Coordinates-based Resource Allocation Through Supervised Machine Learning

Sahar Imtiaz, *Student Member, IEEE,* Sebastian Schiessl, *Member, IEEE,*
Georgios P. Koudouridis, *Member, IEEE,* and James Gross, *Senior Member, IEEE.*

***Abstract*—Appropriate allocation of system resources is essential for meeting the increased user-traffic demands in the next generation wireless technologies. Traditionally, the system relies on channel state information (CSI) of the users for optimizing the resource allocation, which becomes costly for fast-varying channel conditions. In such cases, an estimate of the terminals' position information provides an alternative to estimating the channel condition. In this work, we propose a coordinates-based resource allocation scheme using supervised machine learning techniques, and investigate how efficiently this scheme performs in comparison to the traditional approach under various propagation conditions. We consider a simple system setup as a first step, where a single transmitter serves a single mobile user. The performance results show that the coordinates-based resource allocation scheme achieves a performance very close to the CSI-based scheme, even when the available user's coordinates are erroneous. The performance is quite consistent, especially when complex learning frameworks like random forest and neural network are used for resource allocation. In terms of applicability, a training time of about 4 s is required for coordinates-based resource allocation using random forest algorithm, and the appropriate resource allocation is predicted in less than 90 μs with a learnt model of size <1 kB.**

***Index Terms*—Wireless communication system, resource allocation, position information, machine learning.**

I. Introduction

Efficient allocation of system resources among the users of a network is the key to improved system performance. The resource allocation refers to, for example, the allocation of resource blocks in time or frequency domain, the allocation of transmit power, or modulation and coding scheme for payload transmission, etc. to a specific user in the system. The task of determining the appropriate resource allocation relies on the information about propagation environment acquired by the transmitter(s), as well as on the available computational power in the system. Traditionally, the users' channel state information (CSI) available at the transmitters is utilized for resource allocation, where heuristic approaches can be used to optimize the practical implementation [1]. However, the CSI acquisition on an instantaneous basis contributes to a performance overhead, especially for high mobility users. The use of limited feedback from users with 'good' channel conditions, as proposed in [2] and [3], is a beneficial and valid approach for scenarios where the user channels are uncorrelated, which is not the case for high mobility users [4], [5]. Consideration of user mobility implies channel correlation, which reduces the multi-user diversity gain and thus the performance advantage becomes significant asymptotically only when limited feedback is applied for a setup with large number of users [6]. In case full-band pilots are used for channel estimation, efficient resource allocation implies increased computational complexity. This motivates finding alternate approaches for resource allocation, such that the overhead for acquiring the propagation environment's information is minimal and at the same time, the required computational complexity for optimization is within certain constraints.

According to a recent survey [7], several research works in the last decade have focused on the idea of applying various machine learning frameworks for resource allocation in wireless systems. In addition to the related work mentioned in [7], the work in [8] applies deep neural network for cognitive radio modulation recognition. In [9], a convolutional neural network is used for successive position estimation of vehicles based on their past location data, and a genetic algorithm is designed for organizing services at the edge of the network. The authors in [10] combine unsupervised feature learning with supervised classification techniques to design an efficient quality-of-experience-based video admission control and resource allocation scheme. Deep reinforcement learning is used in [11] to propose a power-efficient resource allocation scheme for a wireless system based on cloud radio access network architecture. Another work in [12] applies deep reinforcement learning for flexible resource allocation to various radio access network slices. All these works highlight the advantages of exploiting machine learning for resource allocation in wireless systems, without compromising the computational capacity of the system.

Besides relying on CSI for optimizing the performance of wireless systems, various research works propose the use of position information in this context. The majority of these works discuss position-based schemes for improving

S. Imtiaz and J. Gross are with the Division of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 10044. (e-mail: sahari@kth.se, james.gross@ee.kth.se)

S. Schiessl was with the Division of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 10044. (e-mail: sebastian.schiessl9@gmail.com)

G. P. Koudouridis is with Wireless Systems Lab, Stockholm Research Center, Huawei Technologies Sweden AB, Kista, Stockholm, Sweden. (e-mail: george.koudouridis@huawei.com)

Manuscript received - November 19, 2020

the system performance at different layers of the protocol stack [13, & references therein], but only a handful of those consider position information for resource allocation. The position information can be acquired through narrow-band uplink pilots, or can be predicted based on the speed of a terminal moving along a trajectory, or estimated through camera-based sensors, etc. The work in [14] provides an overview of the position-based resource allocation to improve the performance of wireless systems, stressing the fact that the true potential of position-based techniques needs to be evaluated by comparison with the CSI-based methods. In [15], position-related information (i.e. angle-of-arrival) of mobile terminals is utilized for spatial filtering in an ultra-dense network to maximize throughput. A similar approach is used in [16], where the authors propose location-aided beamforming by exploiting the distance between the base station and the mobile relay for beam selection to serve high-speed users. Authors in [17] present an extension of the location dependent resource allocation scheme to the multicell deployments, where the terminals' locations are used to allocate the resource blocks efficiently between the primary and the underlay device-to-device (D2D) networks. The concept of spatial deep learning is proposed in [18], where the geographical locations of the transmitters and receivers are used for scheduling links in D2D network. Other works like [19] and [20] consider location-based resource allocation in conjunction with CSI. To the best of our knowledge, none of the previous works have considered resource allocation solely based on position coordinates of the mobile terminal. Furthermore, most of the above studies do not account for stochastic variations affecting the position-related information, rather assume that the position is perfectly known without considering estimation errors. Some initial investigation results in our recent work [21] highlight the feasibility of a coordinates-based resource allocation scheme. In this work, we address the main question of the associated implementation constraints.

In order to demonstrate the concept of coordinates-based resource allocation through machine learning, to show its feasibility and to investigate its limits, we focus on a simple system comprising a single transmitter serving a single mobile terminal, with dominant line-of-sight (LoS) communication link. This implies that the channel characteristics of the propagation environment of the mobile terminal are related to its position information. We apply a supervised machine learning framework to learn this relationship and design a coordinates-based resource allocation scheme that maximizes the transport capacity of the system. Our previous work [21] discussed a possible way to design a coordinates-based resource allocation scheme and showed that such a scheme can work well with a synthetic dataset formulation, where the dataset generation was done with a state-of-the-art channel simulator. In this work, we focus on the applicability aspects of a coordinates-based resource allocation scheme for realistic implementation in a communication system. For this purpose, we first discuss the design and working of the coordinates-based resource allocation scheme, followed by the possible choices for supervised learning framework that can be applied. Then we highlight the different possibilities of dataset formulation for the proposed scheme and their associated challenges. To evaluate the performance of the coordinates-base resource allocation scheme, we consider two scenarios for generating the user's positions; one with random drop and another with mobility model. The former is used for generic performance evaluation of the coordinates-based resource allocation scheme, where in addition to the LoS scenario, we also briefly discuss the performance of the proposed scheme when non-LoS (NLoS) link exists between the transmitter and the mobile terminal. The latter scenario is used to evaluate the feasibility of implementing the proposed scheme, where we focus on the various advantages and computational trade-offs when the coordinates-based resource allocation scheme is applied in a realistic system setup. More specifically, the main contributions of our work are as follows:

- This work uses the idea of a coordinates-based resource allocation scheme presented in [21], with a modified dataset formulation. In contrast to [21], we consider a dataset formulation with the output variable based on the signal-to-interference-and-noise ratio (SINR) of the base station-user link, which leads to a better learning performance of the machine learning frameworks.
- In addition to the supervised learning frameworks applied in [21], we evaluate the performance results for a learning model based on feed-forward neural network with the dataset formulation mentioned before. These results are evaluated for different stochastic variations in system characterization as well as for the case when terminal's position estimates are erroneous, assuming a random drop method for generating the terminal's position.
- The prime novel aspect of this work lies in the discussion around the implementation constraints of the coordinates-based resource allocation scheme. We present an analysis of the time required to train the proposed coordinates-based resource allocation scheme through machine learning. Particularly, we consider simulation data with mobility model, and determine the training time of the different machine learning frameworks necessary for the proposed scheme to achieve a system performance comparable to the CSI-based resource allocation scheme.

The rest of the paper is structured as follows: Section II describes the system model considered in this work, while Section III discusses the design of the proposed resource allocation scheme, along with the details of the supervised machine learning frameworks used in this work. Initial analysis of dataset, as well as the associated challenges in dataset formulation, are discussed in Section IV, with a description of the random drop method and mobility model used for generating the terminal's positions. Section V presents the results and relevant discussions for the random drop scenario, whereas the applicability aspects of the proposed scheme for dataset with mobility model scenario are discussed in Section VI. Section VII concludes the paper.

II. System Model

In this section we first describe the basic system model, followed by a formal definition of the resource allocation problem. This problem definition is based on an optimization function which aims at maximizing the transport capacity of the system.

A. Basic System Model

We focus on the downlink communication between a single base station (BS), and a single user terminal, where the BS is connected to the baseband processing unit, as shown in Fig. 1. We assume that the system operates on time frames of duration T_f [ms]. Orthogonal frequency division multiplex (OFDM) waveform is considered for payload transmission, with the BS utilizing a bandwidth W spanning a number of sub-carriers N. The BS is equipped with A_r transmit antennas, collectively operated with a constant transmit power denoted by P_r [W]. The user terminal is assumed to have A_u receive antennas.

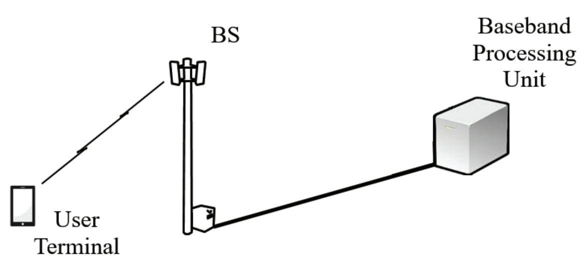

Fig. 1. The system model

We consider a time-varying wireless communication channel between the BS and the terminal, subject to pathloss, shadowing and fading. Let us denote by $\boldsymbol{H}(t,n)$ the multiple input, multiple output (MIMO) channel matrix between all transmit and receive antenna elements at time t and for sub-carrier n. We assume that the MIMO channel stays constant during a single time frame and that the user terminal is not exposed to any interference. With these assumptions in mind, if the BS applies a transmit beam $\boldsymbol{v}$ at time t to transmit the symbol $s(t,n)$ over sub-carrier n, the received signal at the terminal, when the terminal applies a receive filter $\boldsymbol{u}$, will be given by:

$$y(t,n) = \sqrt{P_\mathrm{r}} \cdot (\boldsymbol{u}(t))^\dagger \cdot \boldsymbol{H}(t,n) \cdot \boldsymbol{v}(t) \cdot s(t,n) + z, \quad (1)$$

where z represents the additive white Gaussian noise, and $(\boldsymbol{u}(t))^\dagger$ represents the Hermitian of $\boldsymbol{u}(t)$. Based on the received signal in (1), the signal-to-noise ratio (SNR) for the terminal served by BS at time t for sub-carrier n, with noise power σ_z^2, is given by:

$$\gamma(t,n) = \frac{P_\mathrm{r} \cdot |(\boldsymbol{u}(t))^\dagger \cdot \boldsymbol{H}(t,n) \cdot \boldsymbol{v}(t)|^2}{\sigma_z^2}. \quad (2)$$

For the transmission of payload to the receiver, the BS applies a modulation and coding scheme (MCS) m over all the N sub-carriers. The chosen MCS is applied uniformly for all N, meaning that the system does not feature adaptive MCS per sub-carrier. We assume a full-buffer state at the BS with respect to the terminal. Therefore, the payload transmission based on the choice of MCS carries a maximum possible number of bits, denoted by $b(m(t))$. Due to noise, the transmitted bits can be received erroneously at the terminal, which can be modelled by an error function ϵ. This results in transport capacity (or goodput) $\mathcal{T}(t)$ at the terminal, by which we measure the performance, given by:

$$\mathcal{T}(t) = (1-\epsilon) \cdot b(m(t)). \quad (3)$$

B. Resource Allocation and the Position Information

We consider the maximization of transport capacity as the optimization problem in this work. The transport capacity at the terminal depends on the allocation of available system resources between the BS-terminal pair. We denote the resource allocation by $\mathfrak{r}(t)$, which comprises the following: (a) the transmit beam $\boldsymbol{v}(t)$ applied by the BS, (b) the receive filter $\boldsymbol{u}(t)$ applied by the terminal, and (c) the MCS $m(t)$ chosen by the BS. The transmit beam $\boldsymbol{v}(t)$ and the receive filter $\boldsymbol{u}(t)$ is chosen from the finite sets $\mathbb{V}$ and $\mathbb{U}$, respectively. The sets $\mathbb{V}$ and $\mathbb{U}$ are pre-determined using geometric beamforming, with an angular separation $\theta°$, at both the BS and the user terminal, respectively. The MCS value $m(t)$ is applied from the finite set $\mathbb{M}$, that is based on the link-to-system mapping given in [22]. Since the downlink communication between a single BS and user terminal is the focus of our work, we allocate all the time and frequency resources, as well as the full transmit power, to the single terminal. Consequently, the optimization problem for resource allocation per downlink frame can be stated as:

$$\max_{\mathfrak{r}(t)} \mathcal{T}(t) = (1-\epsilon) \cdot b(m(t)), \quad (4)$$

where resource allocation $\mathfrak{r}(t)$ is the parameter of interest that maximizes the transport capacity $\mathcal{T}(t)$ of the system.

Traditionally, the resource allocation problem is solved based on the CSI of the BS-terminal pair, which requires the CSI to be perfectly known at the BS. This means that the system has to apply full-bandwidth signals, with frequent signaling, but in reality the signaling bandwidth is limited and signaling resources are scarce. The signaling overhead from frequent CSI estimation would drastically lower the data transmission rate, whereas less frequent CSI estimation would lead to outdated CSI. An outdated CSI results in a highly inefficient resource allocation and, consequently, a deterioration of the system performance. This is prevalent in scenarios with many users, and fast channel changes due to user mobility and high density of scatterers. To mitigate these problems, an alternate approach is needed for efficient resource allocation that maximizes the system performance.

In this work, we consider systems where an estimate of the terminal's position can be obtained at the BS in addition to CSI acquisition. This position estimate can be determined, for example, by Kalman filtering of the direction-of-arrival and time-of-arrival of the specifically sent positioning beacons in the uplink [15]. These positioning beacons are in fact narrow-band signals, which pose significantly lesser overhead compared to CSI estimation beacons for high user density scenarios, as mentioned in [23]. Without restricting our work to a specific method for position acquisition, we assume that an estimate of position coordinates of the terminal $\hat{\boldsymbol{p}}(t)$ is available at the BS in time slot t, whereas $\boldsymbol{p}(t)$ denote the true position coordinates of the terminal. In case a dominant LoS link exists between the BS and the terminal, with no interference exposed at the terminal, the channel characteristics remain fairly constant, and are related to the terminal's position estimate known at the BS which can be used for efficient resource allocation. However, this relationship is affected by the fact that the position of terminal can not be accurately known at the BS at all times, as well as by the presence of scatterers in the propagation area. In this work, we first investigate under which conditions a relationship between the position estimate of terminal and the channel state exists, and if so, how this relationship can be exploited to maximize $\mathcal{T}(t)$. This question forms the basis of our research problem, and is presented in the next sub-section.

C. Problem Statement

As mentioned before, maximum transport capacity $\mathcal{T}(t)$ is achieved when resource allocation is determined optimally based on the perfectly known CSI at the BS. However, depending on the propagation scenario, the instantaneous CSI acquisition can be costly or the available CSI estimates can be outdated, both of which are detrimental to CSI-based resource allocation. For the propagation scenarios where dominant LoS exists between the BS-terminal pair, the relationship between the estimated position coordinates of the terminal $\hat{\boldsymbol{p}}(t)$ and the downlink channel state can be exploited to determine the resource allocation for BS-terminal pair for solving the optimization problem (4). Intuitively, with the estimated position of the mobile terminal $\hat{\boldsymbol{p}}(t)$ available at the BS, geometric beamforming can be applied to determine the resource allocation $\mathfrak{r}(t)$. This means that the transmit beam $\boldsymbol{v}(t)$ and the receive filter $\boldsymbol{u}(t)$ are determined based on $\hat{\boldsymbol{p}}(t)$, whereas the MCS $m(t)$ can be determined based on the distance between the BS and the terminal. But this approach suffers from the inaccurate position information availability at the BS from time to time, in addition to being affected by the presence of scatterers in the propagation area. Furthermore, the geometric approach suffers from ignoring the antenna radiation profiles and the antenna orientation at both the BS and the terminal. In this work, we instead apply supervised machine learning to design a coordinates-based resource allocation scheme to solve the capacity maximization problem. In particular, we discuss how supervised machine learning can be used to determine the resource allocation $\mathfrak{r}(t)$ from terminal's position estimates $\hat{\boldsymbol{p}}(t)$, and how such a coordinates-based resource allocation scheme will be implemented in a wireless communication system. Furthermore, we will investigate the computational cost of the proposed scheme by focusing on the simulated data for realistic system implementation.

III. Coordinates-based Resource Allocation Using Machine Learning

In this section, we present the design and working of the coordinates-based resource allocation scheme. We then discuss the representation of the input and output variables for implementing the proposed scheme, along with the details of the different supervised machine learning algorithms used in our work, and the motivation for their choice.

A. Design and Working of the Proposed Scheme

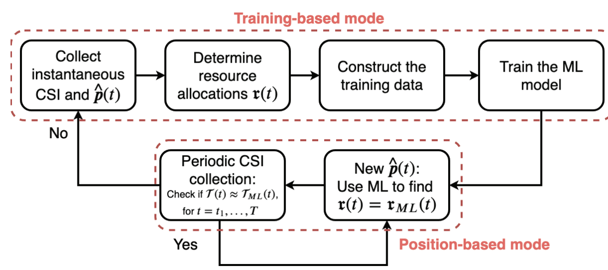

Fig. 2. Working of the coordinates-based resource allocation through supervised machine learning (ML)

We apply a supervised machine learning framework for designing the coordinates-based resource allocation scheme. This implies that the samples collected for training the machine learning framework will consist of input parameters and an output, which will be predicted by the learnt model. Fig. 2 shows the design and working of the coordinates-based resource allocation scheme. The scheme consists of two modes which exist in separate time durations, namely the training-based mode and the position-based mode. In the beginning, the system operates in the training-based mode, where the data collection process is performed simultaneously with the CSI-based resource allocation. In this mode, both the estimate of terminal's coordinates $\hat{\boldsymbol{p}}(t)$ as well as its CSI are collected by the system for a period of time to construct the training samples. This information is processed offline, where the CSI for each collected sample is used to determine the resource allocation $\mathfrak{r}(t)$ that maximizes the system's transport capacity $\mathcal{T}(t)$. Once all the samples are processed, the training dataset is formulated by associating each position estimate $\hat{\boldsymbol{p}}(t)$, the *input*, with the corresponding resource allocation $\mathfrak{r}(t)$, the *output*. These samples are used to train the supervised machine learning frameworks, and the corresponding learning models are then used for prediction by the system.

After the training process is complete, the system operates in position-based mode. In this mode, only the terminal's position coordinate for each time frame is assumed to be known. This estimate is passed to the trained machine learning model for determining the resource allocation based on the model's prediction. To ensure efficient performance, the predictions from the learnt model need to be checked with the baseline CSI-based resource allocation from time to time. In case the goodput computed by the predicted resource allocations is not inline with the one given by the CSI-based scheme, the system switches back to training-based mode to retrain the machine learning model. The exact modelling of this mode-switching process is out of the scope of this work. However, we provide an intuition about (a) the time needed to collect a sufficient amount of training samples, and (b) the training time of machine learning model for a stable learning performance, at the end of this paper.

B. *Selection of the Machine Learning Framework(s)*

Before discussing the selection of supervised machine learning frameworks applied for the proposed scheme, we discuss the input and output representation of the training dataset. Generally speaking, the input variable is the position of the mobile terminal, which can be represented in different ways: Considering a bigger picture, the propagation scenario can be treated as a binary-coded image, where the terminal's position estimate is marked with a 1, while the rest of the image is coded as 0's. Another possibility is to use the coordinates of the estimated terminal's position as the input vector. The former choice entails a highly sparse representation given the considered scenario, as it will indicate the position estimate of only a single terminal, implying that a huge amount of samples will be required for an appropriate learning model. On the other hand, coordinates-based input representation is more intuitive, albeit low dimensional, where the input variables can be stored as floating point numbers in the system. Inspired by this input representation, we choose to encode $\mathfrak{r}(t)$ as a decimal-base equivalent of the binary string stored by the system, where the different parts of the string encode the individual resource variables' information. Note that the chosen input and output representation implies a multi-class classification problem.

We now focus on the selection of the supervised machine learning frameworks. The supervised machine learning domain mainly comprises the following well-known algorithms, ranging from lowest to highest possible complexity: K-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), and neural network (NN). SVM is typically used for binary classification [24], and is, therefore, not applied in this work. Neural networks include a wide variety of structures, but to avoid the bias-variance trade-off with low-dimensional input vector representation [25], we resort to a simple feed-forward neural network in our work instead of a deep learning structure. In summary, we applied KNN, RF and feed-forward NN for the proposed scheme; the details of these learning frameworks are given below:

1) K-Nearest Neighbor Algorithm: KNN is the simplest machine learning framework [26] and does not build an explicit model for predicting the classes. Instead, the whole training dataset itself is used for class prediction. For a given sample of the test dataset, KNN first determines the K samples in the training data that are closest to the test data sample. Then it performs a majority vote on the class associated with the K nearest neighbors to predict the outcome for the given test data sample. In the context of coordinates-based resource allocation, given a training dataset with sufficient sampling of the terminal's coordinates, the KNN can capture the spatial relationship between the terminal's position coordinates and the respective resource allocation quite accurately.

2) Random Forest Algorithm: Random forest [27] is a complex supervised learning algorithm, which builds a learning model for class prediction, as opposed to KNN. The model consists of an ensemble of randomized binary decision trees, where each decision tree is constructed using a dataset consisting of samples taken from sampling with replacement on the available training data. An individual sample in the training data is called an input feature vector, and consists of the input features $\mathfrak{f}$ along with the output variable(s). Each decision tree in the forest consists of a root node, several interior nodes and terminal leaf nodes. The thresholds for each node are determined based on a subset of randomly selected input features $\mathfrak{f}'$, which induces randomness in each decision tree of RF. Overall, Ω_{t} trees are constructed, and each tree is either grown to a maximum depth of Ω_{d} or till all the classes are perfectly separated. The predicted class is based on a majority vote on the classes predicted by all trees in RF.

For the coordinates-based resource allocation, whenever a new estimate of terminal's position is available to RF, it is parsed through all trees in the forest. The resource allocation prediction is made by taking the mode of resource allocations, i.e. the classes, predicted by all trees in the forest. The ensemble of decision trees provides robustness to the learnt model, and therefore, RF is more robust to noisy inputs compared to other machine learning frameworks. This property makes the choice of RF even more attractive for the cases where noisy estimates of terminal's position coordinates are available for determining the resource allocation that maximize the system performance. The randomness introduced by random selection of input features for constructing an individual tree prevents the RF algorithm from over-fitting on the training dataset. Due to these reasons, RF algorithm performs better than KNN for the cases when the terminal's position estimates are erroneous or when the propagation scenario involves randomness in the channel between BS-terminal pair, as shown by the evaluation results in sections V and VI.

3) Neural Network: A feed-forward neural network consists of a number of layers of neurons (or nodes), where the output from each layer is the input to the successive layer

[28]. The first layer of neurons is called the input layer, followed by one or more hidden layers, terminating at the output layer. Each neuron is connected to all the neurons in the successive layer, and hence forms a fully-connected network. The basic operation at each of the hidden layers is given by a non-linear function applied to the weighted sum of the inputs. The non-linear function is also called an activation function, since the output from a given neuron is fed forward (or activated) only if it is above a certain threshold. The output for a given input sample is determined as the probability of belonging to a certain class, calculated by applying a non-linear function on the weighted sum of the neurons in the last hidden layer of the neural network. A detailed mathematical representation of feed-forward neural networks can be found in [24].

The prime purpose of training a neural network is to learn the weights applied by each neuron at each layer of NN. To speed up the training process, the input data is used in the form of small batches, where each batch input defines an iteration. Unlike other supervised learning frameworks, not all of the training samples are used for training the NN; instead, a small amount of samples are left out in training and are used to validate the performance of the trained NN. Once the accuracy of training and validation datasets starts converging, the training of NN is considered to be complete. Samples from the test dataset are then fed as input to the trained NN and the corresponding predicted classes are determined. For the proposed scheme, we reserve about 20% of the training samples for validating the trained NN, such that a sufficient number of samples is available for appropriately training the NN. The accuracy of the classes predicted for both the training and validation samples is calculated for several iterations, and finally the NN is trained for a fixed number of iterations for all the considered scenarios. The non-linear activation functions enable the neural networks to learn intricate input-output relationships, provided that sufficient number of training samples are available and are used to train the network for ample number of iterations. That is why NN performs better than the rest of the supervised learning frameworks, even though it requires more training time than the other frameworks, as discussed in the later sections of the paper. After selecting the supervised machine learning frameworks for the coordinates-based resource allocation scheme, we now focus on the dataset itself; its generation, formulation and basic analysis.

IV. Initial Analysis of the Dataset

In this section, we will first describe the propagation scenario considered in this work, along with the different models for simulation based on the system model described in Section II. Next we present the methodology for simulating the channel model to generate the dataset, and also the two scenarios considered for generating the user terminal's positions. Afterwards, we discuss the dataset formulation and its analysis for some baseline propagation scenarios.

TABLE I
Parameter Settings

Parameter	Value
Center frequency (f_c)	3.5 GHz
System bandwidth (W)	200 MHz
Number of BS antennas (A_r)	$\{8, 4\}$
Number of terminal antennas (A_u)	$\{2, 1\}$
Height of BS antennas (h_r)	10 m
Height of terminal antennas (h_u)	1.5 m
Transmit power per sub-carrier (P_r)	1μW
Transmission time interval (T_f)	0.2 ms
Number of sub-carriers (N)	833

A. Scenario Description

We consider a small street section of 6×25 m^2 with a single BS serving a single mobile terminal as the propagation scenario, shown in Fig. 3. The BS is placed at the right lower end of the street, 3 m off-roadside. We define a parameter ρ [/m^2] to specify the maximum density of scatterers in the considered setups. As an example, if $\rho \le 0.05$/m^2, up to 5 scatterers will be randomly placed in the propagation environment. The placement as well as the number of scatterers will vary for each random-drop of the terminal. The BS is equipped with $A_\mathrm{r} = \{8, 4\}$ transmit antennas, while the terminal has $A_\mathrm{u} = \{2, 1\}$ receive antennas. Each antenna element at both the BS and the terminal is a Hertzian dipole, and forms a uniform linear array oriented along the $x-$axis. The BS antenna elements are collectively operated with a power of $P_\mathrm{r} = 1\mu$W, and are placed at a height of 10 m from the ground. The terminal antennas are at a height of 1.5 m from the ground, which remains constant for all the simulation scenarios. The system operates on a center frequency f_c of 3.5 GHz over a bandwidth $W = 200$ MHz. The transmission time interval of the system is set to $T_\mathrm{f} =$ 0.2 ms. All the simulation parameters are listed in Table I.

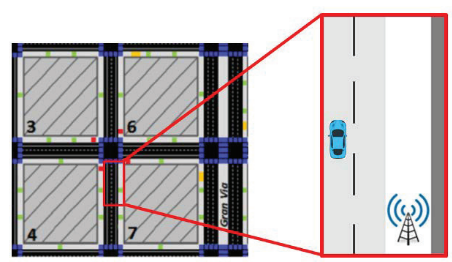

Fig. 3. The simulation scenario for generating random drop scenario's datasets.

To calculate the transport capacity $\mathcal{T}(t)$ in (4), we apply the link-to-system mapping error function for ϵ, $e(m(t), \gamma_{\text{eff}}(t))$, that emulates the erroneous reception of the transmitted bits at the receiver. Here, γ_{eff} is the exponential effective SNR mapping [29], which converts the SNR value $\gamma(t, n)$ per sub-carrier into an equivalent SNR over all the N sub-carriers, with respect to the considered communication scenario. The link-to-system mapping error function provides an error rate that is specific to a range of γ_{eff} values for a given MCS $m(t)$. Hence, the link-to-system model $e(m(t), \gamma_{\text{eff}}(t))$ represents the relationship of block error rate, effective SNR, MCS as well as transmitted payload size.

In terms of the system resources, the sets of transmit beams $\mathbb{V}$ and receive filters $\mathbb{U}$ are determined using geometric beamforming, with an angular separation of $\theta = 3°$ and $12°$, respectively. The finite set of MCS values $\mathbb{M}$ comprises 15 different values, and is based on the link-to-system mapping in [22]. The optimal resource allocation $\mathfrak{r}(t)$ for each terminal position is determined by solving Problem (4) through exhaustive search.

Another important parameter is the error in the position estimate for the system model presented in section II-A. The position estimation error $(\boldsymbol{p}(t) - \hat{\boldsymbol{p}}(t))$ is modelled as a Gaussian zero-mean random variable with variance σ^2. With respect to the communication system, this position error depends on the accurate estimation of the direction of arrival and the time of arrival parameters, where the former depends on the antennas' geometry, while the latter is related to the pathloss between the BS-terminal pair. In addition to the aforementioned modelling parameters, the utmost important choice is that of the channel model, which we present in the next sub-section.

B. The Channel Model

We resort to simulations for generating the datasets to determine the resource allocation using estimated position coordinates of the terminal to solve the optimization problem (4). To the best of our knowledge, only the received signal strength related information is available in the publicly accessible traces on various open-source platforms, with no details about the position information of the terminal. Therefore, one of the major challenges for dataset generation is to choose the simulation models that emulate the real-time measurements as closely as possible. Hence, we utilize the ray-tracer channel model [30] to generate the MIMO channel matrix $\boldsymbol{H}(t, n)$ for various parametrizations. This channel model has been validated for different propagation scenarios, as mentioned in [30], and is a state-of-the-art channel model for next-generation wireless communication systems.

The ray-tracer channel model considers a number of multipath components k existing in the downlink communication between the BS-terminal pair, for each time t and sub-carrier n. These multipath components arise due to different wave propagation phenomena, including reflection, diffraction and scattering, which are affected by the presence of scatterers in the propagation environment. The radiation patterns of the BS and terminal antennas are also taken into account by the ray-tracing model. We denote by $\tilde{h}_{k,a_\text{u},a_\text{r}}$ the impulse response for multipath component k, between each BS antenna element a_r and each terminal's antenna element a_u, which captures all the aforementioned propagation effects in addition to the relevant pathloss. The channel impulse response $H_{a_\text{u},a_\text{r}}(t, n) \in \boldsymbol{H}(t, n)$ is then the sum of the impulse responses of all the k different multipath components, i.e.

$$H_{a_\text{u},a_\text{r}}(t, n) = \sum_{k=1}^{K} \tilde{h}_{k,a_\text{u},a_\text{r}} \cdot \exp^{\frac{j2\pi d_k(t)}{\lambda}} \exp^{-j2\pi f_n \tau_{k,a_\text{u},a_\text{r}}(t)} . \tag{5}$$

Here, K is the total number of multipath components, λ is the wavelength corresponding to the center frequency f_c, and f_n is the frequency of the sub-carrier n. d_k is the total distance for multipath k at time t, and $\tau_{k,a_\text{u},a_\text{r}}$ denotes the delay for multipath k. A detailed implementation of this channel model can be found in [15].

The channel matrix $\boldsymbol{H}(t, n)$ is obtained by the above mentioned ray-tracer model for a set of user positions that are generated assuming two scenarios. For the initial evaluation results, we consider a *random drop scenario* where the mobile terminal is placed randomly over the street, with uniform distribution over the entire street section. For each user position generated using random drop, a single snapshot of the channel impulse response is obtained using (5), which is then used for dataset formulation. In essence, each channel snapshot obtained for the random drop scenario exhibits space and frequency correlation, while a dataset with multiple snapshots does not exhibit time correlation (as the snapshots do not relate to a time order). Thus, the resulting dataset for this scenario encapsulates the behavior of a perfect dataset where the information about every possible channel behavior is present, with the highest degree of freedom for user position sampling as well as for the scatterers' locations.

In contrast to that, a more realistic dataset would result from sample collection as the user terminal moves across the street. This is obtained from the *mobility model scenario*, where the mobile terminal moves with a certain speed in a linear trajectory and the placement of the scatterers remains unchanged during the course of the trajectory. In this scenario, time correlation between the samples for a single trajectory is implicit, and is in addition to the spatial and frequency correlation existing in the considered channel model. In the following, for basic evaluation of the proposed resource allocation scheme, we focus on the random drop scenario and will discuss the mobility model scenario towards the end of the paper.

C. Formulation of the Dataset

Based on the scenario description and the choice of the channel model, we define the following baseline cases to generate the primary datasets, for $A_\text{r} \times A_\text{u} = 8 \times 2$ MIMO system:

- Case 1: When no scatterers are present in the propagation environment ($\rho = 0/\text{m}^2$) and accurate position estimates for the terminal are available. This represents a deterministic channel generation, i.e. the channel between a given terminal position and BS always results in the same $\boldsymbol{H}(t, n)$.

- Case 2: When erroneous position estimates are available (specifically, $\sigma = 0.4$ m) with no scatterers in the propagation scenario ($\rho = 0/\text{m}^2$).
- Case 3: The position estimates of the terminal are known accurately for $\rho \leq 0.05/\text{m}^2$ that are randomly placed in the propagation environment.

We use exhaustive search, performed offline, to determine the optimal resource allocation $\mathfrak{r}(t)$ for each estimate of terminal's coordinates $\hat{\boldsymbol{p}}(t)$ in the generated dataset. Depending on the propagation conditions, for the system model specified earlier, multiple resource allocations can yield the same transport capacity value, which is optimal for the BS-terminal pair during time t. As an example, this can occur when different placements of scatterers result in a range of transmission rates that lead to similar error probability for payload transmission. Therefore, for certain scenarios, a vector $\boldsymbol{\mathfrak{r}}(t)$ results in an optimal $\mathcal{T}$ value at time t, instead of a scalar $\mathfrak{r}(t)$. Using all the optimal resource allocations determined by the exhaustive search as the output variable will result in an unconventional dataset formulation for supervised learning frameworks. In our previous work [21], we initially resolved this issue by considering only the first optimal solution for capacity maximization problem using exhaustive search to form the dataset. This guarantees a unique input-output association in the dataset formulation. From system point of view, the maximum transport capacity for a single BS-terminal link is achieved when γ_{eff} is maximum. Therefore, in this work, we consider only the resource allocation of the transport capacity maximization problem that relates to the highest value of γ_{eff} for formulating the dataset. Hence each input variable $\hat{\boldsymbol{p}}(t)$ in the dataset is associated with a unique output $\mathfrak{r}(t)$ corresponding to the highest value of γ_{eff} that maximizes $\mathcal{T}(t)$.

In terms of system implementation, the dataset generation for initial analysis of each case specified above is performed by considering the random drop scenario, where an estimate of user position $\hat{\boldsymbol{p}}(t)$ as well as its CSI is collected after certain time intervals over a long period of time. For a given user position, the channel matrix is obtained through the ray-tracer channel model, which is then used to compute the SNR and the effective SNR values based on the three resource allocation variables, i.e. the transmit beam $\boldsymbol{v}(t)$, the receive filter $\boldsymbol{u}(t)$ and the MCS $m(t)$. In the formulated dataset, each sample $\boldsymbol{d}_i$ consists of an input vector and an output value. The input vector comprises the $x-$ and $y-$coordinates of the terminal's position estimate $\hat{\boldsymbol{p}}(t)$ (the $z-$coordinate is not used due to the same receive antenna height assumption for all the samples). The output value for $\mathfrak{r}(t)$ is a number denoting the binary sequence, where the sequence encodes the index of $\boldsymbol{v}$, $\boldsymbol{u}$ and m corresponding to the resource allocation. Here, a resource allocation denotes a class to be learnt by the learning algorithm. A collection of such $\boldsymbol{d}_i$ samples constitutes the whole dataset $\boldsymbol{D}$, which is then divided into a training dataset $\boldsymbol{D}'$ and a test dataset $\boldsymbol{D}''$ for applying machine learning.

The above process is repeated multiple times by setting the system parametrization variables differently, to get various datasets for different cases, with the assumption of accurate user position availability. For the case of erroneous position estimates, the inputs in the datasets for accurate user position are replaced by erroneous position estimates modelled by a zero-mean Gaussian with specific variance σ^2, while the outputs are kept the same as the ones in the accurate position dataset.

D. Analysis of the Dataset

After generating the datasets for various cases considering random drop scenario, we analyze their input-output associations to have an intuition about the learning performance for the specified propagation scenario. A total of $I = 125{,}000$ position estimates are generated for a dataset for each of the three cases mentioned in Section IV-C. Two thirds of these position estimates are used for constructing training datasets $\boldsymbol{D}'$ and their analysis is presented here, while the rest of the samples are used for test dataset construction. (Note that for the analysis, we do not make a distinction between the training and validation datasets; this will only be applied for training the feed-forward neural network.)

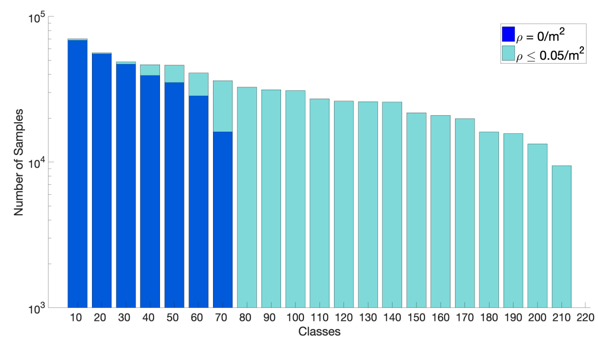

Fig. 4. Distribution of the number of samples per class in training dataset $\boldsymbol{D}'$, for (a) case 1 (ρ = 0/m^2), and (b) case 3 ($\rho \leq 0.05/\text{m}^2$), for 8×2 MIMO system.

We start by presenting the number of samples per class (a class represents a unique resource allocation) distribution for the generated datasets. Note that the number of samples per class are the same for case 1 and 2, since they only differ in the input values. Fig. 4 shows the distribution of number of samples per class for case 1 and 3 (i.e. with $\rho = 0/\text{m}^2$ and $\rho \leq 0.05/\text{m}^2$, respectively), where the x-axis is batched in groups of ten classes for illustration purpose. We observe an equitable distribution of the number of samples per class for both the cases. In addition, the number of classes for case 1 with deterministic channel generation (70) is almost $3\times$ lesser than that for case 3 (209), though being significantly smaller than the total number of generated data samples (125,000). The increased number of classes for case 3 is due to the fact that more scatterers result in more multipath components, which vary with scatterers' placement, introducing more randomness in the channel. This means that for a fixed user position with different placement of the scatterers, the channel response varies, and therefore, different resource allocations corresponding to the highest effective SNR are obtained from the exhaustive search.

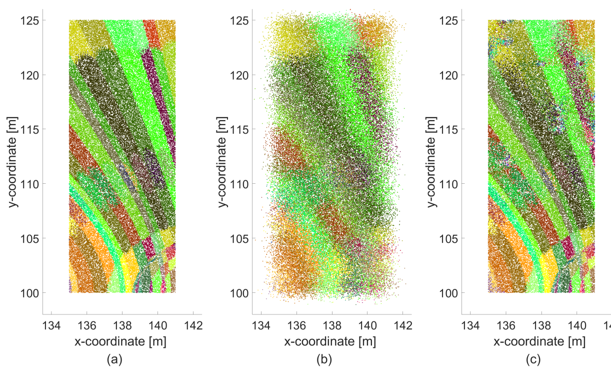

Fig. 5. Distribution of the classes with respect to user positions in $\boldsymbol{D}'$, for (a) case 1: ρ = 0/m^2, $\sigma = 0$ m, (b) case 2: ρ = 0/m^2, $\sigma = 0.4$ m, and (c) case 3: $\rho \leq 0.05$/m^2, $\sigma = 0$ m, for 8×2 MIMO system.

We now analyze the distribution of classes in relation to the input parameters. Fig. 5 shows the distribution of classes with respect to the terminal position estimates for cases 1, 2 and 3. Note that each resource allocation is depicted by a unique color, which is consistent across all the cases. For case 1, i.e. the deterministic channel, we see that the different classes are distinctly separated with clearly defined boundaries. These class boundaries become dispersed due to the erroneous position estimates for case 2, posing a challenge for the learning algorithms in terms of robustness. Considering the channel characteristics, we compare Fig. 5(a) and (c) and notice that the random effects in the propagation scenario do not impact the dataset as severely as the erroneous position estimates do. Though the number of classes in case 3 is thrice as that in case 1, the class boundaries are still distinctly separated across the street section. Therefore, it is expected that the performance of the learning algorithm will be affected primarily by the erroneous position estimates, compared to the presence of random scatterers in the propagation scenario. In general, the analysis of the dataset for different propagation scenarios with random drop consideration for terminal's position estimates shows that strong spatial clustering exists, which supports the feasibility of coordinates-based resource allocation through supervised machine learning frameworks.

V. Random Drop Scenario's Evaluation Results and Discussion

We now evaluate the applicability of coordinates-based resource allocation (RA) under different propagation conditions and system constraints for the random drop scenario. Based on the scenario description in Sections IV-A and IV-B, the performance of the following schemes is evaluated:

- **CSI-based RA scheme:** This represents the traditionally used RA scheme which relies on the instantaneous CSI of the BS-terminal pair.
- **Coordinates-based RA scheme using NN:** This scheme applies feed-forward NN for learning the training dataset. The details about the structure of NN and parameter settings are presented in Section V-A.
- **Coordinates-based RA scheme using RF:** As discussed in Section III-B, this scheme uses the random forest algorithm for learning the training dataset. The details for selecting the different parameters of RF model are given in Section V-A.
- **Coordinates-based RA scheme using KNN:** This is the simplest machine learning-based scheme, as mentioned in Section III-B, where we consider K = 1.
- **Geometric-based RA scheme:** This is the benchmark scheme, where the transmit beam and receiver filter are chosen from the geometric beamforming sets $\mathbb{V}$ and $\mathbb{U}$, respectively, based on $\hat{\boldsymbol{p}}(t)$. The MCS is determined statistically based on the terminal's position coordinates in relation to the geometry of propagation scenario.

The above performance evaluation is done for different channel characterizations: When no scatterers are present in the propagation environment, or when a number of scatterers up to 5 per 100 m^2 are randomly placed in the propagation environment, and accurate position estimates for the terminal are available to the system. Recall from Section IV-C that the former case represents a deterministic channel generation, while the latter refers to a varying channel generation case. The performance comparison is also done for a fairly deterministic channel generation, where the scatterer density ρ varies up to 1 scatterer per 100 m^2. To investigate the performance limit of the proposed coordinates-based RA scheme, we consider different degrees of variation in the error associated with the estimated coordinates of the terminal. This variation is defined by $\sigma = 0, 0.25, 0.4$ and 1 m. We now we discuss the structure of NN and its parameters' setting, as well as the tuning of RF algorithm for generating the results related to various propagation conditions for the random drop scenario, followed by the performance results and relevant discussions. At the end, we discuss the coordinates-based RA scheme in the context of a realistic system setup, i.e. for mobility model scenario, and also comment on the computational time needed for its implementation.

A. Details of the Parameter Settings for the Supervised Learning Frameworks

Correct parametrization of a learning model is the key to optimize its performance. We apply KNN as the baseline supervised learning framework, and therefore, do not optimize for the number of neighbors, K, and fix this as K = 1. To optimize the performance of RF, we need to tune its parameters. Specifically, we need to decide on the number of trees Ω_t that make up the forest as well as the maximum depth Ω_d up to which each tree has to be built while training the model. In terms of the number of randomly selected input features $\mathbf{f}'$ for building each node of the tree, we resort to the conventional practice, i.e. we choose $\mathbf{f}' = \sqrt{\mathbf{f}}$. Fig. 6

TABLE II
PARAMETER SETTINGS FOR THE FEED-FORWARD NN

Hidden Layer	Number of Nodes	Weight Initialization	Bias Initialization	Activation Function
1	10	$\mathcal{N}(0, 0.1)$	Zeros	ReLU
2	100	$\mathcal{N}(0, 0.1)$	Zeros	ReLU
3	200	$\mathcal{N}(0, 0.1)$	Zeros	ReLU

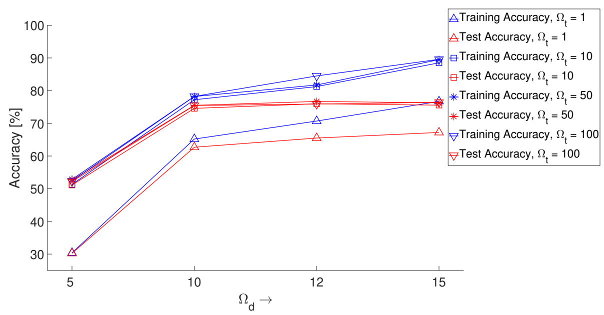

Fig. 6. Accuracy results for different parametrization of Random Forest.

shows the training and test accuracy obtained for different number of trees, at varying maximum depth per tree, for RF. These tuning results are shown for the maximum scatterers' density, i.e. $\rho \leq 0.05/\text{m}^2$. Comparing the two accuracy metrics in Fig. 6, we observe that the depth of trees has the most impact on the learning performance. A shallow tree depth, such as $\Omega_\text{d} = 5$, is not sufficient to learn the relationship between $\hat{\boldsymbol{p}}(t)$ and $\mathfrak{r}(t)$, but the depth can be increased up to a certain extent, such as $\Omega_\text{d} = 15$, to prevent the model to over-fit the training dataset. In addition, the performance does not seem to be affected by the number of trees, as long as it is sufficiently high. A higher number of trees introduces variance in the learnt model, which means that the model can learn various classes even with a small number of associated training samples. Note that even for higher number of trees, the model requires only a little more training time and storage memory by the system. Based on the above observations, we choose RF parametrization for random drop scenario to be $(\Omega_\text{t}, \Omega_\text{d}) = (100, 15)$.

For designing the feed-forward NN, two things need to be selected for the basic structure: One being the number of hidden layers and the other being the number of nodes in each hidden layer. The analysis of the dataset for the base-case scenarios in Section IV-D reveals intricate non-linear relationship between the input and output variables, therefore, one or two hidden layers are not sufficient for learning. In addition, the input vector is low dimensional as discussed in Section III-B, so deep learning model will not be a good choice. We design the NN with three hidden layers, where the number of nodes in each layer is selected as given in Table II. Each layer uses rectified linear unit (ReLU) as the activation function, except the output layer where softmax function is used for normalizing the prediction probabilities for each class. Adam optimizer is used to train the network, with an exponentially decaying learning rate after every 20,000 iterations, with an initial learning rate of 0.002. The network is trained using batch normalization, with a batch size of 100, where 20 epochs of training are applied having 10,000 iterations in each training epoch. The detailed parameter settings for NN used for random drop scenario are given in Table II. The designed NN achieves a test accuracy of 75.97% for the dataset representing case 3, i.e. when $\rho \leq 0.05/\text{m}^2$.

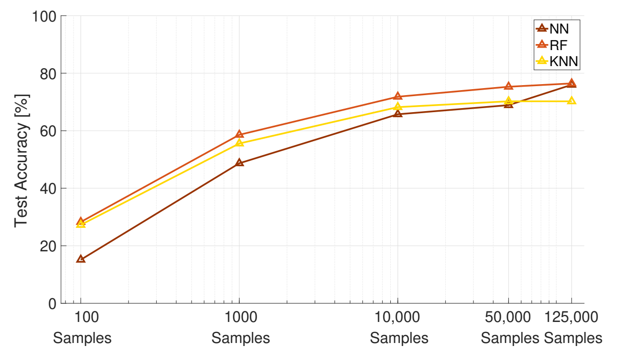

Fig. 7. Variation in test accuracy of the learning frameworks for different number of samples in the dataset.

Another important factor to consider while training a learning model is to determine the number of training samples required to achieve a reasonable performance. Fig. 7 shows the test accuracy of the learning frameworks for different number of samples in the overall dataset. The results show that the test accuracy almost saturates for 10,000 samples in the dataset, for all the learning frameworks. With very small number of samples, KNN performs equally well as RF does, but as the number of samples increase, RF consistently shows comparable performance to KNN. The advantage of random selection with replacement is not beneficial for RF when very small number of samples are available, but with increased number of samples, RF can achieve up to 10% better accuracy than KNN. NN relies on the number of samples available for training and hence performs the best when the maximum possible number of samples are used to train the model. Overall, the test accuracy is the highest for a total of 125,000 samples in the dataset, and that is why, we evaluate the performance of all the schemes for a dataset size of 125,000 samples in the next sub-section.

B. Performance Results and Discussion

We start with the performance comparison between the different RA schemes, shown in Fig. 8, when the number of scatterers varies in the propagation environment. Here,

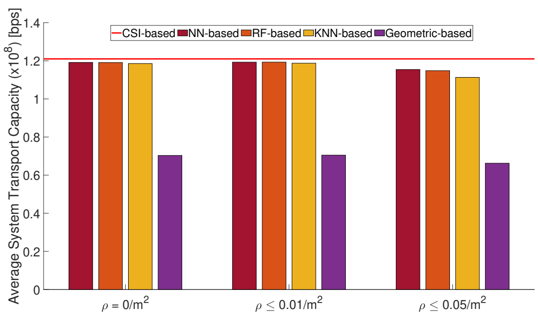

Fig. 8. Average transport capacity for 8×2 MIMO system, with accurate position estimates for different density of scatterers.

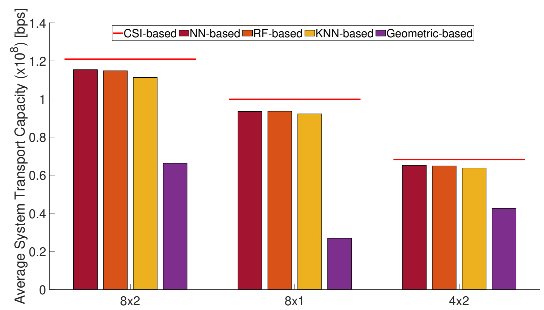

Fig. 10. Average transport capacity for different antenna configurations, with accurate position estimates for $\rho \leq 0.05/\text{m}^2$.

the position estimates of the terminals are assumed to be accurately known to the system. The results show that all the learning-based RA schemes are robust to the variation in scatterers' density compared to the CSI-based RA scheme, where a higher scatterers' density of $\rho \leq 0.05/\text{m}^2$ leads to a performance difference of about 5% compared to $\rho = 0/\text{m}^2$, when the channel is deterministic. NN-based scheme, owing to the complex learning model, shows slightly better performance than RF-based RA scheme. RF-based RA scheme performs better than KNN-based scheme due to the inherent randomness in random trees that constitute RF, and thus can cater for the random channel behavior due to randomized scatterers' placement, typically for the case of $\rho \leq 0.05/\text{m}^2$. In terms of overall comparison, the proposed coordinates-based RA schemes achieve a transport capacity almost twice as much as the one achieved by the benchmark geometric-based RA scheme. The geometric-based RA scheme relies on only the position estimate of the terminal to determine the resource allocation, disregarding the presence of scatterers in the propagation environment, and thus suffers from deteriorated performance. In general, the coordinates-based RA scheme using machine learning can be applied for determining appropriate resource allocations under favorable propagation scenarios, without relying on CSI-collection. We now discuss the impact on the performance of the proposed scheme when either different antenna configurations are considered or when erroneous position estimates are available in the system.

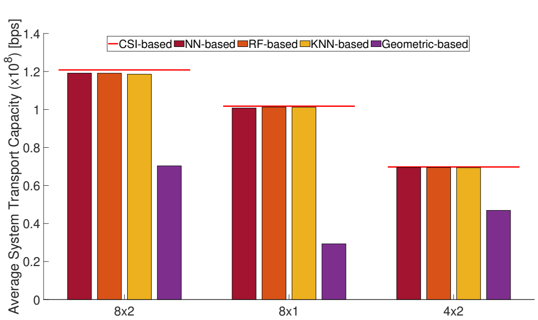

Fig. 9. Average transport capacity for different antenna configurations, with accurate position estimates for $\rho = 0/\text{m}^2$.

Fig. 9 shows the average transport capacity for different antenna configurations when no scatterers are present in the propagation environment, while Fig. 10 shows the same when the scatterer density varies up to $0.05/\text{m}^2$. The first observation is that the average system capacity drops by 50% when the number of transmit antennas are reduced by half, due to the wider beam pattern which is a consequence of reduced number of antennas. The transport capacity decreases also when the number of receive antennas is reduced to one, since no receive beamforming can be applied to enhance the received power for a given position of the terminal. The performance of the proposed scheme, however, is not affected by the antenna configuration in general. For the case with no scatterers, as shown in Fig. 9, the coordinates-based RA scheme performs on par with the CSI-based scheme for all antenna configurations, whereas the scheme performs consistently well when the scatterers' density varies for any antenna configuration (Fig. 10).

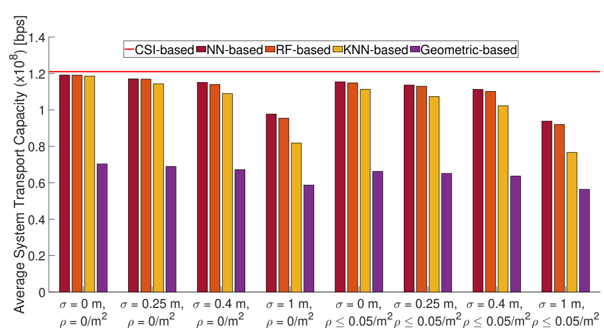

Fig. 11. Average transport capacity for 8×2 MIMO system with different position inaccuracies for $\rho = 0/\text{m}^2$, and $\rho \leq 0.05/\text{m}^2$.

Another important aspect of investigation relates to the accuracy of the acquired position estimates of the terminals. Fig. 11 shows the average transport capacity of the system when the position estimates are known with varying error margins, for both the deterministic channel case as well as the randomly varying channel. Both NN- and RF-based RA schemes are very robust to the different degrees of error in the acquired position estimates, compared to KNN-based RA scheme, because of the inherent randomness in the respective trained learning models. A performance loss of about one-fifth of transport capacity is observed when the acquired

TABLE III
PERFORMANCE COMPARISON BETWEEN LOS AND NLOS SCENARIOS FOR PERFECT POSITION INFORMATION

Comparison Scheme	LoS Scenario	NLoS Scenario
CSI-based RA scheme	1.2096×10^8 [bps]	0.18366×10^8 [bps]
NN-based RA scheme	1.1542×10^8 [bps]	0.13203×10^8 [bps]
RF-based RA scheme	1.1483×10^8 [bps]	0.1315×10^8 [bps]
KNN-based RA scheme	1.113×10^8 [bps]	0.12006×10^8 [bps]
Geometric-based RA scheme	0.66256×10^8 [bps]	0.0347×10^8 [bps]

position estimates are highly erroneous, i.e. $\sigma = 1$ m, for NN- and RF-based RA schemes, but is consistent for an error $\sigma \leq 0.5$ m. KNN-based RA scheme also performs consistently well for position estimates having an error up to $\sigma \leq 0.5$ m, but results in a performance loss of one-third of the transport capacity compared to the CSI-based scheme when $\sigma = 1$ m. The geometric-based RA scheme shows the poorest performance, with a performance loss of about 40% which increases to 50% as the error in position estimates increases to 1 m. We also investigated the performance for extreme error in the acquired position estimates, for up to 5 m, where a consistent performance loss is observed for all RA schemes irrespective of the scatterers' density ρ, as shown in Fig. 12. Here again we observe a substantial performance gain over geometric-based RA scheme even for a propagation scenario with high density of scatterers. All these results clearly show the efficacy of applying the coordinates-based RA scheme that learns the relationship between the terminal position and the appropriate resource allocation, instead of applying the naïve geometric-based resource allocation, for maximizing the performance of the system.

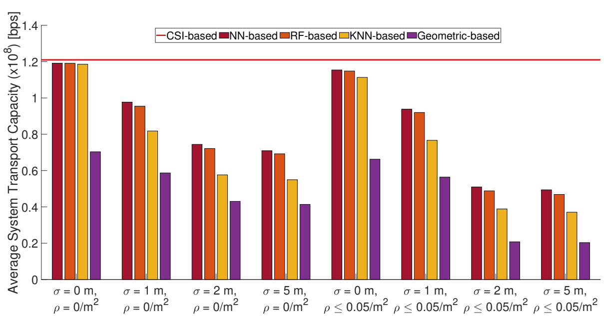

Fig. 12. Average transport capacity for 8×2 MIMO system with greater position inaccuracies for $\rho = 0/\text{m}^2$, and $\rho \leq 0.05/\text{m}^2$.

Although the proposed scheme is designed keeping in mind the relationship between the terminal's position and the channel characteristics when dominant LoS exists, we also investigated the performance of coordinates-based RA scheme for random drop scenario when NLoS exist between the BS and the user terminal. For this purpose, we considered a different simulation ssetup where the user terminal is obstructed by a building to ensure NLoS link with the BS at all times, as shown in Fig. 13. For fair comparison of results, we generated the same amount of samples as used in the LoS case, i.e. 125,000, for dataset formulation.

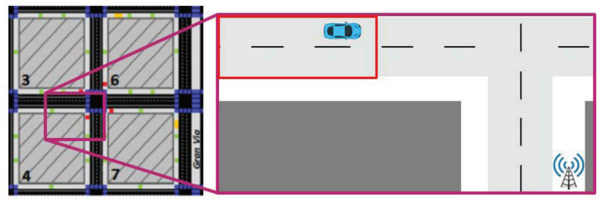

Fig. 13. The simulation setup for NLoS setting, with $\rho \leq 0.05/\text{m}^2$. The simulation area is highlighted with red box.

The performance results for all the comparison schemes with NLoS consideration are shown in Fig. 14. Here, the main observation is a drastic reduction in the average transport capacity of the system compared to the LoS case, as highlighted in Table III. In terms of the comparison schemes, we see a performance loss of about 30-35% for the coordinates-based RA scheme compared to the CSI-based scheme, which is not huge considering that the learning frameworks are provided only the user's position coordinates to estimate the channel characteristics with no dominant LoS path for appropriate resource allocation. This also shows that no link adaptation (i.e. same MCS allocation for all n sub-carriers) is playing in favor of the coordinates-based RA scheme, even when NLoS case is considered for training the machine learning model. And overall, the performance is quite stable when various position inaccuracies are considered and is significantly better than that of the geometric-based RA scheme.

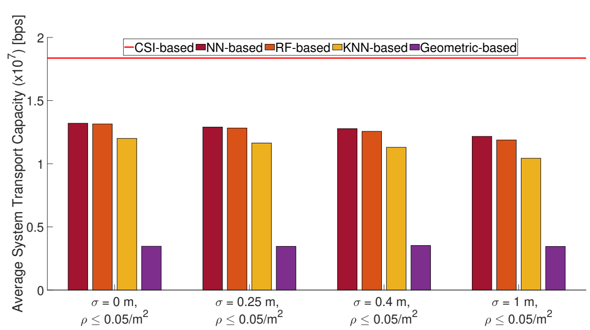

Fig. 14. Average transport capacity for 8×2 MIMO system, with NLoS link, with different position inaccuracies for $\rho \leq 0.05/\text{m}^2$.

In general, the coordinates-based RA scheme using machine learning performs at par with the legacy CSI-based scheme and is robust to the randomness introduced by the presence of scatterers for favorable propagation environment. The proposed scheme is also not affected by the considered antenna configuration and is quite robust to the erroneous position estimates acquired by the system, unless the position estimates are highly erroneous. All these observations are based on a sizeable amount of data acquired for samples with random user drop. What remains to be seen is the applicability of the coordinates-based RA scheme when the data samples are correlated for mobility model scenario, which we discuss in the next section.

VI. Mobility Model Scenario's Evaluation Results and Discussion

After observing the feasibility of the coordinates-based RA scheme through machine learning on the random drop scenario's datasets, we now evaluate its performance on a dataset for mobility model scenario, as discussed in Section IV-B, that captures the characteristics of a real-time environment. In the mobility model scenario, the data samples collected during the training-based mode of the proposed RA scheme (see Fig. 2) are gathered on a continual basis, implying that the collected samples have correlated channels associated with the terminal's position estimates $\hat{\boldsymbol{p}}(t)$. These samples are then used to train the machine learning model, which is used for predicting $\mathfrak{r}(t)$ for a newly available position estimate to the system during the operation in position-based mode. In this case, we are interested in the following key aspects regarding the applicability of the proposed RA scheme: (a) What impact does correlation between the training samples have on the performance of the proposed scheme? Do we need many more samples, acquired over a longer period of time, to appropriately train the learning models? and (b) how long does it take to build the RA prediction model and subsequently, how quickly does the model predict $\mathfrak{r}(t)$ for a new $\hat{\boldsymbol{p}}(t)$?

We resort to simulation-based setup, with the main idea of selecting a channel model that captures realistic channel behavior as accurately as possible. As mentioned before, the ray-tracer based METIS channel model [30] has been validated for different propagation scenarios and is the state-of-the-art channel model available to date. Therefore, we use this channel model for the small street section, with the same system parametrization as considered in all the other experiments mentioned before. Instead of using random-drop, the terminal follows a mobility model, with movement along a straight line across the street, so that the collected samples have correlated channel. The starting position of the terminal is generated randomly, and the subsequent samples are collected by updating only the $y-$coordinates of the terminal's position. We call the movement of the terminal along the street as a trace, and collect several traces for generating the dataset. Each sample is collected after a time period of 1 ms in a single trace, and a collection of these samples is then used to construct the training dataset according to the dataset formulation outlined in Section IV-C. To model a real-time environment, we assume the scatterers' density to be $\rho \leq 0.05/\text{m}^2$, where the number of scatterers as well as their placement varies independently

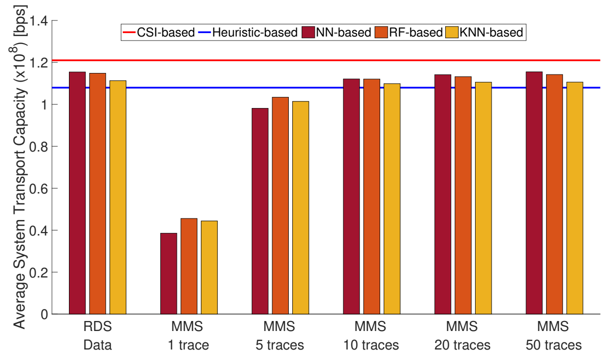

Fig. 15. System performance for 8×2 MIMO system with datasets for random drop scenario (RDS) and mobility model scenario (MMS), for $\rho \leq 0.05/\text{m}^2$ with perfect position information. The number of traces in the mobility model varies from 1 to 50.

over each trace. Overall, 50 traces were generated to have the training dataset size comparable to that of the random drop dataset, for fair evaluation. The same NN structure is used for learning in this case. For RF model, we use $(\Omega_\text{t}, \Omega_\text{d}) = (50, 12)$, chosen after performance comparison for various parametrization settings.

To evaluate the performance of the trained learning models for mobility model scenario, we use the test dataset $\boldsymbol{D}''$ for random drop scenario, to emulate the real-time data acquisition when the system operates in the position-based mode. We compute the average system transport capacity for the proposed coordinates-based RA scheme with KNN, RF and NN models, and compare it to the one obtained for the CSI-based scheme as well as a limited CSI heuristic-based RA scheme. The latter scheme is implemented using the idea proposed in [31], along with limited CSI feedback. In [31], the authors propose a two-phase beam search approach as an alternate to the exhaustive search approach in a MIMO system. Based on the proof of Lemma 1 in [31], we first select the transmit beam $\boldsymbol{v}(t)$ and the receive filter $\boldsymbol{u}(t)$ using the two-phase search, and then apply the MCS $m(t)$ corresponding to the payload size $b(m(t))$ that maximizes the transport capacity $\mathcal{T}(t)$. For limited CSI feedback, we assume that the CSI is updated in the system after about 50 transmission time intervals have passed. This results in an application shift of the resource allocation (derived from a mobility model) of 0.75 m. The transport capacity calculated using this application shift is averaged over the number of samples in $\boldsymbol{D}''$, and is labelled as heuristic-based RA scheme in the comparison results presented afterwards.

A. Impact of the Mobility Model on the Performance

Fig. 15 shows the performance results for the various RA schemes when either random drop scenario with uncorrelated samples are used for training or when different number of traces in the mobility model scenario's dataset are used for training the machine learning models. The heuristic-based scheme being a practical but a sub-optimal approach represents a realistic performance limit for the proposed coordinates-based RA scheme based on any ML model. Note that the results of the heuristic scheme are rather pessimistic, the reason being that the 0.75 m shift is the maximum shift a terminal experiences, but typically it is smaller. We observe that a small number of traces are sufficient for generating the training dataset to achieve a performance comparable to the CSI-based RA scheme, and can potentially outperform the heuristic-based scheme by a slight margin. Specifically, training dataset size of ∼10,000 samples, collected over a period of about 17 seconds, is capable of achieving a performance very close to the theoretical upper bound and the heuristic approach. Though NN achieves a better performance than that of RF-based scheme, but it requires about 250k bytes of memory storage, owing to the complex structure, compared to RF model. RF model, on the other hand, needs only 937 bytes of memory storage for predicting the resource allocation $\mathfrak{r}(t)$. This is a surprising result: In terms of applicability, the proposed coordinates-based scheme needs only a couple of seconds to collect the training data, and provides a system performance very close to the CSI-based scheme with only a small-sized learnt model.

Efficient data collection process is vital for implementing the proposed coordinates-based RA scheme. This means that we also need to determine how frequently the training samples need to be acquired by the system. We apply different rates of undersampling on the mobility model scenario data with 10 traces, to see how stable is the performance when fewer number of samples are available to train the machine learning frameworks. Fig. 16 shows the average system transport capacity obtained on the random drop scenarios' test samples when the machine learning frameworks are trained with datasets of decreasing sample size. The results show that all learning-based RA schemes have stable performance, unless extreme rate of undersampling is applied. NN-based RA scheme shows a stronger deterioration in performance compared to RF- and KNN-based schemes, because of the small number of training samples that are insufficient for appropriately training the NN model. An important observation here is that the performance of RF is at par with KNN even when the undersampling rate is as small as 10 ms. However, as the extent of undersampling increases, KNN shows a slightly better performance than RF. This happens because the RF model looses the generalization property due to lack of sufficient data for accurate classification.

All of these results for mobility model scenario's datasets are generated with the assumption that the terminal position is accurately known by the system. But in reality, the estimated position is inaccurate, involving some degree of error. We now assume that the acquired terminal's position estimates are erroneous, with the error modelled as a zero-mean Gaussian, with $\sigma = 0.4$ m. Fig. 17 shows the resulting performance when erroneous positions are used for training the machine learning frameworks using 10 traces in the dataset for mobility model scenario, with different rates of undersampling. The performance is quite robust, even

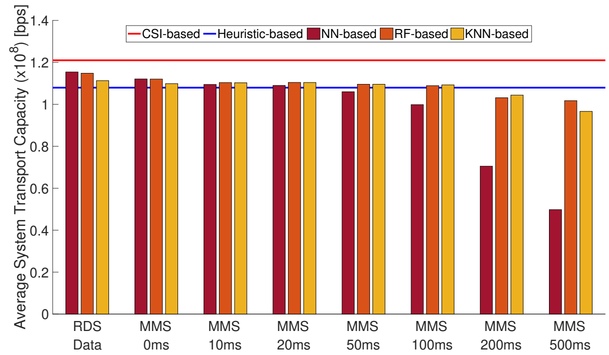

Fig. 16. System performance for 8×2 MIMO system with datasets for random drop scenario (RDS) and mobility model scenario (MMS) having 10 traces for $\rho \leq 0.05/\mathrm{m}^2$ with perfect position information. The undersampling in the mobility model scenario's data varies from 0 to 500 ms.

when a small undersampling rate is applied, compared to the CSI-based and heuristic-based RA schemes. As with the perfect positions' dataset, NN-based scheme shows stronger deterioration in performance with more and more undersampling. KNN-based RA scheme performs at par with RF-based scheme for reduced number of samples in the training dataset. In general, the above results show that correlation between the training samples is a plus: The model does not need to retrain for a longer time to appropriately allocate the resources for a terminal position not available in the training data. Furthermore, sufficient training data can be acquired within a couple of seconds to train the learning frameworks, which achieve a performance comparable to the legacy CSI-based RA scheme.

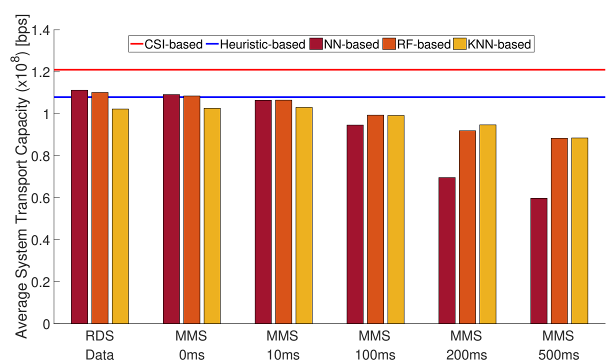

Fig. 17. System performance for 8×2 MIMO system with datasets for random drop scenario (RDS) and mobility model scenario (MMS) having 10 traces for $\rho \leq 0.05/\mathrm{m}^2$ and $\sigma = 0.4$ m. The undersampling in the dataset for mobility model scenario varies from 0 to 500 ms.

B. *Computational Aspects of the Proposed RA Scheme*

After observing the impact of mobility model consideration on the performance of the proposed scheme, we now focus on the computational aspects of the coordinates-based RA scheme. Specifically, we calculate the training time required by NN and RF models, as well as the prediction time per sample, when different number of traces

TABLE IV
TRAINING TIME AND PER-SAMPLE PREDICTION TIME FOR NN AND RF MODELS, FOR DIFFERENT SIZES OF REAL-TIME DATASET

Parameters	50 Traces	20 Traces	10 Traces	10 Traces with 10 ms Undersampling
Training Time (NN)	515.34 s	314.24 s	265.84 s	211.39 s
Per-sample Prediction Time (NN)	3.867μs	3.799μs	3.735μs	3.65μs
Training Time (RF)	3.74 s	1.61 s	0.899 s	0.188 s
Per-sample Prediction Time (RF)	83.67μs	68.8μs	62.34μs	54.4μs

and undersampling rate is considered for generating the dataset for mobility model scenario. For the construction of the training samples, CSI-based schemes used in the next generation communication systems like 5G can be applied, which rely on cloud data centers to determine an appropriate resource allocation [32] within the transmission time interval of the system. Table IV presents the time required to train the two models and the prediction time per sample they take, for different sizes of the dataset with mobility model consideration. The training for NN is performed using NVIDIA RTX 2060 GPU, whereas RF is trained using no GPU support. NN model takes about 4 to 9 minutes to train, depending on the size of the training dataset. For the mobility model scenario's dataset having $\sim$10,000 samples, NN takes about 5 minutes to train. In contrast, training the RF model takes <1 s, with $(\Omega_{\mathrm{t}}, \Omega_{\mathrm{d}}) = (50, 12)$, for a dataset size of $\sim$10,000 samples that needs <1 kB of memory storage. This is another surprising result: If the propagation environment changes rapidly, mainly because of the changing scatterers' position and their density, the system can switch to the training-based mode for only a couple of seconds to construct a learnt model adapted to the changed propagation scenario, and switch back to the position-based mode for predicting $\mathfrak{r}(t)$.

In terms of the per-sample prediction time, NN model shows significantly reduced value, since with GPU supported system, the NN model performs prediction in batches which implies parallelization, resulting in significantly smaller prediction times per sample. Overall, the per-sample prediction time for both the models is significantly lesser than the system's transmission time interval $T_{\mathrm{f}} = 0.2$ ms. All these results point towards the feasibility of implementation of the proposed coordinates-based RA scheme using supervised machine learning frameworks: Depending on the learning model's choice, the system can switch from the training-based mode to the position-based mode within a few minutes, and does not need to be re-trained frequently since the performance of the proposed scheme is quite stable compared to the CSI-based scheme as shown by the results presented in this work.

VII. CONCLUSION

This paper presented a detailed design of coordinates-based resource allocation scheme using machine learning frameworks. We used supervised machine learning to learn the relationship between the terminal's position coordinates and the associated resource allocation to maximize the transport capacity of the system. Specifically, we used K-nearest neighbors as the simplest approach, in addition to the more complex models like random forest and neural network. The average system transport capacity is used for performance comparison between the proposed coordinates-based resource allocation scheme, a simple geometric-based scheme and a legacy CSI-based resource allocation scheme. The results show that the proposed scheme outperforms the geometric-based scheme by a significant margin, irrespective of the antenna configuration considered in the system. The proposed scheme performs consistently well in comparison to the CSI-based resource allocation scheme and is robust to the different stochastic variations in the system. These results are consistent for all the supervised learning frameworks used, with the more complex ones showing better stability in the performance results. These results hold for both the scenarios, i.e. when random drop or mobility model is considered, for generating the terminal's position estimate for dataset formulation. Surprisingly, the proposed scheme needs a training dataset with samples collected over a couple of seconds to achieve a transport capacity of 95% compared to the CSI-based resource allocation scheme. In terms of the system resources, the learnt model using random forest needs <1 s to train and requires <1 kB of memory storage to predict an appropriate resource allocation for a given terminal's position estimate.

The results from this study are very encouraging to establish the feasibility of coordinates-based resource allocation for the communication link between an individual base station and a mobile terminal. In future work, we will extend our investigation by applying the coordinates-based resource allocation scheme to an interference-limited system, i.e. multiple base stations serving multiple mobile terminals. The interference posed by both the interfering terminals as well as by the neighbouring base stations will bring up new challenges for designing the coordinates-based resource allocation scheme. It will also be interesting to see how the training time as well as the size of the machine learning model scales with the multiple-transmitters, multiple-users system. Another interesting aspect is the performance evaluation of the proposed scheme with non-orthogonal multiple access consideration for sub-carrier allocation.

ACKNOWLEDGMENT

A major part of the computations in this work was performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC, KTH.

REFERENCES

[1] Y. Arikawa, H. Uzawa, T. Sakamoto, and S. Shigematsu, “High-Speed Radio-Resource Scheduling for 5G Ultra-High-Density Distributed Antenna Systems,” in *8th International Conference on Wireless Communications Signal Processing*, Oct 2016, pp. 1–5.

[2] J. Chen, R. A. Berry, and M. L. Honig, “Large System Performance of Downlink OFDMA With Limited Feedback,” in *2006 IEEE International Symposium on Information Theory*, 2006, pp. 1399–1403.

[3] J. Leinonen, J. Hämäläinen, and M. Juntti, “Performance Analysis of Downlink OFDMA Resource Allocation with Limited Feedback,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 2927–2937, 2009.

[4] J. Wu and P. Fan, “A Survey on High Mobility Wireless Communications: Challenges, Opportunities and Solutions,” *IEEE Access*, vol. 4, pp. 450–476, 2016.

[5] Y. Li, M. Zhao, and W. Wang, “Internode Mobility Correlation for Group Detection and Analysis in VANETs,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4590–4601, 2013.

[6] I. Bistritz and A. Leshem, “Asymptotically Optimal Resource Block Allocation With Limited Feedback,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 34–46, 2019.

[7] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, “Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2020.

[8] N. E. West and T. O’Shea, “Deep Architectures for Modulation Recognition,” in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2017, pp. 1–6.

[9] A. Dalgkitsis, P. Mekikis, A. Antonopoulos, and C. Verikoukis, “Data Driven Service Orchestration for Vehicular Networks,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[10] A. Testolin, M. Zanforlin, M. De Filippo De Grazia, D. Munaretto, A. Zanella, M. Zorzi, and M. Zorzi, “A Machine Learning Approach to QoE-based Video Admission Control and Resource Allocation in Wireless Systems,” in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pp. 31–38.

[11] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, “A Deep Reinforcement Learning Based Framework for Power-Efficient Resource Allocation in Cloud RANs,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[12] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, “Flexible Resource Block Allocation to Multiple Slices for Radio Access Network Slicing Using Deep Reinforcement Learning,” *IEEE Access*, vol. 8, pp. 68 183–68 198, 2020.

[13] R. Di Taranto *et al.*, “Location-Aware Communications for 5G Networks: How Location Information can Improve Scalability, Latency, and Robustness of 5G,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, Nov 2014.

[14] D. Slock, “Location Aided Wireless Communications,” in *5th International Symposium on Communications, Control and Signal Processing*, May 2012, pp. 1–6.

[15] P. Kela *et al.*, “Location Based Beamforming in 5G Ultra-Dense Networks,” in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sept 2016, pp. 1–7.

[16] X. Chen, J. Lu, P. Fan, and K. B. Letaief, “Massive MIMO Beamforming With Transmit Diversity for High Mobility Wireless Communications,” *IEEE Access*, vol. 5, pp. 23 032–23 045, 2017.

[17] M. Botsov, M. Klügel, W. Kellerer, and P. Fertl, “Location-based Resource Allocation for Mobile D2D Communications in Multicell Deployments,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 2444–2450.

[18] W. Cui, K. Shen, and W. Yu, “Spatial Deep Learning for Wireless Scheduling,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, 2019.

[19] S. H. Cha, J. S. Kim, and M. Y. Chung, “Coordinated-Beam Selection Scheme Using Mobility Pattern of Mobile Device in 5G Mobile Communication Systems,” in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, ser. IMCOM ’17. New York, NY, USA: ACM, 2017, pp. 53:1–53:6.

[20] S. Sand, R. Tanbourgi, C. Mensing, and R. Raulefs, “Position Aware Adaptive Communication Systems,” in *43rd Asilomar Conference on Signals, Systems and Computers*, Nov 2009, pp. 73–77.

[21] S. Imtiaz, G. P. Koudouridis, and J. Gross, “On the Feasibility of Coordinates-Based Resource Allocation through Machine Learning,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec 2019, pp. 1–7.

[22] A. Afifi, K. M. F. Elsayed, and A. Khattab, “Interference-Aware Radio Resource Management Framework for the 3GPP LTE Pplink with QoS Constraints,” in *IEEE Symposium on Computers and Communications (ISCC)*, Jul 2013, pp. 693–698.

[23] S. Imtiaz, G. P. Koudouridis, H. Ghauch, and J. Gross, “Random Forests for Resource Allocation in 5G Cloud Radio Access Networks Based on Position Information,” *EURASIP Journal on Wireless Communications and Networking*, pp. 142–157, 2018.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[25] S. Geman, E. Bienenstock, and R. Doursat, “Neural Networks and the Bias/Variance Dilemma,” *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification and Scene Analysis,” *2nd ed: Wiley Interscience*, 1995.

[27] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] Nielsen, Michael A, *Neural Networks and Deep Learning*. Determination press San Francisco, CA, 2015, vol. 2018.

[29] E. Tuomaala and H. Wang, “Effective SINR Approach of Link to System Mapping in OFDM/Multi-Carrier Mobile Network,” in *2nd Asia Pacific Conference on Mobile Technology, Applications and Systems*, Nov 2005.

[30] V. Nurmela *et al.*, “Deliverable D1. 4 METIS Channel Models,” in *Proc. Mobile Wireless Communication Enablers Inf. Soc.(METIS)*, 2015.

[31] Y. Zhu, Q. Zhang, and T. Yang, “Low-Complexity Hybrid Precoding With Dynamic Beam Assignment in mmWave OFDM Systems,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3685–3689, 2018.

[32] M. Kangas, S. Glisic, Y. Fang, and P. Li, “Resource Harvesting in Cognitive Wireless Computing Networks With Mobile Clouds and Virtualized Distributed Data Centers: Performance Limits,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 3, pp. 318–334, 2015.