

# Deep Learning-Based Decoding and AP Selection for Radio Stripe Network

Aman Kumar Mishra and Vijayakumar Ponnusamy\*

Department of ECE, SRMIST, Chengalpattu, 603203, India

\*Corresponding Author: Vijayakumar Ponnusamy. Email: vijayakp@srmist.edu.in

Received: 19 June 2021; Accepted: 04 August 2021

**Abstract:** Cell-Free massive MIMO (mMIMO) offers promising features such as higher spectral efficiency, higher energy efficiency and superior spatial diversity, which makes it suitable to be adopted in beyond 5G (B5G) networks. However, the original form of Cell-Free massive MIMO requires each AP to be connected to CPU *via* front haul (front-haul constraints) resulting in huge economic costs and network synchronization issues. Radio Stripe architecture of cell-free mMIMO is one such architecture of cell-free mMIMO which is suitable for practical deployment. In this paper, we propose DNN Based Parallel Decoding in Radio Stripe (DNNBPDRS) to decode the symbols of User Equipments (UEs) in the uplink in a parallel fashion to reduce computational complexity by reducing delay in processing. Moreover, to solve the issue of Access Point (AP) selection in radio stripe networks, we propose a Channel link-based AP selection (CLBAPS) algorithm to choose the best APs in terms of channel link quality. The proposed DNNBPDRS framework not only improves Symbol Error Rate (SER) performance when compared to counterparts but is also proved to be comparatively far lesser computational complex. Moreover, the numerical result indicates the proposed AP selection algorithm CLBAPS performs better than random selection of AP in radio stripe networks.

**Keywords:** Cell-free massive MIMO; deep learning decoding; beyond 5G; radio stripe; AP selection

## 1 Introduction

Massive MIMO (mMIMO) is a key wireless physical layer technology in 5G and beyond 5 G (B5G) networks [1], given the features it offers, very high spectral efficiency (SE) by at least ten times [2], improve the radiated energy efficiency [3–6]. However, the adoption of mMIMO (in the current cellular network) faces two major issues. Firstly, as more numbers of antenna is being installed at the base station (mMIMO), inter-cell interference is becoming major bottle neck, this is because inter-cell interference is inherent to cellular paradigm [7]. Secondly, mMIMO suffers from large variations in signal-to-noise ratio between cell-center and cell-edge users [8].

The above-mentioned issues restrict the adoption of mMIMO in beyond 5G (B5G) networks, which is resolved by Cell-Free mMIMO network wherein Central processing unit (CPU) is connected to access point



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

(APs), which jointly serves the given set of User Equipments (UEs) (or all UEs) in the network [9–11]. Authors in [11] discuss the different architecture of cell-free massive MIMO, where they claim centralized implementation of cell-free massive MIMO ‘Level 4’, which performs best in terms of spectral efficiency compared to other architectures. Here, all APs sent their respective received signals in the uplink to CPU over parallel front-haul, which process all the received signals., this requires dedicated front-haul and power supply to each AP, which imposes huge economic costs. Moreover, attaining network synchronization in Cell-Free mMIMO network would be extremely challenging since each AP is connected to CPU over parallel front-haul.

One promising architecture for practical implementation of Cell-free network is using radio stripe [12]. Radio stripes can be easily deployed at public place, stadium, transport (train, bus, etc.) to provide truly ubiquitous connectivity everywhere. Moreover, radio stripe is extremely flexible to deploy, does not require highly trained personnel, needs one connection either to fronthaul or directly to CPU [12].

### **1.1 Related Works**

Given the architecture of radio stripe, where APs are connected sequentially and CPU being connected to last AP (daisy chain architecture) the sequential processing becomes obvious choice. Therefore, authors [13] propose sequential uplink processing for cell-free massive MIMO based on radio stripe. Here, each AP computes the local channel estimate in the uplink, make soft estimate of the transmitted signal and thereby forwards local CSI estimate, soft estimate and error statistic to next AP. Upon receiving the information from the previous AP, the AP improves its own soft estimate. This sequential processing goes on till the last AP, the final estimate is forwarded to CPU for final decoding. Authors [13] claim the proposed processing mechanism achieves comparable performance to ‘Level 4’, while using 90% less signaling in the front-haul. In [8], authors propose Optimal Sequential Linear Processing (OSLP), an uplink sequential processing algorithm which they prove optimal both in the maximum spectral efficiency sense and minimum Mean Square Error (MSE) sense.

However, sequential processing leads to higher processing latency [14]. Therefore, authors in [14] propose two parallel processing schemes in the Radio Stripe network, namely interference-aware MR processing and distributed regularized zero-forcing (D-RZF). Authors [15], propose Q-LMMSE, which implements MRC processing leading to low front-haul loading, simple AP hardware and fast parallel computations.

### **1.2 Motivation**

All works discussed so far in cell-free massive MIMO based on radio stripe [8,13–15] has huge computational needs (for example channel estimation), since next generation network will involve large numbers of APs and UEs. These high computational costs can only be reduced by the application of Deep Learning (DL). DL replaces the complex computations with the trained model, which reduces the complexity considerably.

Authors in [16] propose DeepSIC, which is a data-driven technique for detecting symbols in the Multi-User MIMO (MU-MIMO) scenario. It is based on an iterative SIC algorithm, a MIMO-based detection scheme. Results show it outperforms Iterative SIC in the presence of CSI uncertainty. Since it is an iterative approach, detection is carried out in iteration in a single base station (BS). Our other works on interference cancellation are for Software Defined Radio (SDR) platform [17]. Motivated by the above [16], in our previous work [18], we propose DNN-based distributed sequential uplink processing (DBDSUP) for detecting symbols in the uplink of Cell-Free massive MIMO System Based on Radio Stripe. Here, each AP consists of  $K$  (numbers of UEs) parallel Soft Estimate Network (SEN), which computes the conditional distribution of each user at each AP (each AP carries out a single iteration).

The conditional distribution of each user consists of the estimation of each symbol in the constellation. The Symbol Error Rate (SER) of the proposed mechanism ‘DBDSUP’ is evaluated. However, the processing here is sequential.

In order to take advantage of DL (to reduce computational complexity) and parallel processing (to reduce latency) there is need to develop a mechanism which is based on DL and does parallel processing.

Moreover, next-generation radio stripe network would involve large numbers of APs (typically in the range of 1000 s in particular give area like train, the bus as shown in Fig. 2), if all APs sequentially decodes transmitted symbol of given user, it would impose huge complexity on the system. Therefore, it becomes extremely important to develop a mechanism for users to select APs in Cell-free massive MIMO based on radio stripe.

Authors in [19] propose an effective channel gain-based algorithm in massive cell-free MIMO to assign users to each AP sequentially. Here author considers two metrics, first measure the channel quality of each user to all APs, and another one is to calculate the effective channel gain between every user and AP in the system, which reduces interferences and leads to a higher sum rate. Other works on AP selection focus on Machine learning (ML) [20].

This paper aims to propose a deep learning-based parallel decoding mechanism and AP selection algorithm for cell-free massive MIMO based on radio stripe.

### 1.3 Contributions

The contribution is stated in the points below.

- We propose DNN based Parallel Decoding in Radio Stripe (DNNBPDRS). This has two benefits, firstly, DNN processing (reduces the computational complexity) and secondly, parallel processing reduces the latency, making it very attractive for beyond 5G (B5G) technology (radio stripe is slated to be 6th generation wireless technology).
- To solve the issue of AP selection in the radio stripe network. We propose a Channel link-based AP selection (CLBAPS) algorithm to choose the best APs (in terms of channel link quality).

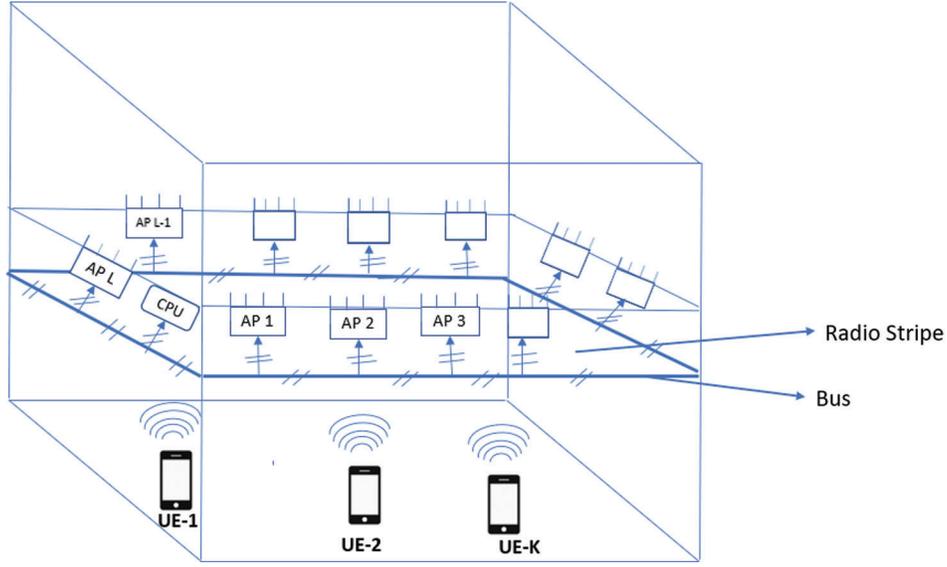
The rest of the paper is organized as follows. In Section 2, we discuss the network model of radio stripe . Section 3, elaborate about different aspect of proposed DNN based Parallel Decoding in Radio Stripe (DNNBPDRS). In Section 4, a detailed description of the proposed AP selection algorithm, Channel link-based AP selection (CLBAPS), is detailed. Simulation setup, training of neural network and numerical results are discussed in Section 5. Finally, the article is concluded in Section 6.

## 2 Radio Stripe Network Model of Cell-Free Massive MIMO

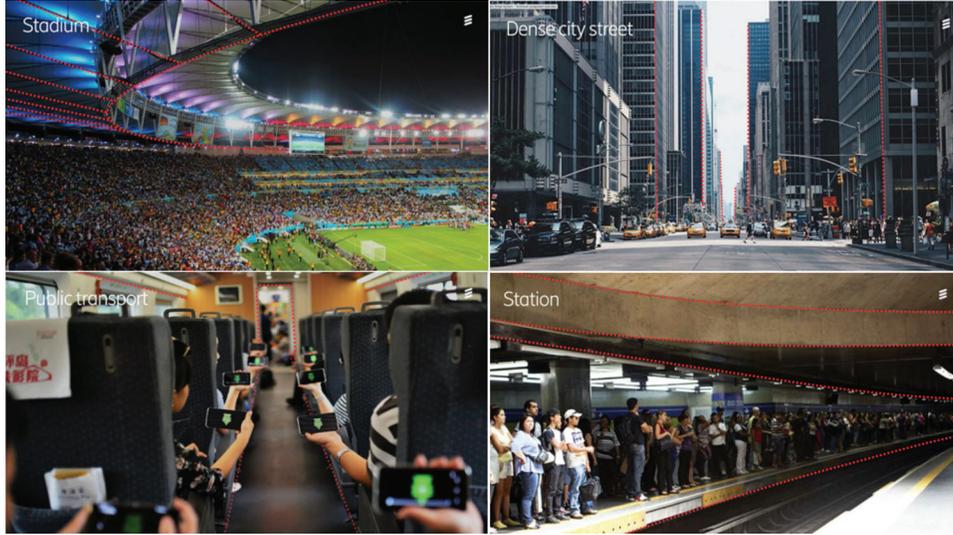
Fig. 1 shows ‘radio stripe’ network model, which consists of L APs (all APs are connected sequentially as shown in Fig. 1), with CPU being connected to the last AP (Lth AP). The fronthaul connections are given by AP1->AP>.....->APL->CPU. K single antenna User Equipments (UEs) are randomly distributed. Each AP consists of N numbers of antenna. As shown in Fig. 1, radio stripe is easy to deploy since it is light in weight and extremely flexible to deploy, which makes it possible to be deployed in any place like office, public place, stadium, etc., as also depicted in Fig. 2.

The channel gain between  $k^{\text{th}}$  UE and  $n^{\text{th}}$  antenna of  $l^{\text{th}}$  AP is given by [21],

$$g_{lnk} = L_{lnk}^{-\alpha} F_{lnk}^2 h_{lnk} \quad (1)$$



**Figure 1:** Radio stripe network model



**Figure 2:** Different scenarios of radio stripe deployment

$L_{lnk} = ||d_{lnk}||$  is the Euclidean distance between  $k^{\text{th}}$  UE and  $n^{\text{th}}$  antenna of  $l^{\text{th}}$  AP,  $b$  is the path-loss exponent.  $F_{lnk} = 10^{\frac{\sigma_{lnk} z_{lnk}}{10}}$  is log-normal shadowing correlation factor, with a standard deviation of  $\sigma_{lnk}$  and  $z_{lnk} = \sqrt{\delta}a_{lnk} + \sqrt{1-\delta}b_{lnk}$ , where  $a_{lnk}$  and  $b_{lnk} \sim N(0, 1)$  are i.i.d random variables (rvs) which represents shadow fading effect at UE  $k$  and  $n^{\text{th}}$  antenna in  $l^{\text{th}}$  AP and  $\delta$  is the transmitter-receiver shadow fading correlation coefficient.  $h_{lnk}$  is the small-scale channel fading between  $k^{\text{th}}$  UE and  $n^{\text{th}}$  antenna of  $l^{\text{th}}$  AP and follows independent but not identically distributed (i.n.d) Nakagami- $M_{lnk}$  distributions with shape and spreading parameters  $M_{lnk}$  and  $\Omega_{lnk}$ . Since  $h_{lnk}$  has Nakagami distribution hence  $|h_{lnk}|^2$  has gamma distribution, i.e.  $|h_{lnk}|^2 \sim G(\alpha_{lnk}, \beta_{lnk})$ , where  $\alpha_{lnk} = M_{lnk}$  is shape parameter and  $\beta_{lnk} = \frac{\Omega_{lnk}}{M_{lnk}}$  is scale parameters respectively.

It is assumed that  $L_{lnk}$  and  $F_{lnk}$  are known. Therefore, we have  $|g_{lnk}|^2 \sim G(\alpha_{lnk}, \beta'_{lnk})$  where

$$\beta'_{lnk} = \frac{\beta_{lnk}}{L_{lnk}^{-2b} F_{lnk}} = \frac{\beta_{lnk} L_{lnk}^{2b}}{F_{lnk}} = \frac{\Omega_{lnk} L_{lnk}^{2b}}{M_{lnk} F_{lnk}}.$$

The signal received at  $l^{\text{th}}$  AP is given by,

$$y_l = g_{lnk} S + N_l \quad (2)$$

where  $S$  is transmitted symbol vector and consists of transmitted symbol from  $K$  UEs which is  $S = [s_1, s_2, \dots, s_K]$ .  $N \in \mathbb{R}^{N \times 1}$  is the noise vector at the  $l^{\text{th}}$  AP whose entries have distribution with zero mean and variance of  $\sigma^2$ , with  $\sigma^2$  being noise power.

If number of antennas in each AP is 1 ( $N=1$ ), Eq. (1) can be modified as

$$g_{lk} = L_{lk}^{-b} F_{lk}^{\frac{1}{2}} h_{lk} \quad (3)$$

As suggested in literatures [8,13–15], sequential processing is done in radio stripe network. The effective channel estimate of  $k^{\text{th}}$  UE channel at the  $L^{\text{th}}$  AP [8] is given below

$$\hat{g}_k = [\hat{g}_{1k}, \hat{g}_{2k}, \hat{g}_{3k}, \dots, \hat{g}_{Lk}] \quad (4)$$

### 3 Deep Neural Network-Based Parallel Decoding in Radio Stripe

In this section, we discuss about, Iterative SIC which is the background for our works, the motivation behind applying deep learning in radio stripe, proposed parallel processing radio stripe network model, then we discuss about Soft Estimate Network (SEN), which forms the core of the proposed parallel decoding framework. Lastly, we discuss about the proposed algorithm.

#### 3.1 Iterative SIC

In this subsection, we briefly discuss about iterative Soft Iterative Cancellation (SIC) [22]. Iterative Soft Interference Cancellation (SIC) is a multi-user detection technique that combines parallel (multi-stage) interference cancellations with soft decoding. In a general sense, iterative SIC is an iteration-based detection scheme wherein each iteration, two operations are carried out in parallel for each user, namely Interference cancellation and soft decoding. Considering  $K^{\text{th}}$  user and  $c^{\text{th}}$  iteration, during the interference cancellation stage, the expected values and variances of  $\{s_u\}_{u \neq k}$  (interfering symbols) are computed based on  $\{\hat{P}_u^{(c-1)}\}_{u \neq k}$ .  $P_k^{(c)}$  is the estimated conditional distribution of  $k^{\text{th}}$  user symbol  $s_k$ . Entries of  $P_k^{(c)}$  are the estimate of the distribution of  $s_k$  for each possible symbol in the constellation.  $\{\hat{P}_u^{(c-1)}\}_{u \neq k}$  is the estimated conditional distribution of interfering symbols obtained in the previous iteration. The expected value of interfering user in  $c^{\text{th}}$  iteration is computed as

$$e_u^{(c-1)} = \sum_{r_w \in S} r_w (\hat{P}_u^{(c-1)})_w \quad (5)$$

where  $\{r_w\}_w = 1$  are the indexed elements of the constellation set  $CS$ . The variance of interfering symbols is given by,

$$V_u^{(c-1)} = \sum_{r_w \in CS} (r_w - e_u^{(c-1)})^2 (\hat{P}_u^{(c-1)})_w \quad (6)$$

The contribution of the interfering symbols from the received signal  $y$  is canceled by replacing it with mean  $\{e_u^{(c-1)}\}$  and subtracting the resultant term. Considering  $h_u$  the  $u^{\text{th}}$  column of  $H$  the elements of  $H$  with respect to user  $u$ . where  $H$  is channel matrix of all  $K$  user where each column represents one user channel vector. Interference canceled output of the channel is given by,

$$G_k^{(c)} = y - \sum_{u \neq k} h_u e_u^{(c-1)} \quad (7)$$

$$= h_k s_k + \sum_{u \neq k} h_u (s_u - e_u^{(c-1)}) + N \quad (8)$$

The second step carried out in parallel for each user is soft decoding. The bijective transformation of  $y$  is  $G_k^{(c)}$ , such that each possible value of  $y$  there is one unique value in  $G_k^{(c)}$ . Therefore,  $P_{s_k|y}(r_w|y) = P_{s_k|G_k^{(c)}}(r_w|G_k^{(c)})$  for each  $r_w \in \text{CS}$ . Therefore, the conditional distribution of  $s_k$  given  $y$  is approximated from the conditional distribution of  $G_k^{(c)}$  given  $s_k$  using Bayes theorem. Each symbol is equiprobable. Conditional distribution is computed as,

$$\left(\hat{P}_k^{(c)}\right)_w = \frac{P_{G_k^{(c)}|s_k}(G_k^{(c)}|r_w)}{\sum_{r'_w \in S} P_{G_k^{(c)}|s_k}(G_k^{(c)}|r'_w)} \quad (9)$$

Symbols are decoded after the final iteration such that symbol which maximizes estimated conditional distribution is decoded as the transmitted symbol for each user.

$$\hat{s}_k = \arg \max \left(\hat{P}_k^{(c)}\right)_w \quad (10)$$

where  $w \in \{1, 2, \dots, W\}$

The algorithm in steps is given in our previous works [18].

### 3.2 Motivation Behind Applying Deep Learning in Radio Stripe

Iterative SIC is leveraged to perform data-driven detection “DeepSIC” in Multi-User MIMO (MU-MIMO) [16], motivated by this work, in our last work, we proposed DNN-based distributed sequential uplink processing (DBDSUP) in cell-free massive MIMO based on radio stripe by exploiting data-driven iterative SIC. However, it considers sequential processing leading to higher latency, which can be reduced given the processing is carried out in a parallel fashion. Thus, we propose DNN based Parallel Decoding in Radio Stripe (DNNBPDERS), which carries out parallel processing in radio stripe networks.

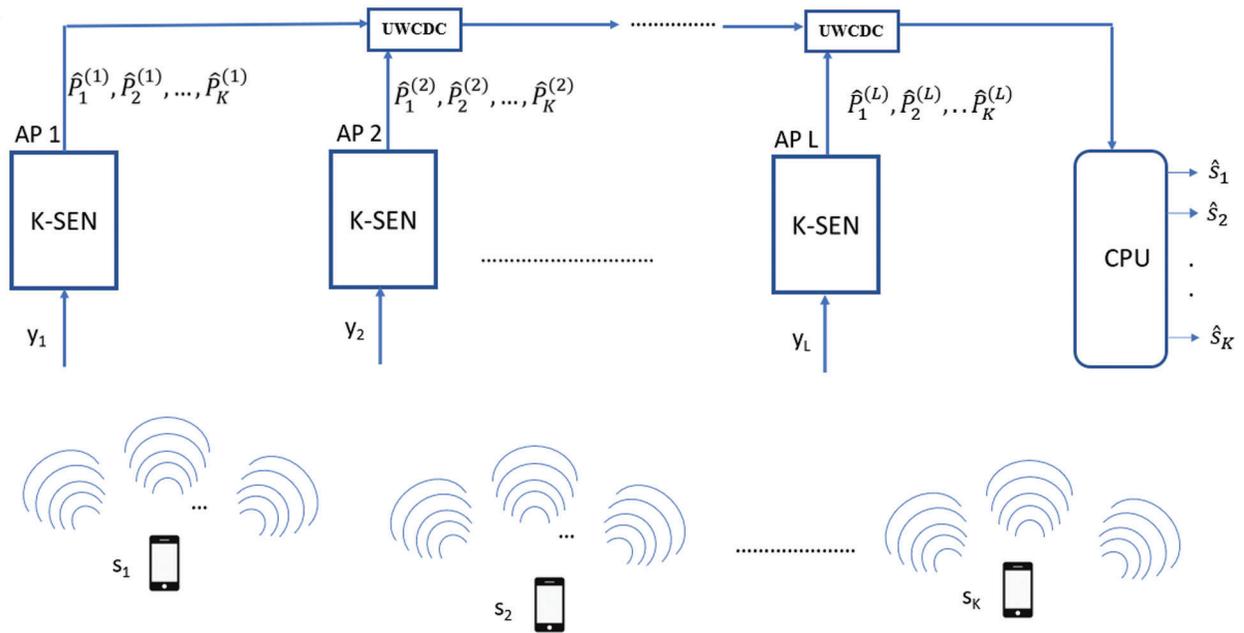
### 3.3 Proposed Parallel Processing Network Model

Fig. 3 shows a simplified proposed parallel processing network model.

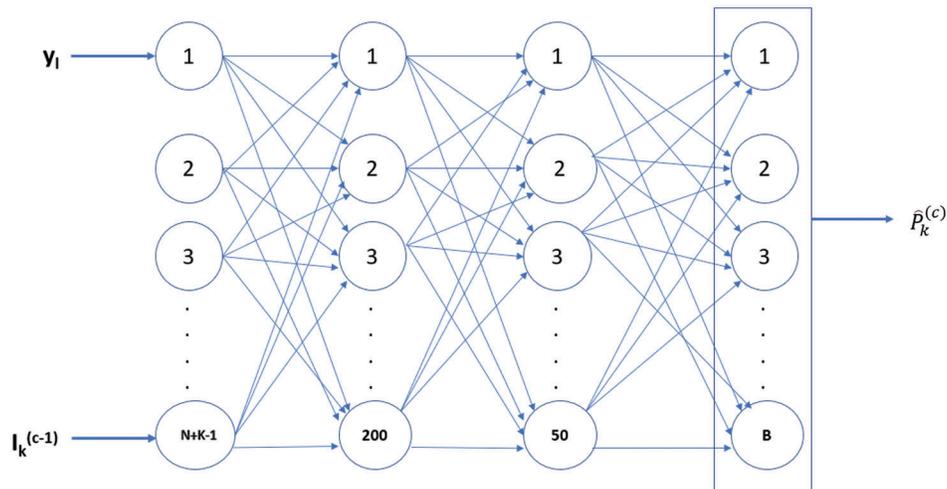
As shown in Fig. 3, each AP consists of  $K$ -parallel Soft Estimate Network (SEN) (which is discussed in detail in Section 3.4). As opposed to sequential processing, here each AP process the received signal simultaneously. The signal received at  $AP_1, AP_2, \dots, AP_L$  is given by  $y_1, y_2, \dots, y_L$ .

### 3.4 Soft Estimate Network

As shown in Fig. 4, Soft Estimate Network (SEN) is a classification-based Neural Network. As explained above, iterative SIC carries out two operations in parallel for each user, namely, interference cancellation and soft decoding. In particular, for  $c^{\text{th}}$  iteration and  $k^{\text{th}}$  user soft decoding estimate  $\hat{P}_K^{(c)}$  is a classification task since entries of  $\hat{P}_K^{(c)}$  is an estimation of the distribution of  $k^{\text{th}}$  user symbols for each symbol in the constellation with the dimension of  $\hat{P}_K^{(c)}$  being  $R^B$ .



**Figure 3:** Simplified proposed parallel decoding mechanism



**Figure 4:** Soft estimate network (SEN)

SEN replaces these complex computations for each iteration. Moreover, interference cancellation and soft decoding are dependent on the underlying channel model. However, SEN does not require any knowledge of the channel model since it is based on dedicated neural network which is trained for different channel scenarios.

Moreover, the inputs to SEN are  $y_1$  and  $I_k^{(c-1)}$ . Here  $y_1$  and  $I_k^{(c-1)}$  are the received signal at the 1<sup>th</sup> AP and interference coefficient of the  $k^{\text{th}}$  user during  $c^{\text{th}}$  iteration, respectively.

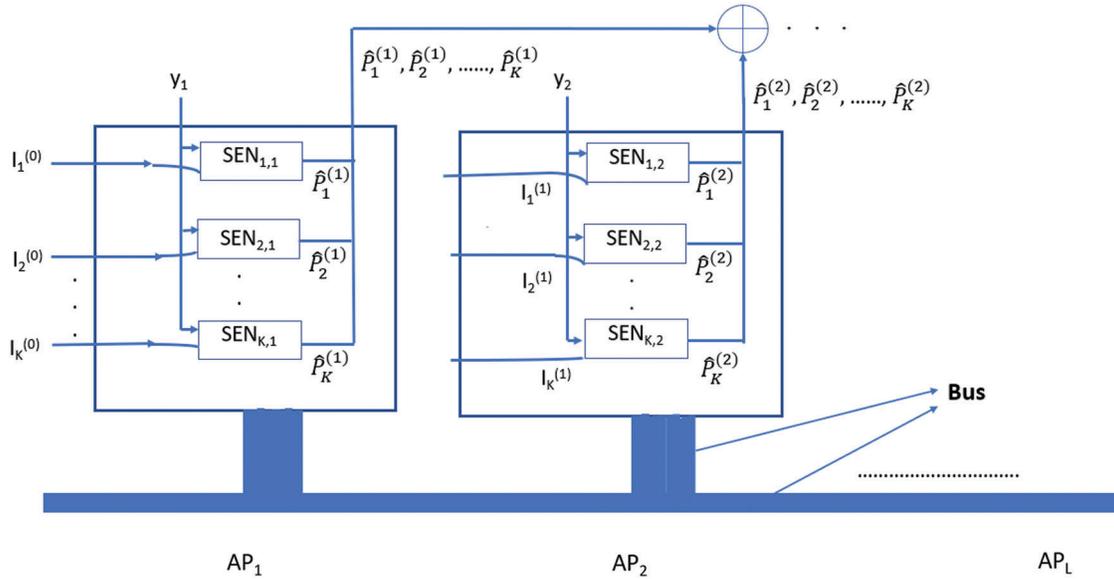
The last layer of SEN is the Softmax layer; it consists of neurons equal to the numbers of symbols in the constellation. For example, if Binary Phase shift Keying (BPSK) is employed, then the Softmax layer consists of 2 (B) neurons.

### 3.5 Deep Neural Network-Based Parallel Decoding in Radio Stripe (DNNBPDRS)

In this subsection, we discuss about the proposed parallel decoding scheme, DNN based Parallel Decoding in Radio Stripe (DNNBPDRS). Fig. 5 shows the K parallel SENS in each AP. As shown in Fig. 5, AP<sub>1</sub> receives y<sub>1</sub>, which forms one of two inputs to SEN<sub>1,1</sub>, SEN<sub>2,1</sub>, ..... SEN<sub>K,1</sub> (first subscript denotes the k<sup>th</sup> user and the second subscript denotes the l<sup>th</sup> AP) while the second input to SENS are interference coefficients I<sub>1</sub><sup>(0)</sup>, I<sub>2</sub><sup>(0)</sup>, ..... I<sub>K</sub><sup>(0)</sup> and takes the value of  $\frac{1}{B}$  [16], which are given by

$$I_1^{(0)} = [\hat{P}_2^{(0)}, \hat{P}_3^{(0)}, \dots, \hat{P}_K^{(0)}] \quad (11)$$

$$I_2^{(0)} = [\hat{P}_1^{(0)}, \hat{P}_3^{(0)}, \dots, \hat{P}_K^{(0)}] \quad (12)$$



**Figure 5:** Processing in proposed radio stripe architecture

Considering SEN<sub>1,1</sub>, the SEN belonging to UE 1 at AP<sub>1</sub>, it receives y<sub>1</sub> and I<sub>1</sub><sup>(0)</sup> to produce  $\hat{P}_1^{(1)}$ , the conditional distribution of UE 1. Similarly, SEN<sub>2,1</sub> and SEN<sub>3,1</sub>, produces  $\hat{P}_2^{(0)}$  and  $\hat{P}_3^{(0)}$  respectively as shown in Fig. 5. Likewise, SEN<sub>1,2</sub> in AP<sub>2</sub> accepts inputs y<sub>2</sub> and I<sub>1</sub><sup>(1)</sup> to produce  $\hat{P}_1^{(2)}$ , where I<sub>1</sub><sup>(1)</sup> is given by

$$I_1^{(1)} = [\hat{P}_2^{(1)}, \hat{P}_3^{(1)}, \dots, \hat{P}_K^{(1)}] \quad (13)$$

In a similar way, all SENS in all L APs compute respective conditional distribution. To sum the conditional distribution of L APs, L-1 numbers of User-Wise Conditional Distribution Combiner (UWCDC) is required.

Finally, the UWCDC performs the following operations,

$$\hat{S}_{1l} = \hat{P}_1^{(1)} + \hat{P}_1^{(2)} + \dots + \hat{P}_1^{(L)} \quad (14)$$

$$\hat{S}_{2t} = \hat{P}_2^{(1)} + \hat{P}_2^{(2)} + \dots + \hat{P}_2^{(L)} \quad (15)$$

and,

$$\hat{S}_{kt} = \hat{P}_k^{(1)} + \hat{P}_k^{(2)} + \dots + \hat{P}_k^{(L)} \quad (16)$$

where,  $\hat{S}_{1t}$ ,  $\hat{S}_{2t}$  and  $\hat{S}_{kt}$  are the total added conditional distribution of 1<sup>st</sup>, 2<sup>nd</sup> and k<sup>th</sup> user and has the dimension of  $1 \times B$  ( $B$  is the size of constellation). For instance, if BPSK is considered then  $B = 2$ .

For each user the CPU performs the operation given below

$$\hat{S}_1 = \max_{cs \in \{+1, -1\}} (\hat{S}_{1t}) \quad (17)$$

$$\hat{S}_2 = \max_{cs \in \{+1, -1\}} (\hat{S}_{2t}) \quad (18)$$

and,

$$\hat{S}_k = \max_{cs \in \{+1, -1\}} (\hat{S}_{kt}) \quad (19)$$

Above max operations, choose the symbol from the constellation which has the maximum value. For example, if BPSK constellation is considered and  $\hat{S}_{1t} = [0.15, 0.85]$ , then  $\hat{S}_1 = 0.85$  and the decoded symbol is +1.

The total numbers of SENs present in radio stripe is given by  $N_{SEN}$ .

$$N_{SEN} = KL \quad (20)$$

The proposed DNN based Parallel Decoding in Radio Stripe (DNNBPDRS) is given in steps below.

---

**Algorithm 1:** DNN based Parallel Decoding in Radio Stripe (DNNBPDRS)

---

1. Iteration-1 (Carried out in  $AP_1$ ): Received signal at  $AP_1$  is  $y_1$ .  $y_1$  is concatenated with an initial guess of conditional distribution  $\hat{P}_k^{(0)}$  and fed to  $SEN_{1,1}$ ,  $SEN_{2,1}$ , .....,  $SEN_{K,1}$ , which produces  $\hat{P}_1^{(1)}$ ,  $\hat{P}_2^{(1)}$ , .....,  $\hat{P}_K^{(1)}$  respectively and passed to UWDCD 1.
  2. Iteration-2 (Carried out in  $AP_2$ ): Received signal at  $AP_2$  is  $y_2$ .  $y_2$  is concatenated with an initial guess of conditional distribution  $\hat{P}_k^{(0)}$  and fed to  $SEN_{1,2}$ ,  $SEN_{2,2}$ , .....,  $SEN_{K,2}$ , which produces  $\hat{P}_1^{(2)}$ ,  $\hat{P}_2^{(2)}$ , .....,  $\hat{P}_K^{(2)}$  respectively and is passed to UWDCD 1.
  3. Iteration-L-1 (Carried out in  $AP_{L-1}$ ): Received signal at  $AP_{L-1}$  is  $y_{L-1}$ .  $y_{L-1}$  is concatenated with an initial guess of conditional distribution  $\hat{P}_k^{(0)}$  and fed to  $SEN_{1,L-1}$ ,  $SEN_{2,L-1}$ , .....,  $SEN_{K,L-1}$ , which produces  $\hat{P}_1^{(L-1)}$ ,  $\hat{P}_2^{(L-1)}$ , .....,  $\hat{P}_K^{(L-1)}$  respectively and is passed to UWDCD L-2.
  4. Iteration-L (Carried out in  $AP_L$ ): Received signal at  $AP_L$  is  $y_L$ .  $y_L$  is concatenated with an initial guess of conditional distribution  $\hat{P}_k^{(0)}$  and fed to  $SEN_{1,L}$ ,  $SEN_{2,L}$ , .....,  $SEN_{K,L}$ , which produces  $\hat{P}_1^{(L)}$ ,  $\hat{P}_2^{(L)}$ , .....,  $\hat{P}_K^{(L)}$  respectively and is passed to UWDCD L-1.
  5. Operations are performed at all the L-1 UWDCD's as given in Eqs. (14)–(16).
  6. After all L iterations at all L APs. CPU performs the operations as shown in Eqs. (17)–(19).
-

### 3.6 Complexity Analysis

This subsection discusses and compares the complexity of proposed DNNBPDRS, DNNBDS [18] and iterative SIC.

#### 3.6.1 Complexity Analysis of DNNBPDRS

Considering the input layer, SEN has four layers therefore, it needs three matrices to represent weight between these layers. Let the numbers of nodes in 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> layers be p, q, r, and s, respectively. The weight matrices are denoted as W<sub>qp</sub>, W<sub>rq</sub>, W<sub>sr</sub>. For example, W<sub>qp</sub> contains weights going from layer p to layer q. Considering there are t examples of data that has to be fed to SEN. Forward propagation from the 1st layer to 2nd layer, we have

$$X_{qt} = W_{qp} * I_{pt} \quad (21)$$

where  $X_{qt}$  is the computed calculation at the second layer,  $I_{pt}$  is the input fed to the first layer of SEN having p nodes. The time complexity is given by

$$O(q * p * t) \quad (22)$$

where O denotes the numbers of the operation carried out at each fully connected layer. After the forward pass, the activation function is applied at the 2nd layer, and since it is an element-wise operation hence time complexity is

$$O(q * t) \quad (23)$$

So, the total time complexity involved in going from the 1st layer of SEN to 2<sup>nd</sup> layer is

$$O(q * p * t + q * t) \quad (24)$$

$$O(q * p * (t + 1))$$

$$O(q * p * t) \quad (25)$$

Similarly, time complexity in going from layer 2 to layer

$$O(r * q * t) \quad (26)$$

Time complexity involved in going from layer 3 to layer 4

$$O(s * r * t) \quad (27)$$

Therefore, the total time complexity for feed-forward propagation is

$$O(q * p * t + r * q * t + s * r * t) \quad (28)$$

$$O(t * (pq + qr + rs)) \quad (29)$$

During the backpropagation, the time complexity is the same as shown in Eq. (29). Hence total time complexity involved in training is as shown in Eq. (29). The proposed DNNBPDRS algorithm consists of  $N_{SEN}$  SENs, however, the processing is carried out in parallel fashion. Moreover, SEN is trained for several iterations (epoch) n. So total time complexity for training SENs is

$$O(n * t * (pq + qr + rs)) \quad (30)$$

However, during the testing and deployment of DNNBPDRS, each input to SEN is passed only once with final detected symbols at the CPU. Therefore, the computation complexity of deployed DNNBPDRS is given by

$$O(pq + qr + rs) \quad (31)$$

### 3.6.2 Complexity Analysis of DNNBDS

Our previous work DNN-based distributed sequential uplink processing (DBDSUP) [18], also employs SENSs, with the only difference being sequential processing.

Therefore, the computational complexity of DNNBDS can be compared with the computational complexity of DNNBPDRS and remains the same till Eq. (31). However, DNNBDS carries out sequential processing and the radio stripe consists of  $N_{SEN}$  SENSs. Therefore, the computation complexity of deployed DNNBDS is given by

$$O(N_{SEN}*(pq + qr + rs)) \quad (32)$$

### 3.6.3 Complexity Analysis of Iterative SIC

In this subsection, we discuss the O complexity of Iterative SIC in order to compare the same with DNNBPDRS and DNNBDS. Iterative SIC mainly consists of 4 stages, expected value computation, variance computation, interference cancellation and soft decoding. The complexity involved in each step is given below.

For expected value, from Eq.(5), the complexity involved to compute the expected value of interfering users for one particular user is given as

$$O(2(K - 1) + B (K - 1)) \quad (33)$$

For K UEs,

$$O(K (2(K - 1) + B (K - 1))) \quad (34)$$

$$O(2K^2 - 2K + BK^2 - BK) \quad (35)$$

Computation of variance involves similar steps hence the complexity involved in computing variance is also,

$$O(2K^2 - 2K + BK^2 - BK) \quad (36)$$

Thirdly, for interference cancellation from Eq. (8), the complexity involved in computing interference cancelled output for one particular user is given by

$$O((K - 2) + (K - 1)) \quad (37)$$

For K UEs,

$$O(2K^2 - 3K) \quad (38)$$

Lastly, Computational complexity involved in computing soft decoding for one user is

$$O(B)$$

For K UEs

$$O(KB) \quad (39)$$

Time complexity involved in four stages at one AP is hence given by

$$O(6K^2 - 7K + 2BK^2 - BK) \quad (40)$$

Since these computations are carried out for C iterations (as given in iterative SIC algorithm) at each AP and carried out at L AP, hence complexity is given as

$$O(LC(6K^2 - 7K + 2BK^2 - BK)) \quad (41)$$

Moreover, iterative SIC requires full CSI, let the time complexity involved in acquiring full CSI for one user be given as CA, therefore time complexity for acquiring CSI for K users at L AP is given by,

$$O(LK(CA)) \quad (42)$$

Hence, total time complexity involved in iterative SIC is

$$O(LC(6K^2 - 7K + 2BK^2 - BK) + LK(CA)) \quad (43)$$

Comparing Eqs. (31), (32) and (43), since Eq. (43) is polynomial, therefore it is very evident that complexity of iterative SIC detector is higher than proposed DNNBPDRS and DNNBDS. Moreover, as seen from Eqs. (31) and (32), DNNBDS has  $N_{SEN}$  times more time complexity than DNNBPDRS.

#### 4 Channel Link-Based AP Selection Algorithm

The SINR in the uplink of user k at lth AP given by,

$$\gamma_{ku} = \frac{\sum_{n=1}^N P_k |\hat{g}_k|^2}{\sum_{n=1}^N [\sum_{u=1, u \neq k}^K P_u |\hat{g}_u|^2] + \sigma_{nu}^2} \quad (44)$$

where,  $P_k$  is transmitted power of user k in uplink and  $\sigma_{nu}^2$  is effective noise power. Hence, Up Link (UL) spectral efficiency is given by,

$$SE_u = \frac{\tau_u}{\tau_c} \sum_{k=1}^K \log_2(1 + \gamma_{ku}) \quad (45)$$

where  $\tau_u$  is UL transmission phase and  $\tau_c$  is coherence interval. Channel link metric-Channel link is basically the numerator component of SINR, which determines the channel gain between UE and AP.

Channel Link between kth user and lth AP is given below,

$$C_{kl} = |g'_{lk}|^2 \quad (46)$$

For each UE k an array of channel link consisting of L elements is, in particular

$$C_k = [CL_{k1}, CL_{k2}, \dots, CL_{kL}] \quad (47)$$

Let the numbers of APs to which UE k is connected in the uplink be given by  $L_a$ . UE k accesses these  $L_a$  AP sequentially.

We propose channel link-based algorithm to choose the most suitable APs. In the step 1 and 2 channel links for each user is calculated and sorted in descending order. Based on the chosen  $L_a$  in step 3, the APs corresponding to first  $L_a$  are chosen for user k in step 4. To ensure fair usage for all users, all four steps are carried in parallel fashion for each user.

The proposed AP selection algorithm is summarized in algorithm 3 below.

---

**Algorithm 2:** Channel Link-Based AP Selection (CLBAPS) algorithm

---

1. Calculate  $C_{kl}$  and hence  $C_k$  as in (1) and (2)  $\forall k = 1, 2, \dots, K$ .
  2. Sort the array  $C_k$  in descending order  $\forall k = 1, 2, \dots, K$ .
  3. Choose  $L_a$ .
  4. Given  $L_a$ , the APs corresponding to first  $L_a$  elements of  $C_k$  are selected for user k.
- 

## 5 Numerical Results

In this section, we discuss on simulation parameters, the performance of the proposed decoding scheme DNNBPDRS, complexity analysis of DNNBPDRS and finally, we discuss the performance of the proposed channel selection algorithm CLBAPS.

### 5.1 Simulation Setup

For large-scale fading, we assume that all users are randomly distributed over a stadium of circular shape with a radius 10 m which corresponds to the total coverage area of 314 m<sup>2</sup>. Radio stripe is installed across the circumference of the stadium at a height of 5 m from the ground.

### 5.2 Training of SEN

The symbols are randomly derived from BPSK Constellation Set (CS)  $\in \{0, 1\}$ , which undergoes operation as given by Eq. (2). For each particular value of  $y_1$  at all the APs, SENs across all APs are trained in a parallel manner. The SENs across all APs are trained on 20 K samples. At 1<sup>th</sup> AP, the received signal is  $y_1$ , based on the initial estimate  $\{\hat{P}_k^{(0)}\}_{k=1}^K$  all K SENs in each AP are trained in a parallel fashion for all 20 K samples.

For a given transmitted symbol vector S, all SENs across all APs are trained in a parallel fashion. Since the proposed DNNBPDRS is trained *via* supervised learning, the label (actual conditional distribution) corresponding to the transmitted symbol vector is available to compute loss.

Since, SEN<sub>1,1</sub> has a softmax layer as the output layer, the output at each neuron of the softmax layer is given as

$$\frac{\exp(x_i)}{\sum_{j=1}^B \exp(x_j)} \quad (48)$$

where  $(x_i)$  and  $(x_j)$  are elements of  $\hat{P}_1^{(1)}$ .

Let  $\varnothing$  represents weight and biases of SEN<sub>1,1</sub>. Since,  $P_1^{(1)}$  consists of B elements,  $P_1^{(1)} = [O_1, O_2, \dots, O_B]$ . The loss function at the output of SEN<sub>1,1</sub> is defined as

$$L(\varnothing) = - \sum_{i=1}^B \left( P_1^{(1)} \right)_i \log \left( \hat{P}_1^{(1)} \right)_i \quad (49)$$

where  $P_1^{(1)}$  and  $\hat{P}_1^{(1)}$  are the actual conditional distribution and estimated conditional distribution respectively. The parameter  $\varnothing$  are updated using Adam optimizer and is defined as

$$\varnothing_{f+1} = \varnothing_f - \frac{\eta m_f}{\sqrt{v_f + \epsilon}} \quad (50)$$

$\eta$  is Learning rate.  $\epsilon$  is a smoothing term which prevents divisions by 0.  $m_f$  and  $v_f$  are the estimate of the mean (first moment) and uncentered variance (second moment) of the gradients and is given as,

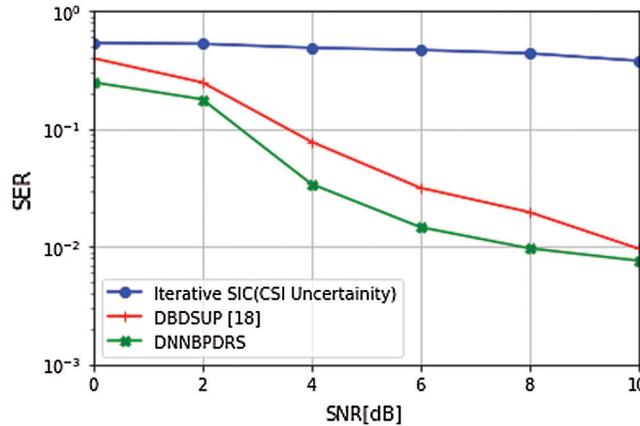
$$m_f = \beta_1 m_{f-1} + (1 - \beta_1) \nabla L(\varnothing_f) \quad (51)$$

$$v = \beta_2 v_{f-1} + (1 - \beta_2) \nabla [L(\varnothing_f)]^2 \quad (52)$$

$\beta_1$  and  $\beta_2$  are the decay rates of the moving average. Similarly, all SENs across all APs are trained in a sequential manner.

### 5.3 Analysis of DNNBPDRS

In this subsection, we analyze the performance of the proposed DNNBPDRS algorithm and compare its performance with Cell-Free massive MIMO ‘Level 4’ implementing iterative SIC [11] under CSI uncertainty and our proposed sequential processing in radio stripe [18]. It is quite evident from Fig. 6, the best performance is achieved by the proposed DNNBPDRS algorithm.



**Figure 6:** Symbol Error Rate (SER) performance of proposed framework

Considering the numerical difference in Symbol Error Rate (SER) performance of the three frameworks are compared, the percentage difference of DNNBPDRS-iterative SIC and DNNBPDRS-DBDSUP are 99.05% and 32.72%, respectively for 2dB Signal-to-Noise Ratio (SNR).

For 8dB SNR, the percentage difference of DNNBPDRS-iterative SIC and DNNBPDRS-DBDSUP are 191.35% and 132.53%, respectively. For all SNRs the difference between DNNBPDRS-iterative SIC is much higher than DNNBPDRS-DBDSUP.

The better performance of DNNBPDRS then DBDSUP is attributed to the fact that in DNNBPDRS, each AP is trained for five iterations, while in DBDSUP each AP is trained for a single iteration. Moreover, in DBDSUP there are chances that errors in conditional distribution estimation may propagate from one AP to next AP, which is eliminated completely in DNNBPDRS.

Since, we consider radio stripe to be deployed across the walls of room/stadium as indicated in Fig. 2, it becomes important to determine the decoding performance of the proposed DNNBPDRS framework for

different numbers of AP  $L$ , given the numbers of UEs present in the radio stripe network. Therefore, we analyze the performance of DNNBPDRS with different numbers of APs  $L$  for a given number of UEs  $K$  in the network.

Simulation is carried out for three different numbers of APs  $L$  using the simulation parameters given in [Tab. 1](#). In particular, simulation is carried out for  $L = 5$ ,  $L = 10$  and  $L = 20$ . [Tab. 2](#) shows the percentage of difference in obtained SER.

**Table 1:** System parameters

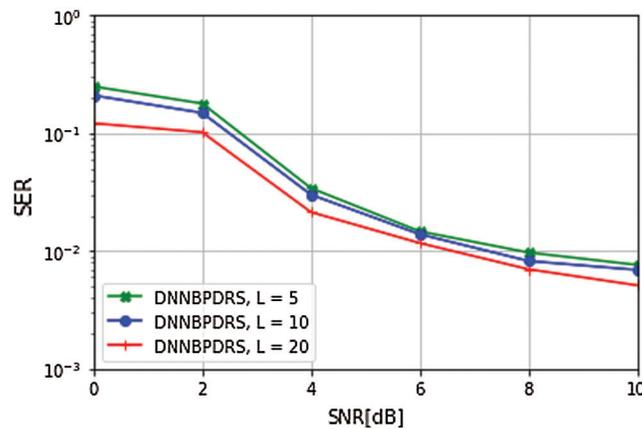
Parameters	Value
Path-loss exponent ( $\alpha$ )	2
Nakagami fading parameter, ( $M, \Omega$ )	(1, 1)
Number of UE ( $K$ )	6
Number of antennas in each AP ( $N$ )	1
Height of AP from ground $h$	5 m

**Table 2:** Comparison table comparing percent difference in SER from [Fig. 7](#)

Variations of numbers of APs $L$ in radio stripe	Percent difference	
	SNR 2dB(%)	SNR 8dB(%)
$L = 5, L = 10$	18.30	16.48
$L = 5, L = 20$	54.84	32.39
$L = 10, L = 20$	37.48	16.13

As the number of AP  $L$  increases, the performance of DNNBPDRS improves. [Tab. 2](#) shows the percentage difference in SER achieved for different combinations. It can be clearly noted for the second combination  $L = 5, L = 20$  (the difference being 4 times), the percentage difference is highest.

Moreover, it can be clearly observed from [Fig. 7](#), that the performance of the proposed DNNBPDRS improves as  $L > K$  inequality increases.



**Figure 7:** SER performance of DNNBPDRS for different numbers of APs  $L$

#### 5.4 Analysis of Channel Link-Based AP Selection

The above four figures (Figs. 8–11) compare the CDF of proposed Channel-link based AP selection (CLBAPS) with CDF of Random AP selection (RAPS) (here AP are selected randomly). Analysis is carried out for different numbers of AP  $L$  in the radio stripe network, keeping  $L_a$  constant. It can be clearly observed the gap between the proposed scheme and random selection increases as the numbers of AP in the network increases. In particular, from Fig. 8, the proposed CLBAPS algorithm gains 0.960 bit/s/Hz as compared to the RAPS with 50% probability. While with the proposed algorithm the SE of the UEs with 50% probability have 5.261 bit/s/Hz gain over random selection, when  $L = 200$ . The reason for these results is quite obvious, as the numbers of APs in the network increases there are more chances of UE being close to AP with excellent channel link. However, there is insignificant impact of increase in numbers of AP  $L$  for random AP selection as it gains 0.0383 bit/sec/Hz when increasing the APs in the network from 10 to 200.

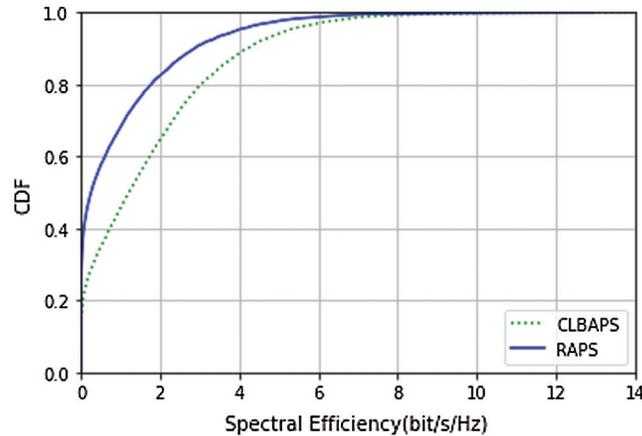


Figure 8: CDF of spectral efficiency with  $L = 10$ ,  $L_a = 4$

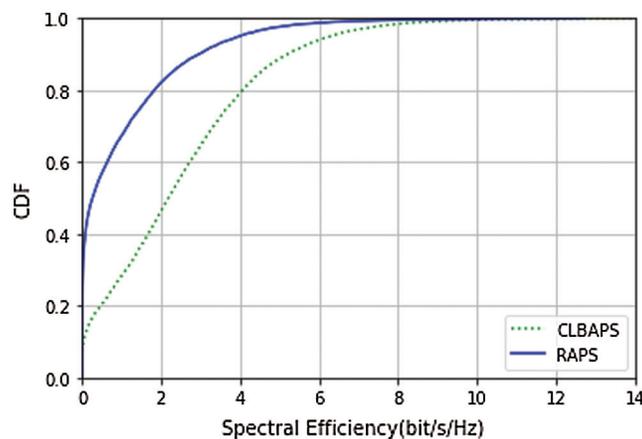
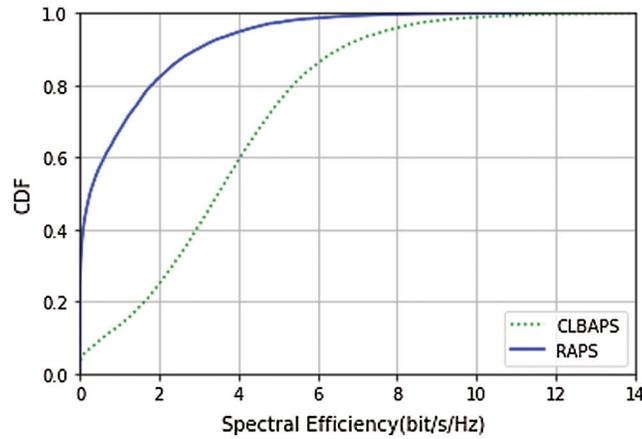
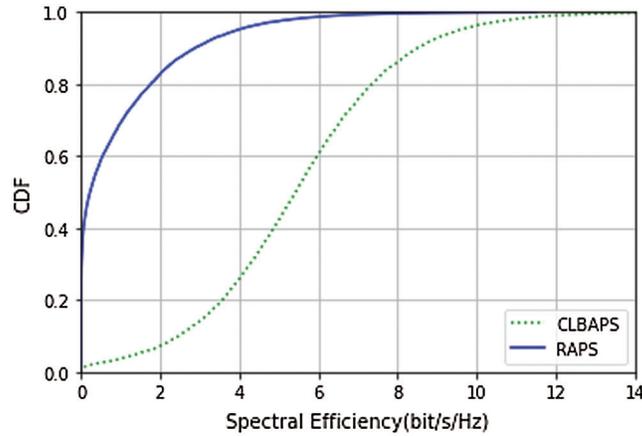


Figure 9: CDF of spectral efficiency with  $L = 20$ ,  $L_a = 4$



**Figure 10:** CDF of spectral efficiency with  $L = 50$ ,  $L_a = 4$



**Figure 11:** CDF of spectral efficiency with  $L = 200$ ,  $L_a = 4$

## 6 Conclusion

Cell-Free massive MIMO is slated to be beyond 5G (B5G) technology; however, its practical adoption issues mar its implementation. Radio stripe, an architecture of cell-free massive MIMO is suitable for practical implementation and is envisioned to be 6G technology by industry experts. In this paper, we propose, deep learning-based parallel decoding scheme in cell-free massive MIMO based on radio stripe namely DNN based Parallel Decoding in Radio Stripe (DNNBPDRS), which gives two benefits. Firstly, DNN processing reduces the computational complexity and secondly, parallel processing reduces the latency, making it very attractive for 6<sup>th</sup> generation wireless technology. Moreover, to solve the issue of AP selection in radio stripe network. We propose Channel link-based AP selection (CLBAPS). In the future, more research can be carried out to optimize the parallel processing in radio stripe, which would further reduce the computational complexity and allow UEs to choose the best APs for uplink processing by considering different simulation setups and different parameters such as distance and mobility.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the study.

## References

- [1] S. Parkvall, E. Dahlman, A. Furuskar, M. Frenne *et al.*, “The new 5G radio access technology,” *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.
- [2] E. Björnson, J. Hoydis and L. Sanguinetti, “Massive MIMO networks: Spectral, energy, and hardware efficiency,” *Foundations and Trends in Signal Processing*, vol. 11, no. 3–4, pp. 154–655, 2017.
- [3] E. G. Larsson, O. Edfors, F. Tufvesson and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.
- [4] H. Q. Ngo, E. G. Larsson and T. L. Marzetta, “Energy and spectral efficiency of very large multiuser MIMO systems,” *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.
- [5] H. Yang and T. L. Marzetta, “Total energy efficiency of cellular large scale antenna system multiple access mobile networks,” in *IEEE Online Conf.on Green Communications*, Piscataway, NJ, USA, pp. 27–32, 2013.
- [6] P. Vijayakumar and S. Malarvizhi, “Hardware impairment detection and prewhitening on MIMO precoder for spectrum sharing,” *Wireless Personal Communications*, vol. 96, no. 1, pp. 1557–1576, 2017.
- [7] A. Lozano, R. W. Heath and J. G. Andrews, “Fundamental limits of cooperation,” *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5213–5226, 2013.
- [8] Z. H. Shaik, E. Björnson and E. G. Larsson, “MMSE-optimal sequential processing for cell-free massive MIMO with radio stripes,” arXiv preprint arXiv: 2012. 13928, 2020.
- [9] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, “Cell-free massive MIMO vs. small cells,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.
- [10] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai *et al.*, “Cell-free massive MIMO: A new next-generation paradigm,” *IEEE Access*, vol. 7, pp. 99878–99888, 2019.
- [11] E. Björnson and L. Sanguinetti, “Making cell-free massive MIMO competitive with MMSE processing and centralized implementation,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2019.
- [12] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger and E. G. Larsson, “Ubiquitous cell-free massive MIMO communications,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–13, 2019.
- [13] Z. H. Shaik, E. Björnson and E. G. Larsson, “Cell-free massive MIMO with radio stripes and sequential uplink processing,” in *IEEE Int. Conf. on Communications Workshops, IEEE*, Dublin, Ireland, pp. 1–6, 2020.
- [14] Y. Ma, Z. Yuan, G. Yu and Y. Chen, “Efficient parallel schemes for cell-free massive MIMO using radio stripes,” arXiv preprint arXiv: 2012. 11076, 2020.
- [15] Z. Yuan, Y. Ma and G. Yu, “LMMSE processing for cell-free massive MIMO with radio stripes and MRC fronthaul,” arXiv preprint arXiv: 2012. 13504, 2020.
- [16] N. Shlezinger, R. Fu and Y. C. Eldar, “DeepSIC: Deep soft interference cancellation for multiuser MIMO detection,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1349–1362, 2020.
- [17] P. Vijayakumar and S. Malarvizhi, “Wide band full duplex spectrum sensing with self-interference cancellation-an efficient SDR implementation,” *Mobile Networks and Applications*, vol. 22, no. 4, pp. 702–711, 2017.
- [18] A. K. Mishra and V. Ponnusamy, “DNN-based distributed sequential uplink processing in cell-free massive MIMO based on radio stripes,” *IET Networks*, vol. 10, no. 3, pp. 118–122, 2021.
- [19] H. T. Dao and S. Kim, “Effective channel gain-based access point selection in cell-free massive MIMO systems,” *IEEE Access*, vol. 8, pp. 108127–108132, 2020.
- [20] S. Biswas and P. Vijayakumar, “AP selection in cell-free massive MIMO system using machine learning algorithm,” in *2021 Sixth Int. Conf. on Wireless Communications, Signal Processing and Networking*, Chennai, India, pp. 158–161, 2021.
- [21] Y. Al-Eryani, M. Akrouf and E. Hossain, “Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1028–1042, 2020.
- [22] W. J. Choi, K. W. Cheong and J. M. Cioffi, “Iterative soft interference cancellation for multiple antenna systems,” *IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 304–309, Conference Record (Cat. No. 00TH8540), 2000.