Deep Learning for Fast and Reliable Initial Access in AI-Driven 6G mmWave Networks

Tarun S. Cousik, Graduate Student Member, IEEE, Vijay K. Shah, Member, IEEE, Tugba Erpek, Member, IEEE, Yalin E. Sagduyu, Senior Member, IEEE, and Jeffrey H. Reed, Life Fellow, IEEE

Abstract—We present DeepIA, a deep neural network (DNN) framework for fast and reliable initial access (IA) for artificial intelligence (AI)-driven 6G millimeter wave (mmWave) networks. DeepIA reduces the beam sweep time compared to a conventional exhaustive search-based IA process by utilizing only a subset of the available beams. DeepIA maps received signal strengths (RSSs) obtained from a subset of beams to the beam that is best oriented to the receiver. In both line of sight (LoS) and non-line of sight (NLoS) conditions, DeepIA reduces the IA time and outperforms the conventional IA’s beam prediction accuracy. We show that DeepIA’s accuracy saturates with the number of beams used, and also depends on the particular choice of the beams used. The choice of beams that are selected is consequential and improves accuracy by up to 35% and 70% in NLoS and LoS channels. We find that, averaging multiple RSS snapshots further reduces the number of beams needed and achieves more than 95% accuracy in both LoS and NLoS conditions. We introduce interference into our models to understand impact on performance. Finally, we evaluate the beam prediction time of DeepIA through embedded hardware implementation and show the improvement over the baseline approach.

Index Terms—Beyond 5G, 6G, machine learning, deep learning, mmWave, initial access, beam sweeping, feature selection.

I. INTRODUCTION

A SHIFT to millimeter wave (mmWave) frequencies has been proposed to satisfy the ever increasing bandwidth requirements of wireless communication systems. This shift has forced the adoption of highly directional antennas and/or arrays (HDAs) to combat the path loss associated with these high frequencies.

The adoption of HDAs in current generation 5G systems introduced an alignment window challenge, where the transmitter and the receiver would need to ensure their beams point at one another in order to avoid connection dropouts. Beam alignment is carried out during initial access (IA) and it becomes increasingly difficult when narrower beams are used. Additionally, IA needs to be repeated more frequently due to potential blockage effects that may otherwise result in loss of beam alignment. Thus, this challenge is further exacerbated in the anticipated 6G systems that use larger arrays and narrower beams in the Terahertz (THz) regime in order to develop networks that are capable of handling next-generation applications and services like holographic projection, digital twins, sub-ms latency communications, massive XR/VR applications.

Given that the 6G standardization effort is in a state of nascent, we have developed our work around 5G systems, with the hope that when the standardization effort for 6G matures, our framework could be considered as a guide for IA in 6G systems.

5G NR allocates up to a maximum of 64 synchronization signal blocks (SSBs) for each beam sweep in the mmWave bands. These SSBs contain the relevant information required to establish synchronization between devices. The time allocated to sweep 64 beams is 5 ms and the beam sweeping process occurs periodically every 5-20 ms [1],[2]. Each SSB is typically mapped onto a unique beam (sector) in the spatial domain. Consequently, this fixes the number of beams that we need to evaluate before choosing an optimal beam. However, the coherence time experienced by mobile platforms is much smaller than 20 ms.

The IA time consists of two components, (i) time for beam sweeping where the received signal strengths (RSSs) for different beams are measured, and (ii) time for beam prediction where the beam for a given transmitter-receiver pair to communicate with is identified. The beam sweeping time dominates the overall IA time, and therefore it is essential to improve the IA time by utilizing fewer beams. Transmit power consumption also decreases by utilizing fewer beams in IA.

However, conventional beam sweeping (CBS) approaches sweep all beams in general. If they are extended to select the best beam based on RSSs from a reduced subset of beams, this selection may not be necessarily accurate since CBS cannot predict the correct beams for users located in beams that are not present in the subset that is swept. Hence, the CBS approach cannot realize the advantages that arise from using a subset of beams. In this work, exhaustive beam search is referred to as CBS.
In general, the mapping from the RSSs measured for a subset of beams to the best beam is a complex process due to various channel, antenna, and network topology effects. This limitation motivates us to understand if and how a data-driven approach could be leveraged to predict best beams that include ones that are not part of the beam sweeping process.

Machine learning (ML) provides automated means to learn from spectrum data and perform complex tasks such as spectrum sensing [3], signal classification [4], anti-jamming [5], and waveform design [6]. Amongst the various ML techniques, deep learning has been shown to be a viable solution to capture high-dimensional representations of spectrum data [7] and is one of the most popular ML techniques.

The most widely-expected technological breakthrough for 6G is expected to come from its ability to natively support AI/ML to better optimize the network performance. Independent of the 6G standardization process, the Open Radio Access Network (O-RAN) effort is trying to bolster current generation 5G networks by enabling AI/ML capabilities at the Radio Intelligence Controller (RIC), i.e., either Non-Real-Time RIC (NRT-RIC) or Near-Real-Time RIC (RT-RIC) [8]. This O-RAN effort could serve as a precursor that makes mature AI/ML implementations in 6G networks a reality.

Motivated by the recent advances in algorithmic techniques as well as the encouraging software ecosystem support provided in the forms of NRT-RIC and RT-RIC for AI/ML applications of the future, in this work we present DeepIA, a deep learning-based solution for IA that reduces the beam sweep time by utilizing measured RSSs from a subset of all available beams and maps them to the best oriented beam in the entire set of beams. The DNN trained in DeepIA learns to associate the correct/best oriented beam between the transmitter and receiver with the RSS measured by the receiver. Hyper-parameters of the DNN are carefully selected to avoid both under-fitting and over-fitting. For both line-of-sight (LoS) and non-line-of-sight (NLoS) mmWave channel conditions, we compare CBS and DeepIA using various number of beams and show that, DeepIA not only provides a faster IA scheme but also a more reliable one in terms of prediction accuracy. For example, DeepIA predicts the optimal beam with close to 95% accuracy in LoS conditions by sweeping only 7 out of 24 beams. The accuracy of the CBS for the same setting is limited to 24%. While a CBS approach would traditionally not use a subset of beams, nevertheless, we implement it in this fashion to achieve a fair comparison between the accuracy and input size of the two approaches.

We identify that the DNN’s accuracy is not just dependent on the number of beams used in its input, but is also dependent on the specific combination of beams that are used. In other words, if we fix the number of DNN inputs, certain combinations of beams are more effective in accurately predicting the best oriented beams. We propose using a sequential feature selection (SFS) algorithm to help identify better performing combinations of beams for DeepIA’s input. We find that using beams predicted by the SFS algorithm further appreciably improves the accuracy of DeepIA in the LoS case. In LoS conditions, it offers enhanced and robust improvement in accuracy for situations where 8 or fewer beams are utilized. Not much improvement is observed by SFS in NLoS cases. Motivated by the performance gain obtained by feature selection in LoS situation, we investigate alternate solutions to improve the NLoS performance and find that using the averaged RSSs as input further improves the performance in both LoS and NLoS conditions. For instance, a DeepIA network that uses the averaged inputs from 25 measurements is able to achieve a 95% accuracy with just 7 beams in NLoS conditions.

We validate the practicality of our approach by computing the time needed to run DeepIA’s DNN through embedded implementation with the Field Programmable Gate Array (FPGA) and comparing it against the computation time for the CBS. We find that both computations are on the order of a few microseconds. The beam sweep time is on the order of milliseconds for CBS and a few hundred microseconds for DeepIA depending on the channel conditions. This analysis supports the claim that DeepIA decreases the overall time required for initial access.

DeepIA is poised to reduce the overall operational costs as it reduces the number of transmissions/beams each transmitter has to sweep through during IA. Consequently, it also decreases the temporal and spatial footprint of the signal over the air which reduces the interference from/to other ongoing transmissions (such as in spectrum sharing scenarios) as well as the probability of detection/intercept. Thus, DeepIA could potentially improve the resiliency against out-of-network interference and jamming.

The rest of the paper is organized as follows. Section II discusses related work. Section III describes the system model and the assumptions. Section IV describes in detail the DeepIA framework. Section V provides an analysis on prediction accuracy and computation time taken by CBS and DeepIA in LoS and NLoS mmWave channel models. Section VI concludes the paper. A preliminary version of the material in this paper is available in arXiv [9].

II. RELATED WORK

Conventional IA involves an exhaustive beam search. A predefined number of beams divide the azimuth plane into sectors, with each beam covering a unique sector [10], [11], [12]. A common implementation of the exhaustive search involves the transmitter cycling through every sector while the receiver runs in a quasi-omnidirectional mode and listens for the transmitted signal. The receiver (e.g., a user equipment (UE) in 5G) records the RSS for each transmit beam and finally sends back the beam with the highest RSS to the transmitter (e.g., gNodeB in 5G). The transmitter then turns on this particular beam, and the receiver performs the same beam sweeping process from its side to determine its best sector. This exhaustive search is computationally inefficient. If $N_{BS}$ and $N_{UE}$ represent the size of the codebook on the BS and UE, for the exhaustive beam sweep, the total number of beams that are swept ($N_{ex}$) is given by $N_{ex} = N_{BS} \times (N_{UE} + 1)$. Unlike omnidirectional sub-6 GHz systems, this IA process has to be
periodically repeated in mmWave systems in order to ensure that there is no misalignment between transmitter-receiver pairs over time. Another variant of the exhaustive search runs narrow transmitter beams against narrow receiver beams [10].

Iterative search technique uses a combination of wide and narrow beams at the transmitter and receiver to minimize the computational cost of the exhaustive search at the expense of decreasing the detection accuracy [11]. In addition to this, since only a partial set of the antenna elements are used to create the beams, it can decrease the range over which these wide beams can be used. If the number of wide beams at the BS is given by \( N_w \), the number of narrow beams on the UE is given by \( N_{UE} \) (\( N_{UE} = 0 \) for omni-directional case) and the number of narrow beams contained in a wide beam of the BS is given by \( N_n \), then the total number of beams for the iterative approach \( (N_{iter}) \) is given by \( N_{iter} = N_w \times (N_{UE} + 1) + 2 \times N_n \).

A hybrid technique for IA process was introduced in [11], where first the iterative process is performed and then the receiver sends the uplink signals in the best beam. The transmitter finds its best narrow beam after cycling through all its narrow beams against the best receiver beam. The total number of beams for the hybrid approach \( (N_{hy}) \) is given by \( N_{hy} = N_w \times (N_{UE} + 1) + N_n \).

This hybrid IA technique performs equally with the iterative search in terms of accuracy but typically utilizes approximately 80-90% of the time resources. Both the iterative and hybrid approaches are handicapped by the dependency to use two sets of beams. As the number of beams in a system increases, the cost of the memory (typically SRAM based solution) needed to store corresponding antenna element weights increases rapidly, thus discouraging commercial applicability of these types of solutions. Additionally, while they are faster than the exhaustive beam sweep, at their very best they tend to perform as well as exhaustive beam search [11]. Thus, from a performance perspective, an improvement over CBS automatically translates to an improvement over hybrid and iterative techniques.

To understand the impact of timing/number of beams swept in this work, consider using \( (N_w, N_n) \) as its beam search combination where \( N_w \) corresponds to the number of wide beams and \( N_n \) corresponds to the number of narrow beams scanned in each wide beam. Note \( N_{UE} = 0 \) as we consider omni-directional receivers in this work as detailed in Section III. Assume \((0,24), (2,12), (3,8), \) and \((4,6)\) are used as the beam search combinations. Note that \((0,24)\) corresponds to the CBS scheme, where there are no wide beams and all of the narrow beams are swept.

We assume that the time taken to generate/sweep a single beam (whether its wide or narrow) is the same and given by \( t_b \). The best case scenario in terms of beam scanning time would be the \((4,6)\) scenario where a total time of \(10t_b\) would be incurred, while the performance would still be limited to the accuracy of CBS. In our work, we note that we reach considerably higher accuracy compared to CBS (thereby hybrid and iterative schemes) while using \(5 - 7\) beams (i.e., beam scanning time of \(5t_b - 7t_b\), which is a reduction of 30-50% of the overall sweep time relative to the hybrid approach (i.e., \(10t_b\) in the best case scenario). Hence in this work, we limit the performance comparison of DeepIA against the CBS technique, since 1) we do not use a codebook with two sets of beams (that the iterative and hybrid approach use) and 2) from Section V, we note that from a performance and timing perspective DeepIA does better than CBS (therefore Hybrid) and Hybrid (therefore CBS) techniques, respectively.

Using multi-beam analog beamformers which simultaneously overlay several narrow beams in the spatial domain for IA was proposed in [12]. While the transmit time is reduced, computational and temporal requirements for the back-end processing need to distinguish the beams and detection accuracy remains unclear. [13] proposed the use of compressed sensing to leverage the change gradient of RSS/signal-to-noise ratio (SNR) to detect the best beam from a subset of beams. They found that their approach scaled appreciably with an increase in the sparsity of a channel. [14] leveraged the DeepIA algorithm as a means of determining the action-policy in a reinforcement learning framework called DeepIA-DRL. Additionally, they also propose a new algorithm called RapidIA, that attains beam orientation with a single transmission. This is achieved by leveraging transmissions from/to several coordinated base stations/access points (APs) to transmit to the user which roughly translates to transmitting multiple beams but from different spatial regions. It shows good promise in scenarios whenever multiple BSs can exist within each other’s broadcasting range. The subsequent related work that are discussed leverage various forms of context-based information to enhance and accelerate the IA process. The average discovery time by leveraging knowledge obtained from real time arrival statistics of incoming users was reduced in [15]. Gated recurrent neural networks (RNNs) were used in [16] to predict the sequence of the beams swept during IA. Using call records to find the user’s location, [16] used this information to determine the number of users in each sector such that the beams are swept in a descending order of users in each sector, i.e., sector with the highest number of users goes first and so on. Pseudo-directional patterns were considered in [16] since the operations are in the sub-6 GHz band. Leveraging channel state information (CSI) as the input variable for a DNN was investigated in [17] to predict the \(k\) best beams for IA. A context-aware IA approach was proposed in [18], where global positioning system (GPS) data is fed to the UE and BS to help predict the optimal beam choice for IA. The main challenge for this approach is to generate the reliable GPS data for various scenarios including indoors. A context aware online learning algorithm that leverages aggregated received data alongside rough estimates of the user’s location information was developed for beam selection in vehicular mmWave networks [19]. Authors in [20] developed a beam alignment process that leveraged light sensors and changes in light intensity to accurately point beams in the appropriate direction.

### III. System Model

We consider a 6G mmWave network that consists of a directional transmitter and omnidirectional receivers as shown in Fig. 1. Without loss of generality, we consider a 2D plane
where \( R \) receivers are uniformly distributed in a square area and the transmitter is located at the center. No receiver is placed within a 1 m radius of the transmitter to support the close intercept (CI) model explained in Section III-B. First, the transmitter performs the IA process. While it is not explicitly shown, it is assumed that the subsequent processes in this paper are repeated again from the receiver’s side as is the case with any IA strategy outlined in the related work.

### A. Transmitter and Antenna Array Characteristics

We design a \( 10 \times 10 \) antenna array at the transmitter which generates a beam width of about \( 15^\circ \) using the standard planar array formulation [21]. This array points at an azimuth angle \( \phi = 0^\circ \) and an elevation angle \( \theta = 90^\circ \). Since we consider a 2D scenario, we take the azimuthal slice of the array factor at \( \theta = 90^\circ \). We then subtract 10 dB from this standard array slice for azimuth angles between \( 180^\circ \) and \( 360^\circ \) as shown in Fig. 2. This modification serves two purposes. First, antenna designers routinely modify the array/antenna patterns to ensure minimal backlobe radiation [22]. The goal is to avoid transmitting in unnecessary directions, reduce electromagnetic interference to back-end electronics and avoid interfering with users in the backlobe’s direction. Second, using a standard \( 10 \times 10 \) planar array pattern severely degrades the performance of CBS because of pattern symmetry. When the array factor of the backlobe has the same magnitude as that of the front lobe, the CBS process gets confused on which beam to pick. This modified array pattern is consistently used for all of the sectors.

### B. Channel Models

In this work, we consider both LoS and NLoS mmWave channels. Specifically, we use the CI path loss model, where at distance \( d \), the path loss in dB is given by

\[
PL(d) = PL(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{1}
\]

where \( PL(d_0) = 10\log_{10}\left(\frac{\lambda^2}{4\pi d_0}\right)^2 \), \( \lambda \) is the signal wavelength, \( d_0 = 1 \) m, \( n \) is the best fit minimum mean squared error path loss exponent (PLE) (note \( n = 4.5 \) for NLoS conditions and \( n = 1.9 \) for LoS conditions) and \( X_\sigma \) is the shadow factor representing large scale signal fluctuations (note \( X_\sigma \) is a zero-mean Gaussian random variable with a standard deviation of 10 for NLoS conditions and 1.1 for LoS conditions [23]). The shadowing factor is a random process that holds a second order wide-sense stationarity. This ensures that while there may be variation in individual channel realizations, the overall channel statistics do not vary during testing, validation and training phases. If the channel statistics keeps changing over time, then the network would need to be retrained at the speed (or greater) than the speed of changes in channel distribution.

The LoS and NLoS parameters in [23] were obtained from empirical results, thus meriting applicability in practical situations. Other alternative path loss models derived from ray tracing simulations have also been presented in literature. While these ray tracing models offer a complex environmental model, the models they deliver are specific to the locations and situations they are simulated for. Additionally, these models are better suited for experiments that require additional parameters such as the number of reflections and delay characteristics.

### C. Overview of the Baseline IA

The baseline IA that we refer to is CBS approach that uses an exhaustive search (i.e., sweep all beams [10]. We define \( N = \{1, \ldots, N\} \) as the set of all \( N \) beams that the transmitter can sweep. Let \( r_k \) denote the \( k \)th receiver, where \( k \in \{1, 2, 3, \ldots, R\} \), and \( RSS_{ik} \) denote the RSS corresponding to the \( i \)th beam between the transmitter and the receiver \( r_k \). In CBS, all \( N \) beams are swept sequentially and \( i_k \)th beam which provides the highest RSS is selected, i.e., for receiver \( r_k \), the selected beam is \( i_k = \arg\max_{i \in N} RSS_{ik} \). Algorithm 1 shows the baseline CBS algorithm.

### D. Overview of DeepIA

While a detailed description of DeepIA’s inner workings is provided in Section IV, it is beneficial to know that DeepIA uses the global information of receiver positions during the training phase. Given that the receiver’s position is known to
TABLE I
DEEP NEURAL NETWORK ARCHITECTURE

| Layers     | Number of neurons | Activation function |
|------------|-------------------|---------------------|
| Input      | $M = 1, 2, 3, \ldots, 24$ | –                   |
| Dense 1    | 32                | ReLu                |
| Dense 2    | 64                | ReLu                |
| Dense 3    | 128               | ReLu                |
| Dense 4    | 64                | ReLu                |
| Dense 5    | 32                | ReLu                |
| Output     | $N = 24$          | Softmax             |

Algorithm 1: Baseline CBS Algorithm

**Input:** Receiver $r_k$

**Output:** Selected beam $i_k$

1. for $i \in N$ do
2. Measure $RSS_{ik}$
3. end for
4. $i_k = \text{argmax}_{i \in M} RSS_{ik}$

the transmitter during the training stage, the best beam is calculated based on the angle between them; this process is described in Section IV-B. The DNN is trained by feeding the RSS values from a subset of beams, $M$ (where $M$ is the cardinality, i.e., $M = |M|$), as the input, where $M \subseteq N$.

To support fast implementation on an embedded platform (see Section V-F), the DNN uses a simple feedforward neural network (FNN) architecture. The input layer consists of $M$ neurons, each receiving input from one beam from $M$. At the output layer, there are $N$ output neurons corresponding to the $N$ beams and their magnitudes denote the respective probabilities of being the optimal beam. During training, the label, i.e., the best beam for a given transmitter and receiver position, is determined based on the angle between the transmitter and receiver instead of the measured RSS values. The subsets of $M$ beams are uniformly sampled from $N$ and the same subset is used during both the training and test times in our simulations.

IV. DEEP LEARNING FRAMEWORK OF DEEPIA

This section describes the details of DeepIA’s deep learning framework; namely, DNN architecture, beam mapping and label generation, and training and testing phases.

A. Deep Neural Network Architecture

The DNN architecture of DeepIA consists of seven layers including the input and output layers. The input layer has $M$ neurons, and the output layer has $N$ neurons with each neuron providing a likelihood score. The hidden layers have 32, 64, 128, 64 and 32 neurons, respectively. These hidden layer structures are determined after tuning the hyperparameters based on the training and validation performance. In simulations, we set $N = 24$ and select $M$ between 1 and 24. All the hidden layers are activated by a ReLu activation function and Softmax activation function is used at the output layer. The number of neurons in each layer and the activation functions used are shown in Table I. Each layer’s output except that of the output layer is batch normalized before passing it on to the next layer. This architecture was selected for after comprehensive experimentation with various other architectures, after the validation accuracy showed that the other models either over-fit or under-trained the training data.

B. Beam Mapping and Label Generation

As shown in Fig. 3, the angle between the $k$th receiver $r_k$ and the transmitter is denoted as $A_k$. This angle represents the relative orientation between any receiver position and transmitter. $A_k$ can take values between $0^\circ$ and $359.99^\circ$, and is rounded to the lowest integer. Transmitter $A_k$ for every receiver $r_k$ is then mapped to its corresponding beam as follows. When $N = 24$, angles $\{(-8^\circ, 7^\circ], (7^\circ, 22^\circ], (22^\circ, 37^\circ], \ldots, (337^\circ, 352^\circ]\}$ are mapped to beams 1, 2, \ldots, 24 respectively. Note that $-a^\circ$ is the same as $360^\circ - a^\circ$. This is performed by setting $i_k^* = \left\lfloor \frac{A_k + x}{360^\circ} \right\rfloor$, where $i_k^*$ represents the true beam sector that receiver $r_k$ should correspond to. $i_k^*$ also corresponds to the labels that we use against our training, test and validation features. In addition, we use $i_k^*$ in evaluating the accuracy of both the CBS and DeepIA. In practical situations, wireless network providers can obtain these labels from propagation modeling tools that model the localized environment and/or through measurement campaigns in the local area.

C. Deep Neural Network Training

DeepIA uses RSS values from $M$ beams as inputs. The typical training time varies depending on the number of beams, number of epochs as well as the hardware used in training. In simulations, we use GoogleColab Pro to train the DNNs and for any given input of $M$ beams, it takes roughly 2 minutes to train a DNN. Since training will be performed offline, the training time and computational resources can be ignored in subsequent performance comparisons. The input variables are transformed to linear units and normalized using the maximum of the absolute value. Note that the values in Tables I

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We show an example of training data entries in Table II.

| Receiver | Beam 1 | ... | Beam M | True Beam |
|----------|--------|-----|--------|-----------|
| 1        | -30    | ... | -46    | 1         |
| 2        | -51    | ... | -24    | 5         |
|          | -      | ... | -      | -         |
| \( R_{tr} \) | -65   | ... | -68    | 2         |

Table III shows an example of test data entries.

| Receiver | Beam 1 | ... | Beam M | Predicted Beam | True Beam |
|----------|--------|-----|--------|----------------|-----------|
| 1        | -35    | ... | -46    | 1              | 1         |
| 2        | -51    | ... | -64    | 5              | 5         |
|          | -      | ... | -      | -              | -         |
| \( r_k \) | -27   | ... | -71    | 4              | 5         |
|          | -      | ... | -      | -              | -         |
| \( R_{te} \) | -25  | ... | -28    | 2              | 2         |

and III are only written in dB scale, to aid in visualizing the procedure. In reality, the DNN has input features that range from 0 to 1. The DNN is trained to learn to map \( \{ RSS_{ik}\}_{i \in \mathcal{M}} \) values to the corresponding \( i_k \) values. We show an example of how DNN training set would look like in Table II. The receiver column corresponds to the index of different receivers. The true beam column corresponds to the best beam that is determined based on the angle between the transmitter and receiver instead of the measured RSS values.

We separate \( R \) receivers and their corresponding data into 3 sets for training, validation and testing. The number of receivers used for training, validation and testing are given by \( R_{tr}, R_{val} \) and \( R_{te} \), respectively. Therefore, \( R = R_{tr} + R_{val} + R_{te} \). The output of training for beam set \( \mathcal{M} \) is the trained DNN model \( \text{DeepIA}_{\mathcal{M}} : \{ RSS_{ik}\}_{i \in \mathcal{M}} \rightarrow i_k \in \mathcal{N} \).

DNN models are trained for \( M = 1,224 \) and 10 epochs are used to train each model. The backpropagation algorithm is used to train the DNN by minimizing the cross entropy loss function. We experimented with various learning rates and finally chose an initial learning rate of \( 10^{-2} \) and a final learning rate is 0.2. A batch size of 1024 is consistently maintained for all the models. We observed that while the training time for smaller batch sizes (i.e., 512,128,64) took longer they offered no benefit in terms of validation accuracy. Larger batch sizes (i.e., 4096,8192,16384) showed deterioration in performances and we settled for 1024. The hyperparameters of the DNN are determined after evaluating the validation accuracy and epochs where the training loss saturated. After checking a variety of optimizers including ADAM, Adamax, Stochastic Gradient Descent, and Adabound, our results indicate that Adabound converges to the minimum of the loss function the quickest, while the accuracy is practically identical for Adabound and Adam. Therefore, the Adabound optimizer [24] is used during training.

### D. Deep Neural Network Testing

The RSS values from the same \( M \) beams used during training are measured and fed to the DNN model for receiver indices that have not been encountered in the training process. The index of the output with the highest probability is selected as the best spatially oriented beam. This process is shown in Algorithm 2. Table III shows an example test input and output with \( \mathcal{M} \) inputs. For a successful prediction to occur, the predicted beam should exactly match with the true and vice-versa.

#### E. Selection of Features

As discussed above, the input to the DeepIA framework is the subset of beams, \( \mathcal{M} \). Thus, an important question arises at this point on the selection of the beams in the subset \( \mathcal{M} \). We investigate whether certain beams matter more than others when it comes to the actual selection of the beams.

In order to investigate this question, for any subset \( \mathcal{M} \) of beams, we utilize a baseline set that consists of \( M \) beams that are spatially equally separated. We call this set as Manually Selected Beams (MSB) denoted by \( \mathcal{M}_{MSB} \). The MSBs for a given number of beams \( M \), can be calculated as

\[
\mathcal{M}_{MSB}(j) = \left\{ \left\lfloor \frac{24}{M} \right\rfloor | j \in \{0, 1, 2, \ldots, M - 1\} \right\},
\]

where \( \lfloor \cdot \rfloor \) is the floor function and \( \mathcal{M}_{MSB}(j) \) denotes the \( j \)th beam in the set \( \mathcal{M}_{MSB} \).

It is important to identify the best combination of input beams to increase prediction accuracy while minimizing the total number of beams used. In fact, the whole domain of feature selection is developed with the objective of identifying a minimal set of features for a given dataset, that when used, typically minimizes the objective function. Feature selection can be categorized into three basic approaches:

**Filter Methods:** These methods rely on processing input variables with respect to some criterion (e.g., mutual information, Pearson’s correlation, Fischer score, etc.), ranking the processed data and then selecting the \( M \) desired number of input variables that obtain the highest \( M \) ranks [25].

**Embedded Methods:** These methods incorporate the feature selection process as part of the training process; regularization can in essence be thought of as feature selection. Typically algorithms like Lasso/L1 norm based regularization introduce feature sparsity while approaches like Ridge regression help regulate performance in the presence of large outliers [26].

**Wrapper Methods:** Wrapper methods typically involve cycling through the various combination of input variables in the learning algorithm in question, measuring a predetermined performance in the presence of large outliers [26].
V. PERFORMANCE ANALYSIS

A. Simulation Setting

We consider a two-dimensional scenario where transmitters and receivers all lie on the same plane. The location of the transmitter is fixed at (0,0) and the receiver’s X and Y positions each uniformly randomly distributed between $-25$ m and $25$ m. This bounds the simulation cell to an area of $50 \times 50$ m. The transmit power is set to $20$ dBm.

A total of $10^6$ receiver positions are generated, which act as data samples. Note, we do not assume all these receivers are simultaneously operating. Out of total $10^6$ data samples, $65\%$ are used to train DeepIA, $15\%$ of data samples are used for validation, and $20\%$ of data samples are used for testing.

B. Comparison Approaches

We evaluate the performance of DeepIA and compare it to that of CBS in terms of both accuracy and beam prediction time, when varying the subset of beams out of total beams.

Prediction Accuracy: The prediction accuracy is computed as the ratio of receivers for which the predicted beam matches the true beam, namely

$$\text{Prediction Accuracy} = \frac{\sum_{k=1}^{R_{\text{col}}} 1(i_k = \hat{i}_k)}{R_{\text{col}}} \times 100, \quad (3)$$

where $1$ is the indicator function ($1(E) = 1$ if $E$ holds and $0$ otherwise).

IA Time: The IA time is the total time incurred in predicting the correct beam for a certain transmitter-receiver position. It constitutes two parts described below.

1) Beam Sweep Time – the time taken to sweep a desired number of beams. The beam sweep time is directly proportional to the number of beams swept. For instance, 5G NR [1] allows a duration of one half frame or $5$ ms for sweeping across a total of $64$ beams. Using these numbers, it should take $1.875$ ms to sweep $24$ beams, $0.9375$ for $12$ beams, $0.675$ ms for $8$ beams, and so on. In simulations, we consider the same subset of beams to sweep for both DeepIA and CBS. Therefore, we compare the accuracy performance under the same beam sweep time.

2) Beam Prediction Time – the time it takes to process the collected RSS data and predict the beam it belongs to. In CBS, this roughly involves calculating a maximum operation on the RSS collected from $N$ beams. In DeepIA, this involves calculating the time it takes to run the DNN model using $M$ inputs in order to predict the beam a given receiver belongs to.

We show the implementation of the SFS approach for selecting optimal $M$ subset of beams (out of $N$ beams) in Algorithm 3. As shown in line 1, an empty beam subset $\mathcal{M}$ is initialized. Then, beams (or features) are iteratively added to the subset $\mathcal{M}$, on the condition that the beam that is included obtains the highest prediction accuracy amongst all the other available beams (See lines 4 - 14). Beams that are selected and added based on these criteria are permanently retained in the $\mathcal{M}$ and this process is repeated until the required number of features ($M$) are added (See line 2).
In the time analysis section, we discuss the benefits of utilizing DeepIA in terms of time taken for computation.

C. Prediction Accuracy Analysis

Since the exhaustive beam search in CBS relies on scanning the entire spatial plane, its accuracy continuously deteriorates as fewer beams are utilized. The user distribution contributes to the slightly non-linear prediction accuracy curve seen in Fig. 4. In addition, the shadow fading factor in LoS condition also causes incorrect prediction in a few users (explained in later part of this section), thus accuracy is 95% with 24 beams as shown in Fig. 4. On the other hand, in LoS conditions DeepIA demonstrates a more robust and reliable prediction accuracy over the range of beams that are swept as shown in Fig. 4. For example, in the LoS scenario, setting \( M = 7 \) achieves close to 100% prediction accuracy. The performance of DeepIA deteriorates when less than 7 beams are utilized.

The relevance of feature selection becomes evident as we observe that the prediction accuracy by DeepIA is better with a 5 beam input than a 6 beam input, when these input beams are selected based on the MSB criterion. Recall, MSB refers to the Manually Selected Beams which are equally separated from one another as described in (2). If each feature/beam is considered equally important, this finding is obviously counter-intuitive. In other words, DeepIA is able to perform better with 5 beams selected using the MSB criterion, because the ‘quality of the information’ in those beams is more valuable to DeepIA in determining the prediction accuracy. When beams are selected based on the SFS algorithm, we find that the prediction accuracy for DeepIA models that take inputs of 1 to 6 beams are all further improved relative to the MSB case. Furthermore, we note that the beams selected based on the SFS criterion correspond to a rough upperbound to what the MSB can achieve. The maximum accuracy achieved with a 24 beam CBS in LoS conditions is achieved using a DeepIA model trained on just 5 SFS selected beams. In LoS scenario, SFS is able to reasonably discern ‘quality’ beams for DeepIA. Therefore, using more than 8 adds very little benefit in terms of accuracy in the DNN approach under LoS conditions.

With the introduction of the severe shadow fading in the NLoS mmWave channels, DeepIA’s performance deteriorates compared to LoS case as shown in Fig. 5. Nonetheless, DeepIA still outperforms CBS. In fact, the gap in performance increases as the number of input beams increases from 1 to 24. The NLoS fading factor adds considerable amount of fluctuations in the collected RSS values and this in effect leads to the following two phenomena which contribute to the rapid drop in CBS accuracy under NLoS conditions.

Since the fluctuations are random, it can increase or decrease the value of the ‘clean’ RSS measurements where ‘clean’ refers to the calculated RSS values based on the path loss for a given distance excluding the shadow fading factor. The clean RSS values from users positioned in between the main lobes of two adjacent beams decreases rapidly as the user position changes closer to the boundary/intersection of the two beams. Fluctuations due to shadowing can cause the RSS of the incorrect adjacent beam to exceed the RSS of the correct beam, thus resulting in inaccurate predictions for the CBS. These fluctuations directly impact the performance of CBS for both LoS and NLoS conditions by boosting the RSS of the incorrect beam, thus adding another possible culprit that reduces prediction accuracy in CBS.

Fig. 5 also shows that the introduction of sequential feature selection (SFS) does not improve the overall performance by reasonable measure, but the earlier observation we made on the SFS-based DeepIA being the upper limit of the MSB-based DeepIA still holds. Previous work [28] which implemented sequential feature on various other data sets reported that SFS could potentially have higher prediction accuracies for a model trained with lower number of input features and attributed this effect to the forced retention of features in the SFS approach. In this paper, we did not observe that effect and conclude that this effect may be data dependent.

In certain time-constraint cases, it would be practical for LoS scenarios to use a feature selected DeepIA with 7 beams, since it gives close to 100% and performs better than an exhaustive search with 24 beams in CBS while...
utilizing approximately 29% of the total computation time. If the performance is critical and the channel is predominantly NLoS, it makes sense to use DeepIA with 24 beams. Thus, DeepIA ordinarily provides the network operator with a prediction accuracy vs. beam alignment time tradeoff to work with. However, in next subsection, we discuss an additional “averaging” enhancement which allows performing IA with just a subset of beams even in NLoS conditions; thus, realizing faster and reliable IA in both LoS and NLoS conditions.

D. Averaging of Input Beams and Its Effects on Accuracy

It is clear that the larger NLoS shadow fading factor has a significant impact on the performance regardless of the number of beams used. Motivated by the fact that the LoS performance far exceeds the NLoS performance, we consider alternative ways to boost the NLoS accuracy. It is expected that DeepIA in NLoS may benefit from input data fed with less variance, similar to what it got in LoS conditions. Since averaging a signal decreases the variance we build a new dataset that is of size $[R, M, s]$ where $s$ is the number of measurements/snapshots collected for averaging. Recall, $R$ and $M$ are the number of users and the number of beams in the subset, respectively. We select the values of $s$ as 10 and 25 as shown in Figs. 6 and 7. While Fig. 7 shows an increase in performance in the LoS case due to averaging, it is not as significant as the NLoS case. We observe that in NLoS conditions, DeepIA converges to a more than 95% prediction accuracy with just 7 SFS beams if 25 averages are used (and with 11 SFS beams if 10 averages are used). This leads to an interesting observation for our dataset - when fed with data corrupted by AWGN, even if the DeepIA network leverages the higher quality features it tends to perform at the level of the MSB. In other words, while noise brings down the performance of a network fed with MSBs, it tends to affect a DeepIA network fed with SFS features more severely. The lower limit seems to plateau towards the MSB performance.

The confusion matrices in Figs. 8, 9, 10, and 11 show how prediction accuracy is affected at the beam level. The previous prediction degradation due to fluctuations distorting adjacent and main beams can be seen in all the CBS confusion matrices. We can also observe that, in the NLoS CBS case, the back lobe begins to degrade prediction accuracy. In DeepIA, the deep learning network has the ability to recognize this back lobe related performance degradation and circumvents it by relying on the general trend of RSS values for each beam it observes. Nonetheless, it is still susceptible to the effects of fluctuations and sometimes incorrectly selects adjacent beams.
However, prediction accuracies for the users in individual beams are not uniform and do not seem to be related to the input beams fed to the network.

E. Interference Analysis

In this section, we analyze the impact of various levels of interference on the performance of DeepIA and CBS. In Section V.D, we note that the best performance is obtained for $s = 25$ and that DeepIA can obtain an accuracy of 95% with just 5 and 7 beams for LoS and NLoS channel conditions, respectively. Thus to draw a fair comparison on the effect of interference between DeepIA and CBS, we measure the impact of CBS with $s = 25$, while using 5 and 7 beams in LoS and NLoS cases. Now, when considering the DeepIA, for both the LoS and NLoS case, we consider two possibilities,

1) The DeepIA network is trained with no interference present in its training data but it is then exposed to interference during testing/validation stage DeepIA\textsubscript{WoI}.

2) The DeepIA network is trained with the expected interference and is then deployed in a network with the expected interference distribution DeepIA\textsubscript{WI}.

In this analysis, we consider 5 levels of interference. The interference ranges considered while generating the training data were $U[-140, -120)$ dBW, $U[-120, -100)$ dBW, $U[-100, -80)$ dBW, $U[-80, -60)$ dBW, and $U[-60, -40)$ dBW. The phase offset of the interference signal relative to the intended signal is sampled from $U[0, 2\pi]$. Since the receiver antenna is omnidirectional, for any particular user, for a given interference power range, the same realization of random interference is considered for all beams.

Our results show that interference does not significantly impact the performance of DeepIA until the interference levels above $-80$ dBW are encountered in the LoS case as shown in Fig. 12. We also note that in general, the (DeepIA\textsubscript{WI}) model outperforms DeepIA\textsubscript{WoI}, which in turn outperforms CBS. In the NLoS case, we see a steady decrease in accuracy as the amount of interference is increased as shown in Fig. 13. We see that even with interference training, the network shows appreciable deterioration with increasing level of interference. Nonetheless, interference-trained NLoS DeepIA network still outperforms CBS-based approach even at the highest levels of interference. We find that in general, in order to improve the accuracy, there is some benefit in training networks considering the expected interference.

F. Beam Prediction Time Analysis

For practical implementation aspects, 5G allows 5 ms periodicity by default to complete the beam sweeping procedure in the IA. However, coherence time over which the channel...
remains unchanged is much smaller for highly mobile users communicating at high frequencies. Coherence time $T_C$ is inversely proportional to the maximum Doppler spread $D_S = \frac{f_c}{v}$, where $f_c$ is the center frequency, $v$ is the speed of mobile user, and $c$ is the speed of light [29]. As an example, consider $f_c = 28$ GHz. If the user moves with walking speed (e.g., $v = 2$ m/s), then $T_C$ is about than 5 ms. On the other hand, if the user moves with a car speed of $v = 25$ m/s, then $T_C$ is less than 0.43 ms. Therefore, it is important to make the beam prediction decision within this time frame. User mobility controls the channel coherence time (time period over which the channel realization remains constant). In our test time, we assume that the RSS values are collected from each beam are well within the coherence time of for any given user. For a fast moving vehicle that strains these conditions, a worthwhile solution would be to increase number of samples that are collected (by using an ADC with a higher sampling rate) for the calculation of the RSS; by collecting the required number of samples for each RSS faster, we can bring down the time required to sweep the beam.

To better understand the computation time in practice, we consider running the DNN configuration of DeepIA on an embedded platform. For that purpose, the procedure of [30] to convert the trained software model for a deep neural network to the FPGA code is followed and computational aspects of running DeepIA’s DNN are evaluated for the FPGA implementation. Vivado Design Suite [31] is used to simulate and then synthesize the FPGA code with the 16-bit implementation for Xilinx UltraScale FPGA.

We evaluate the processing time associated with beam sweeping and beam prediction as follows. Beam sweeping involves a pilot tone transmitted from each beam at the transmitter one at a time to the receiver. The receiver measures the RSS value for each beam. This process is repeated $N$ times for CBS algorithm and $M$ times for DeepIA. Note $N$ is the total number of beams in consideration, and $M \leq N$ is the number of subset of beams using in DeepIA framework. The selection of the best beam from RSS values can either be performed at the transmitter or receiver. In the former case, the receiver transmits the measured RSS value back to the transmitter after every beam transmission as a feedback. In the latter case, the receiver predicts the best beam itself and transmits a single feedback. Assuming it takes $t_s$ to receive a tone signal at the receiver, the beam sweeping time is reduced from $N t_s$ to $M t_s$ by switching to DeepIA. This is because the duration of beam sweeping is linearly proportional to the number of beams swept, and we find that it is much larger than the beam prediction time that we discuss next.

The duration of beam prediction, however, depends on whether CBS or DeepIA is used. Suppose $M = 7$ beams are swept to fix the beam sweeping time for both CBS and DeepIA. The duration to take the maximum operation in CBS is $M T$ with one comparator, $(\frac{1}{2^2} + 1) T$ with two comparators, and so on, where $T$ is the time for one FPGA cycle at 100 MHz clock. Hence, CBS selects its beam in at most 0.06 $\mu$s. On the other hand, DeepIA runs the RSSs from 6 beams through its DNN. The latency to process one input sample (namely, 6 RSSs) through the DNN architecture given in Table I is 3.85 $\mu$s. Since the beam prediction time in both CBS and DeepIA is at most at microsecond level, the time for beam sweeping dominates the total time spent for IA. Therefore, DeepIA reduces the time for IA significantly and leaves more time for data communications to carry ever-growing traffic demands of 6G. Consequently, it is reasonable to anticipate that the demands for 6G networks would push the beam sweep time to sub-ms; however, this constraint does not impact our approach as we would still utilize only a subset of the total number of available beams that would otherwise have been swept.

When considering averaging the RSS data, it is reasonable to question if there are any additional time costs associated with that. The 5G NR standard specifies a subcarrier spacing of 240 MHz or a symbol duration of 4.16 $\mu$s for the mmWave frequency sync signals [32]. The bandwidth required for the synch transmission is roughly 60 MHz or a time period of 16 ns. Each synchronization signal block (SSB) is 4 symbols long and the length of 1 symbol is 4.16 $\mu$s. The SSB comprises of the Primary Broadcast Channel (PBCCH), Secondary Synchronization Signal (SSS) and Primary Synchronization Signal (PSS). The SSS is responsible for the RSS estimation and is one symbol long. Assuming a sampling speed of 5G samples per second (available from dual channel mmWave ADCs) [33], we note that we can obtain 21840 samples in 1 symbol/SSS duration (4.16 $\mu$s). If 25 averages are performed within the 1 symbol duration, and if RSS is estimated using the same number of samples, we would need about 870 samples for each RSS value. Note that the number of I/Q samples used for capturing the RSS data is specific to the chip/radio being implemented. We limit the scope of this work to investigating how the prediction accuracy scales with the number of samples used in averaging. There are other radio design constraints that may dictate the number of samples that are needed per average, so the basis of this work should not be used to determine that. If the number of averages we need is not served in 1 symbol duration, the averaging would be done over multiple SSS, however this impacts both CBS and DeepIA.

VI. CONCLUSION

In this paper, we developed the deep learning-based IA solution, DeepIA, for 5G and 6G mmWave networks. DeepIA collects measurements only from a small number of beams to predict the best beam for data communications. It is suggested as an alternative to CBS that iteratively searches for the best beam from among all possible beams. DeepIA can successfully capture the complex patterns of transmitter-receiver locations and beam patterns, and can predict the best beam with very high accuracy even when a small number of beams are swept in the IA. Compared to CBS, DeepIA significantly reduces the time for IA and leaves more time for communications. Our simulation results show that DeepIA is able to predict the optimal beam with nearly 100% accuracy even when
7 out of 24 beams are used in LoS conditions, while the beam prediction accuracy of CBS drops to around 24% with the same number of beams utilized. We identified that certain beams are more valuable in determining accuracy and leveraged this to increase prediction accuracy to close to 100% using just 5 beams in LoS. In NLoS we found that the significant amount of fluctuations present prevented significant features from contributing to further increase in accuracy. We also found that by using the averaged RSS we could further boost the performance of both LoS and NLoS conditions and that NLoS can exceed 95% prediction accuracy. Finally, we assessed the beam prediction time of DeepIA through embedded implementation on FPGA and showed that DeepIA significantly reduces the time needed for IA.

REFERENCES

[1] J. Reed and N. Tripathi, “5G cellular communications: Journey & destination,” 2019 [Online]. Available: https://thewirelesssuniversity.com

[2] S. Lien, S. Shieh, Y. Huang, B. Su, Y. Hsu, and H. Wei, “5G new radio: Waveform, frame structure, multiple access, and initial access,” IEEE Commun. Mag., vol. 55, no. 6, pp 64–71, Jun. 2017.

[3] K. Davasioliou and Y. E. Sagduyu, “Generative adversarial learning for spectrum sensing,” in Proc. IEEE Int. Conf. Commun., 2018, pp. 1–6.

[4] Y. Shi, K. Davasioliou, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, “Deep learning for RF signal classification in unknown and dynamic spectrum environments,” in Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw., vol. 5, no. 1, pp 2–14, Mar. 2019.

[5] T. Erpek, Y. E. Sagduyu, and Y. Shi, “Deep learning for launching and mitigating wireless jamming attacks,” IEEE Trans. Cogn. Commun. Netw., vol. 5, no. 1, pp 1–6, Jul. 2019.

[6] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Physical layer deep learning of encodings for the MIMO fading channel,” in Proc. IEEE 55th Annu. Allerton Conf. Commun., Control, Comput., 2017, pp. 76–80.

[7] T. Erpek, T. J. O’Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, “Deep learning for wireless communications,” in Development and Analysis of Deep Learning Architectures, Berlin, Germany: Springer, 2020, pp. 223–266.

[8] S. Niknam et al., “Intelligent O-RAN for beyond 5G and 6G wireless networks,” 2020, arXiv:2005.08374.

[9] T. S. Cousik, V. K. Shah, J. H. Reed, T. Erpek, and Y. E. Sagduyu, “Fast initial access with deep learning for beam prediction in 5G mmWave networks,” in Proc. IEEE Mil. Commun. Conf., 2021, pp. 664–669.

[10] L. Wei, Q. C. Li, and G. Wu, “Initial access techniques for 5G NR: Omni/beam SYNC and RACH designs,” in Proc. IEEE Int. Conf. Comput., Netw. Commun., 2018, pp. 249–253.

[11] L. Wei, Q. Li, and G. Wu, “Exhaustive, iterative and hybrid initial access techniques in mmWave communications,” in Proc. IEEE Wireless Commun. Netw. Conf., 2017, pp. 1–6.

[12] M. Jasim, J. E. Pezoa, and N. Ghani, “Simultaneous multi-beam analog beamforming and coded grating lobes for initial access in mmWave systems,” in Proc. IEEE CHILEAN Conf. Electrical, Electron. Electron. Inf. Technol., 2017, pp. 1–6.

[13] I. Akyin and M. Krunz, “Efficient beam sweeping algorithms and initial access protocols for millimeter-wave networks,” IEEE Trans. Wireless Commun., vol. 19, no. 4, pp. 2504–2514, Apr. 2020.

[14] I. Ismath, K. B. Shashika Manosha, S. Ali, N. Rajatheva, and M. Latwa-aho, “Deep contextual bandits for fast initial access in mmWave based user-centric ultra-dense networks,” IEEE 93rd Vehicular Technol. Conf., 2021, pp. 1–7.

[15] H. Soleimani, R. Parada, S. Tomasin, and M. Zorzi, “Fast initial access for mmWave 5G systems with hybrid beamforming using online statistics learning,” IEEE Commun. Mag., vol. 57, no. 9, pp. 132–137, Sep. 2019.

[16] A. Mazzin, M. Elkouri, and R. D. Gillin, “Accelerating beam sweeping in mmWave standalone 5G new radios using recurrent neural networks,” in Proc. IEEE 88th Veh. Technol. Conf., 2018, pp. 1–4.

[17] 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.

[18] R. Zia-ul Mustafa and S. A. Hassan, “Machine learning-based context aware sequential initial access in 5G mmWave systems,” in Proc. Globecom Workshops, 2019, pp. 1–6.

[19] A. Asadi, S. Müller, Gek Hong Sim, A. Klein, and M. Hollick, “FML: Fast machine learning for 5G mmWave vehicular communications,” in Proc. IEEE Conf. Comput. Commun., 2018, pp. 1961–1969.

[20] M. K. Haider, Y. D. G. Koutsonikolas, and E. W. Knightly, “LiSteer: MmWave beam acquisition and steering by tracking indicator leads on wireless ads,” in Proc. 24th Annu. Int. Conf. Mobile Comput. Netw., 2018, pp. 273–288.

[21] C. A. Balanis, Antenna Theory: Analysis and Design. Hoboken, NJ, USA: Wiley, 2016.

[22] H. Alias, M. T. Ali, S. SubahirN. Ramli, M. A. Sulaiman, and S. Kayat, “A back lobe reduction of aperture coupled microstrip antenna using DGS,” in Proc. IEEE 10th Int. Conf. Elect. Eng/Electron., Comput., Telecommun. Inf. Technol., 2013, pp. 1–5.

[23] T. S. Rappaport, G. R. MacCartney, M. K. Samimi, and S. Sun, “Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design,” IEEE Trans. Commun., vol. 65, no. 9, pp. 3029–3056, Sep. 2015.

[24] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” 2019, arXiv:1902.09843.

[25] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” Comput. Elect. Eng., vol. 40, no. 1, pp. 16–28, 2014.

[26] R. Grigonv, V. Cevher, and M. E. Davies, “Compressible distributions for high-dimensional statistics,” IEEE Trans. Inf. Theory, vol. 58, no. 8, pp. 5016–5034, Aug. 2012.

[27] I. Lembhardi, F. Ruan, L. Abraham, and R. Tibshirani, “LassoNet: A neural network with feature sparsity,” J. Mach. Learn. Res., vol. 22, no. 127, pp. 1–29, 2021.

[28] J. Reunane, “Overfitting in making comparisons between variable selection methods,” J. Mach. Learn. Res., vol. 3, pp. 1371–1382, 2003.

[29] D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[30] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davasioliou, H. Deng, and T. Erpek, “Real-time and embedded deep learning on FPGA for RF signal classification,” in Proc. IEEE Mil. Commun. Conf., 2019, pp. 1–6.

[31] “Vivado ML Overview,” Xilinx [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html.

[32] J. Liu et al., “Initial access, mobility, and user-centric multi-beam operation in 5G new radio,” IEEE Commun. Mag., vol. 56, no. 3, pp. 35–41, Mar. 2018.

[33] “ADC12JD5200RF data sheet,” Texas Instrument Incorporated. [Online]. Available: https://www.ti.com/product/ADC12JD5200RF

Tarun S. Cousik: Graduated student member, IEEE, is currently working toward the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA. He was previously at the antenna design, wireless testing, and systems engineering for jump trading, bose corporation, analog devices, and nokia siemens networks. His research interests include wireless positioning, RF array systems design, and machine learning for wireless communication. He was the Co-recipient of the 2020 AT & T VURI Award.

Vijay K. Shah (Member, IEEE) is currently an Assistant Professor with Cybersecurity Engineering Department, George Mason University, Fairfax, VA, USA. Before joining Mason, he was a Research Assistant Professor with the Bradley Department of Electrical and Computer Engineering (ECE), Virginia Tech, Blacksburg, VA, USA, and affiliated with Wireless@Virginia Tech. His research interests include next-G wireless, O-RAN architecture, robust AI/ML for communications, and wireless tested development and prototyping – with applications to Internet of Things (IoT) systems, autonomous systems (e.g., V2X) and cyber-physical systems (e.g., power grids).
Tugba Erpek (Member, IEEE) received the B.S. degree in electrical and electronics engineering from Osmangazi University, Eskisehir, Turkey, the M.S. degree in electrical and computer engineering from George Mason University, Fairfax, VA, USA, and the Ph.D. degree from Virginia Tech, Arlington, VA, USA. She is currently a Research Associate Professor with National Security Institute, Virginia Tech. Before joining Virginia Tech, she was a Lead Scientist and Network Communications Technical Area Lead with Intelligent Automation Inc., Rockville, MD, USA. Her R&D work covers wireless communications and networks, resource allocation, nextG, physical-layer security, cognitive radio, machine learning, and adversarial machine learning. She has authored or coauthored extensively in these areas. She is also a TPC Member and Reviewer for major IEEE conferences and journals.

Yalin E. Sagduyu (Senior Member, IEEE) received the B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA. He is currently a Research Professor with the National Security Institute of Virginia Tech, Arlington, VA, USA. He is also a Visiting Research Professor with the Department of Electrical and Computer Engineering, University of Maryland. Previously, he was the Director of Networks and Security Division with Intelligent Automation Inc. and BlueHalo Company, Rockville, MD, USA. His research interests include wireless communications, networks, security, and machine learning. He was the Track Chair of IEEE PIMRC, IEEE GlobisIP, and IEEE MILCOM, and was with the organizing committee of IEEE GLOBECOM and IEEE MILCOM. He chaired workshops with ACM MobiCom, ACM WiSec, IEEE CNS, and IEEE ICNP. He is the Editor of IEEE Transactions on Communications. He was the recipient of the IEEE HST 2018 Best Paper Award.

Jeffrey H. Reed (Life Fellow, IEEE) is currently the Founder of Wireless@Virginia Tech, and was its Director until 2014. He is also the Founding Faculty Member of the Ted and Karyn Hume Center for National Security and Technology and was its Interim Director when founded in 2010. His book, Software Radio: A Modern Approach to Radio Design was published by Prentice Hall and his latest textbook Cellular Communications: A Comprehensive and Practical Guide was published by Wiley-IEEE in 2014. He is the Co-Founder of Cognitive Radio Technologies, a company commercializing the cognitive radio technologies, Federated Wireless, a company developing technologies for 5G systems, and for PFP Cybersecurity, a company specializing in security for embedded systems. In 2005, he became Fellow to the IEEE for contributions to software radio and communications signal processing and for leadership in engineering education. In 2013, he was the recipient of the International Achievement Award by the Wireless Innovations Forum. In 2012, he was on the President’s Council of Advisors of Science and Technology Working Group that examine ways to transition federal spectrum to allow commercial use and improve economic activity and he is also a former Member of CSMAC a group that provides advice to the NTIA on spectrum issues.