Some New Perspectives on the Multi-user Detection in Uplink Grant-Free NOMA using Deep Neural Network

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

29-11-2021 / 16-12-2021

CITATION

Hasan, Shah Mahdi; Mahata, Kaushik; Hyder, Md Mashud (2021): Some New Perspectives on the Multi-user Detection in Uplink Grant-Free NOMA using Deep Neural Network. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.17091560.v2>

DOI

[10.36227/techrxiv.17091560.v2](https://doi.org/10.36227/techrxiv.17091560.v2)

Some New Perspectives on the Multi-user Detection in Uplink Grant-Free NOMA using Deep Neural Network

Shah Mahdi Hasan, Kaushik Mahata, Md Mashud Hyder

School of Engineering

The University of Newcastle

Corresponding author email: shahmahdi.hasan@uon.edu.au

Abstract—To support the explosive growth of the Internet of Things (IoT), Uplink (UL) grant-free Non-Orthogonal Multiple Access (NOMA) emerges as a promising technology. It has the potential of offering scalable and low-cost solutions for the resource-constrained Massive Machine Type Communication (mMTC) systems. In principle, the grant-free NOMA enables small signaling overhead and low access latency time by circumventing complicated grant-access based procedures which is commonly found in the legacy wireless networks. In a UL grant-free system, a complete Multi-User Detection (MUD) algorithm not only performs the Active User Detection (AUD) but also the Channel Estimation (CE) and the Data Detection (DD). By exploiting the naturally occurring sparse user activity in the mMTC systems, the MUD problem can be solved using a wide range of Compressive Sensing based algorithms (CS-MUD). However, some alternative routes have been explored in the literature as well. The utility of these algorithms, in general, revolve around some assumptions about the channel or the availability of perfect channel information at the Base Station (BS). How these assumptions are met in a practical circumstance is, however, an important concern. In this work we devise an end-to-end MUD using Deep Neural Network (DNN) where we relax these assumptions. We approximate an ensemble of trained DNN based MUD using Knowledge Distillation (KD) to enable fast AUD at the Base Station (BS). Furthermore, using the inter-resource correlation, we estimate the channels of the active users which is an ill-posed problem otherwise. We carry out elaborate numerical investigation to validate the efficacy of the proposed approach for the UL grant-free NOMA systems.

Index Terms—Massive machine-type communication (mMTC), grant-free, active user detection, non-orthogonal multiple access (NOMA), channel estimation, deep learning

I. INTRODUCTION

A. Background and Motivation

The next generation of wireless network is increasingly incorporating a wide range of distinct use cases. Among those, mMTC has been able to garner widespread attention due to its applications in the smart factories, smart metering, fleet management, and the Industry 4.0 [1], [2]. Contrary to the other service categories of the 5g wireless networks where maximizing peak rates (Enhanced Mobile Broadband (eMBB)) or lowering latency (Ultra-Reliable Low Latency Communications (URLLC)) are the primary design concerns, mMTC focuses on providing scalable connectivity for a large number of low-complexity, low power-backup, and low-rate devices [1], [3]. An mMTC system can be characterized by several

unique properties in terms of operation and traffic model which are rather uncommon in Human Type Communication (HTC). These are: sporadic transmission of small packets, uplink dominated transmission, battery constrained devices, low data-rates, and large number of devices in each cell. Due to these design criteria, it is prohibitive to directly adopt complex grant-access based legacy networks for UL mMTC systems (e.g., LTE-M and NB-IoT). According to the study in [4], the signalling overhead in an mMTC system with grant-access based framework can reach up to 30%, which leaves a lot of room for improvement.

In recent years grant-free NOMA [4]–[8] emerged as a solution to realize UL mMTC in resource constrained scenarios. In principle, grant-free NOMA addresses the aforementioned challenges by combining the classical random access mechanism with NOMA. In a UL grant-free NOMA system, all the User Equipments (UE) share the same set of pre-configured radio resources (e.g., subcarriers of an OFDMA system) in non-orthogonal manner. During a Random Access (RA) opportunity, a small set of UEs activate themselves to transmit their data symbols in uplink over the shared resources by spreading their information over a uniquely assigned spreading sequence (SS) without performing any complex grant-access procedures. In this way, access-latency and overheads are reduced while securing spectral efficiency and throughput improvement [4]. At the BS, an MUD algorithm must extract the set of active UEs from the superimposed received signals. This is commonly referred as active UE detection (AUD). However, a complete MUD must perform channel estimation (CE), followed by data detection (DD) as well. Intermittent and sporadic network activity coupled with the large inactivity period exhibited by the UEs in an mMTC system cause the channel information stored at the BS to be frequently outdated. Hence, the CE and DD steps are essential stages for an MUD in an mMTC system.

B. Related Works

According to some empirical study [9], the number of active UEs during an RA opportunity never exceed 10% of the total UEs, i.e., the UE activity is sparse. Furthermore, in practice, the UE activities remain unchanged across the entire RA opportunity which is denoted as *framewise sparsity* in

the literature. Exploiting these properties, the MUD problem can be readily formulated as a joint-sparse recovery problem. Subsequently it is solved using variants of greedy Compressed Sensing (CS) based techniques [10]–[18]. The algorithms in [10]–[12], [14], [15], [17] assumed the availability of perfect channel information at the BS for carrying out AUD. Several works pursued different route other than CS to develop efficient MUD algorithms. For example, in [18], the authors formulated the AUD task as a block-sparse recovery problem and proposed using Sparse Bayesian Learning (SBL) for active UE detection and channel estimation. In [19] the authors proposed an Expectation Propagation (EP) based MUD for jointly carrying out AUD, CE and DD. In this work the sparse channel vector with computationally intractable Bernoulli-Gaussian distribution was approximated by a multivariate Gaussian vector. An entirely different approach was explored in [20] where the AUD problem was formulated as a frequency estimation problem by using sinusoidal sequences as the spreading sequences. It enabled the application of highly accurate yet low complexity non-iterative algorithms from classical signal processing for carrying out AUD.

Another emerging MUD design paradigm is the family of Deep Learning based MUD. In the context of UL grant-free NOMA systems, the deep learning based algorithms have been proposed primarily for the constellation design, resource allocation, and throughput optimization [21]–[23]. However, several applications of deep learning based MUD can be found in the literature [24]–[28]. In [25] the authors considered a composite traffic model where the BS, which has perfect channel information, serves the UEs with two types of activation pattern: random and periodic. A Long Short-Term Memory (LSTM) network was developed to predict the activity of the UEs with periodic activation. Using the predicted activity vector, an OMP based detector was initialized for detecting the activity of the UEs with random activation. Another deep learning based MUD, namely DeepMuD, was developed in [26] which proposed a novel data-frame design. However, the DeepMuD raises some scalability concerns. This is because, the number of frames allocated for pilot transmission in the proposed data-frame grows linearly with the number of UEs in the system, increasing the signalling overhead as a result. In [27] the authors developed the D-AUD, which is an ensemble of Deep Neural Networks (DNN) trained to learn the non-linear mapping between the indices of the active UEs and the received signal at the BS. Furthermore, an additional sparsity estimation step was included in order to extract the set of active UEs from the softmax output.

C. Contributions

The use of softmax in the output layers are well-suited for the family of multi-class classification problems which concern with learning from training examples which are associated with a *single label* from a set of disjoint labels [29]. During an RA opportunity in an UL grant-free NOMA system, more than one UE can become simultaneously active. From that perspective, the formulation of AUD as a multi-class classification problem [27] might not be an accurate depiction.

On this premise, the contributions put forth by this work are following

- 1) We propose formulating the AUD problem as a *multi-label* classification problem since it concerns with learning from training examples associated with a set of labels, *i.e.*, the set of active UE indices. Subsequently, we design a DNN based MUD scheme, namely DNN-MUD, which learns the mapping between the sparse binary activity vector and the received NOMA signals during an RA opportunity over frequency selective Rayleigh fading channels. One immediate advantage of the proposed scheme is it neither requires prior information about the sparsity nor requires any additional sparsity estimation stage. Furthermore, numerical experiments reveals that the DNN-MUD converges with substantially less number of iterations than its multi-class counterpart
- 2) We apply DNN compression methods like knowledge distillation [30], [31] to approximate an ensemble of trained DNN-MUD which offers a mean to trading-off between the computational overheads and the performance
- 3) We also estimate the channel gains of the active UEs using the inter-resource correlation which is an ill-posed problem otherwise. Finally we carry out the DD using the estimated channel gains
- 4) We carry out elaborate numerical investigations to examine the various performance aspects of the proposed scheme. We finally conclude by discussing some challenges in the application of the DNN for MUD as well as outlining some future research directions.

II. SYSTEM MODEL

Let us consider a UL grant-free NOMA system with a BS equipped with K antenna serving N UEs. An OFDMA system with O subcarriers is used to set up the NOMA system by allocating $M < N$ subcarriers for RA. The indices of these subcarriers are given by $M_0 + 1, M_0 + 2, \dots, M_0 + M$. Each UE $n \in \{0, 1, 2, \dots, N - 1\}$ is assigned with a spreading sequence (SS) ψ_j which is an M -dimensional complex vector. During an RA opportunity, which consists of J timeslots, a small fraction of the UEs become active and spread their information over M subcarriers using the designated SS. Specifically, at the j -th timeslot, where $j \in \{1, 2, \dots, J\}$, the n -th UE forms the vector $\beta_{n,j}\psi_j$ where $\beta_{n,j}$ is the data symbol transmitted by the n -th UE at the j -th timeslot. Subsequently, it uses the m -th orthogonal resource to transmit the m -th component of $\beta_{n,j}\psi_j$ in UL. Let $\mathcal{N}$ denote the set of active UE indices. We define $\mathcal{N} := \{a_1, a_2, \dots, a_{|\mathcal{N}|}\}$ where $|\cdot|$ denotes the cardinality. Moreover, $\{a_n\}_{n=1}^{|\mathcal{N}|}$ are the active UE indices. The ratio $p_a := |\mathcal{N}|/N$ is defined as the activation probability in the literature. It has been empirically verified that $p_a < 0.1$, making the UE activities during an RA opportunity sparse. The BS treats and processes the signal received during an RA opportunity at the antenna $k \in \{0, 1, \dots, K - 1\}$ as a block. The received signal block at the k -th antenna consists of J complex valued M -dimensional vectors $\{\mathbf{y}_{j,k}\}_{j=1}^J$. The m -th component of $\mathbf{y}_{j,k}$ can be given by

$$\mathbf{y}_{j,k}(m) = \sum_{n \in \mathcal{N}} \beta_{n,j} \mathbf{h}_{n,k}(m) \psi_n(m) + \mathbf{w}_{j,k}(m), \quad (1)$$

where $\mathbf{h}_{n,k} \in \mathbb{C}^M$ is the complex channel gain between the k -th antenna and the n -th active UE. Moreover, $\mathbf{w}_{j,k}$ is the vector valued additive noise composed with mutually independent and identically distributed zero mean complex Gaussian random variables. The covariance matrix of $\mathbf{w}_{j,k}$ is given by $\sigma^2 \mathbf{I}$ where σ is not known to the BS. The active UEs $n \in \mathcal{N}$ reserve the first data symbol as the pilot in order to carry out channel estimation. Without any loss of generality, we set

$$\beta_{n,1} = 1, \quad n \in \mathcal{N}. \quad (2)$$

Using the vector form (1) can be written as

$$\begin{aligned} \mathbf{y}_{j,k} &= \sum_{n \in \mathcal{N}} \beta_{n,j} \text{diag}(\boldsymbol{\psi}_n) \mathbf{h}_{n,k} + \mathbf{w}_{j,k} \\ &= \boldsymbol{\Psi} \mathbf{q}_{j,k} + \mathbf{w}_{j,k}, \end{aligned} \quad (3)$$

where $\boldsymbol{\Psi}$ is a $M \times M|\mathcal{N}|$ matrix formed by the SS of the active UEs, i.e.,

$$\boldsymbol{\Psi} = [\text{diag}(\boldsymbol{\psi}_{a_1}) \text{diag}(\boldsymbol{\psi}_{a_2}) \cdots \text{diag}(\boldsymbol{\psi}_{a_{|\mathcal{N}|}})]. \quad (4)$$

Moreover, $\mathbf{q}_{j,k}$ is the combined data symbol and the complex channel vector of the active UEs at the k -th antenna during j -th timeslot, i.e.,

$$\mathbf{q}_{j,k} = [\beta_{a_1,j} \mathbf{h}_{a_1,k}^\top \quad \beta_{a_2,j} \mathbf{h}_{a_2,k}^\top \quad \cdots \quad \beta_{a_{|\mathcal{N}|},j} \mathbf{h}_{a_{|\mathcal{N}|},k}^\top]^\top \quad (5)$$

We collect the vectors $\{\mathbf{y}_{j,k}\}_{j=1}^J$ received during an RA at the k -th antenna in the following matrix

$$\mathbf{Y}_k = [\mathbf{y}_{1,k} \quad \mathbf{y}_{2,k} \quad \cdots \quad \mathbf{y}_{J,k}] = \boldsymbol{\Psi} \mathbf{Q}_k + \mathbf{W}_k \quad (6)$$

where $\mathbf{Q}_k = [\mathbf{q}_{1,k} \quad \mathbf{q}_{2,k} \quad \cdots \quad \mathbf{q}_{J,k}]$ and $\mathbf{W}_k = [\mathbf{w}_{1,k} \quad \mathbf{w}_{2,k} \quad \cdots \quad \mathbf{w}_{J,k}]$. In what follows next, the BS forms the received signal block to be processed for AUD by collecting $\mathbf{Y}_k$ from all $k \in \{0, 1, \dots, K-1\}$ as following

$$\mathbf{Y} = [\mathbf{Y}_0 \quad \mathbf{Y}_1 \quad \cdots \quad \mathbf{Y}_{K-1}] = \boldsymbol{\Psi} \mathbf{Q} + \mathbf{W} \quad (7)$$

where $\mathbf{Q} = [\mathbf{Q}_0 \quad \mathbf{Q}_1 \quad \cdots \quad \mathbf{Q}_{K-1}]$ and $\mathbf{W} = [\mathbf{W}_0 \quad \mathbf{W}_1 \quad \cdots \quad \mathbf{W}_{K-1}]$. Using the property of vectorization of the matrix products, we work out the following

$$\bar{\mathbf{y}} = \text{vec}(\mathbf{Y}) = (\mathbf{I} \otimes \boldsymbol{\Psi}) \text{vec}(\mathbf{Q}) + \text{vec}(\mathbf{W}) \quad (8)$$

where $\otimes$ denotes the well-known Kronecker product. Using the definition of the Kronecker product we find

$$\bar{\mathbf{y}} = \begin{bmatrix} \boldsymbol{\Psi} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Psi} \end{bmatrix} \begin{bmatrix} \mathbf{q}_{1,0} \\ \mathbf{q}_{2,0} \\ \vdots \\ \mathbf{q}_{J,K-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{1,0} \\ \mathbf{w}_{2,0} \\ \vdots \\ \mathbf{w}_{J,K-1} \end{bmatrix} \quad (9)$$

Let $\boldsymbol{\lambda}$ is a binary indicator vector of length N such that

$$\begin{cases} \lambda(n) = 1, & \text{if } n \in \mathcal{N} \\ \lambda(n) = 0, & \text{if } n \notin \mathcal{N} \end{cases} \quad (10)$$

In what follows next, we estimate $\boldsymbol{\lambda}$ using $\bar{\mathbf{y}}$ using the proposed DNN based MUD.

III. ACTIVE UE DETECTION

It was established in [32] that the multilayer feedforward networks, e.g., DNNs can be classified as universal function approximators. The formal definition of a fully connected feedforward neural network can be given as following [33]

Definition 1. A fully connected feedforward neural network (NN) can be given by its architecture $\mathcal{A} = (\mathcal{M}, \varrho)$, where $\mathcal{M} \in \mathbb{N}^{L+1}$, and $\varrho : \mathbb{R} \rightarrow \mathbb{R}$. Here, $L+1$ is the number of layers, and ϱ is the so called activation function. Furthermore, we refer to $\mathcal{M}(0)$, $\mathcal{M}(L)$, and $\mathcal{M}(l)$, $l \in \{1, 2, \dots, L-1\}$, as the number of neurons in the input, output, and the l -th hidden layer, respectively. The number of neurons in the l -th layer $\mathcal{M}(l)$ is also denoted as the width of the l -th layer. Let us denote the set of parameters as $\theta = (\theta^{(l)})_{l=1}^L = ((\mathbf{W}^{(l)}, \mathbf{b}^{(l)}))_{l=1}^L$, where $\mathbf{W}^{(l)}$ are the weight matrices of dimension $\mathcal{M}(l) \times \mathcal{M}(l-1)$. Furthermore, $\mathbf{b}^{(l)}$, $l \in \{1, 2, \dots, L-1\}$ are the bias vectors of length $\mathcal{M}(l)$. Then the number of total parameter in the feedforward NN can be calculated as

$$\mathcal{Q} := \sum_{l=1}^L \mathcal{M}(l) \mathcal{M}(l-1) + \mathcal{Q}, \quad (11)$$

where $\mathcal{Q} \geq 0$ is the number of parameters which depends on the architecture under consideration. Let us define the corresponding realization function $\mathcal{F}_a : \mathbb{R}^{\mathcal{M}(0)} \times \mathbb{R}^{\mathcal{Q}} \rightarrow \mathbb{R}^{\mathcal{M}(L)}$ which satisfies for all input $\mathbf{x} \in \mathbb{R}^{\mathcal{M}(0)}$ and parameters θ that $\mathcal{F}_a(\mathbf{x}, \theta) = \mathcal{F}^{(L)}(\mathbf{x}, \theta)$, where

$$\mathcal{F}^{(1)}(\mathbf{x}, \theta) = f(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}), \quad (12)$$

$$\bar{\mathcal{F}}^{(l)}(\mathbf{x}, \theta) = \varrho(\mathcal{F}^{(l)}(\mathbf{x}, \theta)), \quad l \in [L-1], \quad (13)$$

$$\hat{\mathcal{F}}^{(l)}(\mathbf{x}, \theta) = g(\bar{\mathcal{F}}^{(l)}(\mathbf{x}, \theta)), \quad l \in [L-1], \quad (14)$$

$$\mathcal{F}^{(l+1)}(\mathbf{x}, \theta) = f(\mathbf{W}^{(l+1)} \hat{\mathcal{F}}^{(l)}(\mathbf{x}, \theta) + \mathbf{b}^{(l+1)}), \quad l \in [L-1], \quad (15)$$

and ϱ is an elementwise operation. Here, $\mathcal{F}^{(l)}(\mathbf{x}, \theta)$ and $\bar{\mathcal{F}}^{(l)}(\mathbf{x}, \theta)$ are the pre and post activation vectors, respectively, for the l -th layer. $f(\cdot)$ and $g(\cdot)$ denote additional transformations of the pre-activation and post-activation vectors respectively (e.g., batch normalization, dropout, et cetera). The architecture is classified as deep if $L > 3$.

In this work, we approximate the mapping between the received NOMA signals $\bar{\mathbf{y}}$ and the sparse binary activity vector $\boldsymbol{\lambda}$ using an appropriately trained DNN, namely DNN-MUD. We iteratively update the parameters θ of the DNN-MUD using backpropagation [34] by minimizing a carefully chosen loss function. Next, we describe the architecture of the proposed DNN-MUD.

A. DNN-MUD Architecture

Let us define the following vector

$$\mathbf{z} := [\text{Re}(\bar{\mathbf{y}}^\top) \quad \text{Im}(\bar{\mathbf{y}}^\top)]^\top, \quad (16)$$

where $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ denote the real and imaginary part, respectively. It can be verified that the real valued vector $\mathbf{z}$

Fig. 1. An UL mMTC scenario with the architecture of the proposed DNN-MUD

has $2JKM$ elements. Then, the AUD problem in the context of a grant-free NOMA system can be written as a DNN based function approximation problem using the definition provided in Definition 1, *i.e.*,

$$\hat{\lambda} = \mathcal{F}_a(\bar{\mathbf{z}}^{(0)}, \theta), \quad (17)$$

where

$$\bar{\mathbf{z}}^{(0)}(n) := \frac{\mathbf{z}(n) - \mathbb{E}[\mathbf{z}]}{\sqrt{\text{Var}[\mathbf{z}]}} , \quad n \in \{0, 1, \dots, 2JKM - 1\} \quad (18)$$

The superscript (0) denotes that the vector under consideration is an input for the 0-th layer, *i.e.*, the input layer. This is evident from (17) that the number of neurons in the input and the output layers of the DNN-MUD are $\mathcal{M}(0) = 2JKM$ and $\mathcal{M}(L) = N$, respectively. In Figure 1 we present the block diagram of the proposed DNN-MUD delineating its various building blocks. In the subsequent sections we detail the function of each of these blocks in the context of DNN-MUD training.

B. Building blocks of a hidden layer

Let $\mathbb{T}$ be the set of training samples which is composed of T complex NOMA signal vectors $\{\bar{\mathbf{y}}_t\}_{t=0}^{T-1}$ generated using

the equality (3), (6), (7), and (8). The practical utility of using synthetic training samples for training DNN based MUD was discussed in detail in [27]. During the training stage, we first split the training samples into real and imaginary part followed by the normalization process given in (16) and (18). Then we feed the resultant training vectors $\{\bar{\mathbf{z}}_t^{(0)}\}_{t=0}^{T-1}$ into the input layer of DNN-MUD which is a fully connected (FC) layer. For the t -th training sample, the output of the FC layer yields

$$\bar{\mathbf{z}}_t^{(1)} = \mathbf{W}^{(1)}\bar{\mathbf{z}}_t^{(0)} + \mathbf{b}^{(1)} \quad (19)$$

In the sequel, we discuss the remaining building blocks of a hidden layer and the roles associated with them.

1) *Batch Normalization*: Due to the grant-free access coupled with the sparse activity of the UEs, the distribution of the network activations are subjected to very large variations [27]. This phenomenon is known as the *Internal Covariate Shift* which was explored in detail in [35]. A direct consequence of the internal covariate shift is it restricts the use of the large learning rate during training. Furthermore, it also requires careful parameter initialization [35]. This inevitably leads to several difficulties in the training stage, *e.g.*, slower training of the network and the loss function plateauing. In order to

tackle this phenomenon, batch normalization was introduced in [35] with a goal to enforce fixed mean and variance over the pre-activation distribution. The steps involved in the batch normalization are as following:

- First, the outputs of the FC layers are stacked in the batches of size B , where B is a hyperparameter. For example, the batch formed by the first B FC layer outputs can be given as $\mathbf{B} = [\bar{\mathbf{z}}_0^{(1)} \ \bar{\mathbf{z}}_1^{(1)} \ \bar{\mathbf{z}}_2^{(1)} \ \cdots \ \bar{\mathbf{z}}_{B-1}^{(1)}]$.
- Afterwards, the batch normalization for the batch $\mathbf{B}$ is performed as following

$$\tilde{\mathbf{z}}_i(n) = \iota \left(\frac{\bar{\mathbf{z}}_i^{(1)}(n) - \mu_{B,n}}{\sigma_{B,n}} \right) + \kappa, \quad i \in \{0, 1, \dots, B-1\}, \quad (20)$$

where $n \in \{0, 1, \dots, 2JKM - 1\}$. Moreover,

$$\begin{aligned} \mu_{B,n} &= \frac{1}{B} \sum_{i=0}^{B-1} \bar{\mathbf{z}}_i^{(1)}(n) \\ \sigma_{B,n}^2 &= \frac{1}{B} \sum_{i=0}^{B-1} (\bar{\mathbf{z}}_i^{(1)}(n) - \mu_{B,n})^2. \end{aligned} \quad (21)$$

The parameters ι and κ are known as the scale and the shift parameters, respectively. These parameters are optimized during the training of DNN along with the parameters θ . Subsequently, batch normalization is carried out for the remaining batches. Afterwards, the normalized training vectors $\{\tilde{\mathbf{z}}_i(n)\}_{i=0}^{T-1}$ pass through the activation layer which is detailed in the following section.

2) *ReLU Activation*: Over the choices of the non-linear activation functions, we opt for the Rectified Linear Unit (ReLU) activation function in this work. Applying ReLU activation on the n -th element of the i -th batch normalization output yields the following

$$\begin{cases} \text{ReLU}[\tilde{\mathbf{z}}_i(n)] = \tilde{\mathbf{z}}_i(n), & \text{if } \tilde{\mathbf{z}}_i(n) > 0 \\ \text{ReLU}[\tilde{\mathbf{z}}_i(n)] = 0, & \text{if } \tilde{\mathbf{z}}_i(n) \leq 0 \end{cases} \quad (22)$$

The reason behind choosing ReLU as the non-linear activation function is two-fold. First, compared to the family of tanh or sigmoid based activation functions, ReLU possesses a significantly reduced likelihood of vanishing/saturating gradient. Second, as suggested in [36], ReLU allows the network to easily obtain some sparse representations of the inputs. This effectively leads to robustness against small input changes via information entanglement (see [36] for details). Both of these properties greatly accelerate the convergence of the stochastic gradient descent based optimizers during the model training [37]. Next, let $\{\tilde{\mathbf{z}}_i(n)\}_{i=0}^{T-1}$ denote the post-activation training vectors.

3) *Dropout Layer*: To tackle the tendency of overfitting exhibited by the family of DNN architectures, there exist several regularization techniques (see [38] for an overview). Dropout is one of the widely adopted regularization techniques which aims to prevent the DNN from overfitting by randomly dropping out neurons along with the associated connections [39]. In essence, during training stage, dropout randomly samples from a set containing exponentially large number of *thinned* networks. On the other hand, during prediction

it approximates the effect of averaging all of those thinned networks with one unthinned network having smaller weights [39].

Let $\mathbf{u}^{(l)}$ be the binary dropout vector of length $\mathcal{M}(l)$ associated with the l -th hidden layer. The elements of $\mathbf{u}^{(l)}$ are the Bernoulli random variables with the dropout probability P_D , i.e.,

$$\mathbf{u}^{(l)}(n) \sim \text{Bernoulli}(P_D), \quad n \in \{0, 1, \dots, \mathcal{M}(l) - 1\} \quad (23)$$

Continuing the example of the first hidden layer, the output vector after the dropout layer can be given as

$$\hat{\mathbf{z}}_i^{(1)} = \tilde{\mathbf{z}}_i^{(1)} \odot \mathbf{u}^{(1)} \quad (24)$$

C. Output layer

As described earlier, each of the $L - 1$ hidden layers constitutes an FC layer followed by the batch normalization, the ReLU layer, and finally, the dropout layer. On the other hand, the output layer constitutes an FC layer only, which outputs a vector of length $\mathcal{M}(L) = N$. Continuing the example of the i -th training sample, we get

$$\bar{\mathbf{z}}_i^{(L)} = \mathbf{W}^{(L)} \hat{\mathbf{z}}_i^{(L-1)} + \mathbf{b}^{(L)} \quad (25)$$

Afterwards, the binary indicator vector $\hat{\boldsymbol{\lambda}}_i$ is estimated using the output vector $\bar{\mathbf{z}}_i^{(L)}$. For example, $\bar{\mathbf{z}}_i^{(L)}$ can be passed through a softmax layer to generate a probability distribution of the UE activities over the set of UE indices [27]. The use of softmax layer turns the underlying classification problem into a multi-class one. The resultant probability distribution can be used in conjunction with an ad-hoc threshold-based policy for selecting the UE indices having the largest probabilities. However, under the proposed scheme, the mapping between the output vector of the DNN $\bar{\mathbf{z}}_i^{(L)}$ and the estimated support vector $\hat{\boldsymbol{\lambda}}_i$ is carried out in two steps. First, we apply the element-wise sigmoid function on $\bar{\mathbf{z}}_i^{(L)}$ to constrain each element in the open interval $(0, 1)$. Then the estimated support vector $\hat{\boldsymbol{\lambda}}$ is determined by applying an element-wise threshold on the resultant vector found in the first step, i.e.,

$$\begin{cases} \hat{\boldsymbol{\lambda}}_i(n) = 1, & \text{if } \mathcal{S}_i(n) \geq 0.5 \\ \hat{\boldsymbol{\lambda}}_i(n) = 0, & \text{if } \mathcal{S}_i(n) < 0.5, \end{cases} \quad (26)$$

where $n \in \{0, 1, \dots, N - 1\}$. Moreover,

$$\mathcal{S}_i(n) = \frac{1}{1 + \exp\left(\frac{-\bar{\mathbf{z}}_i^{(L)}(n)}{\xi}\right)} \quad (27)$$

Some recent works denoted $\mathcal{S}_i$ as the *soft output* whereas $\xi \geq 1$ is the so called temperature [30] parameter which controls the softening of the output. An immediate advantage of this approach is it does not require any additional sparsity estimation stage, nor does it require any information about the sparsity level, i.e., the number of active UEs $|\mathcal{N}|$ in the RA opportunity under consideration. As can be seen from the above discussion, the received NOMA signals pass through all the layers $\{\mathcal{M}(l)\}_{l=0}^L$ followed by a vector-valued sigmoid function to produce the soft output. This process is commonly referred as the forward-propagation.

D. Training the DNN-MUD

Using the training samples $\bar{\mathbf{z}}_0^{(0)}, \bar{\mathbf{z}}_1^{(0)}, \dots, \bar{\mathbf{z}}_{T-1}^{(0)}$, the learning algorithm iteratively pursues the closest approximation of the optimal mapping function $\mathcal{F}_a^*$, parameterized by θ^* . In order to pose the underlying MUD problem as a multi-label classification problem, we choose the Binary Cross-Entropy (BCE) based loss function to minimize the well known Kullbeck-Liebler divergence between the binary UE indicator vector and the output vector produced by the forward-propagation. On this premise, the BCE loss function for a given set of parameters θ can be given as

$$\mathcal{L}(\mathcal{S}, \lambda, \theta) := \lambda^\top \log \mathcal{S} + (1 - \lambda)^\top \log(1 - \mathcal{S}) \quad (28)$$

In (28) $\mathbf{1}$ is a vector of length N where all elements are equal to 1. Moreover, the log is taken element-wise. This loss function is minimized via backpropagation [34] where the parameters θ in the w -th iteration are updated according the steepest descent principle, *i.e.*,

$$\theta_w = \theta_{w-1} - \eta \nabla \mathcal{L}(\mathcal{S}, \lambda, \theta_{w-1}) \quad (29)$$

In (29) ∇ denotes the gradient, and η is the learning rate. We do not employ any l_2 regularization since both batch-normalization and dropout act as the regularizers [35], [39]. Several stochastic gradient descent based optimizers are available in the literature [40], [41] for minimizing the left hand side of the equality (28). We detail the choice of the optimizer in a later section.

E. Ensembling and Knowledge Distillation

Some classic works on the neural networks showed that the generalization error can be significantly reduced by invoking ensembles of networks with similar architecture which are independently trained on different training sets [42], [43]. Suppose, E is the number of DNN-MUD independently initialized by the parameters $(\theta_0^1, \theta_0^2, \dots, \theta_0^E)$. Moreover, let $\mathbb{T}_1, \mathbb{T}_2, \dots, \mathbb{T}_E$ denote the E training sets. During the MUD stage, the received signal $\bar{\mathbf{y}}$ is fed to each of the trained networks within the ensemble. These networks independently generate the soft outputs $\{\mathcal{S}^{(e)}\}_{e=1}^E$. By simply taking the arithmetic mean of these soft outputs we calculate the ensemble soft output as following

$$\mathcal{S}^{\text{en}} = \sum_{e=1}^E \mathcal{S}^{(e)} \quad (30)$$

Finally, using (26) we can estimate the binary indicator vector.

While using an ensemble of DNN-MUD helps to achieve better performance metrics, the implementation of this scheme comes with additional computational overhead. To enable a trade off between the performance metrics and the computational overhead, we propose using a technique called Knowledge Distillation (KD). KD is a form of model compression which was first pioneered by [30], later by [31], where it was shown that the representation learned by an ensemble of networks, namely, the teacher, can be transferred into a single model, namely, the student. In essence, an ensemble of the trained DNNs can be approximated by a single DNN with the

same architecture. The student network is then trained on a different training set, known as the transfer set, by optimizing the following weighted loss function

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{BCE}}(\mathcal{S}^{\text{S}}, \lambda, \theta^{\text{S}}) + \zeta \mathcal{L}_{\text{CE}}(\mathcal{S}^{\text{S}}, \mathcal{S}^{\text{T}}) \quad (31)$$

In the loss function above, $\mathcal{S}^{\text{T}} = \mathcal{S}^{\text{en}}$ is the soft output of the teacher, which is an ensemble of the DNN-MUD in this case. $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{CE}}$ are the binary cross-entropy loss given the student parameter θ^{S} and the cross-entropy loss function, respectively. Moreover, ζ is a tunable parameter to balance both of the loss functions. In the subsequent sections we denote this student network as the KD-DNN-MUD.

IV. CHANNEL ESTIMATION AND DATA DETECTION

As mentioned in the Section II, the active UEs $n \in \mathcal{N}$ allocate the first symbol for sending pilots in UL. Let us remind the reader that the received pilot signal, *i.e.*, $j = 1$ at the k -th BS can be written as

$$\mathbf{y}_{1,k} = \Psi \mathbf{q}_{1,k} + \mathbf{w}_{1,k} \quad (32)$$

where $\mathbf{q}_{1,k} = [\mathbf{h}_{a_1,k}^\top \mathbf{h}_{a_2,k}^\top \dots \mathbf{h}_{a_{|\mathcal{N}|},k}^\top]^\top$ since $\beta_{n,1} = 1$ for all $n \in \mathcal{N}$ (see (5)).

Let $\hat{\mathcal{N}}$ denote the estimated set of active UE indices *i.e.*, $\hat{\mathcal{N}} = \{\hat{a}_1 \hat{a}_2 \dots \hat{a}_{|\hat{\mathcal{N}}|}\}$. Using $\hat{\mathcal{N}}$ we form the matrix of spreading sequences as following

$$\hat{\Psi} = [\text{diag}(\psi_{\hat{a}_1}) \text{diag}(\psi_{\hat{a}_2}) \dots \text{diag}(\psi_{\hat{a}_{|\hat{\mathcal{N}}|}})]. \quad (33)$$

Using (5) and (33), the channel estimation can be done by solving the following linear least square problem

$$[\hat{\mathbf{h}}_{\hat{a}_1,k}^\top \hat{\mathbf{h}}_{\hat{a}_2,k}^\top \dots \hat{\mathbf{h}}_{\hat{a}_{|\hat{\mathcal{N}}|},k}^\top]^\top = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{y}_{1,k} - \hat{\Psi} \mathbf{x}\|_2^2 \quad (34)$$

where $\hat{\mathbf{h}}_{n,k}$ is the estimated channel gain vector for the UE $n \in \hat{\mathcal{N}}$ at the k -th antenna.

The estimator in (33) can be shown to be the minimum mean square error (MMSE) estimator provided that the elements of $\mathbf{w}_{1,k}$ are independent and identically distributed random variables. Since $\hat{\Psi}$ has M rows and $M|\hat{\mathcal{N}}|$ columns, the least square problem in (34) is, however, an ill-posed problem without the presence of any additional information, and has infinitely many solutions. In what follows next, we show how the inter-resource correlation of the channel gains can be utilized to turn this ill-posed problem into a well defined one.

Let P denote the maximum delay spread of the system under consideration. Furthermore, the Channel Impulse Response (CIR) of an UE $n \in \mathcal{N}$ during an RA opportunity is denoted as a P dimensional complex valued vector $\mathbf{r}_n$ where the p -th component $\mathbf{r}_n(p)$ is the gain of the p -th channel tap. Hence, the channel gain experienced by the UE n at the $M_0 + m$ th subcarrier can be given as

$$\begin{aligned} \mathbf{h}_n(m) &= \sum_{p=0}^{P-1} \mathbf{r}_n(p) \exp\left(j \frac{2\pi(M_0 + m)p}{O}\right) \\ &= \mathbf{F} \mathbf{r}_n \end{aligned} \quad (35)$$

where $\mathbf{F}$ is an $M \times P$ matrix. The element in its m -th row and $p + 1$ -th column is given by

$$\mathbf{F}(m, p + 1) = \exp\left(i \frac{2\pi(M_0 + m)p}{O}\right) \quad (36)$$

Note that the matrix $\mathbf{F}$ is independent of the UEs. Using this property, one can estimate the channel taps instead of the channel gains as following

$$[\hat{\mathbf{r}}_{\hat{a}_1, k}^\top \ \hat{\mathbf{r}}_{\hat{a}_2, k}^\top \ \cdots \ \hat{\mathbf{r}}_{\hat{a}_{|\hat{\mathcal{N}}|}, k}^\top]^\top = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}_{1, k} - \tilde{\Psi}\mathbf{x}\|_2^2 \quad (37)$$

where $\tilde{\Psi}$ is an $M \times P|\hat{\mathcal{N}}|$ matrix, *i.e.*,

$$\tilde{\Psi} = [\operatorname{diag}(\psi_{\hat{a}_1})\mathbf{F} \ \operatorname{diag}(\psi_{\hat{a}_2})\mathbf{F} \ \cdots \ \operatorname{diag}(\psi_{\hat{a}_{|\hat{\mathcal{N}}|}})\mathbf{F}]. \quad (38)$$

In practice it is reasonable to expect that $M > P|\hat{\mathcal{N}}|$ due to the sparse nature of the UE activity. Hence, one can first estimate the channel taps using the linear least squares as $\tilde{\Psi}^\dagger \mathbf{y}_{1, k}$ where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo-inverse operation. Using these estimates, the channel gains can then be estimated using the relationship given in (35).

After estimating the channel tap gains using (37), we form the so called $M \times |\hat{\mathcal{N}}|$ equivalent channel matrix for the k -th antenna as following

$$\check{\Psi} = [\operatorname{diag}(\psi_{\hat{a}_1})\mathbf{F}\hat{\mathbf{r}}_{\hat{a}_1, k} \ \operatorname{diag}(\psi_{\hat{a}_2})\mathbf{F}\hat{\mathbf{r}}_{\hat{a}_2, k} \ \cdots \ \operatorname{diag}(\psi_{\hat{a}_{|\hat{\mathcal{N}}|}})\mathbf{F}\hat{\mathbf{r}}_{\hat{a}_{|\hat{\mathcal{N}}|}, k}]. \quad (39)$$

Let us define the Nearest-Neighbour decoding as

$$\operatorname{NN}(x) = \underset{\beta \in \mathcal{A}}{\operatorname{argmin}} |x - \beta|, \quad (40)$$

where $\mathcal{A}$ is a constellation of choice (*e.g.*, QPSK). Then, for $j > 1$, the data symbol detected at the k -th antenna can be found via the Nearest-Neighbour decoding

$$\hat{\beta}_n^k = \operatorname{NN}(F(n)), \quad (41)$$

where $F = \check{\Psi}^\dagger \mathbf{y}_{j, k}$ and $n \in \hat{\mathcal{N}}$. One can detect the data symbol transmitted by the n -th active UE during the j -th timeslot at all $k \in \{0, 1, 2, \dots, K-1\}$ antenna and exploit the receiver diversity by employing further processing (*e.g.*, majority voting) to refine the detection.

V. NUMERICAL SIMULATION

A. Simulation setup

For numerical investigation, we consider two UL mMTC systems. The number of subcarriers allocated for grant-free NOMA for these two systems are $M = 40$ and $M = 64$, respectively. Unless otherwise specified, these are single antenna systems, *i.e.*, $K = 1$. These mMTC systems are serving a cell having the radius of 200 m where $N = 2M$. These N UEs are randomly deployed within the cell. Let d_n be the distance between the BS and the n -th UE where $10 \leq d_n \leq 200$. The pathloss exponent (in dBm) between the n -th UE and the BS is modeled as specified in [19], [27], [44]:

$$\tau_n = -128.1 - 36.7 \log_{10}(d_n) \quad (42)$$

In (42) the unit of d_n is in km.

Fig. 2. Training and validation loss for training set with 5×10^5 samples for a UL GF-NOMA system with $M = 64$, $N = 128$

The frequency selective channel gains are modeled using zero mean and unit variance complex Gaussian random variables. Furthermore, the transmission bandwidth of the system is set at 1 MHz. The power spectral density is -174 dBm/Hz. Unless otherwise specified, the activation probability is set at $p_a = 0.05$. We use low-density signature codes [45] as the spreading sequence with independent and identically distribute complex normal random variable as the non-zero elements. Let ρ denote the number of non-zero elements in each sequences. We set the ratio $\rho/M = 4$ while generating the sequences. The performance metrics are evaluated using 5000 independent channel realizations. In the simulation, the KD-DNN-MUD is an approximation of an DNN-MUD ensemble with $E = 12$ unless otherwise specified.

B. Hyperparameter selection

The proposed DNN-MUD involves several hyperparameters to be tuned to ensure stable model training and improved performance metrics once deployed. The hyperparameters include the number of layers L , the width of the hidden layers $\{\mathcal{M}(n)\}_{n=1}^{L-1}$, the batch size B , the dropout probability P_D , and the learning rate η . For simplifying the search space, instead of directly searching $\{\mathcal{M}(n)\}_{n=1}^{L-1}$, we search the vector of integers $\mathcal{T} \in \mathbb{Z}^{L-1}$ such that $\{\mathcal{M}(n)\}_{n=1}^{L-1} = 2JKM\mathcal{T}$. Let $\mathcal{H}$ denote the set of the hyperparameters to be tuned, *i.e.*, $\mathcal{H} = \{L, \mathcal{T}, B, P_D, \eta\}$. For finding the optimal set of hyperparameters $\mathcal{H}^*$ based on cross-validation, one can perform random or grid search. In this work we use Optuna [46] which is a hyperparameter optimization framework that uses Covariance Matrix Adaptation (CMA) based sampling for searching the optimal set of hyperparameters [47]. We list the final selection of the hyperparameters in the Table I.

C. Model Training

In this work we use $T = 5 \times 10^5$ training samples for training the DNN-MUD. We would like to emphasize that this training set is substantially smaller than some of the related works. For the similar simulation setup and similar choice of the spreading sequences, the model training did not converge for a training set with $\sim 10^5$ training samples in [27]. In our

TABLE I
LIST OF HYPERPARAMETERS AND THEIR VALUES FOR $M = 40$, $M = 64$

Hyperparameter	$M = 40$	$M = 64$
L	5	7
$\mathcal{T}$	$[1\ 2\ 2\ 1]^\top$	$[1\ 1\ 1\ 1\ 1\ 1]^\top$
B	500	500
P_D	0.1	0.25
η	5×10^{-4}	5×10^{-4}

Fig. 3. (a) AER of the DNN-MUD as a function of transmit power t_p for $M = 40$ (a) AER of the DNN-MUD as a function of transmit power t_p for $M = 64$

experiment with the optimizers, the ADAM [40] stood out among the candidates. The candidate optimizers include the stochastic gradient descent (SGD), the RMSProp, the ADAM, and the AMSGrad [48]. We set the epoch at 200. Moreover, we use Kaiming initialization for initializing the parameters θ [49]. In Figure 2 we plot the training and validation loss for $M = 64$, $N = 128$, and $p_a = 0.1$ for the DNN-MUD with the hyperparameters listed in the Table I.

D. Simulation Results

In our simulations we use the Activity Error Rate (AER), and the Bit Error Rate (BER) as the performance metrics. The definition of the AER can be given as following

$$\text{AER} = \left(|\{n \in \mathcal{N} : n \notin \hat{\mathcal{N}}\}| + |\{n \in \hat{\mathcal{N}} : n \notin \mathcal{N}\}| \right) / |\mathcal{N}|.$$

The AER is defined in such a way so that it incorporates both missed detection of the active UEs and false inclusion of

Fig. 4. (a) AER of the DNN-MUD as a function of the activation probability p_a for $M = 40$ (a) AER of the DNN-MUD as a function of the activation probability p_a for $M = 64$

the inactive UEs. On the other hand we use the conventional definition of BER.

In Figure 3 we evaluate the AER as a function of the transmit power t_p where $p_a = 0.05$, $K = 1$. Specifically, we evaluate the AER performance of the DNN-MUD ensembles with $E = 2, 4, 6, 10$. In this plot KD ($E = 12$) denotes a KD-DNN-MUD which is trained using a teacher network that is an ensemble of DNN-MUDs with $E = 12$. It can be seen that, for both $M = 40$ and $M = 64$, the bigger ensembles yield improved AER. However, the improvement of AER against E is not linear and it saturates as E increases. It can be seen that the KD-DNN-MUD consistently outperforms the ensemble with $E = 6$ as t_p varies from 0 dBm to 18 dBm.

In Figure 4 we evaluate the AER performance of the DNN-MUD against various activation probabilities p_a . In this experiment we set $t_p = 18$ dBm, $N = 2M$, and $K = 1$. From the figure it is evident that the DNN-MUD rapidly loses its AER performance as the activation probability increases. This is an expected outcome since larger p_a yield higher inter-UE interference. However, the larger ensembles and the KD-DNN-MUD exhibits some resilience against the increasing p_a . For example, $E = 12$ delivers 40% and 119% gain in the AER compared to $E = 2$ for $M = 40$ and $M = 64$, respectively. In Figure 5 and Figure 6 we plot the BER of the proposed MUD using the similar setting in Figure 3 and Figure 4.

(a)

(b)

Fig. 5. (a) BER of the DNN-MUD as a function of transmit power t_p for $M = 40$ (a) BER of the DNN-MUD as a function of transmit power t_p for $M = 64$

Since the MUD jointly conducts AUD, CE and DD, the BER performance is strongly correlated with the AER performances as shown in Figure 3 and Figure 4.

In Figure 7 we evaluate the BER performance of the proposed scheme for $K = 3, 5, 7$. We set $p_a = 0.05$, and $N = 2M$ for this experiment. Moreover, we use the KD-DNN-MUD for generating the performance curves. As can be seen from the figure, the BER performance improves as the number of antenna K increases from 3 to 7. In this case the improvement in the BER performance can be attributed to the following two-fold phenomena. Firstly, according to (9), (16), and the Definition 1, a larger antenna system employs a larger network for the same number of neurons in the output layer, *i.e.*, the number of UEs N . Hence, the AER reduces, resulting into a performance gain in the BER. Secondly, the receiver diversity contributes to the further reduction to the BER.

Finally, we carry out an experiment on the algorithm runtime for the DNN-MUD ensembles of different sizes. In this experiment, we carry out end-to-end MUD (*i.e.*, AUD, CE, and DD) for 20 RA opportunities. Moreover, the experiment was run using PyTorch on an Intel Corei7 2600, 8-core computer clocked at 3.40 GHz. It can be seen that the algorithm runtime grows moderately as the size of the ensembles E increases for a given number of antenna K . However, across various multi-antenna configuration (*i.e.*, $K = 1, 3, 5$) the algorithm

(a)

(b)

Fig. 6. (a) BER of the DNN-MUD as a function of the activation probability p_a for $M = 40$ (a) BER of the DNN-MUD as a function of the activation probability p_a for $M = 64$

runtime differs vastly. For example, the runtime for $K = 3$ and $K = 5$ compared to the runtime $K = 1$ increases by the factors of 9.5 and 29.53, respectively, for $E = 6$. In this case, one can see how model compression techniques (*e.g.*, KD in this work) can be essential to enable fast MUD at the BS in the multi-antenna setting.

VI. CONCLUSIONS

In this work, we have explored the utility of the DNN based MUD for the grant-free NOMA based UL mMTC systems. We have shown that formulating the AUD problem as a multi-label classification problem simplifies the training and architecture of the DNN-MUD. By incorporating model compression techniques like knowledge distillation greatly alleviates the computational overhead required for the end-to-end MUD using DNN, especially in multi-antenna settings. We also have demonstrated how the inter-resource correlations can be exploited to enable channel estimation in UL with frequency selective fading channels. In our numerical investigations, we have considered a practical NLOS environment and inspected the AER, SER, and the algorithmic runtime of the proposed MUD in both single and multiple antenna settings. In those simulations, the proposed DNN-MUD has shown promising AER and BER results in highly sparse scenarios and showed resilience in the low transmit power scenarios. However, the

Fig. 7. (a) BER of the KD-DNN-MUD as a function of transmit power t_p for $M = 40$ for different multi-antenna configuration (a) BER of the KD-DNN-MUD as a function of transmit power t_p for $M = 64$ for different multi-antenna configuration

Fig. 8. Algorithm runtime as a function of E for $K = 1, 3, 5$

performance of the DNN-MUD in moderately high activation probability region is not that promising. Our future course of research will include assessing the utility of the Bayesian Neural Networks and Transfer Learning to design scalable DNN based MUD.

REFERENCES

- [1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, September 2016.
- [2] C. Bockelmann, N. K. Pratas, G. Wunder, S. Saur, M. Navarro, D. Gregoratti, G. Vivier, E. D. Carvalho, Y. Ji, C. Stefanovic, P. Popovski, Q. Wang, M. Schellmann, E. Kosmatos, P. Demestichas, M. Raceala-Motoc, P. Jung, S. Stanczak, and A. Dekorsy, "Towards Massive Connectivity Support for Scalable mMTC Communications in 5G Networks," *IEEE Access*, vol. 6, 2018.
- [3] H. Tullberg, P. Popovski, Z. Li, M. A. Uusitalo, A. Hoglund, O. Bulakci, M. Fallgren, and J. F. Monserrat, "The METIS 5G System Concept: Meeting the 5G Requirements," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 132–139, December 2016.
- [4] K. Au, L. Zhang, H. Nikopour, E. Yi, A. Bayesteh, U. Vilaipornsawai, J. Ma, and P. Zhu, "Uplink contention based SCMA for 5G radio access," in *2014 IEEE Globecom Workshops (GC Wkshps)*, December 2014, pp. 900–905.
- [5] (2018) Technical specification group radio access network; Study on non-orthogonal multiple access (NOMA) for NR (Release 16). 3GPP TR 38.812 V16.0.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3236>
- [6] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-Free Non-Orthogonal Multiple Access for IoT: A Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1805–1838, 2020.
- [7] M. Elbayoumi, M. Kamel, W. Hamouda, and A. Youssef, "NOMA-Assisted Machine-Type Communications in UDN: State-of-the-Art and Challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1276–1304, 2020.
- [8] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive Non-Orthogonal Multiple Access for Cellular IoT: Potentials and Limitations," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 55–61, September 2017.
- [9] J. Hong, W. Choi, and B. D. Rao, "Sparsity Controlled Random Multiple Access With Compressed Sensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 998–1010, February 2015.
- [10] A. T. Abebe and C. G. Kang, "Iterative Order Recursive Least Square Estimation for Exploiting Frame-Wise Sparsity in Compressive Sensing-Based MTC," *IEEE Communications Letters*, vol. 20, no. 5, pp. 1018–1021, 2016.
- [11] B. Wang, L. Dai, T. Mir, and Z. Wang, "Joint User Activity and Data Detection Based on Structured Compressive Sensing for NOMA," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1473–1476, July 2016.
- [12] C. Wei, H. Liu, Z. Zhang, J. Dang, and L. Wu, "Approximate Message Passing-Based Joint User Activity and Data Detection for NOMA," *IEEE Communications Letters*, vol. 21, no. 3, pp. 640–643, 2017.
- [13] Y. Du, B. Dong, W. Zhu, P. Gao, Z. Chen, X. Wang, and J. Fang, "Joint Channel Estimation and Multiuser Detection for Uplink Grant-Free NOMA," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 682–685, 2018.
- [14] Y. Du, C. Cheng, B. Dong, Z. Chen, X. Wang, J. Fang, and S. Li, "Block-Sparsity-Based Multiuser Detection for Uplink Grant-Free NOMA," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 7894–7909, 2018.
- [15] J. Zhang, Y. Pan, and J. Xu, "Compressive Sensing for Joint User Activity and Data Detection in Grant-Free NOMA," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 857–860, June 2019.
- [16] J. Liu, G. Wu, S. Li, and O. Tirkkonen, "Blind detection of uplink grant-free SCMA with unknown user sparsity," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [17] N. Y. Yu, "Multiuser Activity and Data Detection via Sparsity-Blind Greedy Recovery for Uplink Grant-Free NOMA," *IEEE Communications Letters*, vol. 23, no. 11, pp. 2082–2085, November 2019.
- [18] Y. Zhang, Q. Guo, Z. Wang, J. Xi, and N. Wu, "Block Sparse Bayesian Learning Based Joint User Activity Detection and Channel Estimation for Grant-Free NOMA Systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9631–9640, 2018.
- [19] J. Ahn, B. Shim, and K. B. Lee, "EP-Based Joint Active User Detection and Channel Estimation for Massive Machine-Type Communications," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 5178–5189, July 2019.
- [20] S. M. Hasan, K. Mahata, and M. M. Hyder, "Uplink Grant-Free NOMA With Sinusoidal Spreading Sequences," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3757–3770, 2021.
- [21] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8440–8450, 2018.

- [22] R. Huang, V. W. S. Wong, and R. Schober, "Throughput optimization in grant-free noma with deep reinforcement learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [23] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of the uplink grant-free noma system," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6369–6379, 2020.
- [24] A. Emir, F. Kara, and H. Kaya, "Deep learning-based joint symbol detection for noma," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.
- [25] X. Miao, D. Guo, and X. Li, "Grant-Free NOMA With Device Activity Learning Using Long Short-Term Memory," *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 981–984, 2020.
- [26] A. Emir, F. Kara, H. Kaya, and H. Yanikomeroglu, "Deepmud: Multi-user detection for uplink grant-free noma iot networks via deep learning," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1133–1137, 2021.
- [27] W. Kim, Y. Ahn, and B. Shim, "Deep Neural Network-Based Active User Detection for Grant-Free NOMA Systems," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2143–2155, 2020.
- [28] J. Pan, N. Ye, and X. Li, "A Deep Learning-Aided Detection Method for FTN-Based NOMA," *Wireless Communication and Mobile Computing*, vol. 2020, 2020.
- [29] G. Tsoumakas and I. Katakis, "Multi-Label Classification: An Overview," *International Journal of Data Warehousing and Mining (IJDDWM)*, vol. 3, pp. 13–31, 2009.
- [30] C. Buciluă, R. Caruana, and A. Niculescu-Mizil, "Model Compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.
- [31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [32] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0893608089900208>
- [33] J. Berner, P. Grohs, G. Kutyniok, and P. Petersen, "The Modern Mathematics of Deep Learning," *arXiv e-prints*, p. arXiv:2105.04026, May 2021.
- [34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, pp. 533–536, 1986.
- [35] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv e-prints*, p. arXiv:1502.03167, Feb. 2015.
- [36] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: <https://proceedings.mlr.press/v15/glorot11a.html>
- [37] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.
- [38] I. Nusrat and S.-B. Jang, "A Comparison of Regularization Techniques in Deep Neural Networks," *Symmetry*, vol. 10, no. 11, 2018. [Online]. Available: <https://www.mdpi.com/2073-8994/10/11/648>
- [39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.
- [41] A. Graves, "Generating Sequences With Recurrent Neural Networks," *arXiv e-prints*, p. arXiv:1308.0850, Aug. 2013.
- [42] L. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [43] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. MIT Press, 1995. [Online]. Available: <https://proceedings.neurips.cc/paper/1994/file/b8c37e33defde51cf91e1e03e51657da-Paper.pdf>
- [44] W. Kim, Y. Ahn, and B. Shim, "Evolved Universal Terrestrial Radio Access (E-UTRA): Radio Frequency (RF) requirements for LTE Pico Node B," *ETSI TR 136 931*, vol. 9, no. 4, pp. 2143–2155, 2011.
- [45] R. Hoshyar, F. P. Wathan, and R. Tafazolli, "Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1616–1626, 2008.
- [46] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.
- [47] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [48] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of ADAM and Beyond," *arXiv preprint arXiv:1904.09237*, 2019.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.