman filter for millimeter wave communication," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1658–1662, 2019.
- [11] M. Zamanipour, "A survey on deep-learning based techniques for modeling and estimation of massivemimo channels," *arXiv preprint arXiv:1910.03390*, 2019.
- [12] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [13] T. E. Bogale, X. Wang, and L. B. Le, "Machine intelligence techniques for next-generation context-aware wireless networks," *arXiv preprint arXiv:1801.04223*, 2018.
- [14] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2018.
- [15] X. Cao, R. Ma, L. Liu, H. Shi, Y. Cheng, and C. Sun, "A machine learning based algorithm for joint scheduling and power control in wireless networks," *IEEE Internet of Things Journal*, 2018.
- [16] D. Burghal, R. Wang, and A. F. Molisch, "Band assignment in dual band systems: A learning-based approach," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 7–13.
- [17] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*. IEEE, 2016, pp. 341–346.
- [18] D. Hu, Y. Zhang, L. He, and J. Wu, "Low-complexity deep-learning-based doa estimation for hybrid massive mimo systems with uniform circular arrays," *IEEE Wireless Communications Letters*, 2019.
- [19] S. Navabi, C. Wang, O. Y. Bursalioglu, and H. Papadopoulos, "Predicting wireless channel features using neural networks," *arXiv preprint arXiv:1802.00107*, 2018.
- [20] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1688–1692, 2019.
- [21] Z.-M. Liu, C. Zhang, and S. Y. Philip, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.
- [22] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-d multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.
- [23] H2020-ICT-671650-mmMAGIC/D2.2, "mmMAGIC D2.2- measurement results and final mmMAGIC channel models," Tech. Rep., 2017.
- [24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.