Random Forests Resource Allocation for 5G Systems: Performance and Robustness Study

Sahar Imtiaz[*], Hadi Ghauch[*], George P. Koudouridis[†], James Gross[*]
[*]Department of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
[†]Radio Network Technology, Huawei Technologies Sweden AB, Stockholm, Sweden
email: sahari@kth.se

***Abstract*—Next generation cellular networks are expected to improve aggregate multi-user sum rates by a thousand-fold, implying the deployment of cloud radio access networks (CRANs) that consist of a dense set of radio heads. Such a densification of the network inevitably results in high interference coordination complexity and is associated with significant channel state information (CSI) acquisition overhead. The main hypothesis behind this study is that both the coordinated resource allocation complexity and the signaling overhead can be significantly reduced by exploiting explicit knowledge about a terminal's position to make resource allocation predictions. More specifically, we present a design of a learning-based resource allocation scheme for 5G systems that uses Random Forests as multi-class classifier to predict the modulation and coding scheme of a terminal at any given position served by the CRAN. Through performance evaluations it is shown that the signaling overhead is significantly reduced while the learning-based resource allocation scheme can achieve a comparable spectral efficiency to CSI-based schemes. We demonstrate the robustness of the proposed scheme for a varying accuracy of users' positions, showing that even for quite large variations the learning-based approach can still exhibit good performance.**

***Index Terms*—5G, CRAN, resource allocation, machine learning, Random Forests.**

I. Introduction

As compared to Long Term Evolution-Advanced (LTE-A) systems, 5G systems are expected to provide a 1000× increase in system capacity [1], a round trip time over the radio interface of less than 1 ms [2], and support for medium to high mobility users, with high throughput and always-on connectivity requirements. Meeting the capacity and delay demands can be achieved by means of network densification, resulting in an increased interference in the system.

Today's LTE system architecture cannot handle the emerging interference associated with an increased network density. For this purpose, the cloud radio access network (CRAN) [3] has been devised. In CRAN, the radio access units, which constitute transmission points formed from distributed antenna systems, are separated from the central processing units via high-capacity links, that handle all the baseband processing. Each unit of this distributed antenna system is called remote radio head (RRH), which when densely placed with other RRHs in an area is formally known as ultra-dense network (UDN) deployment [4]. Although CRANs may be ideal for achieving tight interference coordination between RRHs, the overhead for channel state acquisition in such UDNs becomes excessive as the densification increases [5], potentially destroying the gains from interference coordination.

The traditional approach utilizes the users' channel state information (CSI) for doing resource allocation in a centralized or distributed fashion for multi-antenna systems [6], [7]. Note that the same approach can be applied for a 5G system, combined with the advantage of coordination between RRHs provided by the CRAN architecture. However, the resulting system overhead can lead to deteriorating performance for increased number of users in the system. Moreover, the CSI-based resource allocation leads to increased computational complexity with large number of users to be served in the system. In [8], it is shown that resource allocation based on CSI is much more expensive in terms of system overhead compared to location-based resource allocation in the context of device-to-device communications.

The objective of this research study is to design an efficient resource allocation scheme that reduces the signaling overhead in CRAN based 5G system, and maximizes the sum-goodput of the CRAN. In lieu of our previous work [9], we design a resource allocation approach using a supervised machine learning algorithm to combine the acquired users' position information with different correlated system parameters. We then compare the resulting sum-goodput with that obtained from the traditional CSI-based resource allocation scheme. We use the Random Forests algorithm as a *multi-class* classifier, as opposed to the binary classifier implementation used in [9]. Our proposed scheme alleviates the problem of system overhead while achieving a sum-goodput comparable to the CSI-based scheme. Furthermore, the proposed method is robust to errors in the estimated position of the terminals, naturally present in the system.

The remaining paper is structured as follows: Section II presents the system model while the machine learning-based approach for resource allocation is presented in Section III, along with brief introduction to the Random Forests algorithm. Performance of the proposed scheme is evaluated in Section IV and conclusions are drawn in Section V.

II. System Model

Fig. 1 presents the CRAN system, consisting of an aggregation node, AN, which performs all the baseband processing in the entire system. A number of R remote radio heads (RRHs) arranged in a dense deployment are connected to this

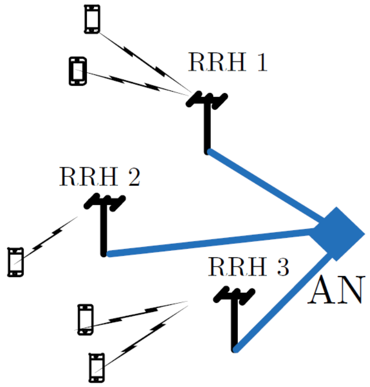

Fig. 1. The CRAN architecture for 5G system

aggregation node by a fast backhaul. The RRHs serve N users spread out across the entire CRAN system in the downlink. The CRAN system operates in time division duplex (TDD), where time frames of duration T_{f} are used for a communication link. Each time frame consists of a number of L sub-frames, each of duration T_{sub}. Each sub-frame further consists of a number of symbols S_{total} and f_{sc} sub-carriers. The operating frequency of the system is f_{c}, with a system bandwidth W. Users are roaming freely within the area with varying velocities and directions. Each RRH and user is equipped with A_{Tx} and A_{Rx} antennas, respectively.

The aggregation node is the resource allocation unit in the considered CRAN system. First, it assigns each RRH to the users present in the system, such that each RRH serves at least one user within a given frame. Once the user assignment is done, the aggregation node then selects a transmit beam $\boldsymbol{v}_r^t$, and a receive filter $\boldsymbol{u}_n^t$, to be used by the assigned RRH and its corresponding user, respectively. The transmit beams and receive filters are taken from the pre-defined sets of beams, $\mathbb{V}$ and $\mathbb{U}$, that are available at each RRH and user, respectively. Finally, the aggregation node selects a modulation and coding scheme (MCS), m_n^t, from the set $\mathbb{M}$, for data transmission between each RRH-user link. The resource allocation is done by the aggregation node on per-frame basis, where the objective is to maximize the system goodput.

For simplicity, we consider the case where each RRH is serving only one user in a given time frame. The propagation scenario is interference-limited, having densely deployed RRHs within the area of interest. Given a certain allocation of users to RRHs in the system, combined with the selection of transmit beams and receive filters for each allocation, the signal-to-interference-and-noise ratio (SINR) of a user n for time t can be written as:

$$\gamma_n^t = \frac{P_{r,n}^t}{\sigma^2 + \sum\limits_{q=1,q\neq r}^{R} P_{q,n}^t}, \tag{1}$$

where, $P_{r,n}^t$ is the signal power received by user n, from its serving RRH r, at time t and is given by:

$$P_{r,n}^t = P_{\text{Tx}} \cdot \ |(\boldsymbol{u}_n^t)^\dagger \boldsymbol{H}_{r,n}^t \boldsymbol{v}_r^t|^2. \tag{2}$$

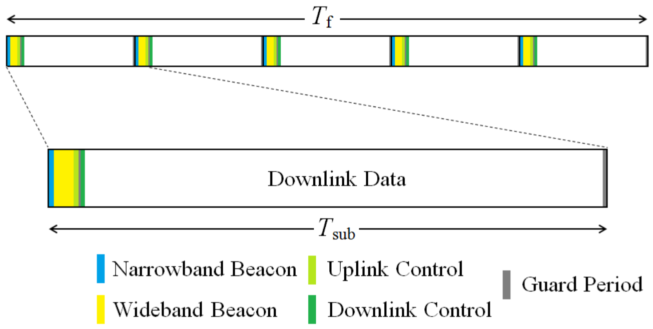

Fig. 2. The frame structure

Here, P_{Tx} denotes the transmit power allocated per RRH. $\boldsymbol{H}_{r,n}^t$ is the $A_{\text{Rx}} \times A_{\text{Tx}}$ downlink MIMO channel matrix for the time frame t between user n and its serving RRH r. σ^2 is the noise power, whereas $(.)^\dagger$ denotes the Hermitian of a vector. Throughout the time frame we assume this SINR to remain constant in time and frequency.

Each element of the channel matrix $\boldsymbol{H}_{r,n}^t$ represents the complex polarimetric channel impulse response between each transmitter antenna element a_{Tx} and receiver antenna element a_{Rx}, denoted by $H_{a_{\text{Rx}},a_{\text{Tx}}}(t,\tau)$. In reality, this channel response is the combination of different path components, i.e. reflection, diffraction, and scattering, which can be modelled as k different multipath components, and is given by:

$$\begin{aligned} H_{a_{\text{Rx}},a_{\text{Tx}}}(t,\tau) = \sum_{k=1}^{K} &\tilde{\boldsymbol{h}}_{k,a_{\text{Rx}},a_{\text{Tx}}}(t) \cdot \\ &\cdot e^{\frac{j2\pi d_k(t)}{\lambda}} \delta(\tau - \tau_{k,a_{\text{Rx}},a_{\text{Tx}}}(t)). \end{aligned} \tag{3}$$

Here, K is the total number of multipath components. $\tilde{\boldsymbol{h}}_{k,a_{\text{Rx}},a_{\text{Tx}}}(t)$ is the impulse response of k-th multipath, including the relevant pathloss. λ denotes the wavelength, and d_k is the total distance for multipath k at time t. $\delta(\tau - \tau_{k,a_{\text{Rx}},a_{\text{Tx}}}(t))$ is the dirac delta function representing the evolution of channel impulse response with respect to different multipath delays $\tau_{k,a_{\text{Rx}},a_{\text{Tx}}}$.

Given the SINR per time frame, the system utilizes the MCS m_n^t chosen by the resource allocation unit, i.e. the aggregation node, to convey backlogged information to the corresponding user. This results in a certain spectral efficiency combined with a block error rate $e(\gamma_n^t)$. Thus, assuming full-buffer at the aggregation node, the choice of MCS determines a corresponding payload size b_n^t that can be sent over the channel, and depending on the resulting block error rate, the goodput for corresponding link at time t can be calculated as:

$$G_n^t = \frac{(1 - e(\gamma_n^t)) \cdot b_n^t}{T_{\text{f}}}. \tag{4}$$

The determination of user assignments and resource allocations is done by the aggregation node utilizing either the position estimates of the users present in the system, or their CSI. The acquisition of either comes at a certain expense, which we model as a system overhead. We assume a time frame structure, as shown in Fig. 2, where the frame duration T_{f} is 1 ms, and it comprises of 5 sub-frames, each of duration T_{sub}. The first few symbols of each sub-frame are used for acquiring users' position or CSI estimates. Since TDD-based system operation is assumed, channel reciprocity holds and therefore, the uplink pilots can be used for users' positions or CSI estimates in the downlink transmission.

The first symbol of each sub-frame is used for position acquisition, since narrow-band pilots are sufficient for such purpose [2]. The next four consecutive symbols in each sub-frame are the full-band pilots used for CSI acquisition. The number of pilots needed for position or CSI estimation depend on the number of users present in the system; larger number of users result in larger number of full-band pilots needed for CSI estimation, but the number of narrow-band pilots typically spans a few time symbols within a frame T_{f}. It should be noted that the adjacent CSI-sensing pilots are scheduled based on the cyclic-prefix compensation distance, as explained in [2], to avoid inter-carrier interference. Also, if any of the pilots in the sub-frames within a frame are not used for position or CSI sensing, they can be used for data transmission in downlink.

Based on these parameters, the percentage overhead for position acquisition per frame can be calculated as:

$$OH_{\mathrm{pos}} = \frac{S_{\mathrm{pos}} \times f_{\mathrm{sc,pos}}}{S_{\mathrm{total}} \times f_{\mathrm{sc,total}}} \times 100\%. \tag{5}$$

Here, S_{pos} is the number of symbols used for position estimation of users in the system, and $f_{\mathrm{sc,pos}}$ denotes the number of sub-carriers used in the positioning beacon. S_{total} and $f_{\mathrm{sc,total}}$ is the total number of symbols and sub-carriers available in the time frame T_{f}, respectively.

Similarly, for CSI acquisition per frame, the percentage overhead can be computed as:

$$OH_{\mathrm{CSI}} = \frac{S_{\mathrm{CSI}} \times f_{\mathrm{sc,CSI}}}{S_{\mathrm{total}} \times f_{\mathrm{sc,total}}} \times 100\%, \tag{6}$$

where S_{CSI} and $f_{\mathrm{sc,CSI}}$ denote the number of symbols and the number of sub-carriers, used for CSI acquisition of users present in the system in a transmitted frame, respectively.

III. Learning-based Resource Allocation Scheme

We use the Random Forests algorithm [10], a *supervised* machine learning technique, for designing the learning-based resource allocation scheme. In general, Random Forests algorithm is well known for its properties of inherent robustness and capability of handling missing data values over other known supervised learning techniques [10], and is the reason for our choice in the design of this proposed scheme.

The algorithm uses a combination of multiple 'random' binary decision trees, which make up the *forest*, for predicting one (or a set of) outcome(s). Being a supervised learning technique, Random Forests algorithm relies on provision of a training dataset to generate the decision trees. The training dataset $\boldsymbol{D}$ consists of two parts: a set of data characteristics or features $\boldsymbol{F}$, and a set of output variables $\boldsymbol{Y}$. Each instance $\boldsymbol{d}_i$ of the training dataset is called an *input feature vector*. The algorithm then constructs Ω_t binary random trees, each with a depth Ω_d, using the different features (selected randomly) in the training dataset. Each tree typically consists of a root node, one or more interior nodes and terminates at leaf nodes, as shown in the sample tree in Fig. 3. The leaf nodes store the output variable(s), technically called a 'vote', and the output variable predicted by the algorithm is the mode of those votes from all trees in the Random Forest. It is worthy to mention here that the algorithm attempts to learn the input-output correlation so as to maximize the overall accuracy of prediction, irrespective of the distribution of individual values of the output variable y. Therefore, care has to be taken that the set of output variables $\boldsymbol{Y}$ is not 'biased' towards some particular values in the training data. Once a forest has been trained (during the operational phase), the input features of a new (and potentially unknown) instance is presented to the decision forest, leading to a prediction (through the voting of the trees) of the output variable [10].

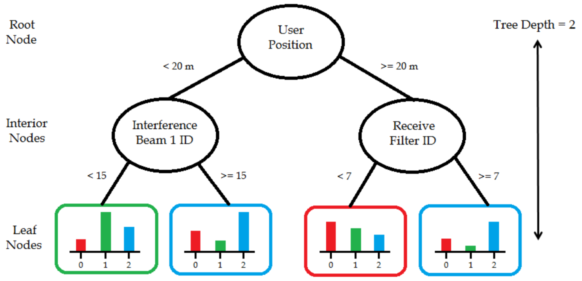

Fig. 3. An example of a binary random decision tree

As mentioned earlier, the main motivation for designing the proposed approach is to use the acquired users' position estimates for allocating system resources efficiently. The resources include transmit beam, $\boldsymbol{v}_r^t$, per RRH-user link, receive filter, $\boldsymbol{u}_n^t$, per user, and the appropriate MCSs' selection, m_n^t, for each RRH-user assignment. We consider fixed RRH-user assignments, where each RRH serves only one user throughout the operational phase. We apply a simplistic approach for designing the training dataset: we use the acquired users' position estimates in combination with some system resources as inputs, and train the algorithm for predicting the MCS as an output. Specifically, the input parameters are the following:

- The estimated position for user n at time t, $\mathcal{P}_n^t$. In reality, this is acquired with certain precision using an extended Kalman filter, along with the direction of arrival and time of arrival estimates of user n [11].
- Transmit beam $\boldsymbol{v}_r^t$ used for serving the dedicated user. Essentially, they belong to a set of fixed beams $\mathbb{V}$ based on geometric beamforming, with a certain angular separation.
- Receive filter $\boldsymbol{u}_n^t$ used by the user terminal to receive transmitted data in a specific direction. These filters are also geometric beams, with an angular separation dependent on the number of antennas at the terminal.

TABLE I
PROPERTIES OF THE INPUT FEATURES & OUTPUT VARIABLE

Variable Name	Data Type	Range
`MS_ID`	integer	1 - 4
`Position_x`	float	273.5m - 291.5m
`Position_y`	float	67m - 203m
`Tx_beam`	integer	1 - 28
`Rx_filter`	integer	1 - 7
`Tx_IF1,Tx_IF2,Tx_IF3`	integer	1 - 28
Output: `MCS`	integer	1 - 8

- Interfering transmit beams $\boldsymbol{v}_q^t, q \neq r$ used by the interfering RRHs, for $n\text{-}th$ system user.

Random Forests algorithm is trained on these 'input features' to predict the MCS, m_n^t, such that the sum-goodput is maximized. Table I shows details of the input variables used for constructing the input feature vector for training the Random Forests model, along with the output variable. Here, '`Tx_IF1`' refers to the interfering transmit beam from the first interfering RRH, '`Tx_IF2`' is the interfering transmit beam for the second interferer, and so on.

Essentially, we deploy the Random Forests as *multi-class classifier*, where the output variable m_n^t has multiple values, or classes. For constructing the training dataset, we use 'exhaustive search' to determine the optimal allocation of resources i.e. $\boldsymbol{v}_r^t$, $\boldsymbol{u}_n^t$ and m_n^t for a user location $\mathcal{P}_n^t$, to be served by a pre-allocated RRH. In a realistic system, a heuristic approach can be used to determine the optimal resource allocation for the acquired user position estimates, to construct the training dataset. Once the training of the Random Forests is complete, we use it as a 'scheduler' for predicting the resource variable, MCS m_n^t, for a newly acquired user position estimate to test its performance. This is done by constructing a test dataset, where the newly acquired position information is compared to the position estimates known during the training process. The input features for the closest matching user position are then combined with the new position estimate to construct the test data sample, for which a prediction of MCS is obtained from the trained Random Forests' data structure. Using the predicted MCS, the sum-goodput is computed by combining all RRH-user links' goodput, calculated using (4), for evaluating the overall system performance.

IV. PERFORMANCE EVALUATION

A. Evaluation Methodology

The performance evaluation of the proposed learning-based resource allocation scheme is done by performing simulations using the discrete event simulator *Horizon* [12]. Fig. 4 shows the simulation scenario, comprising 4 RRHs, each serving a single user. This represents a simpler multi-RRH, multi-user scenario for 5G CRAN system where only inter-RRH interference exists. The user position estimates can either be accurate or can be erroneous, where the error in position is modelled using normal distribution with zero mean and a given standard deviation. A fixed set of transmit beams is designed using geometric beamforming, with an angular separation of 3°. The receive filters are designed in the same way, but the angular separation is kept as 12°. Other parameter settings for the simulation set up are given in Table II. Since downlink communication is assumed in the system model, therefore, channel coefficients for TDD-based downlink are extracted for each RRH-user link in the simulation scenario, using the map-based METIS channel model for Madrid grid [13]. A ray-tracer based channel model was implemented for this purpose, the details for which can be found in [14]. Note that in Table II h_{Tx} refers to the height of the RRH antennas from the ground, while h_{Rx} refers to the user's antenna height from the ground.

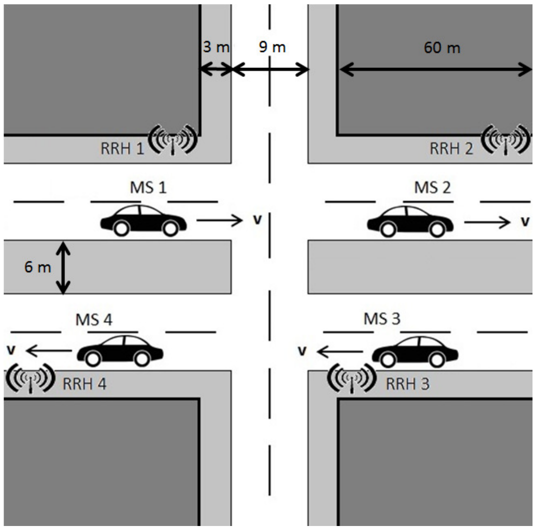

Fig. 4. The simulation scenario; each RRH serves one user

TABLE II
PARAMETER SETTINGS

Parameter	Value
f_c	3.5 GHz
BW	200 MHz
R_{Tx}	8
N_{Rx}	2
h_{Tx}	10 m
h_{Rx}	1.5 m
P_{Tx}	1 mW
T_{f}	1 ms
T_{sub}	0.2 ms
v_{Rx}	30 m/s

Depending on the investigation scenario, the training datasets are constructed as the next step, using the procedure outlined in Section III. These training datasets are used to construct the multi-class Random Forests model, where we rely on the implementation provided by OpenCV [15]. A total of 100 positions, per user, are selected randomly from a set of 1000 user positions generated by Horizon, to create training datasets of 0.25 million samples for each investigation scenario. The output from the Random Forests model is used to compute the user goodput per time instance t, using (4). The system goodput is computed by taking the sum of user goodput per time instance, and its average over all considered 100 user positions is used for system-based performance evaluation.

TABLE III
TRAINING AND TEST ACCURACY FOR DIFFERENT PARAMETRIZATION OF RANDOM FORESTS MODEL

T_n	T_d	Training Accuracy (%)	Test Accuracy (%)
5	3	89.7	90.5
10	3	89.75	90.25
10	5	94	92.75
10	10	99.5	93
50	5	94.75	92.75
50	10	99	93.5
100	5	94	92.75
100	**10**	**99.5**	**93.25**
200	10	99.5	93.25
300	10	99.5	93.25

B. Performance Analysis of Random Forests Algorithm

Random Forests algorithm essentially optimizes the forest data structure for accuracy over the whole training dataset. Here, an important aspect relates to the dimensioning of the forest itself, as it impacts the training and test accuracy. Dimensioning relates to the depth of the trees as well as the number of trees to be used in the forest. For optimized forest design, the number of random features selected for creating a node split in each tree is chosen to be $\sqrt{|F|}$ according to the analysis study presented in [10].

The training accuracy is obtained by using a subset of training data for validation of the constructed Random Forests model. Once a sufficiently dimensioned forest structure has been found, a test dataset is then used to compute the test accuracy of the model by passing each instance of the test dataset through each of the random trees in the model. In this work, the number of test samples is the same as that of the training samples, i.e. 0.25 million, with 100 user positions drawn randomly from among 1000 user positions' data. In terms of performance evaluation, the accuracy of the data structure built by the Random Forests algorithm is an important metric; the higher the number of correctly predicted output by the model (whether for the validation dataset, or the test dataset), the higher will be the accuracy. However, having a very high training accuracy is not an indicator of an appropriately learned forest structure. It could be the case that the learned structure works perfectly for the training dataset, but shows a low accuracy for test dataset. Such a Random Forests structure is then an over-fit to the training data.

For building a robust learning model, we need to vary the number of trees Ω_t in the forest, as well as the depth of the trees Ω_d, in such a way that the model achieves a fairly high accuracy on both the training and test datasets. Hence, for some data collected from a first system set-up (see the next sub-section for details), we study in Table III the training and test accuracy obtained for different parametrization of the Random Forests structure. Based on these investigations, we used the best possible Random Forests model for the design of the learning-based resource allocation scheme, with 100 trees, each with a maximum depth of 10. Once we have the learnt model, we use it for predicting the output variable for the test dataset generated for the considered CRAN system.

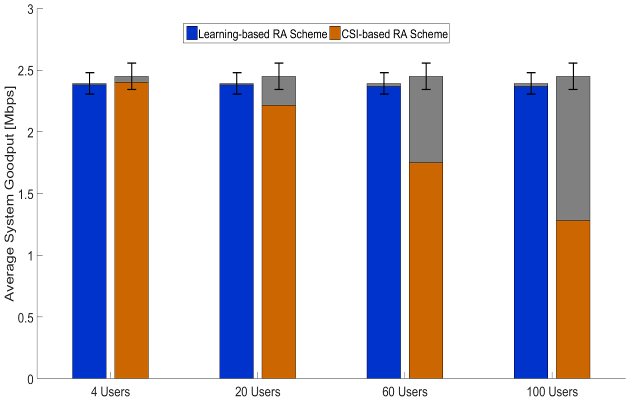

Fig. 5. Effect of overhead on average system goodput for different resource allocation schemes, for perfect information estimates of all users

C. Evaluation Results for the Proposed Learning-based Resource Allocation Scheme

Taking the system overhead into account, we set the transmission time-frame duration to 1 ms and assume 5 subframes to be used for position, or CSI, acquisition and data transmission for all users present in the system. This serves as basic parametrization for the overhead calculations presented in (5) and (6). Our goal is to study the impact of overhead on the performance of the learning-based and CSI-based resource allocation (RA) schemes as the number of users in the system (for which the state information needs to be collected) grows. Note that we consider at this step still all state information to be perfectly accurate (i.e. the position information as well as the CSI).

Fig. 5 shows the results for average system goodput obtained using accurate user position information at all RRHs for the learning-based and CSI-based resource allocation (RA) schemes. The colored bars show the effective average system goodput, i.e. the system goodput obtained after taking into account the effect of system overhead due to position beaconing or CSI sensing, while the underlying gray bars represent the system performance without taking the overhead into account. Overall, the proposed scheme achieves about 96% of the system goodput achieved by the CSI-based scheme, without considering any overhead. However, if the system overhead is accounted for, we observe that the proposed scheme is either at par or better in performance than the CSI-based scheme for all possible number of users present in the system. In particular, as we increase the number of users in the system, the number of narrow-band beacons for acquiring users' position estimates increases gradually per time-frame, and thus the overhead scales up only marginally for the learning-based resource allocation scheme. In contrast, the overhead for the CSI-based scheme grows much stronger with the increase in the number of users present in the system (reaching up to 48% of the frame time), showing that effective system performance degrades severely if CSI-based scheme is used for allocating resources

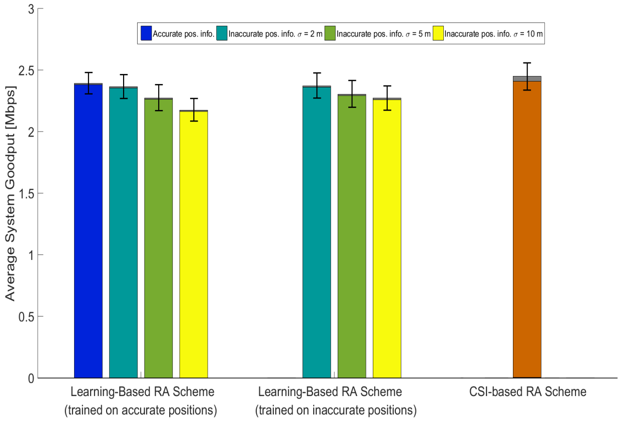

Fig. 6. Average system goodput for different schemes and various possibilities of available position information

in a system with high user density.

These two initial results are quite striking: Firstly, with respect to pure spectral efficiency, a learning-based resource allocation scheme using position information only can achieve quite a good performance already in comparison to a CSI-based scheme. This holds at least for the considered system scenario, which nevertheless has been designed carefully and contains a typical level of detail for a system-level simulation of a 5G network. Secondly, if the overhead or the state acquisition is factored in, due to the high cost of the CSI acquisition, the learning-based resource allocation scheme can significantly outperform CSI-based approaches (up to 100% performance improvement).

D. Robustness of Learning-Based Resource Allocation Scheme

The afore-mentioned performance advantage motivates a more thorough study on the robustness of our proposed learning-based scheme. We start with considering the most obvious potential source of inaccuracy influencing the proposed scheme, namely the accuracy of position information. Fig. 6 shows results for the average system goodput obtained when a random error is involved in the position estimation of the users being served by RRHs. It can be seen that the classifier trained on perfect user position information is enough to guarantee good system performance up to a certain degree of error involved in the position estimation. However, if the error margin in the user position estimates exceeds 2 m, the learning-based resource allocation (RA) scheme trained on perfect user position estimates fails to provide satisfactory system performance. Better system goodput can be obtained using the learning-based scheme trained on inaccurate position estimates, but the traditional CSI-based scheme still provides about 10% better effective system performance. This shows the robustness of the proposed scheme for small degrees of error involved in acquired user position information, but when the error margin becomes excessively large, the CSI-based resource allocation scheme provides better effective system performance, when the best-case user density scenario is considered.

V. Conclusions

In this paper, Random Forests algorithm is used for designing a learning-based resource allocation scheme that serves the different user terminals using only their position information. The proposed learning-based scheme operates with much lower system overhead and complexity than the traditionally used CSI-based resource allocation scheme. It further exhibits either comparable or significantly better effective system performance compared to the CSI-based scheme for different user densities in the system. In terms of the design parameter variations, the proposed scheme is fairly robust to the inaccuracy involved in the users' position estimation.

References

[1] Q. Li, H. Niu, A. Papathanassiou, and G. Wu, "5G Network Capacity: Key Elements and Technologies," *Vehicular Technology Magazine, IEEE*, vol. 9, no. 1, pp. 71–78, March 2014.

[2] P. Kela, J. Turkka, and M. Costa, "Borderless Mobility in 5G Outdoor Ultra-Dense Networks," *Access, IEEE*, vol. 3, pp. 1462–1476, 2015.

[3] M. Hadzialic, B. Dosenovic, M. Dzaferagic, and J. Musovic, "Cloud-RAN: Innovative Radio Access Network Architecture," in *Proceedings ELMAR-2013*, Sept 2013, pp. 115–120.

[4] N., NGMN, "5G White Paper," 2015.

[5] J.-C. Shen, J. Zhang, K.-C. Chen, and K. B. Letaief, "High-Dimensional CSI Acquisition in Massive MIMO: Sparsity-Inspired Approaches," *arXiv preprint arXiv:1505.00426*, 2015.

[6] T. E. Bogale and L. Vandendorpe, "Weighted sum rate optimization for downlink multiuser mimo coordinated base station systems: Centralized and distributed algorithms," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1876–1889, April 2012.

[7] A. Chiumento, C. Desset, S. Pollin, L. V. der Perre, and R. Lauwereins, "Impact of CSI Feedback Strategies on LTE Downlink and Reinforcement Learning Solutions for Optimal Allocation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 550–562, Jan 2017.

[8] M. Botsov, S. StaĎńczak, and P. Fertl, "Comparison of Location-based and CSI-based Resource Allocation in D2D-Enabled Cellular Networks," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 2529–2534.

[9] S. Imtiaz, H. Ghauch, M. M. U. Rahman, G. Koudouridis, and J. Gross, "Learning-Based Resource Allocation Scheme for TDD-Based 5G CRAN System," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '16. New York, NY, USA: ACM, 2016, pp. 176–185.

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

[11] J. Werner, M. Costa, A. Hakkarainen, K. Leppänen, and M. Valkama, "Joint User Node Positioning and Clock Offset Estimation in 5G Ultra-Dense Networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, December 2015, pp. 1–7.

[12] G. Kunz, O. Landsiedel, S. Götz, K. Wehrle, J. Gross, and F. Naghibi, "Expanding the Event Horizon in Parallelized Network Simulations," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 172–181.

[13] T. Jamsa *et al.*, "Deliverable D1.2 Initial Channel Models Based on Measurements," *METIS Project Deliverable*, 2014.

[14] P. Kela *et al.*, "Location Based Beamforming in 5G Ultra-Dense Networks," in *Proc. Vehicular Technology Conference (VTC Fall), 2016 IEEE 84th*, September 2016, accepted for publication.

[15] G. Bradski, "Open Source Computer Vision Library," *Dr. Dobb's Journal of Software Tools*, 2000.