Efficient Cross-Frequency Beam Prediction in 6G Wireless Using Time Series Data

Vaibhav Bhosale[§1], Navrati Saxena[2], Ketan Bhardwaj[1], Ada Gavrilovska[1], Abhishek Roy[3]
[1]*School of Computer Science*, *Georgia Institute of Technology*, Atlanta, USA
[2]*Department of Computer Science*, *San Jose State University*, San Jose, USA
[3]*Advanced Communication Technology*, *Mediatek Inc*, San Jose, USA
vbhosale6@gatech.edu, navrati.saxena@sjsu.edu, ketanbj@gatech.edu, ada@cc.gatech.edu, Abhishek.Roy@mediatek.com

2024 International Symposium on Networks, Computers and Communications (ISNCC) | 979-8-3503-6491-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISNCC62547.2024.10759051

***Abstract*—Next generation 6G wireless envisions a much higher data rate and a lower latency compared to 5G wireless networks. Directional antennas with narrow beams across high mmWave frequencies hold the key to achieving such high data rates. However, Beam Management (BM), which is the process of finding appropriate transmit and receive beams, offers significant challenges. Dynamic channel variation, user mobility, and narrow beams in high frequency mmWave channels further complicate these challenges. Efficient Machine Learning (ML) strategies can be used to alleviate this overhead. In spite of their underlying differences, sub-6 GHz and high frequency mmWave channels share some similarities in array geometry, number of paths, and surrounding environment. As sub-6 GHz channel characteristics are relatively easier to acquire and learn, we introduce a new machine learning framework, using transformer and LSTM to learn sub-6 GHz channel information over time for efficient beam prediction across high frequency mmWave channels. System Level simulation results point out that when using time-series data-based learning of beam patterns with transformers or LSTMs in sub-6 GHz channels, our proposed scheme achieves up to 99.5% top-5 beam prediction accuracy while reducing the BM overhead by over 50% compared to existing work.**

***Index Terms*—6G wireless, beam management, machine learning, transformer, LSTMs, channel measurements**

I. Introduction

6G wireless, driven by the insatiable demand for high data rates and ubiquitous connectivity, aims to utilize high-frequency millimeter wave (mmWave) bands to provide an order of magnitude more spectrum and ubiquitous connectivity for emerging new applications, like eXtended Reality (XR) and Metaverse. However, mmWave bands face challenges such as high path loss, blockage, and oxygen absorption. Dense network deployment with a large number of small-size, directional antennas is expected to alleviate these challenges. This will be aided by beam management (BM) to find the (near) optimal transmit/receive beams between the next-generation base station nodes (gNB) and mobile User Equipment (UE).

One major challenge [1] in BM lies in inefficient beam sweeping for the downlink (DL)/uplink (UL) beam correspondence. Beam sweeping involves an exhaustive search to find the beam/beam-pair having the strongest signal strength. Naturally, performing an exhaustive search in all possible directions is quite complex and expensive (both in terms of latency and power). This already complex problem becomes even more complicated in the mmWave bands, located in the frequency spectrum ranging from 30 to 300 GHz. MmWave bands provide the benefits of a big chunk of available and un-explored frequencies. However, these benefits come with a high attenuation cost in free space. To address the high attenuation, a large number of highly directional MIMO antennas are used, whose gain compensates for the path loss. The use of these highly directional MIMO antennas demands precise and efficient beam selection methods to ensure the required application data rate and meet the strict delay requirements. Another challenge such bands pose is the low diffraction capacity and severe blocking caused by most materials. Furthermore, beam searching across these large number of narrow beams in high mmWave frequencies incurs more delay and computational power. This makes BM in mmWave more complex and expensive compared to BM in sub-6 GHz bands. Therefore, 6G wireless will continue using sub-6 GHz bands, also known as frequency range 1 (FR1) for outdoor to indoor connectivity, while exploiting mmWave bands or frequency range 2 (FR2) for high data rates in indoor and outdoor environments [2].

Artificial intelligence (AI) and machine learning (ML) are expected to play a key rile in alleviating the complexity of 6G wireless. Major wireless standard bodies, like the 3rd Generation Partnership Project (3GPP), anticipate native AI/ML support in 6G wireless. 3GPP's efforts encompass AI/ML applications for networks and radio interfaces, with the eventual goal of standardization [3]. More recently, 3GPP has recognized the use of AI/ML techniques for efficient BM as a key use-case. As many fundamental relationships in wireless communications are often non-linear, deep neural networks (DNNs) have recently been adopted in various fields of wireless communications including BM [4]–[12].

Interestingly, while different types of DNN models [11], [12] have been recently used for beam prediction such as using past channel state information to predict future channels [9], using sub-6 GHz channels to predict mmWave beams [11], [12], time series data for cross-frequency BM is not yet explored. As BM in wireless involves non-linear mappings across time-varying wireless channels, we believe that exploring time series data for cross-frequency BM has sufficient potential for improving its efficiency and accuracy. This is where we believe that models, such as transformers [13] and long short-term

[§]This work was done as an intern at Mediatek Inc

memory (LSTMs) [6] come into play[§]. Similar to recurrent neural networks (RNNs) [5], LSTM is a deep learning model architecture capable of learning long-term dependencies for sequence prediction problems. A transformer, on the other hand, is another deep learning model architecture that avoids recurrence and instead relies entirely on an *'attention'* mechanism, differentially weighing the significance of each part of the input data for obtaining global dependencies between input and output. Removing recurrence in favor of attention mechanisms endows transformers with significantly higher parallelization, compared to other popular DNN methods like RNNs and LSTMs.

In this paper, we introduce a new time series data-based beam prediction using models, such as LSTMs and transformers that exploit channel characteristics of a sub-6 GHz channel to choose a mmWave beam. The use of LSTMs incurs a lower number of FLOPs, resulting in lower energy consumption, while transformers can serve inference faster due to the presence of their underlying attention mechanism. Our framework lets 6G developers choose which model they need to use based on their specific requirements. Assuming a sub-6 GHz connection is already established, we learn its underlying channel information over time by using LSTM/transformer learning. While learning over sub-6 GHz information aids in reducing the search space needed for the initial mmWave beam establishment, the use of LSTMs/transformers over time series helps in the efficient capturing of time-varying wireless channel characteristics.

The rest of the paper is organized as follows: In Section II we take a look at the major related works on beam prediction and point out our motivation behind using time series information. Section III explains the system model and problem formulation. Subsequently, we explain our proposed time series based beam prediction method in Section IV. Simulation results in Section V show the efficacy of our solution. Section VI concludes the paper with pointers to future research.

II. Related Works and Motivation

The BM problem has recently raised an increased research interest, especially for the high frequencies (e.g., mmWave and THz) 6G wireless networks. These high frequency networks are known to suffer from significant path loss and blockage [14], and hence dense network deployments with large-scale antenna arrays are used to achieve high beamforming gains to compensate this loss. However, the increased scale of antenna arrays introduces high overhead when UEs try to periodically measure the signal quality of different beams for reporting the beam measurements for beam alignment and potential handovers [15].

Adaptive learning of time varying wireless channels using AI/ML techniques have recently been used to address this challenge and enable reliable and efficient beam management [16], [17]. The work in [4] uses reinforcement learning to develop a multi-armed bandit framework for beam tracking.

[§]RNN was also a candidate, but we did not consider it due to its vanishing/exploding gradients for longer sequences.

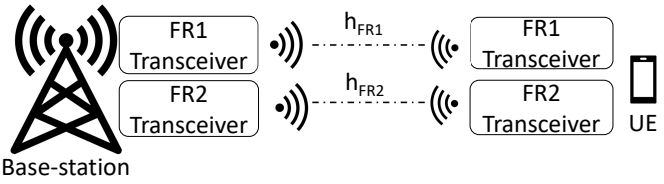

Fig. 1: Depicting the system setup where the FR1 and FR2 antenna arrays are colocated on the base-station and UE.

A data-driven strategy that fuses the reference signal received power (RSRP) with orientation information using an RNN is developed in [7]. The authors in [8] use LSTMs to support large-scale antenna array enabled hybrid, directional BM in hotspot-based virtual small cells. [9] explores LSTMs to utilize the past channel state information (CSI) for efficient prediction of future channels in vehicular mmWave systems. Deep learning techniques, similar to Natural Language Processing (NLP) is used in [10] to predict the best serving beams of the mobile UEs. Interestingly, the work in [11] points out the efficacy of fully-connected neural networks (FCNN) to closely approximate these mapping functions, required for predicting the optimal mmWave beams from sub-6 GHz channels. [12] also uses deep neural networks for exploring the power delay profile (PDP) of a sub-6 GHz channel to predict the optimal mmWave beams.

Broadly a lot of this work has focused on predicting beam parameters by either (i) looking at fewer beam parameters, (ii) parameters over time, (iii) parameters in a different frequency band, etc. Our focus is on the latter two classes of problems, where AI/ML has been recently used to share information along the temporal and frequency domains individually. However, to the best of our knowledge, exploiting time series data for beam prediction in different frequency bands is not yet explored. As BM in high frequency mmWave bands are more complex and expensive compared to BM in sub-6 GHz bands and wireless channels are inherently time varying, we posit that there is additional value in combining the information across the two domains (time and frequency) to use temporal information for improving the prediction accuracy in the frequency domain beam prediction. Fortunately, LSTMs and transformers provide an efficient tool to capture such time series information [18]. Although originally developed for Natural Language Processing (NLP) [6], [13], over the next few sections we will show how time series based learning can delve into time varying wireless channels information to assist and improve cross-frequency BM in 6G wireless.

III. System and Channel Models

We consider a system where the uplink communication happens in FR1 and the downlink in FR2 as shown in Figure 1. We assume M_{FR1} antennas for uplink and M_{FR2} antennas for downlink. We follow a similar system operation and channel models as defined in [11]. We use the uplink channel vector as

$h_{FR1}[k] \in \mathbb{C}^{M_{FR1} \times 1}$ at the kth subcarrier with $k = 1, \ldots, K$ (K is the total number of subcarriers) and the signal received at the gNB sub-6 GHz or FR1 array as:

$$y_{FR1}[k] = h_{FR1}[k]s_p[k] + n_{FR1}[k], \tag{1}$$

where $s_p[k]$ is the uplink pilot signal satisfying $\mathbb{E}\|s_p[k]\|^2 = \frac{P_{FR1}}{K}$, with P_{FR1} as the uplink transmit power, and $n_{FR1}[k] \sim \mathbf{N_C}(0, \sigma^2, I)$ is the noise at the gNB FR1 array. Similarly, defining the downlink beamforming vector as $f \in \mathbb{C}^{M_{FR2} \times 1}$ leads to the signal received by the UE as:

$$y_{FR2}[\bar{k}] = h_{FR2}^T[\bar{k}]fs_d + n_{FR2}[\bar{k}] \tag{2}$$

where $h_{FR2}[\bar{k}] \in \mathbb{C}^{M_{FR2} \times 1}$ represents the downlink channel from the mobile UE to the gNB FR2 array at the $\bar{k}^{th}$ subcarrier, $\bar{k} = 1, 2, \ldots, \bar{K}$. Owing to the hardware constraints in the FR2 analog beamforming vectors, these vectors are generally selected from quantized codebooks. Thus, it can be assumed that the beamforming vector f takes one of the candidate values collected in the codebook F, i.e., $f \in F$, with cardinality $\|F\| = N_{CB}$. We adopt a geometric (physical) model for the FR1 and FR2 channels [19]. Using this allows us to model the FR2 channel as

$$h_{FR2}[k] = \sum_{d=0}^{D_C - 1} \sum_{l=1}^{L} \alpha_l e^{-j\frac{2\pi k}{K}} p(dT_s - \tau_l)\mathbf{a}(\theta_l, \phi_l), \tag{3}$$

where L is the total number of channel paths, $\alpha, \tau, \theta, \phi$ are the path gains (including the path-loss), the delay, the azimuth angle of arrival (AoA), and elevation AoA, respectively, and $\mathbf{a}$ is the steering vector for the lth channel path. T_s represents the sampling time while D_C denotes the cyclic prefix length (assuming that the maximum delay is less than $D_C T_s$). Using this channel model and the system setup mentioned above, the achievable downlink rate for an FR2 channel h_{FR2} and beamforming vector $\mathbf{f}$ can be written as:

$$R(\{h_{FR2}[\bar{k}], \mathbf{f}\}) = \sum_{\bar{k}=1}^{\bar{K}} \log_2(1 + SNR\|h_{FR2}^T\mathbf{f}\|^2) \tag{4}$$

where the SNR is defined as $SNR = \frac{P_{FR2}}{K\sigma_{FR2}^2}$. The optimal beamforming vector is the one that maximizes the $R(\{h_{FR2}[\bar{k}], \mathbf{f}\})$, which we denote by $f^\star$. Predicting $f^\star$ requires estimating the channel h_{FR2} or an online exhaustive beam training, which incurs large training overhead, especially for high-frequency FR2 or mmWave channels.

[11], [12] recently pointed out that it is feasible to use the FR1 channel information to predict the beamforming vector in FR2. Moreover, they have also shown that there exists a mapping from the FR1 channels to the achievable rate in FR2 channel.

$$\Phi^n : h_{FR1} \rightarrow R(h_{FR2}, f_n), n = 1, 2, \ldots, \|F\|, \tag{5}$$

and that there exists a mapping function that can be leveraged to obtain the optimal mapping between the FR1 and the FR2 channels. Let's define the optimal mapping function as $\eta_t^\star$ at timestep t as

$$\eta^\star = argmax_{n \in \{n=1,2,\ldots,\|F\|\}} \Phi_{FR1}^n(h_{FR1}) \tag{6}$$

The works in [7]–[10] showed that temporal FR2 channel information can be utilized to predict the optimal FR2 beamforming vector at the next time-step, thereby giving

$$\Psi_t : \{h_{FR2}^{t-m}, h_{FR2}^{t-m+1}, \ldots, h_{FR2}^{t-1}\} \rightarrow R(h_{FR2}^t, f^\star) \tag{7}$$

Combining with Equation 7 with Equation 6, gives us

$$\begin{aligned}\Psi_t : \{&\eta^\star(h_{FR1}^{t-m}), \eta^\star(h_{FR1}^{t-m+1}), \ldots, \eta^\star(h_{FR1}^{t-1})\} \\ &\rightarrow R(h_{FR2}^t, f^\star)\end{aligned} \tag{8}$$

We term this composition Ω mapping from the FR1 channels for m time-steps to the optimal beamforming vector at the latest time-step.

$$\Omega_t : \{h_{FR1}^{t-m}, h_{FR1}^{t-m+1}, \ldots, h_{FR1}^{t-1}\} \rightarrow R(h_{FR2}^t, f^\star) \tag{9}$$

Prior works have shown thatthe high beam training time raises significant challenges with solving these functions using traditional (non-ML) approaches. Hence, we decided to look at machine learning techniques that can help solve this problem. We observe that this is quite similar to the mapping used to solve time series problems [18] in AI/ML. Different ML approaches, like RNNs, LSTMs, and transformers have been used to solve these types of time series problems. We next look at which of these approaches will be suitable for this problem.

IV. Cross Frequency BM using Time Series Data

As we discussed earlier, ML can help to drastically reduce the time associated in BM by incorporating information about the channel and predicting the best *k* beams. Prior work [11], [12] focused on using the FR1 measurements at the current timestep to exploit the correlation between the FR1 and FR2 measurements. We observed that this correlation can be composited with the correlation between measurements at consecutive time steps i.e., the time series data of FR1 measurements helps with better prediction of FR2 beams.

The most common techniques to deal with time series data include RNNs, LSTMs and transformers. We ruled out RNNs as a potential solution since they suffer from vanishing/exploding gradients [6] and hence won't be able to capture the time series nature of this data. We observe comparable accuracy results for transformers and LSTMs (Section V) with the transformer doing marginally better. However, transformers have a faster inference speed due to their ability to replace recurrence to entirely focus on *attention* making them faster while also increasing the parallelism of the model. On the other hand, LSTMs incur a less number of Floating Point Operations (FLOPs) [20], [21] that will result in lower energy utilization to run the model. Our contribution in this paper is to provide a framework to use time series data, so we provide the various options through which this data can be utilized for beam management, while also outlining the tradeoffs associated with the same.

- **Model Structure:** Table I and Table II show the structure of our LSTM and transformer models. We use standard LSTM blocks of the Keras library. The LSTM model consists of 5 LSTM cells and the transformer model consists of 4 transformer blocks (Figure 2) to process sequential data. These layers are followed by dense layers that help to obtain the FR2 data values for the requisite

Layer	Output Shape	Parameters
Input	(10,504)	0
LSTM 1	(10,256)	779264
LSTM 2	(10,256)	525312
LSTM 3	(10,128)	197120
LSTM 4	(10,64)	49408
LSTM 5	(10,32)	12416
Dense	(128,)	4224
Dense	(504,)	65016

TABLE I: Different layers used in the design of LSTM

Layer	Output Shape	Parameters
Input	(10,504)	0
Transformer Block 1	(10,504)	256159
Transformer Block 2	(10,504)	256159
Transformer Block 3	(10,504)	256159
Transformer Block 4	(10,504)	256159
Average Pooling	(504,)	0
Dense	(128,)	64640
Dropout	(128,)	0
Dense	(504,)	65016

TABLE II: Different layers used in the design of transformer

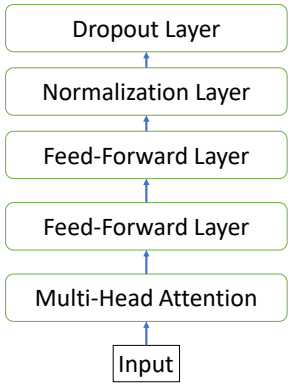

Fig. 2: The structure of every individual transformer block

channels (504 in our current implementation) that help us identify the k best beams for BM.

- **Training Procedure.** Both these models are trained using the data generated by our system-level simulation (more details in Section V). We use sparse categorical cross-entropy as the loss function and the Adam optimizer with a learning rate of 0.0001 for training. The training process for both the models required 40-60 epochs for our dataset.

We describe the steps involved in our cross-frequency beam management algorithm in Algorithm 1. At every step, the FR1 data for the previous 10 timesteps is collated to act as the input to the algorithm.

1) First, the FR1 data is normalized which helps to maintain the numerical stability of the data.
2) Next, we run this data through a trained time series model (transformer / LSTM) to provide the top-k beams.
3) These k beams are then measured, ensuring that a tiny subset of all the beams needs to be measured thus, speeding up the beam management process.
4) Finally, the optimal beam from these k beams is chosen.

Algorithm 1 Beam Management function for cross-frequency prediction

function CROSSFREQUENCYBM(FR1Data)
  $ProcessedData \leftarrow DataProcessing(FR1Data)$
  $topKBeams \leftarrow TimeSeriesModel(ProcessedData)$
  $topKMeasurements \leftarrow Measure(topKBeams)$
  return $ArgMax(topKMeasurements)$
end function

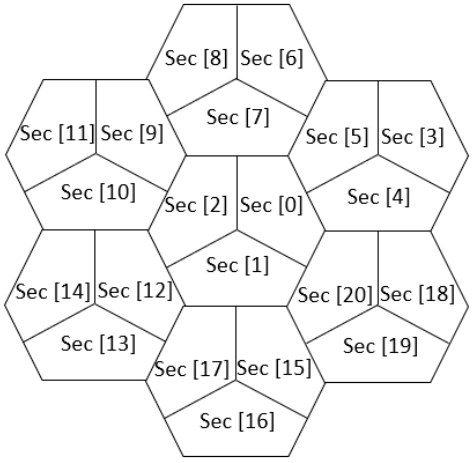

Fig. 3: The layout of different cells used in our simulation environment

The design of our algorithm provides flexibility to choose the value of k which provides a tradeoff between the measurement overhead and accuracy. As we show in Section V, the top-5 beams can provide over 99% accuracy using LSTMs and transformers.

V. Simulation Experiments and Results

In this section, we first briefly present our simulation setup and go through the overall simulation methodology. Subsequently, we discuss the simulation results obtained by carrying out the simulation experiments.

A. Simulation Setup.

As ML-enabled 6G modem chipsets are not yet widely available for experimental design and verification, we resort to MediaTek's custom-built System-Level Simulator (SLS) to generate data using 3D ray tracing and subsequently use 3GPP-specified simulation parameters [22]. We use the two frequencies 4 GHz (for sub-6 GHz/FR1) and 30 GHz (for mmWave/FR2). Our SLS simulates an outdoor environment consisting of 21 cells (as shown in Figure 3) for 1000 timesteps, with each timestep 5 ms apart. At every cell, we consider a base station equipped with two co-located uniform linear arrays for both frequencies. We run our simulations for

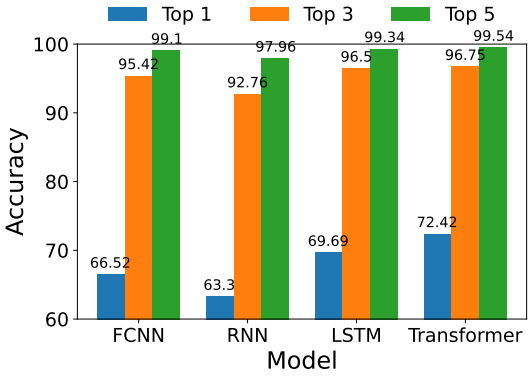

Fig. 4: Top-1, Top-3, and Top-5 accuracy for different models.

a total of 100 UEs. We consider both stationary and mobile UEs (up to 30km/h).

B. Simulation Methodology.

We compare our methodology of using time-series data with the prior work [11], [12] that suggested the use of fully connected neural networks (FCNN) to determine the best beam in FR2. The FCNN model consists of an input layer and four hidden layers all having the same number of neurons. All the layers are activated with the ReLu function [23] followed by the dropout function [24], except the last layer which is activated with the softmax function [25]. In addition to using the FCNN model, we also show the results for an RNN model as a baseline comparing with our proposed LSTM and transformer models. All the models used in our simulation are developed using Keras [26]. While we use the entire models for our simulation study, using Keras provides us the benefit of using the TensorFlow Lite Converter which can easily convert our models to run efficiently on UEs.

We use a 90-10 train-test split for our simulations. During the training phase, we use an 80-20 train-validation split. We compare the models based on the Top-1 accuracy, Top-3 accuracy, Top-5 accuracy, and the number of floating point operations (FLOPs) required for every instance of inference. We obtain the FLOPs for the FCNN and transformer models using the keras-flops library [27]. However, as there is no way to find the FLOPs for RNN and LSTM, we do not share the FLOPs for those models.

C. Results

We measure the accuracy for the scenario using all 21 cells from our simulation environment in Figure 4. We observe that the transformer model outperforms all the other models (FCNN, RNN and LSTM). While RNN performs poorer compared to the FCNN model due to its difficulties in capturing patterns in longer sequences, LSTM does indeed perform better than the FCNN, but slightly worse than a transformer. Further, as we see in Figure 5, a transformer requires far fewer number of FLOPs compared to the FCNN. However, as shown in [20], [21], LSTMs will require even fewer FLOPs.

This difference in trends for the FLOPs for the FCNN and

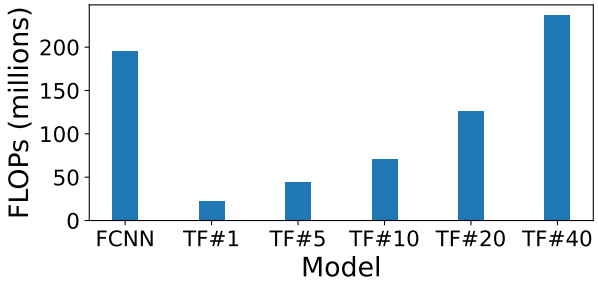

Fig. 5: Comparing the number of Floating Point Operations required for all the different models.

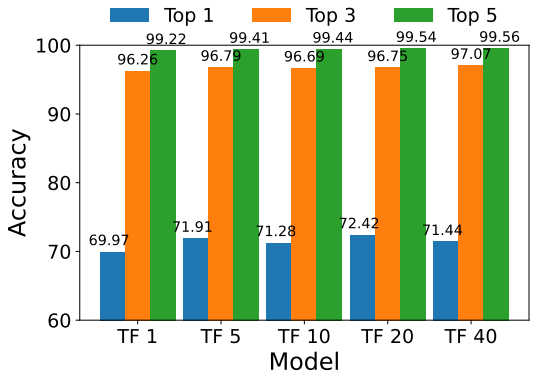

Fig. 6: Top-1, Top-3, and Top-5 accuracy for transformer variants. (TF n refers to TF with window size n)

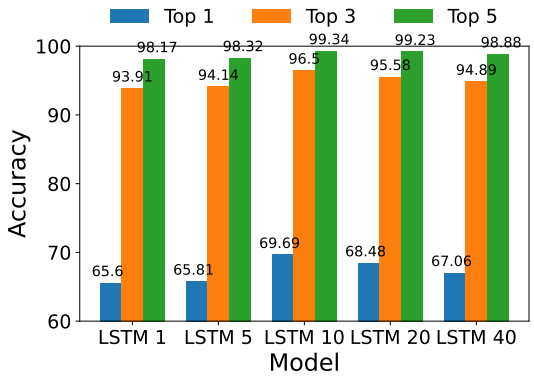

Fig. 7: Top-1, Top-3, and Top-5 accuracy for LSTM variants. (LSTM n refers to LSTM with window size n)

transformer is due to the inherent design of the two models. In the case of an FCNN model, the number of multiply-accumulate operations increases as the number of FLOPs increases quadratically with the increase in the number of neurons [28] whereas, for a transformer, the number of FLOPs increases linearly. Thus, this has greater implications while modifying the size of the model, which is why we advocate the use of a transformer.

Further, we also show a micro-benchmark for the transformer varying the window sizes trying to converge to an

optimal window size. We observe that the accuracy of the model increases rapidly as we increase the window size from 1 to 10, with very little increase from 10 to 20, but then we observe a drop at window size 40, as shown in Figure 6. This tradeoff is important especially since the number of FLOPs required by the model increases as the window size increases as we observed in Figure 5.

Similarly, we also show a micro-benchmark for LSTMs while varying the window size. Similar to the transformer, the accuracy of the model increases while increasing the window size from 1 to 10, but then drops at 20 and 40 (Figure 7). In a way, LSTMs are constrained to utilize less amount of data compared to the transformer.

VI. Conclusion

In this paper, we introduce a new ML framework which explores using time series based transformers and LSTMs to efficiently learn the historical channel measurement of sub-6 GHz wireless channels. As mmWave channel characteristics are more complex and expensive to learn, we subsequently use this sub-6 GHz wireless channel characteristics, learnt over time, to efficiently predict the channel information and BM in high frequency mmWave channels for 6G wireless. The System Level simulation results demonstrate that our time series based learning of beam patterns in sub-6 GHz channels results in up to 99.5% top-5 accuracy for beam prediction and reduces the BM overhead by over 50% in mmWave channels compared to existing approaches. We hope that our work inspires the community to incorporate such learning based on time series data for other aspects of beam management and channel estimations.

References

[1] Y. Heng, J. G. Andrews, J. Mo, V. Va, A. Ali, B. L. Ng, and J. C. Zhang, "Six Key Challenges for Beam Management in 5.5G and 6G Systems," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 74–79, 2021.

[2] Q. Xue, C. Ji, S. Ma, J. Guo, Y. Xu, Q. Chen, and W. Zhang, "A Survey of Beam Management for mmWave and THz Communications Towards 6G," *arXiv preprint arXiv:2308.02135*, 2023.

[3] 3GPP Release 18, "Summary for RAN rel-18 package, 3GPP RAN Plenary 94-e," *Technical Report RAN Chair, RP-213469*, 2022.

[4] I. Aykin, B. Akgun, M. Feng, and M. Krunz, "MAMBA: A Multi-armed Bandit Framework for Beam Tracking in Millimeter-wave Systems," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1469–1478.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] K. N. Nguyen, A. Ali, J. Mo, B. L. Ng, V. Va, and J. C. Zhang, "Beam Management with Orientation and RSRP using Deep Learning for Beyond 5G Systems," *arXiv preprint arXiv:2202.02247*, 2022.

[8] Y. Liu, X. Wang, G. Boudreau, A. B. Sediq, and H. Abou-zeid, "Deep Learning Based Hotspot Prediction and Beam Management for Adaptive Virtual Small Cell in 5G Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 1, pp. 83–94, 2020.

[9] Y. Guo, Z. Wang, M. Li, and Q. Liu, "Machine Learning Based mmWave Channel Tracking in Vehicular Scenario," in *2019 IEEE International conference on communications workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[10] A. Ö. Kaya and H. Viswanathan, "Deep Learning-based Predictive Beam Management for 5G mmWave Systems," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–7.

[11] M. Alrabeiah and A. Alkhateeb, "Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6GHz Channels," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[12] M. S. Sim, Y.-G. Lim, S. H. Park, L. Dai, and C.-B. Chae, "Deep Learning-Based mmWave Beam Selection for 5G NR/6G With Sub-6 GHz Channel Information: Algorithms and Prototype Validation," *IEEE Access*, vol. 8, pp. 51 634–51 646, 2020.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17*, pp. 6000–6010, 2017.

[14] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A Survey of Millimeter Wave (mmWave) Communications for 5G: Opportunities and Challenges," *Wireless networks*, vol. 21, no. 8, pp. 2657–2676, 2015.

[15] 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.

[16] M. Q. Khan, A. Gaber, P. Schulz, and G. Fettweis, "Machine Learning for Millimeter Wave and Terahertz Beam Management: A Survey and Open Challenges," *IEEE Access*, vol. 11, pp. 11 880–11 902, 2023.

[17] K. Ma, Z. Wang, W. Tian, S. Chen, and L. Hanzo, "Deep Learning for Beam-Management: State-of-the-Art, Opportunities and Challenges," *arXiv preprint arXiv:2111.11177*, 2021.

[18] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, G. Rasool, and R. P. Ramachandran, "Transformers in Time-series Analysis: A Tutorial," *arXiv preprint arXiv:2205.01138*, 2022.

[19] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," *IEEE journal of selected topics in signal processing*, vol. 10, no. 3, pp. 436–453, 2016.

[20] W. Li, J. Qin, C.-C. Chiu, R. Pang, and Y. He, "Parallel Rescoring with Transformer for Streaming On-Device Speech Recognition," *arXiv preprint arXiv:2008.13093*, 2020.

[21] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, "Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 744–751.

[22] 3GPP Release 14, "Study on International Mobile Telecommunications (IMT) parameters for 6.425 - 7.025 GHz, 7.025 - 7.125 GHz and 10.0 - 10.5 GHz," *3GPP Technical Report*, 2018.

[23] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv preprint arXiv:1803.08375*, 2018.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[25] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[26] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[27] Keras Flops. [Online]. Available: https://pypi.org/project/keras-flops/

[28] M. Hollemans, "How fast is my model?" https://machinethink.net/blog/how-fast-is-my-model/.