

UCINet0: A Machine Learning based Receiver for 5G NR PUCCH Format 0

Anil Kumar Yerrapragada*, Jeeva Keshav Sattianarayanan †, Radha Krishna Ganti‡

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India, 600036

Email: *anilkumar@5gtiitm.in, †jeevakeshavs@smail.iitm.ac.in, ‡rganti@ee.iitm.ac.in

Abstract—Accurate decoding of Uplink Control Information (UCI) on the Physical Uplink Control Channel (PUCCH) is essential for enabling 5G wireless links. This paper explores an AI/ML-based receiver design for PUCCH Format 0. Format 0 signaling encodes the UCI content within the phase of a known base waveform and even supports multiplexing of up to 12 users within the same time-frequency resources. Our first-of-a-kind neural network classifier, which we term UCINet0, is capable of predicting when no user is transmitting on the PUCCH, as well as decoding the UCI content of any number of multiplexed users, up to 12. Inference results with both simulated and hardware-captured field datasets show that the UCINet0 model outperforms conventional DFT-based decoders across all SNR ranges.

Index Terms—5G, AI/ML, Multi-label Classification, Neural Network, Physical Uplink Control Channel, Format 0

I. INTRODUCTION

THE successful establishment of any wireless communication link between two entities relies on feedback signalling from both ends to indicate the channel quality as well as the status of previous transmissions. The Third Generation Partnership Project (3GPP) standards define the 5G NR (New Radio) Physical Uplink Control Channel (PUCCH) [1], which is the key enabler of such feedback in the uplink direction. It is a dedicated channel on which a User Equipment (UE) can send control information to a Base station (gNB). Uplink Control Information (UCI) carried by the PUCCH includes (1) Hybrid Automatic Repeat Request (HARQ) acknowledgements for prior downlink transmissions (Base station to UE), (2) Scheduling Request (SR) for the subsequent allocation of uplink transmission resources, and (3) Channel State Information (CSI) reports containing channel quality metrics that facilitate link adaptation, precoding, and downlink resource allocation.

In order to encompass the wide range of capacity and latency requirements of various 5G applications, the 5G standards [2]–[4] have provisioned five different formats of the PUCCH transmission (See Table I for a summary of the five formats). Smaller UCI payloads of 1 or 2 HARQ bits and/or an SR are transferred using Formats 0 and 1. Larger payloads, including several SR and HARQ bits, along with CSI reports, are transmitted using Formats 2, 3, and 4. Furthermore,

The first two authors contributed equally to this work. This work was supported by the Indian Ministry of Electronics and Information Technology (MeitY) and the Department of Telecommunications (DoT) under the project “Next Generation Wireless Research and Standardization on 5G and Beyond” with grant number SP21221155EEMEIT008073 through the 5G Testbed Project.

Formats 0 and 2 are suitable for low latency applications since they occupy only 1 or 2 OFDM symbols in the time domain. Formats 1, 3, and 4 are suitable for applications requiring improved coverage and capacity since they extend between 4 and 14 OFDM symbols in the time domain and provide SNR gain.

A. Our Contributions

In this paper, we focus on PUCCH Format 0, a particularly important format used during several steps of the 5G initial attach process [5]. Specifically, we look at the design and performance analysis of a Format 0 receiver using AI/ML techniques such as Neural Network classifiers. We make the following contributions.

- PUCCH Format 0 uses a waveform-based technique for information transfer. It encodes the UCI within phase rotations applied to a pre-defined base sequence. This paper presents a Machine Learning based decoder for Format 0 that uses a Neural Network (NN) to predict the phase rotation. This paper, an extension of our previous work [6], is one of the first attempts at designing such a receiver for 5G PUCCH Format 0.
- PUCCH Format 0 signals can be transmitted either by individual UEs or by multiple UEs (a maximum of 12 UEs [2]) multiplexed in the same time-frequency resources. Conventional decoding methods employ iterative correlation of the coherently combined signals with the known base sequences. This paper shows that a single NN model can replace the correlation operations, thus decoding the UCI from a single UE up to a maximum of 12 multiplexed UEs, with each UE transmitting any combination of HARQ and SR payloads.
- One of the challenges of correlation-based methods is their susceptibility to false detections. Conventional approaches to determine whether a UE has actually transmitted on the PUCCH involve signal power-based thresholds, which are often determined either based on observation or tedious simulations. We show that the NN model described in this paper is robust enough to identify false transmissions, thus eliminating the need for thresholds.
- This paper provides insights into the generation of training and testing wireless signal datasets derived from both MATLAB simulations and real-time over-the-air field captures taken at the indigenous 5G Testbed at IIT Madras [7].

TABLE I: Summary of 5G NR PUCCH Formats

Format	Payload	Duration	Duration in symbols	Resource Blocks	Waveform	Modulation	DMRS	Multiplexing
0	HARQ, SR	Short	1-2	1	CP-OFDM	None (Sequence-based)	No	Yes
1	HARQ, SR	Long	4-14	1	CP-OFDM	BPSK or QPSK	Yes	Yes
2	HARQ, SR, CSI	Short	1-2	1-16	CP-OFDM	QPSK	Yes	No
3	HARQ, SR, CSI	Long	4-14	1-16 ¹	DFT-S-OFDM	±BPSK or QPSK	Yes	No
4	HARQ, SR, CSI	Long	4-14	1	DFT-S-OFDM	±BPSK or QPSK	Yes	Yes

¹Limited to 1-6, 8-10, 12, 15, 16 for ease of DFT-S-OFDM.

- The proposed AI/ML model relies on extracting a map between the received PUCCH samples (input) and the applied phase rotation to the base sequence (output). At least two degrees of randomness are embedded in the input data. The first comes from the phase rotation, which is a function of the random UCI bits. The second comes from the fading channel and thermal noise, which we assume to be Additive White Gaussian Noise (AWGN). In the presence of such randomness, neural networks often converge to the correct input-output relationship while obscuring the path they take to achieve convergence. This paper attempts to offer some insights into the interpretability of the model's behavior.
- AI/ML models must adhere to certain complexity, memory, and latency constraints to become deployable on hardware devices such as Field Programmable Gate Arrays (FPGA). To this end, this paper provides a complexity analysis of the NN models presented.

II. BACKGROUND ON 5G NR PUCCH FORMAT 0

PUCCH Format 0 is used to transfer one or two HARQ acknowledgments and/or an SR [4]. In 5G, Format 0 signalling is used as early as the initial attach procedure in which a UE is trying to latch onto a gNB. During the initial attach, there are several critical downlink transmissions, such as the Radio Resource Control (RRC) Setup, that need to be acknowledged, which the UE signals on PUCCH Format 0. It is to be noted that at this early stage of the connection process, neither the UE nor the gNB are expected to have obtained detailed information about the wireless surroundings. Consequently, Format 0 employs a sequence-based transmission in which phase-rotated versions of a pre-defined base sequence are transmitted by the UE. The UCI is encoded in the phase of the base sequence. Since the base sequence is assumed to be known at both ends, Format 0 signalling does not contain any pilot reference signals such as the Demodulation Reference Signals (DMRS), nor does it use Quadrature Amplitude Modulation (QAM) to modulate the UCI bits [8].

A. Time-Frequency Allocation, Sequence Generation and UCI Encoding

The cyclically shifted base sequence is mapped to the Resource Grid, where it occupies one Resource Block (RB) in the frequency domain and either one or two symbols in the time domain. One Resource Block in the frequency domain contains 12 Resource Elements (or 12 sub-carriers of the

OFDM grid i.e., $N_{sc}^{RB} = 12$). Since PUCCH Format 0 can span two symbols, either 12 or 24 Resource Elements can be occupied. The second symbol contains a sequence similar to that of the first symbol and can be used for SNR enhancement at the receiver. In more concrete terms, the PUCCH Format 0 sequence in the frequency domain includes a phase rotation α applied to a low Peak-to-Average Power Ratio (PAPR) base-sequence [2] $r_{u,v}^\alpha(k)$, and is given by,

$$\begin{aligned} r_{u,v}^\alpha(k) &= e^{j\alpha k} \cdot \bar{r}_{u,v}(k) \\ &= e^{j\alpha k} \cdot e^{j\phi(k)\pi/4}, \end{aligned} \quad (1)$$

where $k = 0, 1, 2, \dots, N_{sc}^{RB} - 1$ and $\phi(k)$ is given by Table 5.2.2.2-2 in [2] which is a set of phase factors that lead to the generation of low PAPR sequences. The subscripts u and v represent the group number and the sequence number within the group, respectively. The values of u and v are configured by higher layer group hopping parameters. We note that when PUCCH Format 0 is two OFDM symbols long, intra-slot frequency hopping can be enabled. In this paper, for ease of exposition, we assume that intra-slot frequency hopping is disabled and utilize only one symbol for decoding the UCI bits.

The cyclic shift α applied to the base sequence is given by,

$$\alpha = \frac{2\pi}{N_{sc}^{RB}} \left((m_0 + m_{cs} + n_{cs}(n_{sf}^l, l + l')) \bmod N_{sc}^{RB} \right), \quad (2)$$

where

- $m_0 \in [0, 11]$ is the initial cyclic shift that is used to differentiate various multiplexed users as defined in [2].
- m_{cs} is the UCI-specific cyclic shift [2] shown in Table II.
- $n_{cs}(n_{sf}^l, l + l')$ is a function based on the pseudo-random binary sequence defined in [2].
- n_{sf}^l is the slot number in the radio frame [2].
- l is the OFDM symbol number in the PUCCH transmission where $l = 0$ corresponds to the first OFDM symbol of the PUCCH transmission [2], [4].
- l' is the index of the OFDM symbol in the slot that corresponds to the first OFDM symbol of the PUCCH transmission in the slot [2], [4].

B. PUCCH Format 0 Transmission Scenarios

While m_0 and n_{cs} are drawn from higher layer (L2) configurations, the UE chooses the m_{cs} depending on the UCI content it transfers. Consider the following two examples of possible PUCCH Format 0 transmission scenarios.

I) Single UE Transmission: A single UE scheduled to transmit on a particular time-frequency allocation chooses an m_{cs} value based on its UCI content as defined in Table. II. In the table, NACK and ACK refer to Negative and Positive Acknowledgments, respectively. +ve SR indicates an SR transmission and a -ve SR denotes that an SR is not transmitted despite the PUCCH resource allocation (the UE does not require uplink PUSCH allocations).

TABLE II: Possible m_{cs} values in PUCCH Format 0

UCI Content	Possible m_{cs} values
1 HARQ	0 (ACK), 6 (NACK)
2 HARQ	0 (NACK, NACK) 3 (NACK, ACK) 6 (ACK, ACK) 9 (ACK, NACK)
1 SR	0 (+ve SR)
1 HARQ + 1 SR	0 (NACK, -ve SR) 3 (NACK, +ve SR) 6 (ACK, -ve SR) 9 (ACK, +ve SR)
2 HARQ + 1 SR	0 (NACK, NACK, -ve SR) 1 (NACK, NACK, +ve SR) 3 (NACK, ACK, -ve SR) 4 (NACK, ACK, +ve SR) 6 (ACK, ACK, -ve SR) 7 (ACK, ACK, +ve SR) 9 (ACK, NACK, -ve SR) 10 (ACK, NACK, +ve SR)

2) Multiplexed UE Transmission: In the interest of optimizing radio resources, PUCCH Format 0 has a unique potential to multiplex more than one UE on the same time-frequency locations. This is achieved by allocating different initial cyclic shifts (m_0) to each UE. Since the number of possible values of m_0 is limited to 12, the number of UEs that can be multiplexed is upper-bounded by 12. The maximum number of UEs that can be multiplexed is also driven by the UCI content. A few examples of this are shown in Figure 1. Figure 1a shows how assigning m_0 values of 0 through 5 to a maximum of 6 different UEs allows each of them to pick one of the two possible cyclic shifts and transmit 1 HARQ bit on the same resources. Similarly, Figure 1b shows that m_0 values of 0 through 2 allow for 3 UEs to be multiplexed when each of them transmits 2 HARQ bits (or 1 HARQ and 1 SR bit) with one of the four possible cyclic shifts. Additionally, multiplexing across different types of UCI content is possible, wherever applicable. For example, a UE transmitting 2 HARQ bits and an SR bit cannot be multiplexed with another transmitting the same combination but can be multiplexed with up to 3 more UEs transmitting SR bits or 1 more UE transmitting 2 HARQ bits (Figure 1c). Multiplexing of multiple UEs is possible as long as there are no overlaps in the assigned possible cyclic shifts for each of the UEs.

Accounting for UE multiplexing, the generalized multi-user Format 0 received signal in the frequency domain is given by,

$$y(k) = \sum_{m=0}^{N_{UE}-1} h_m(k) e^{j\alpha_m k} \cdot \bar{r}_{u,v}(k) + w(k), \quad (3)$$

where $k = 0, 1, 2, \dots, N_{sc}^{RB} - 1$, N_{UE} is the number of multiplexed UEs in a given resource block, $h_m(k)$ is the

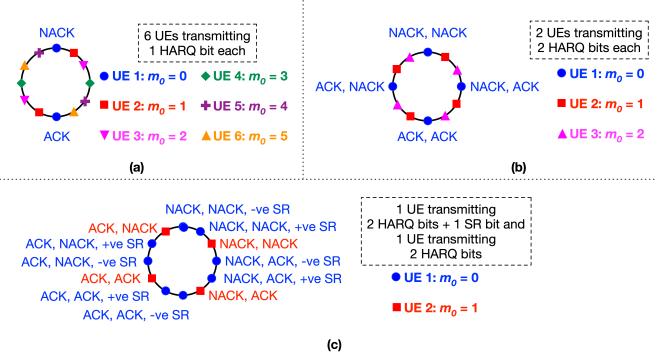


Fig. 1: Example scenarios showing how assigning different initial cyclic shifts to different UEs allows them to be multiplexed on the same time-frequency resources.

channel between the gNB and the UE transmitting the base sequence with cyclic shift α_m in the k^{th} Resource Element (observe that all multiplexed UEs use the same base sequence). The noise is denoted by $w(k)$.

C. Existing Methods For PUCCH Format 0

I) Conventional Approaches: We reiterate that Format 0 has no provision for DMRS (pilots), and hence, there is no channel estimation or equalization possible. Blind correlation methods [9] [10] exist for such scenarios in which channel information is unknown, but these methods are often reserved for scenarios in which the resource allocation information is also not known. One example of such a scenario is the initial time synchronization between the gNB and the UE. Since the allocation of the synchronization sequences is unknown, longer correlations need to be performed. But, by the time the first Format 0 signal is received at the gNB, time and frequency synchronization are already achieved, and the gNB is aware of the UE's transmission [11]. Since the allocation is known and is always 1 RB long, a simpler length 12 correlation is sufficient.

In [12]–[14], various receiver techniques based on correlation are considered. Correlation relies on the fact that the low PAPR base sequence used for encoding the UCI content is known at the receiver. The decoding method involves correlating the received samples with various cyclically shifted versions of the base sequence. The predicted cyclic shift is the cyclic shift that gives the highest correlation magnitude. It should be noted that peak selection, by definition, requires an optimal threshold for determining whether a peak is due to noise or a true transmission.

The work in [12] shows that correlation-based signal detection outperforms methods based on raw signal power measurements even under fading channel scenarios. A normalized correlation-based solution for identifying false detections of HARQ bits is provided in [13]. The solution uses the probability distributions of a normalized correlation peak to determine the optimum threshold for classifying a reception as false. The PUCCH format 0 receiver algorithm proposed in [14] eliminates phase opposition across multiple hops by finding

the sum of correlation magnitudes across symbols instead of correlation values.

In the 5G testbed deployment at IIT Madras [7], the correlation is implemented by an equivalent DFT-based algorithm. Later in the paper, we use this as a baseline for comparison. Such an algorithm proves to be better from a hardware perspective due to its optimized use of resources (avoiding the need to correlate with all the shifted base sequences), reduced latency, and higher throughput. In the IIT Madras 5G testbed, a DFT-based receiver for PUCCH Format 0 has been implemented on custom Field Programmable Gate Array (FPGA) boards for real-time operations. This method

an example in which 3 UEs are multiplexed. In this case, we select the top 3 peaks (at $index = 4, 6$, and 8) as α values.

We observe that while the above works in literature do provide false and missed detection performance analysis, they are based on fixed thresholds that are specific to a certain scenario and derived from lengthy Monte Carlo simulations. Another significant element that is not considered in [12]–[14] and considered only to some extent in [7] is the multiplexing of multiple UEs in the same resources.

It is not clear how the thresholds in the above work and implementations extend to multiplexed UEs. Furthermore, fixed thresholds do not account for the received signal power variations that are caused due to variations in gains of hardware elements such as the LNA and due to varying locations of the multiplexed UEs. For example, a true transmission with large distance-based path loss could be misclassified as false if a fixed threshold is used.

Finding a one-size-fits-all threshold for various channel environments and multiplexed UEs is non-trivial. This has led us to explore an AI/ML-based data-driven approach in which a receiver could be designed directly from the raw frequency domain signals rather than correlation values. We apply similar techniques here. The IQ samples in the received resource block are analogous to pixel values in an image. Hence, models that work for image classification could potentially work here too.

[2) AI/ML Approaches: To our knowledge, there is limited work on applying AI/ML toward the decoding of Format 0 signals. One of the first models for Format 0 decoding was proposed in our previous work [6]. However, this was an initial effort that focused on training the model to predict 1 out of 4 possible m_{cs} values in the specific case of 1 HARQ + 1 SR transmission. The architecture described in this paper extends our previous work to support all the possible UCI payloads shown in Table I and all the possible multiplexing combinations of UEs ($N_{UE} = 0, 1, \dots, 12$). This generalized architecture is designed for potential field deployment in any 5G gNB.

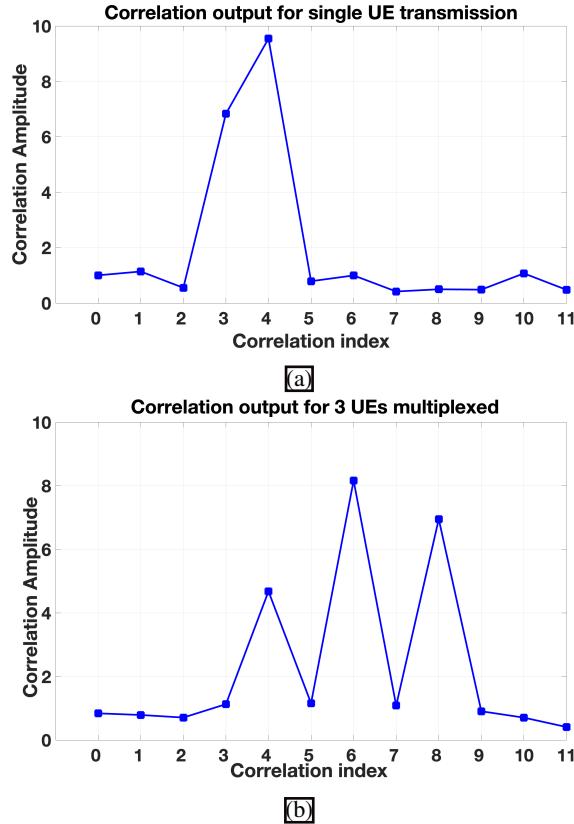


Fig. 2: Illustration of the output of Correlation of received Format 0 signal with the known base sequence for (a) 1 UE transmission and (b) 3 UEs multiplexed

recovers the phase rotation α by taking the 12-point DFT of $y(n) \cdot \bar{r}_{u,v}(n)$. Since $y(n) = e^{j\alpha n} \cdot \bar{r}_{u,v}(n)$, the multiplication of the base sequence $\bar{r}_{u,v}$ with its complex conjugate forces it to unity. The 12-point DFT of the resultant exponential term $e^{j\alpha n}$ results in a peak at α . This is shown in Figure 2a with a correlation peak at $index = 4$. The correlation index corresponding to the maximum correlation amplitude is taken as the estimated α value. Recall from (2) that α is a combination of m_0 , m_{cs} and n_{cs} . Therefore, upon the subtraction of $m_0 + n_{cs}$ from α (subtraction is modulo 12), the UCI-specific cyclic shift m_{cs} remains. In scenarios where multiple users (N_{UE} of them) transmit PUCCH Format 0 signals in the same time-frequency location, the DFT algorithm selects the top N_{UE} peaks from the DFT output and uses them to determine the m_{cs} for each of the N_{UE} multiplexed UEs. Figure 2b shows

III. MACHINE LEARNING RECEIVER FOR PUCCH FORMAT 0 DETECTION

This section lays down the framework for posing the PUCCH Format 0 detection as an AI/ML classification problem. Specifically, we show how a single multi-label neural network classifier can serve as a generalized PUCCH Format 0 decoder.

A. PUCCH Format 0 as an AI/ML Classification Problem

Classification, a widespread use case of machine learning algorithms, is a supervised learning task that involves identifying to which class (label) a given data instance belongs. A predictive model, typically a Neural Network, is trained with a dataset that contains several input instances and the corresponding ground truth class labels. Labeled data helps us "supervise" the NN in learning the correct input-output mapping.

Learning happens by iteratively minimizing a loss function, which is a distance metric between the predicted label and

the ground truth label. In many classification tasks, the input-output mappings are intractable. ML techniques, such as Neural Networks, have proven to be more adept at extracting these mappings.

There are typically three types of classification tasks. The first is Binary Classification, in which the input data falls into one of two classes. An example of this is email classification as spam or not spam. The second is multi-class classification, in which input data could belong to one of more than two classes. A routinely cited example is classifying handwritten digit images into one of ten numbers. The third is multi-label classification, in which input data can independently belong to more than one class. An example of this is the identification of multiple genres a movie belongs to, given a summary of its plot.

For PUCCH Format 0, the classification task is as follows: Given a sequence of received frequency domain Format 0 samples, predict the N_{UE} phase rotation values $\alpha_0, \alpha_1 \dots \alpha_{N_{UE}-1}$ (see Eq. (3)) applied to the base sequence. The obtained α values can then be mapped back to the UCI-specific cyclic shift m_{cs} for each UE by modulo 12 subtraction of the corresponding m_0 and n_{cs} which are provided by the higher layers (L2). In the case where no user is transmitting on the PUCCH ($N_{UE} = 0$), the NN prediction has to reflect this. In summary, the NN classifier has to predict either a single α value, multiple α values, or zero α values. Such a pattern of prediction is a typical use case of multi-label classification.

B. Data Generation And Representation

Dataset generation in AI/ML for wireless communication has a unique challenge not present in other domains - the non-availability of off-the-shelf benchmark datasets. One way of developing such datasets is through state-of-the-art simulation tools, such as the MATLAB 5G Toolbox, which can generate near-accurate datasets under various channel impairments. Simulated data is a good starting point for training neural networks in communication problems. However, we note that including field data, if available, gives us an insight into the generalization performance of AI/ML models across different distributions of data. The inclusion of field data also aligns with 3GPP requirements for 5G Release 18 [15]. For this paper, we use a combination of the above two approaches for data generation. The datasets used in this paper are publicly available at [16].

(1) Simulated Datasets: Using the MATLAB 5G Toolbox, we generate the received waveforms containing the PUCCH Format 0 signals. These generated waveforms include fading channel impairments and Additive White Gaussian Noise (AWGN). For various SNR values (0, 5, 10, 15, and 20 dB), we generate PUCCH signals transmitted over a TDL-C Channel and store the noisy received samples. These samples are the input to the NN. For each input, the corresponding output label is the applied phase rotation α for each of the N_{UE} UEs in multi-hot encoded format.

The pseudocode for generating PUCCH Format 0 datasets in MATLAB is shown in Algorithm 1. We first place PUCCH Format 0 signals on all 14 symbols in a slot and 12 resource

blocks in each symbol spaced 20 resource blocks apart. This results in 168 allocations per resource grid. We generate 1000 iterations of such resource grids. These 1000 grids are generated for each value of N_{UE} in the range 0 to 12. The process is further repeated for the five SNR values.

Algorithm 1: Training dataset generation

```

NUE = {0, 1, 2, ..., 12}
SNR = {0, 5, 10, 15, 20} dB
Iter = {1, 2, ..., 1000}
foreach NUE, SNR, Iter do
    nUE = 0, 1, 2, ..., NUE
    foreach u ∈ nUE do
        foreach c ∈ {1, 2, ..., 168} do
            bit_len_harq ∈ {0, 1, 2}
            bit_len_sr ∈ {0, 1}
            Generate UCI bits and label (α)
            Generate Format 0 signal and place it in
            the resource grid
            OFDM Modulation of the TX OFDM grid
            Transmit over channel
        Receive combined signal from NUE UEs (Eq. 3)
        Perform OFDM Demodulation to obtain the RX
        grid
    foreach c ∈ {1, 2, ..., 168} do
        Extract and save Format 0 signal
    
```

(2) Real-Time Over-The-Air Datasets: We use hardware captures derived from the state-of-the-art 5G testbed at IIT Madras [7] for more realistic testing. The hardware captures help represent channel states and hardware impairments not included in the training dataset. The setup (see Figure 3) consists of an N5182B Vector Signal Generator (VSG) for transmitting the 5G signal at a center frequency of 3.49986 GHz (one of the sub-6 GHz channel raster in the n78 band). In our setup, the VSG acts like a UE, and it uses a commercial omnidirectional wideband monopole antenna to transmit the signals. The VSG is connected to the antenna through 2 SMA cables with 1.9 dB wire loss each. Here, commercial UEs could not be used because it is not easy to extract the actual transmitted information, i.e., ground truth labels for the UCI transmissions made by the UE. Also, the dynamic nature of the allocations made by the L2 scheduler makes it difficult to capture the data at the gNB for that particular time duration.

A multi-channel 5G Remote Radio Head (RRH) with a dual-polarized antenna at the gNB receives the signals from over the air (for the purpose of the paper, we utilize only one antenna and one transceiver chain). The RRH operates in the n78 band with 100 MHz bandwidth. It is ORAN-compliant and follows the 7.2b split as defined in [17]. Other receiver components of the RRH include an in-house Low Noise Amplifier (LNA) with 60dB gain at the receiver front end and an ADRV 9009 RF transceiver. We place the transmitter antenna one meter away from the receiver. The PUCCH Format 0 signal is transmitted from the VSG through an antenna, over the air, and then received at the RRH antenna, followed by the LNA and the

transceiver. The signal out of the 16-bit ADC is then collected and used for testing.



Fig. 3: IIT-Madras 5G testbed setup with the Remote Radio Head used as a receiver (gNB) and the VSG used as a transmitter (Emulated UE).

C. Neural Network Architecture

Given a problem statement, there is no formula for knowing apriori, the best neural network architecture. For most problems, initial architectures are determined based on prior experience and domain knowledge. These architectures are then tuned by experiment. Figure 4 shows the UCINet0 architecture that showed the best performance in our experiments. In subsequent sections, we also comment on the trade-off between high complexity and performance.

1) Neural Network Structure: The input to the NN has two components. The first is the received PUCCH Format 0 signal, and the second is metadata indicating the number of multiplexed UEs that the receiver expects. Since the PUCCH Format 0 signals occupy a single Resource Block, there are 12 received complex samples. We separate the real and imaginary parts of each complex number and concatenate them, resulting in a length 24 real sequence. As shown in Eq. (3), a given PUCCH resource block at the receiver could contain the sum of signals from multiple UEs. To aid the neural network, we also feed the number of multiplexed UEs as a 25th metadata input. The number of multiplexed UEs that a receiver should expect to see is often provided by an L2 scheduler as an upper bound, meaning that the received signal could contain less than or equal to the number of UEs indicated by L2. We denote the difference between the actual number of UEs multiplexed and the number of UEs indicated by L2 as Δ . One example of a scenario where this is observed is when the L2 scheduler allocates multiple UEs to transmit SR, but not all of the UEs

have any data to transmit. In such cases, UEs that do not require PUSCH resources do not transmit the SR.

Owing to the relatively small dimension of the input, our proposed UCINet0 is a Fully Connected Neural Network (FCN). As shown in Figure 4, there are 3 dense layers, each containing 512 neurons. The output layer contains twelve neurons. Each output neuron acts as an indicator for the transmission of the base sequence which is cyclically shifted by α_m . We arrived at the final UCINet0 architecture of 3 layers and 512 neurons after having evaluated the performance of other architectures with a varying number of layers and neurons. Figure 5 shows the UCI decoding accuracies(defined in IV IV-B) obtained with 20 variations of FCN architectures. We considered combinations of 1, 2, 3 and 4 layer models with 64, 128, 256, 512 and 1024 neurons.

From Figure 5, we observe the following:

- Model performance generally improves as we increase the number of neurons from 64 to 1024 for any number of layers and for values all of Δ .
- For $\Delta = 0$, both 1024 and 512 neurons result in almost the same accuracy values for a given number of layers. For example, 2 layers \times 1024 and 2 layers \times 512 result in similar accuracy values.
- For higher values of Δ , the accuracy drops as the number of layers increases beyond a certain value as the model begins to overfit. For $\Delta = 4$ and 1024 neurons, the accuracy drops beyond 2 layers, whereas for 512 neurons, the accuracy drops beyond 3 layers.

Hence, considering the balance between the number of layers, number of neurons, and accuracies for various Δ values, 3 layers with 512 neurons architecture seems to be the optimal choice.

2) Dropout: Dropout [18] is used during training to "drop out" or cut off certain neurons and their connections randomly with a certain probability during training. It serves two purposes - Ensemble Learning [19] and Regularization [20]. Ensemble learning helps improve predictive performance by combining the outputs of multiple AI/ML models or multiple instances of the same AI/ML model trained on different subsets of data. However, training a large number of models is extremely prohibitive both in terms of time and computational resources. With dropout, in each backward pass, only a subset of weights are updated, effectively resulting in a new neural network. The subset of weights that are updated varies with each forward and backward pass, thus creating the combined effect of training several neural networks without the significant computational overhead. Dropout also has a regularizing effect on training in that it makes training noisy and, as a result, doesn't allow neurons to co-adapt. This means that neurons in one layer will learn not to depend on or compensate for other neurons because these very neurons may randomly be dropped out. This encourages neurons to become more independent, thus preventing overfitting. All the models described in this paper use dropout with a probability of 0.5 in each of the hidden layers.

3) Activations and Backpropagation: The hidden layers use a ReLU activation function, and the output layer uses a Sigmoid activation function. The use of the sigmoid activation

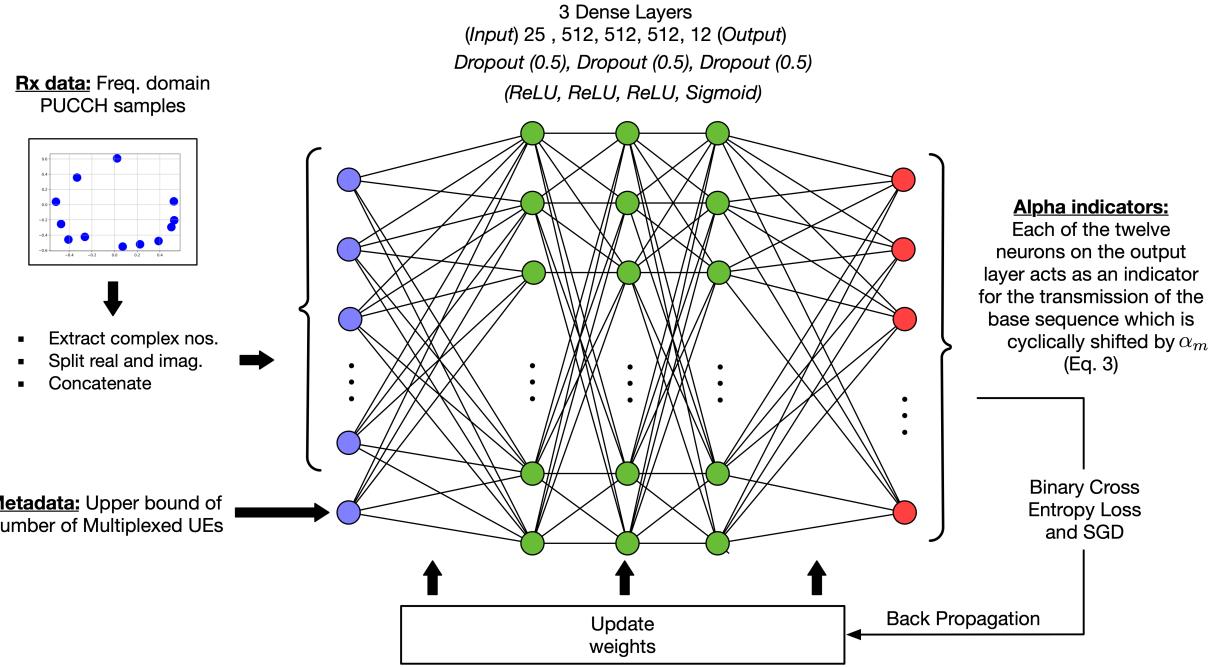


Fig. 4: UCINet0 Architecture for PUCCH Format 0 decoding with 25 neurons in the input layer, 512 neurons in each of the three hidden layers and 12 neurons in the output layer.

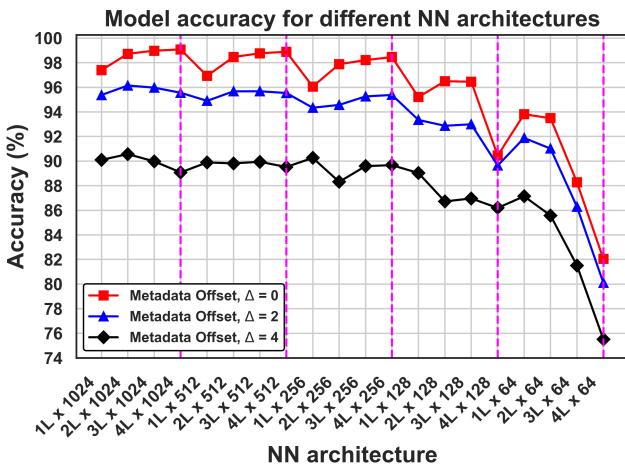


Fig. 5: Model accuracy for different FCN architectures at various values of Δ . Here, Δ represents the maximum metadata offset between the true value and L2's upper bound for the number of multiplexed UEs. We use these results to arrive at the final UCINet0 architecture presented in this paper.

Descent (SGD) with a learning rate of 10^{-2} and a momentum parameter of 0.9.

D. Training and Testing

We consider received signal data corresponding to 5 SNR values i.e., 0, 5, 10, 15, 20 dB. The training instances are designed such that they encompass all the possible cases of multiplexed UEs including the cases where no UEs are transmitting. During training, we assume that the metadata input always holds the correct number of UEs present in the signal. Furthermore, based on the insights gained from our previous work [6], we train the neural network only using data with a middle SNR of 10dB, rather than all the SNRs. We have found that this model achieves good generalization across the entire range of SNR during inference/testing. 75% of the data is used for training from which a further 30% is used for validation and fine-tuning of hyper-parameters. We train the model for 150 epochs(an epoch is 1 pass of the entire dataset through the neural network). The training and testing was done using Python and TensorFlow. The code is available at [16] for download.

In the inference phase, we test the model using data from all SNRs. We also incorporate the fact that the number of multiplexed UEs in a real-time deployment may not be exactly known. The L2 scheduler in a gNB merely provides a value for the upper bound of the number of UEs a receiver could possibly expect in a given PUCCH allocation. This is usually a consequence of one of the following scenarios:

- A UE does not transmit any SR if no uplink grants are required because of lack of data into buffer
- A UE does not transmit feedback (on HARQ resources) if the prior downlink message is not detected.

function is motivated by the fact that in a multi-label classifier, the sum of all the output probabilities need not be 1. Each individual output neuron can be thought to be a part of an independent binary classifier. In other words, since each output neuron corresponds to an α value, the probabilities of each output neuron indicate the confidence with which the NN detects the presence of a certain UE with a certain α value. The output probabilities are used to compute a binary cross entropy loss that is back-propagated using Stochastic Gradient

During testing, we feed this upper bound value as the metadata input to the neural network, which is offset from the true value. To test the robustness of the trained model, we consider various cases of the maximum metadata offset Δ between the true value and L2's upper bound for the number of multiplexed UEs. For example, $\Delta = 0$ indicates that the number of UEs provided as the metadata input to the NN matches the number of UEs actually transmitting in the given allocation. $\Delta = 2$ indicates that for each test instance, the number of UEs provided as the metadata input to the NN could be offset by any value between 0 and 2 with 0 being the best case (no offset) and 2 being the worst. In this paper, we show results for $\Delta = 0, 2, 4$.

Eq. (4) shows the offset added to the true value of the number of multiplexed UEs, N_{UE} to obtain \tilde{N}_{UE} which is the expected number of multiplexed UE at the receiver, as indicated by L2.

$$\tilde{N}_{UE} \in [N_{UE}, N_{UE} + \Delta] \quad (4)$$

Note that since the maximum number of UEs that can be multiplexed in a given resource block is 12, the value of \tilde{N}_{UE} is chosen accordingly.

IV. RESULTS AND DISCUSSION

In this section, we describe the performance results of the UCINet0 Neural Network configuration shown in Figure 4.

A. Performance Metrics and their Motivation

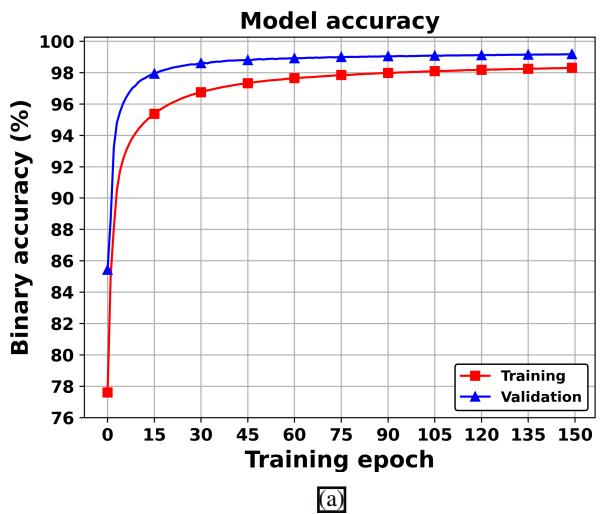
We use model test accuracy as the main performance metric. We also show confusion matrices to indicate the distribution of model predictions. Wherever appropriate, we show comparisons of the model performance with the corresponding performance shown by the DFT algorithm.

Accuracy is simply the number of correct predictions in relation to total predictions. Though it is a useful high-level metric to gauge model performance, in cases where the classes in a dataset are unequally distributed or when, as is the case in this paper, there are more than 2 classes, accuracy alone as a metric for an NN classifier does not offer the full picture of model performance. It is helpful to know if the model classifies all classes equally well or if it is more "confused" by certain classes compared to others. Hence, calculating a confusion matrix gives a better insight into any patterns that may exist in both correct and wrong classifications made by the model.

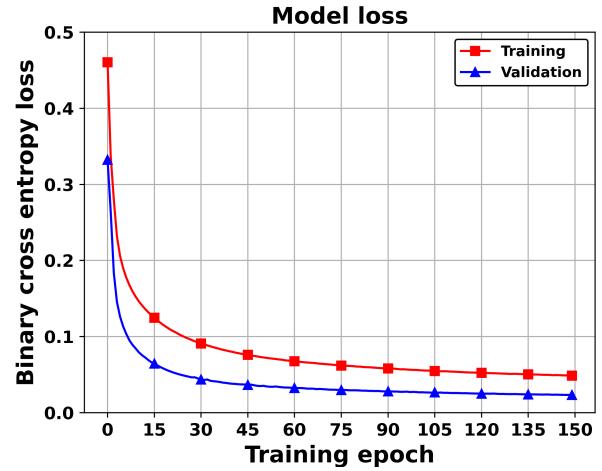
In this paper, we show 3 types of confusion matrices: (1) A multi-label confusion matrix that shows how well each α is predicted; (2) A confusion matrix showing how well the model predicts the correct number of multiplexed UEs, and (3) A confusion matrix showing how each value of α (including the cases when no α is selected) is classified. We present this last metric in the form of a column chart for ease of visualization. These 3 types of confusion matrices provide a sense of interpretability of the inner workings of the NN model.

B. Accuracy and Loss

Figure 6a shows the training and validation accuracy with respect to the training epochs. Figure 6b shows the training and validation loss. Both curves follow the expected pattern for a well-fit model. The accuracy/loss gradually increases/decreases with epoch and eventually converges. We also note that the validation accuracy and loss values are slightly better than the corresponding training values. One possible explanation for this could be the use of dropout. As stated above, a dropout with probability 0.5 drops neurons 50% of the time during training. However, during validation and testing, the entire model is used, leading to a higher accuracy/lower loss.



(a)



(b)

Fig. 6: Training was performed only using simulated data at SNR = 10 dB. (a) Model Accuracy vs. Epoch for simulated data. (b) Model Loss vs. Epoch for simulated data.

Figure 7a shows the test accuracy of the model versus SNR for simulated data. Figure 7b shows the test accuracy of the model versus SNR for hardware-captured data. In both cases, the test dataset for each SNR contains a combination of all possible values of $N_{UE} = \{0, 1, 2, \dots, 12\}$. Both Figures 7a and 7b show that model accuracy increases with SNR. For the

simulated data, the model significantly outperforms the DFT algorithm for all values of Δ .

For the hardware captured data, when $\Delta = 0$, the model performs slightly better than the DFT algorithm. But as the SNR increases, the accuracies of the model and that of the DFT algorithm converge. This is because $\Delta = 0$ constitutes a best-case scenario in which both algorithms are aided by the fact that the receiver knows exactly how many UEs are multiplexed. At higher values of Δ , the convergence is not seen, and the NN model shows a significant gain compared to the DFT algorithm.

Furthermore, from Figures 7a and 7b, we observe an interesting difference between the performance of the neural network with simulated data versus hardware captured data. For all values of Δ , there is a significant SNR gain with the hardware data compared to simulated data. For example, when $\Delta = 0$, the accuracy with hardware captured data reaches 100% at as low an SNR as 5 dB. In contrast, the accuracy with simulated data requires more than 15dB SNR to reach 100%. This difference could be attributed to the characteristics of the datasets themselves. The simulated dataset includes a strong fading component that impairs the transmitted signal. However, owing to the short propagation distance, such a strong fading component is absent in the current 5G testbed setup that was used to obtain the hardware captures. We note that the captures still incorporate other real-world impairments caused by front-end hardware elements like filters, DACs, ADCs, up/down converters, and amplifiers. In summary, the hardware dataset models impairments due to front-end elements well. It also models over-the-air impairments but does not encapsulate all characteristics of a typical communication link between a gNB and UE that might span hundreds of meters. On the other hand, the simulated data models over-the-air characteristics well but it doesn't model impairments due to hardware elements. We also point out that the IIT Madras 5G Testbed is moving towards deployment of Remote Radio Heads at on-campus sites which will allow the inclusion of even more realistic field data into future work.

Another observation from Figures 7a and 7b is that the DFT algorithm performs worse on hardware data, showing that it is negatively affected not only by impairments from hardware elements but also by even the slightest over-the-air fading. On the other hand, the NN Model outperforms the DFT method in all scenarios of hardware and simulated datasets.

Figure 8a shows the test accuracy of the NN model versus the number of multiplexed UEs (N_{UE}) for simulated data. Figure 8b shows the test accuracy of the NN model versus the number of multiplexed UEs (N_{UE}) for hardware-captured data. In both cases, the test dataset for each N_{UE} contains a combination of all possible values of SNR = {0, 5, 10, 15, 20} dB. It is clear that the accuracy decreases as the maximum offset Δ increases. Note that no value is plotted when $N_{UE} = 0$ and $\Delta = 0$ since this case corresponds to the scenario in which no UEs are transmitting on a given Resource Block and the receiver knows apriori that no UEs are scheduled as well. In this case, the receiver need not even run. When $N_{UE} = 12$, all the multiplexed UEs transmit only Scheduling Requests (SR). For both the simulated and hardware datasets,

the DFT algorithm selects all 12 values from the 12-point DFT and assigns one value to each UE. Since all UEs are transmitting SRs, the order of assignment doesn't matter, and hence the accuracy is always 100%. The NN model also seems to recognize that the only scenario in which 12 UEs can be multiplexed is when they all transmit exactly the same information, i.e., SRs.

Lower values of N_{UE} tend to result in lower accuracy values. This is because the DFT algorithm picks up the top N_{UE} values of the 12-point DFT, and the probability of a true value lying in one of the $12 - N_{UE}$ locations increases as N_{UE} decreases. We observe a similar pattern in the case of decoding with the proposed UCINet0 as well.

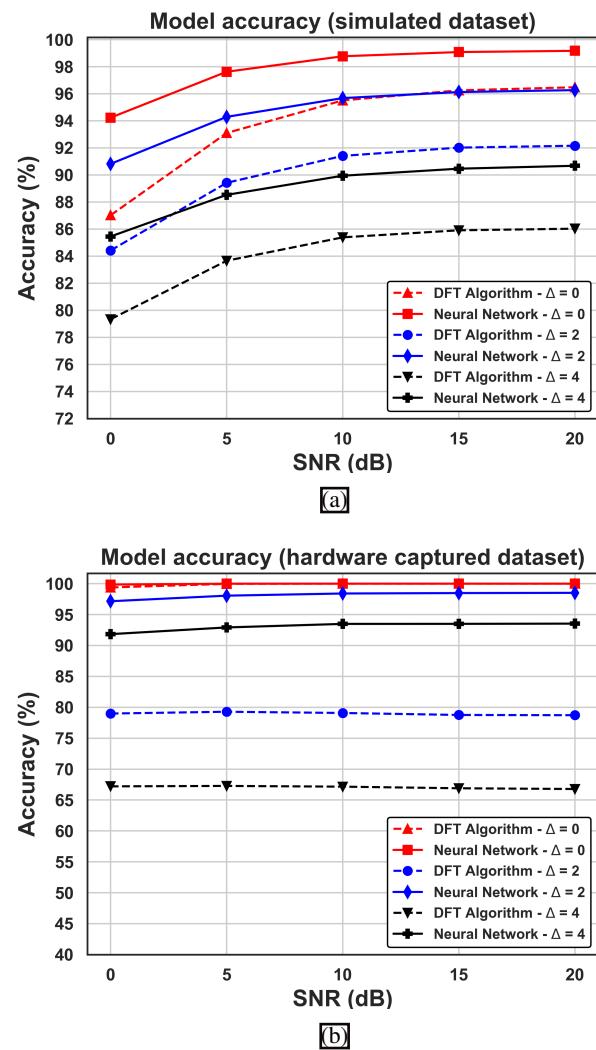


Fig. 7: Testing Accuracy vs SNR for (a) simulated data and (b) hardware captured data. The test dataset for each SNR contains a combination of $N_{UE} = \{0, 1, 2, \dots, 12\}$.

C. Confusion Matrices and Column Charts

The confusion matrices in Figures 9 and 10 indicate the overall multi-label classification performance of the model. Figures 9 and 10 show the multi-label confusion matrix for

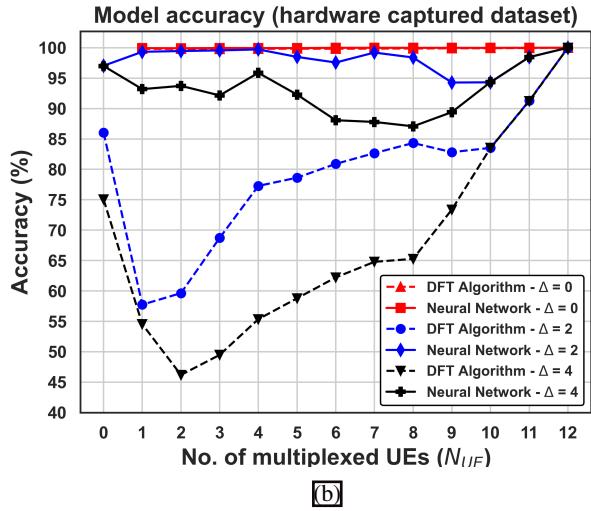
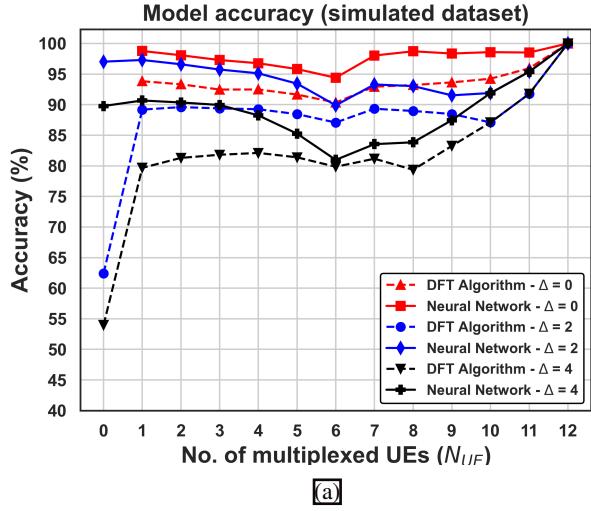


Fig. 8: Testing Accuracy vs Number of Multiplexed UEs (N_{UUE}) for (a) simulated data and (b) hardware captured data. The test dataset for each value of N_{UUE} contains a combination of $\text{SNR} = \{0, 5, 10, 15, 20\} \text{ dB}$.

the best ($\Delta = 0$) and worst ($\Delta = 4$) cases of 20dB SNR respectively, for both simulated data and hardware-captured data. Multi-label classification can be thought of as a combination of several independent binary classifiers. Diagonally dominant confusion matrices indicate that each of the binary classifiers is well-trained. For a given SNR, as Δ increases, the model leans more towards false positives because the model (whose 25th input is $N_{UUE} \in [N_{UUE}, N_{UUE} + \Delta]$) is trying to predict N_{UUE} number of UEs. The value of Δ is always positive, meaning that the number of expected UEs, as indicated by L2, is always greater than or equal to the actual number of UEs transmitting on a given resource allocation. Hence, the model almost always over-predicts.

However, it should be noted that the multi-label confusion matrix does not show the exact details of misclassifications. To gain a deeper understanding of the mistakes made by the model, consider the matrices in Figures 11 to 14. For each instance of the received PUCCH signal, there are two



Fig. 9: Multi-label confusion matrix at SNR = 20 dB and $\Delta = 0$ for (a) simulated test dataset and (b) hardware captured test dataset. Here, α represents the UCI specific cyclic shift.

characteristics to correct classification. The first characteristic is correctly identifying the number of multiplexed UEs (N_{UUE}) embedded in the signal. Even though this is not an explicit output of the NN model, we believe that the model is learning to estimate the number of multiplexed UEs. Our belief is based on the fact that when the number of possible multiplexed UEs is not given as metadata to NN input, the accuracy drops significantly. The second characteristic of correct classification is derived from the explicit output of the NN model, which is the correct prediction of the value of α for each of the multiplexed UEs.

Figures 11 and 12 show confusion matrices for the number of multiplexed UEs for both simulated data and hardware-captured data at an SNR of 0 dB and Δ values of 0 and 4 respectively. Figures 13 and 14 confusion matrices for the number of multiplexed UEs for both simulated data and hardware-captured data at an SNR of 20 dB and Δ values of 0 and 4 respectively. We notice that all the confusion matrices for the prediction of the number of multiplexed UEs are largely diagonally dominant. Furthermore, for any SNR, when $\Delta = 0$, the error in the predicted number of UEs fluctuates between +1 and -1 in the majority of instances, confirming the high accuracy values in Figure 7. As Δ increases, the error tends to grow on the positive side, meaning that the NN model predicts more UEs than actually present. As stated above, the

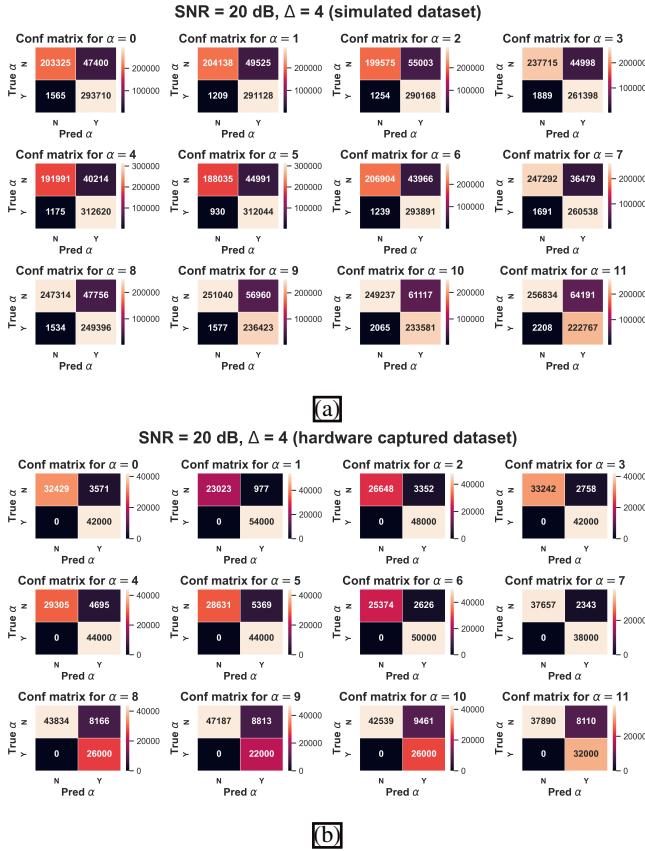


Fig. 10: Multi-label confusion matrix at SNR = 20 dB and $\Delta = 4$ for (a) simulated test dataset and (b) hardware captured test dataset.

overprediction is because Δ is always positive.

The column charts in Figure 15 and Figure 16 show the percentage of correct and incorrect predictions for the transmission of every α from 0 through 11. The result of the NN model's prediction for the scenario where a particular α has not been transmitted by any UE is also captured in the last column labeled *No Tx*. Correct prediction in the last column indicates that the model was able to correctly identify the non-transmission of a particular α value, and incorrect detection determines the instances where a particular α has not been transmitted but incorrectly predicted.

From Figure 15 and Figure 16, it is evident that the NN model is not biased towards any particular α and the cyclic shifts are correctly identified in the majority of instances.

D. Implications of Model Performance on a 5G System

In general, both false detections and missed detections are detrimental to PUCCH Format 0 performance. However, from the context of sustaining a communication link between a gNB and a UE, a small number of false detections are more tolerable than missed detections. For example, when a gNB falsely detects a Scheduling Request, it allocates a small uplink bandwidth to the UE. Since the UE does not have any valid data to transmit, it would populate the uplink packet with padding information and indicate its lack of data in the Buffer

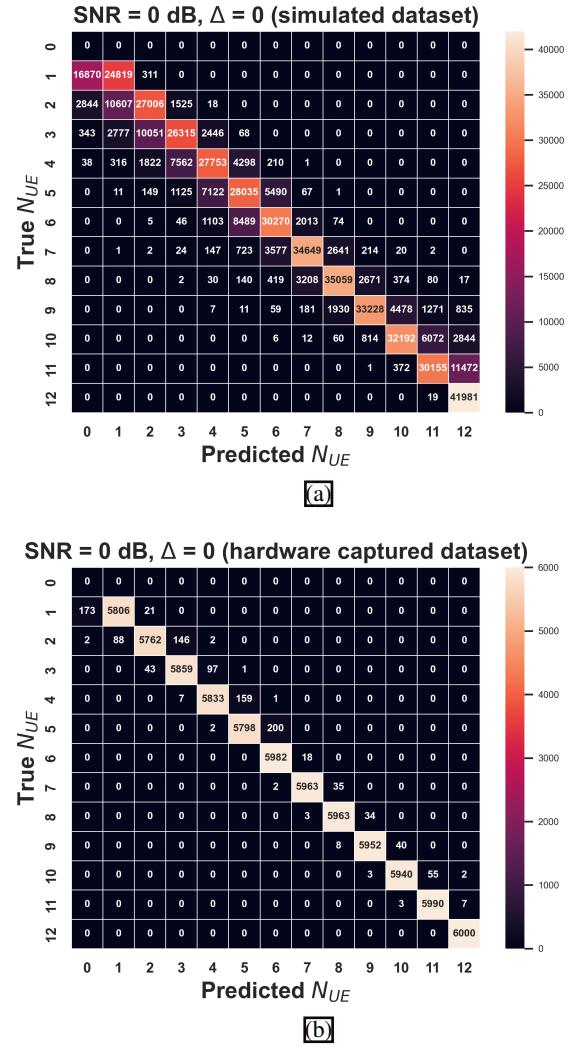


Fig. 11: Confusion matrix for the number of multiplexed UEs N_{UE} , at SNR = 0 dB and $\Delta = 0$ for (a) simulated test dataset and (b) hardware captured test dataset.

Status Report (BSR) [21]. No further uplink grants will be allocated to the UE.

On the other hand, if a gNB misses the detection of an ACK, it will assume that the prior downlink transmission was not decoded by the UE. In this scenario, a significant downlink bandwidth will be allocated to the UE for the retransmission. Successive missed detection of ACKs will lead to the termination of the radio link (referred to as Radio Link Failure) between the gNB and the UE.

Hence, occupying a small uplink bandwidth (due to false detection of an SR) is preferable to occupying a large downlink bandwidth (due to missed detection of ACKs) since recovery from a Radio Link Failure (RLF) involves heavy signaling for the UE to reattach.

As shown in Figures 10, 12 and 14, at low SNRs or high Δ values, the proposed UCINet0 tends to overpredict the number of UEs actually present, rather than underpredict. This shows that the model, even when it makes a wrong prediction, is good at avoiding scenarios leading to RLFs.

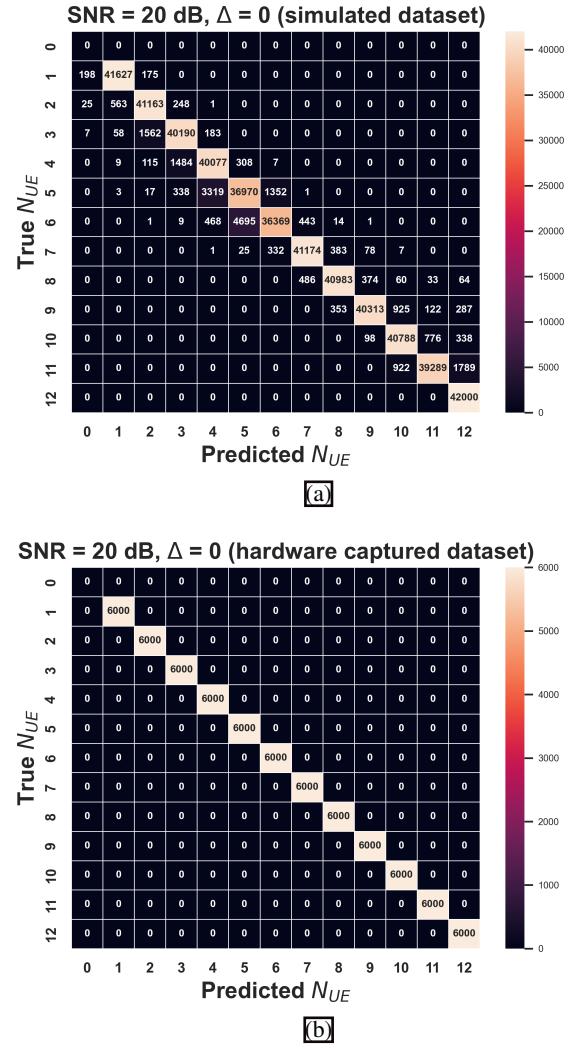
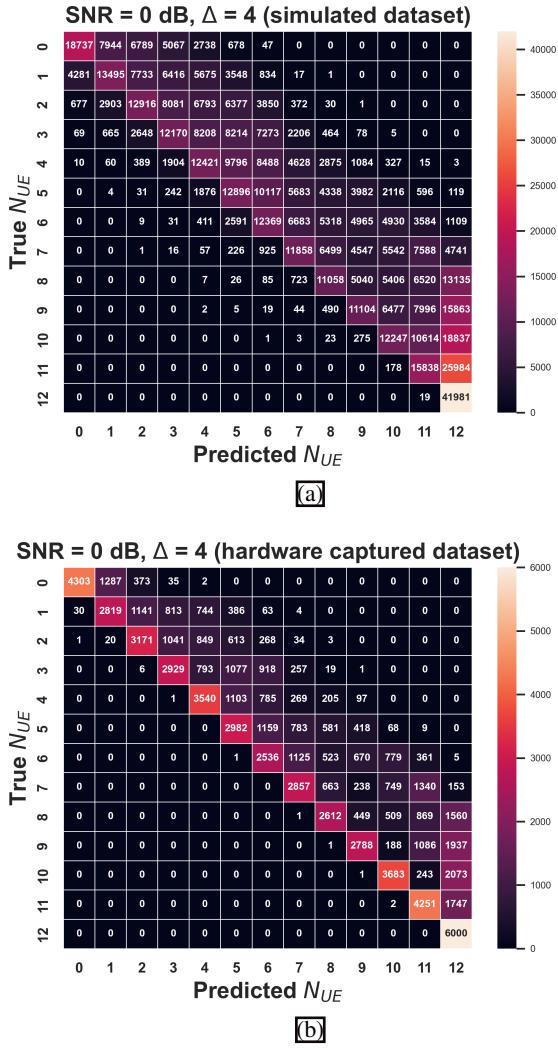


Fig. 12: Confusion matrix for the number of multiplexed UEs N_{UE} , at SNR = 0 dB and $\Delta = 4$ for (a) simulated test dataset and (b) hardware captured test dataset.

Fig. 13: Confusion matrix for the number of multiplexed UEs N_{UE} , at SNR = 20 dB and $\Delta = 0$ for (a) simulated test dataset and (b) hardware captured test dataset.

E. Model Complexity Analysis

Figure 17 shows the number of trainable parameters for different FCN architectures. The number of trainable parameters increases with the number of layers and neurons. An increase in the number of trainable parameters leads directly to an increase in the latency incurred due to fetching the weights from memory and neural network computations. Current 5G systems function with fundamental time units on the order of hundreds of microseconds. To cater to this requirement, the processing of all the physical layer channels allocated to each user must be completed in a duration of the order of tens of microseconds. For example, in our 5G Testbed [7], we use a subcarrier spacing of 30kHz, which translates to a slot duration of $500\mu s$. Hence, the physical layer processing of all the users allocated in that slot needs to be completed within this time frame. If 10 PUCCH Format 0 instances are allocated in an uplink slot, each of them should be processed in no more than $50\mu s$. This time constraint also applies to AI/ML-

based approaches such as neural networks. Hence, FPGAs are a promising choice for hardware implementation of such time-constrained neural networks.

In order to deploy a version of UCINet0 on hardware devices such as FPGAs, the trade-offs between model performance (Figure 5) and model complexity (Figure 17) must be considered. These two parameters help us choose the best-suited model based on the available memory and compute resources on the hardware. The memory and bandwidth requirements for storing and retrieving the weights of Neural Networks like UCINet0 can be met by Xilinx's latest Versal series FPGAs [22], which contain AI engines to perform matrix multiplications efficiently.

CONCLUSION

In this paper, we have designed a generalized Machine Learning based PUCCH Format 0 receiver that decodes all the combinations of UCI bits across several multiplexed users. The proposed UCINet0 model is a multi-label classifier

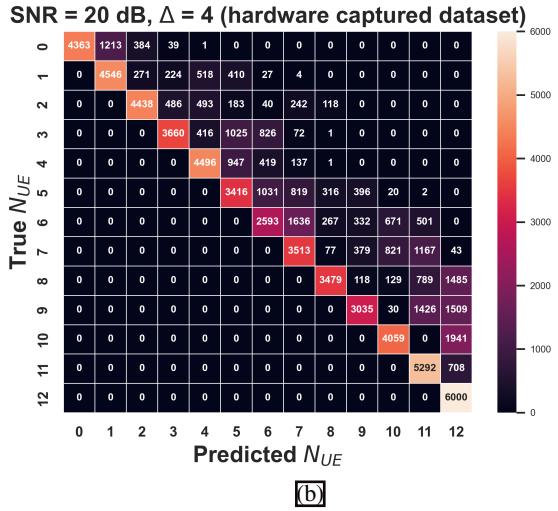
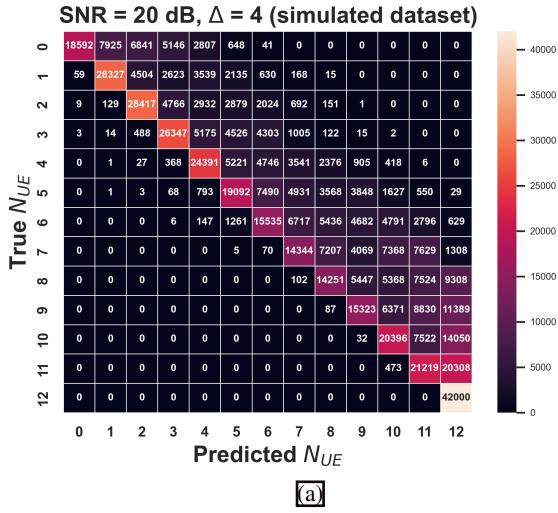


Fig. 14: Confusion matrix for the number of multiplexed UEs N_{UE} , at SNR = 20 dB and $\Delta = 4$ for (a) simulated test dataset and (b) hardware captured test dataset.

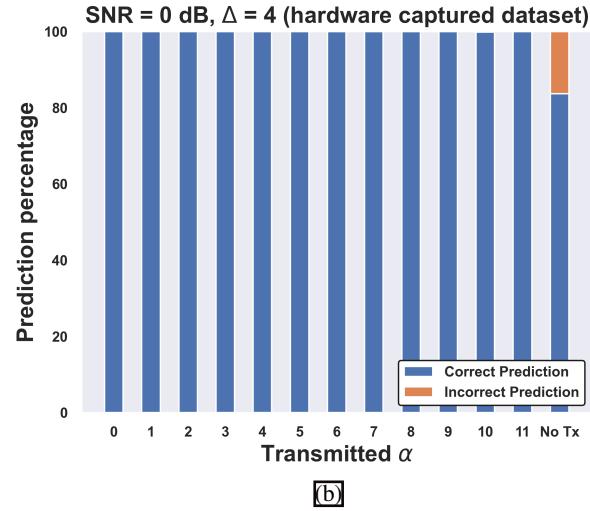
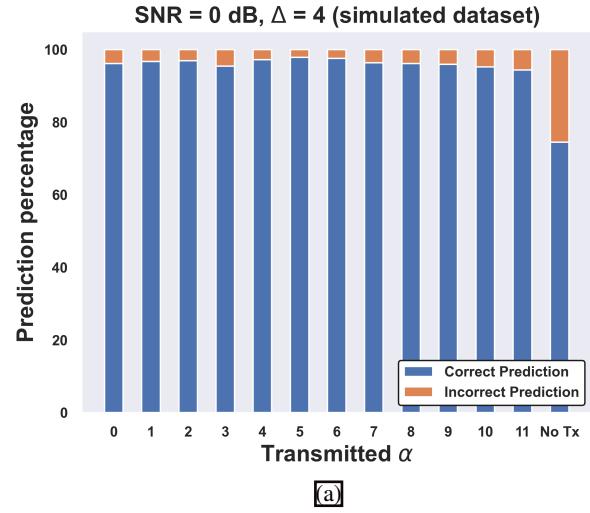


Fig. 15: Bar graph for the cyclic shift α at SNR = 0 dB, $\Delta = 4$, and with $N_{UE} = \{0, 1, 2, \dots, 12\}$ for (a) simulated test dataset and (b) hardware captured dataset.

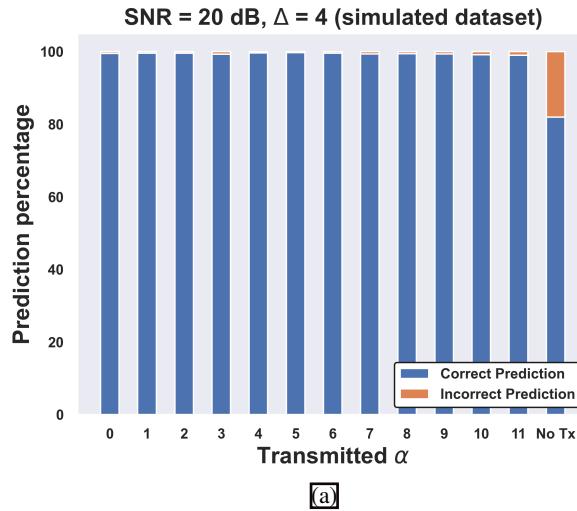
that showed superior performance compared to conventional correlation-based approaches by a significant margin. The inclusion of field datasets in the testing of the model ensures robustness and the ability of the network to work in a wide range of scenarios. In addition to standard AI/ML metrics like accuracy, we have also provided insights into the interpretability of the NN model through confusion matrices and column charts of detection percentages of individual classes.

ACKNOWLEDGMENT

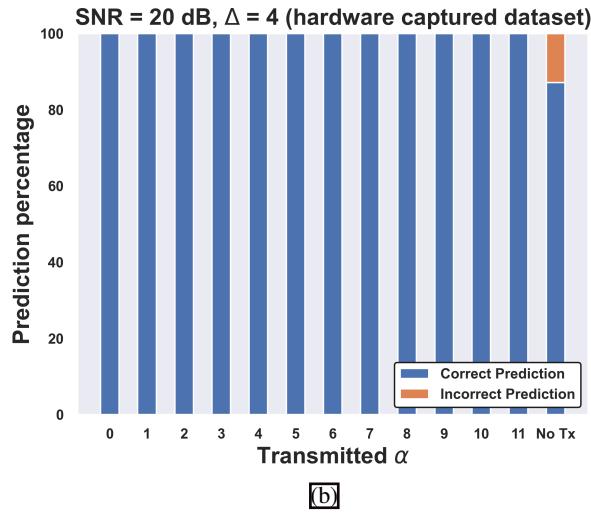
The authors would like to thank the Department of Telecommunications (DOT), India for funding the 5G Testbed project and the Ministry of Electronics and Information Technology (MeitY) for funding this work through the project "Next Generation Wireless Research and Standardization on 5G and Beyond".

REFERENCES

- C. Johnson, *5G New Radio in Bullets*, 1st ed. Farnham, England: Self-Published, 2019.
- 3GPP, "Physical Channels and Modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 2021, version 16.5.0.
- 3GPP, "Multiplexing and Channel Coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 2021, version 16.5.0.
- 3GPP, "Physical Layer Procedures for Control," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.213, 2021, version 16.5.0.
- J. Ryu. (2019) 5G/NR - Initial Attach. Accessed: Oct 30 2023. [Online]. Available: https://www.sharetechnote.com/html/5G/5G_CallProcess_InitialAttach.html#SA_InitialAttach_OverallSignalingFlow
- A. K. Yerrapragada, J. K. Sattianarayanan, A. Gautam, and R. K. Ganti, "Machine Learning Decoder for 5G NR PUCCH Format 0," in 2023 National Conference on Communications (NCC), 2023, pp. 1–6.
- IIT Madras. (2019) 5G Testbed IIT Madras. Accessed: Oct 30, 2023. [Online]. Available: <http://www.5gbiitm.in>
- L. Kundu, G. Xiong, and J. Cho, "Physical Uplink Control Channel Design for 5G New Radio," in 2018 IEEE 5G World Forum (5GWF), 2018, pp. 233–238.



(a)



(b)

Fig. 16: Bar graph for the cyclic shift α at SNR = 20 dB, $\Delta = 4$, and with $N_{UE} = \{0, 1, 2, \dots, 12\}$ for (a) simulated test dataset and (b) hardware captured dataset.

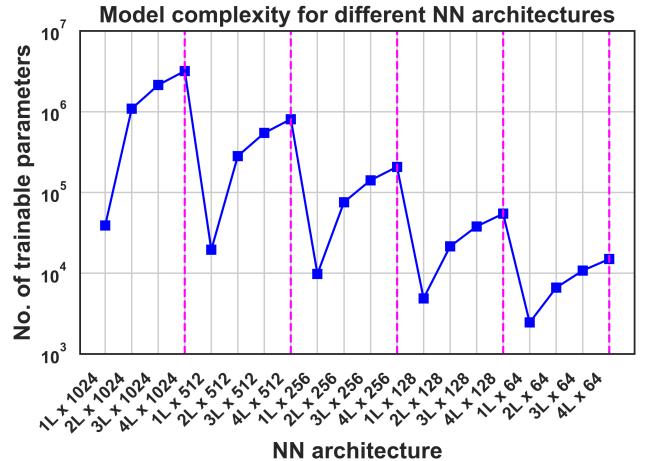


Fig. 17: Line graph showing the number of trainable parameters for different FCN architectures. These parameters increase with the number of layers and neurons.

- [16] 5G Testbed, IIT Madras, “PUCCH Format 0 AI/ML Source Code and Datasets,” <https://aifor6g.com/available-after-review>, accessed: 07 Jan. 2024]
- [17] Work Group-4, “O-RAN Working Group 4 (Open Fronthaul Interfaces WG) Control, User and Synchronization Plane Specification,” Open RAN Alliance, Technical Specification (TS), 2022, version 11.00.
- [18] 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. 1, pp. 1929–1958, 2014.
- [19] K. Hara, D. Saithoh, and H. Shouno, “Analysis of Dropout Learning Regarded as Ensemble Learning,” in *Artificial Neural Networks and Machine Learning – ICANN 2016*, 2016, pp. 72–79.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, <http://www.deeplearningbook.org>.
- [21] S. Lin and J. Yu, “Happy Bits Based BSR Mechanism in 5G Networks,” *IOP Conference Series: Materials Science and Engineering*, vol. 466, no. 1, 2018.
- [22] M. Wierse, “Evaluation of Xilinx Versal Device,” Bachelor Thesis, ETH Zurich, 2023.

- [9] B. Park, H. Cheon, E. Ko, C. Kang, and D. Hong, “A blind OFDM synchronization algorithm based on cyclic correlation,” *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 83–85, 2004.
- [10] S. Ma, X. Pan, G.-H. Yang, and T.-S. Ng, “Blind Symbol Synchronization Based on Cyclic Prefix for OFDM Systems,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1746–1751, 2009.
- [11] M. Wahlgqvist, R. Larsson, and C. Ostberg, “Time synchronization in the uplink of an OFDM system,” in *Proceedings of Vehicular Technology Conference-VTC*, 1996.
- [12] Y.-H. Kim, H. Ju, C. B. Jeong, and M.-S. Lee, “Performance comparison of DTX detection schemes for 5G NR PUCCH,” in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1391–1394.
- [13] S. Tadavarty and N. K. Chaval, “Performance Analysis of 5G NR PUCCH Format 0 Receiver,” in *2021 IEEE 18th India Council International Conference (INDICON)*, 2021, pp. 1–5.
- [14] D. Phan and T. H. P. Nguyen, “An Enhanced Method to Improve Signal Reception Performance for 5G NR Physical Uplink Control Channel with Frequency Hopping Configuration,” in *2021 International Conference on Advanced Technologies for Communications (ATC)*, 2021, pp. 13–17.
- [15] 3GPP, “Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface,” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.843, 2023, release 18.