User Allocation in 5G Networks Using Machine Learning Methods for Clustering

Christos Bouras(✉) and Rafail Kalogeropoulos

Computer Engineering and Informatics Department, Patras, Greece
bouras@cti.gr, kaloger@ceid.upatras.gr

Abstract. The rapid increase in the volume of devices connected to mobile networks poses unprecedented demands on existing networking infrastructures. Machine Learning is a promising technique, already applied in various sectors of our everyday lives. It enables decision making not with the use of traditional programming but rather by using data to train models to cope with various problems without explicit programming on how to do so. The integration of Machine Learning techniques is deemed necessary in as many processes as possible to help the network face congestion and enable efficient real time decision making. In this paper we present two Machine Learning based mechanisms for improving real time user allocation on the network as well as predicting the best positioning scheme for Smallcell Base Stations to provide effective utilization of the network's resources.

1 Introduction

The introduction of more capable mobile devices has set the pace for a complete redesing of all cellular networks and their infrastructures. These new devices are able to sustain a plethora of applications. Their computing power always allows for the development of new apps while the establishment of communication standards allows them to connect with a multitude of sensors and devices. This motivates the integration of new devices such as mobile phones and sensors into the network. They can be used for a variety of purposes such as medical [9] or agricultural use [2].

Wireless networks as a result face an unprecedented rise in the number of connected devices that seek to utilize their resources, while the volume of data transfered through the network's infrastructure increases as well. In contrast to previous generations of networks that mostly consisted of a Base Stations (BSs) of the same type, namely Macro cell Base Stations (McBSs), in latest generations we saw the introduction of Smallcell Base Stations (SBSs), smaller BSs that can be further categorized based on their transmit power.

In previous generations, McBSs prevailed due to a higher transmit power, ensuring high Signal to Interference plus Noise Ratio (SINR), especially in the Downlink (DL) direction, while performance in the Uplink (UL) direction was

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
L. Barolli et al. (Eds.): AINA 2021, LNNS 225, pp. 13–24, 2021.
https://doi.org/10.1007/978-3-030-75100-5_2

deemed secondary. All McBSs feature the same characteristics throughout the entire network and are also capable of supporting a similar amount of users [12]. As a result, User Equipment (UE) were associated on the same BS for both directions, based solely on DL performance. These networks, called Homogeneous Networks were applicable since the required throughput for DL was traditionally higher than UL, resulting in an asymmetric utilization from the two traffic directions [10]. This approach is now completely dated, with the plethora of new devices uploading in the UL direction.

The fifth generation, considers the DL and UL direction as separate networks and does not demand users to connect to the same BS for both of them [6]. Such a technique is called Downlink and Uplink Decoupling (DUD). These networks are called heterogeneous networks (HetNets). Such networks consist of prominent McBSs and various SBSs that are scattered among a McBS's vicinity [8]. They are necessary to ensure accessibility for all connected devices and guarantee a respectable Quality of Service (QoS) in real-time.

In HetNets, SBSs become more prevalent since more traffic is generated on the UL direction. Most network connected mobile devices are battery powered with identical transmit power, though with different demands. For example autonomous cars rely on sensors that produce data needed for autonomous driving. They are extremely dependent on accuracy to function properly and demand high QoS while other applications such as music streaming usually have less QoS demands. This increases dramatically the complexity of the UL direction.

The use of Machine Learning (ML) offers a significant opportunity to refine existent network applied methods and it can even work supplementary to them to improve real time performance. So far network application of ML techniques contains the work of [1], where the authors explore an application of ML techniques to explore unknown guest user dynamics for a network of users with different demands. Self-organizing maps applied to cellular networks have been proposed to dynamically shape the connectivity of the network to the prospective demands in [4]. The survey in [3], discusses multiple open issues on ML applications for computer networks and potential problems that arise with them.

In this paper we will propose two ML based mechanisms. Our first proposal is based on Decision Trees, that can be trained on data produced by already deployed networks, pinpoint patterns on optimal user allocation and produce promising results without the need to assess all these metrics that are necessary for traditional user allocation techniques. Our second proposal can function supplementary to our first proposal, and predict the optimal positioning of SBSs based on user distribution across the network. This mechanism utilizes the k-means clustering algorithm to produce cluster centers, which we consider as candidates for positioning SBSs. Our proposals can be utilized for Efficient user allocation in networks and can be proven extremely helpful in research that aims to provide an efficient scheme for enriching network infrastructures.

In remaining sectors, we present our full proposal. Sect. 2 focuses on the system model. In Sect. 3 we present our two ML mechanisms. Section 4 presents our simulation parameters while Sect. 5 presents the simulation results. Finally, in Sect. 6 we draw our conclusions and we make suggestions on how ML can be introduced in future research.

2 System Model

With the emerge of new technologies like the Internet of Things (IoT), we expect radical changes in computer networks, such as a massive rise in the number of UEs that try to utilize the network's resources. As Small Cells become smaller we expect their deployment to be massive in current and next generation networks. These two cases create the need for solutions on how to manage such an increase in the number of connected UEs and how to position and utilize new SBSs to optimally match a UE with the appropriate BS for its UL/DL needs.

Regarding the basic network layout, we consider a Heterogeneous Network. Our network consists of BSs (McBSs and SBSs) and users (UEs). McBSs are denoted as M (M=1,...,$|M|$), SBSs are denoted as S (S=1,...,$|S|$) and the set of UEs is denoted as U (U=1,....,$|U|$). Network users attempt to transmit and receive data. Traffic can be split into two networks for transmitting and receiving data (UL Network and DL network). In HetNets, they are considered as separate channels. All BSs have limited resources, meaning they have a bound on the maximum number of users they can serve simultaneously. All BSs of the same type are considered to have the same resources.

To compute the number of RBs that a user (suppose user j) demands from a specific BS, for achieving their desired DR, we will use the following equation:

$$RB_{\mathrm{j,i}} = [\frac{T_j}{B_{\mathrm{RB}} * log_2(1 + SINR_{\mathrm{j,i}})}], \tag{1}$$

where T_j denotes the UE throughput demands and DR_i the desired Data Rate for the user j, B_{RB} is the bandwidth of a RB and $SINR_{\mathrm{j,i}}$ is the SINR between a BS and an associated user.

Next we shall define the equations for calculating Pathloss (PL) and Signal to Interference and Noise Ratio (SINR). We use the distance dependent Path loss model, to calculate pathloss, and thus we have two different equations, one for McBSs and one for SBSs. PL is used to measure the signal loss we can expect from a signal as the user receives it, relatively to when it was emitted from the BS. The equations are given below, both for McBSs and SBSs:

$$PL_M = 128.1 + 37.6 * log_{10} d, \tag{2}$$

$$PL_S = 140.7 + 36.7 * log_{10} d, \tag{3}$$

where d is the distance with a user and its serving BS.

Next we shall define SINR for DL and UL. $SINR_{\mathrm{j,i}}$ corresponds to the SINR between a user j and its corresponding BS i. This metric represents a ratio, namely the strength of the signal received by the receiving antenna and can be calculated as:

$$SINR^{\mathrm{UL}}_{\mathrm{i,j}} = \frac{P_{\mathrm{BS}}}{N + I}, \tag{4}$$

$$SINR_{\mathrm{i,j}}^{\mathrm{UL}} = \frac{P_{\mathrm{ue}}}{N + I}, \tag{5}$$

Here, P_{ue} corresponds to Transmit Power (TP) of the UE, while P_{BS} is the TP of the BS, whether it is a McBS or a SBS. Regarding noise and interferences, N corresponds to the Noise power while I corresponds to the total interferences [7]. We try to simulate a typical metropolitan area network scenario. Considering that most real life networks suffer from great Non Line Of Sight (NLOS) issues, we assume that our network features more SBSs compared to McBSs. These SBSs aim to alleviate congestion in areas with high user density.

For our simulation, the main focus of applying ML is associating users and BSs, minimizing the amount of calculations necessary. Our goal is to reduce time complexity and enable better real time decision making. Network load should be efficiently distributed among all BSs, retaining an acceptable QoS level for all users. For our simulations we utilize two ML techniques that are quite different, yet incorporating them into the network can yield promising results. The two techniques are:

2.1 Decision Trees

A decision tree is a decision tool that creates a tree like presentation to model decisions and their possible outcomes. The decision tree consists a rooted tree, since it has a node called "root" with zero incoming edges while all other nodes feature exactly one incoming edge. Nodes with no outgoing edges are called leaves (and in this case they can also be called decision nodes). In a decision tree, based on a designated function each internal node splits the instance space into two or more sub-spaces.

Each leaf may be assigned to one class or it may hold a vector indicating the probability of the target attribute assigned to a certain value. The classification is produced by navigating from the tree root all the way down to a leaf [5]. In other words, following this path we can see all decisions made by the Decision Tree to come up with the depicted result (depicted on the leaf).

2.2 K-Means Clustering

Clustering is the process of grouping a set of points into "clusters", where points with a small distance belong to the same cluster, where points with a large distance are placed in different clusters. K-means algorithm assumes a Euclidean space and a predifined number of clusters. These two characteristics produce two of the most prominent issues with k-means clustering [11]. It is not always easy to know in advance the number of clusters, while it also creates similarly shaped clusters (ball-shaped), since the clusters the based on Euclidean distance.

For the algorithm itself, there are several ways to define the starting cluster centers. They can be random or we can select points from the dataset we want to process. All the points that we want to classify need to be assigned to the nearest cluster, meaning that for each point we need to calculate its Euclidean distance to the cluster center. As new points are added to the clusters, the cluster centers are constantly re-evaluated. The process can stop when we have no more points to be clustered. As an optional step, in the end we can fix the cluster centers and we can re-examine all points. Euclidean distance can be calculated as:

$$D = ||X - Z|| = \sqrt{\sum_{n=1}^{n} (x_i - z_i)^2}, \tag{6}$$

where x_i and z_i, are the coordinates of the points (user and BS) in question.

3 The Mechanisms

In this sector we present our two proposed mechanisms. These mechanisms can stand alone or they can work collectively to provide a prediction mechanism that aims to refine user allocation on BSs and provide an adequate BS positioning scheme that takes into consideration user distribution across the network.

3.1 Mechanism for Predicting User Allocation on BSs

The first mechanism is based on Decision Trees. We expect our Decision tree-based model to able to predict the best matched BS for each user based on a specific metric. We begin by modeling a network where all users are distributed uniformally across the network and allocated to a certain BS using SINR as the preferred allocation metric. For this scenario, when the network model has decided the best matching BS for each user based on our metric, the allocation results are saved on a dataset. The dataset features the coordinates of each user in the network, as well as their matching BS both on the DL as well as the UL direction. This dataset is then utilized by our ML based model for training and testing.

The size of the dataset varies with the number of users that are deployed on the network. We test our mechanisms for various dataset sizes. The produced dataset is split into two different datasets. The first one being the training dataset and the second one is the test dataset. The training dataset is fitted into a ML model that is based on the Decision Trees technique. After the model has been trained, we use the test dataset to assess its capabilities and calculate its precision. We will evaluate the model precision for a variable number of users to see how the dataset size affects the model performance.

Algorithm 1. Pseudocode for the first mechanism

```
U: Denoting the number of users
for i = 1 to U do
    Calculate SINR for all BSs on DL and UL;
    Create BS preference list for DL and UL over SINR;
    Associate to best matched BS for DL and UL;
end for
Produce dataset with User coordinates and DL,UL associated BS
while Precision is low do
    Split produced dataset into training dataset and test dataset
    Fit training dataset into Decision trees model
    Use test dataset to predict associations and calculate model precision
end while
```

3.2 Mechanism for Efficient Placement of SBSs

The second proposed mechanism is based on K-means Clustering and aims to provide a reliable method to optimally position BSs across the network so that we can accommodate as many users as possible to avoid congestion escalation in the network. The network consists of a set of users, distributed uniformly across the network to ensure a realistic deployment scenario. These users can be classified, based on their positions where users that are closer together, will be classified in the same class.

To complete this task we use the k-means clustering algorithm. This algorithm starts with a set of empty clusters (we begin with 42 clusters - the same number as the number of BSs in our network) where users will be allocated. The algorithm then places users into clusters based on their distance from the cluster center. As a new user is allocated into a cluster, the coordinates of the cluster center are calculated again to account for the new user that is now part of the cluster. It is essential that all users must connect to some cluster, so that we can produce accurate coordinates for the cluster centers. When the procedure is completed, for each of these final cluster centers, we calculate their distance from all BSs.

We then identify the minimum distance for all clusters and any BS. Cluster centers that share a minimum distance with McBSs, are ignored while the coordinates of cluster centers that are placed near SBSs, are considered to be the optimal coordinates for placing the SBSs in our network. Finally we compare the network performance using random coordinates for SBS positioning vs the positioning scheme that we proposed. The coordinates of cluster centers that are closer to McBSs are deleted, since McBSs are stationary and cannot be moved and thus we cannot propose a cost-efficient repositioning scheme for them.

Algorithm 2. Pseudocode for the second Mechanism

U: Denoting the number of users
Start with XX empty clusters
for $i = 1$ to U **do**
  Associate user with a specific cluster
  Calculate new cluster center
end for
Calculate cluster centers distance with all BSs
Delete cluster centers closer to McBSs
for clusters centers that are closer to SBSs **do**
  Use calculated cluster centers as the new SBS coordinates
end for

4 Simulation Setup

In this sector we will present the parameters used to model our 5G network. All network simulations and the proposed mechanisms were produced using Python. We used Python because it incorporates functions for a plethora of ML techniques, making it a powerful tool for ML centric simulations. Our produced network, can be seen on Fig. 1. This network consists of 13 McBS, 29 SBSs and 200 UEs. McBS are presented as big triangles placed at the center of the network hexagons, SBSs are presented as "Y" figures, while all users are presented as bullets.

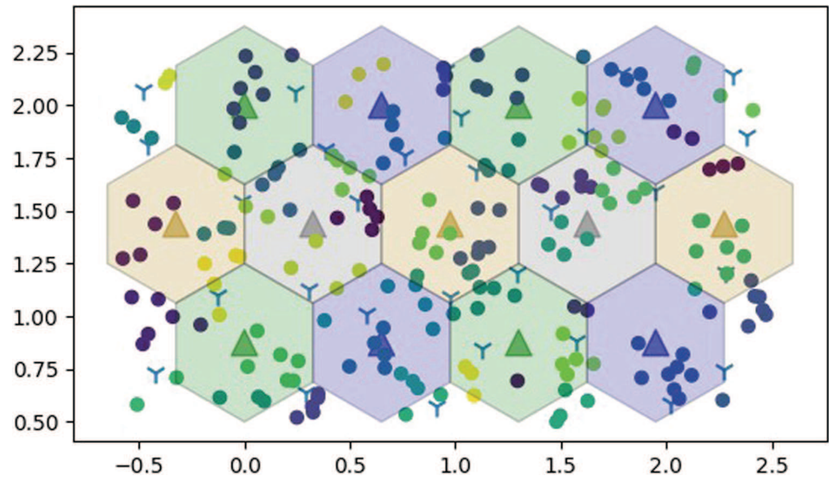

Fig. 1. The Python produced network, consisting of 13 MBSs, 29 SBs and 200 users.

In both of our simulations we have the same number of users, and they are distributed along the network using the uniform distribution to ensure a uniform distribution across the entirety of our network. This ensures a realistic deployment with users wanting to exploit the entirety of our network. In the second mechanism, we aim to produce user clusters. Using the Euclidean distance as the distance metric for the k-means algorithm, all produced clusters seem to revolve around the cluster centers. Our mechanism is provided with the number of wanted clusters, that is equal to the number of BSs in the network, a feature that we can utilize if we want to expand the network with new BSs. All simulation parameters can be seen on Table 1.

Table 1. Simulation parameters

Parameter	Setting
Macro cell transmition power	50 dbm
Macro cell transmition power	24 dbm
User equipment transmition power	20 dbm
McBS Pathloss exponent	4
SBS Pathloss exponent	3.6
Network deployment	13 Mcells and 29 Scells
Number of users	200, 500 and 1000
Stationary user distribution	Uniform distribution

5 Results

In this section we will present the results produced from our two mechanisms. In both cases, we consider a basic network where users are static. For our first scenario, we used a dataset containing the association results from the network model based on the SINR metric, to train a ML model, based on the "Decision Trees" technique.

From the results presented on Fig. 2, we see that the model can be quite precise on the produced predictions. The dataset used is split on two parts, a training set, used to train the model and a test set, used to test its capabilities. Considering a fixed test dataset that is 20% of the size of the original dataset produced from the network model, we test our precision value. Starting from 200 users, we get a precision value of 0.625 for the UL direction and a value of 0.85 for the DL direction. When we increase the number of network users, our produced dataset size increases as well. With a dataset of 500 users, we see an improvement on the produced accuracy that now reaches 0.73 for the UL direction and 0.92 for the DL direction. Finally with a dataset consisting of the results on the allocation of 1000 users we see the accuracy rise to 0.84 for the UL direction and to 0.935 for the DL direction.

These results, show that using a ML technique can be proven extremely efficient in predicting user allocation on the network. Allocating users on BSs can be quite a complicated procedure, since it has to take into account multiple parameters for the association. As we can see from the produced results, the predictions for the DL direction are far superior to the predictions on the UL

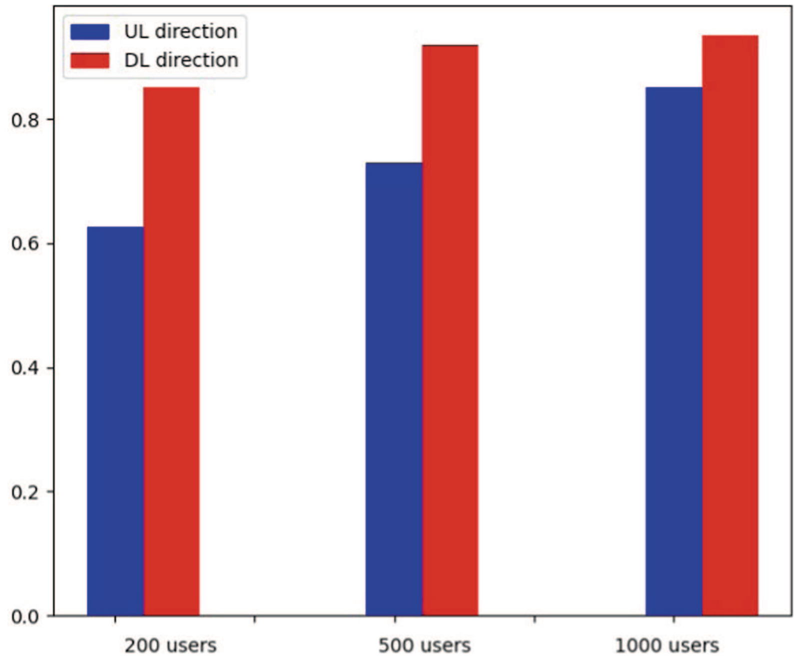

Fig. 2. Prediction accuracy.

direction. This is to be expected, since most users for the DL direction are allocated to McBSs. That means that in the produced dataset, most BSs are McBSs, making it easier for the model to predict the associated BS for each user, since it has to choose from a limited pool of BSs.

It is important though to mention that the size of our training dataset (and the test dataset) massively impacts performance. For complicated models like these, it is important that we fit them with a respectable size training dataset, so that the model can pinpoint all connections between the parameters and produce accurate results. The size of the training set should be appropriate, according to our needs, in order to avoid cases of overfitting and underfitting.

The allocation techniques that exist are quite extensive and very accurate on selecting the BS that best matches each user. This means that they can be used as a pretty good source for creating accurate datasets to use for ML models training. Especially in our case, the produced model can predict with high accuracy the best allocation for each user. In real world scenarios, this can lead to a massive improvement in real time performance for the network. Classic user allocation models take into account multiple metrics and network parameters and as a result they suffer from high complexity. Utilizing ML based models that are trained on datasets produced from classic allocation models, results in better real time decision making since a simpler model features much better time complexity, that can produce reliable results even in cases with limited network processing resources.

For our second scenario, the produced user clusters can be seen on Fig. 3, along with their respective cluster centers. The produced cluster centers are depicted as black bullets, while users that consist a cluster are depicted as bullets that share the same color. In our network we have defined 42 BSs and we also produce 42 user clusters. The position of the McBSs cannot be changed, while the positioning of the SBSs is not definite.

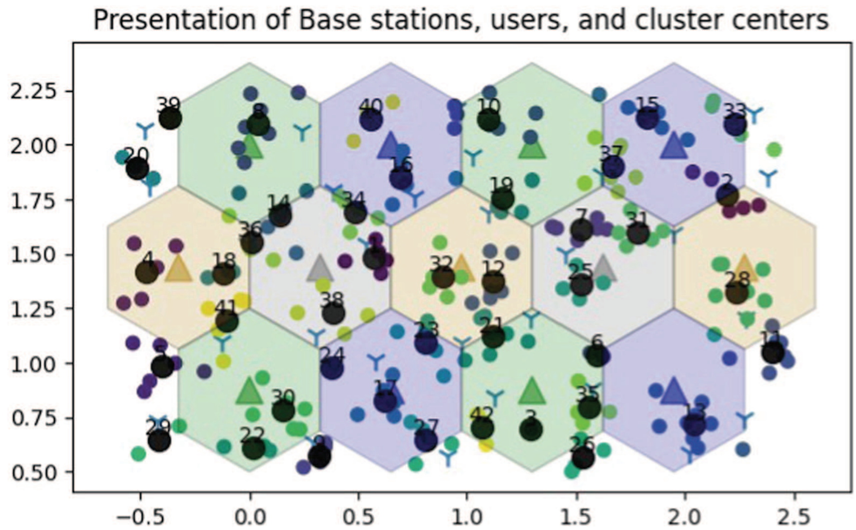

Fig. 3. Produced user clusters

Using the cluster centers as the new coordinates for the SBSs we ran our simulations. We can conclude that our proposal results in smaller distances between cluster centers and the closest available SBS. Smaller distances, result in less pathloss and signal deterioration. This means that our users will be able to enjoy higher DRs. Traditionally, UL direction demands are smaller than DL demands and are satisfied from SBSs. In our simulations, cluster centers that are close to McBSs, result in the same distance from McBSs in both results since McBSs are stationary.

Our proposal yields impressive results for the UL direction. As seen on Fig. 4, after improving the positioning scheme of SBSs, the number of associated users in the UL direction remains high and is usually better than the results with a random positioning of SBSs. This is subject to small changes, considering we can never be certain about user spawn points, and their RB needs. Since most of the associations in the DL direction are with McBSs (which remain stationary across both simulations), DL association results are the same on both cases. The difference varies with the UL direction.

All these users are matched with the BS that offers them the best SINR, meaning that they enjoy the best available QoS. Our results show that our suggestion can improve the network performance, but they ensure that our proposal remains a trustworthy method for pinpointing the best locations for placing new SBSs if we want to enrich our network's infrastructures with new SBSs. That will result in a higher offered QoS for the UL direction as well as the DL direction while the improved number of associated users results in improved total network throughput.

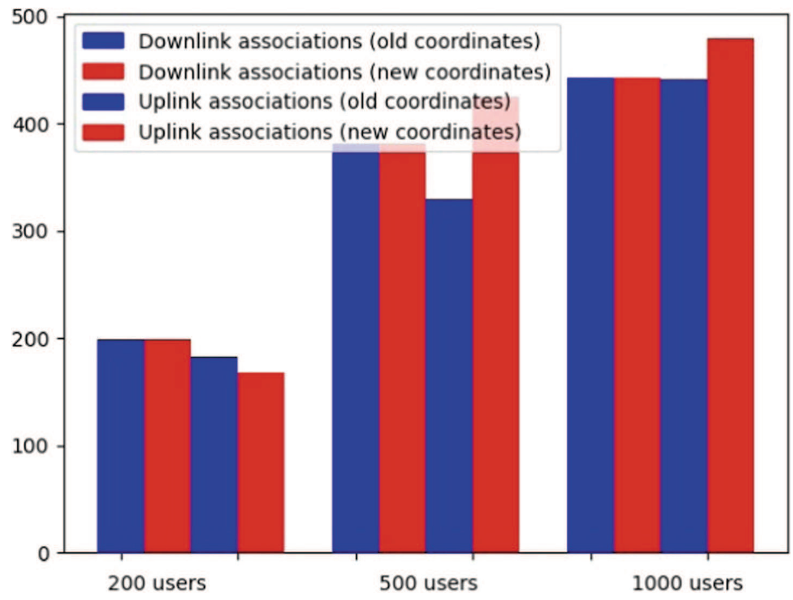

Fig. 4. Associated users with best matched BS

Considering that both pathloss and SINR are indeed heavily dependent on distance, users will see increased DRs that suffer less reduction (due to the lower pathloss). The smaller distance also means that the network resources will be better allocated. Users will enjoy better DRs, but will need less RBs from their matching BSs. That means that less RBs will be consumed by a single user, while more users will be satisfied using the same resources. As result, with our proposal we will see a massive improvement on the achieved channel throughput.

Our two proposals show that using ML we can begin to see important benefits on both user allocation as well as resource management. Our first proposal can allocate users on their best matching BSs, using minimal network resources, ensuring better real time decision making for the network. With our second proposal, we can predict the optimal positioning for SBSs. This supplements our first proposal, ensuring that more users are satisfied (especially in the UL direction) with higher DRs. This improves network performance as well as the perceived QoS, since now more users can utilize the network's resources.

6 Conclusion and Future Work

With the rise of IoT, we expect networks to face an unprecedented rise in the number of UEs that seek to use their resources. This is a crucial issue since it dictates a fruitful utilization of the existent infrastructure and the available resources. With the number of UEs expected, real time performance on networks can also be affected since massive calculations should be completed for optimal user allocation. With our two proposals, we proved that applying ML techniques can help with real time performance by predicting the optimal association between users and BSs, but it can also be used to predict the best positioning system for SBSs. This will significantly improve the existing network performance but can also complement the results produced from our first proposal.

Many ML techniques have been introduced to various scientific sectors and its introduction to networking is deemed necessary. This introduces an opportunity for massively improving existing mechanisms. In the future prediction mechanisms can be introduced to resource management (like frequency allocation), while clustering can also be used to improve the network scalability by finding an optimal system for positioning new BSs to reinforce current infrastructures.

References

1. Yang, J., Wang, C., Wang, X., Shen, C.: A Machine Learning Approach to User Association in Enterprise Small Cell Networks, pp. 850–854 (2018). https://doi.org/10.1109/ICCChina.2018.8641148
2. Mahbub, M.: A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. Internet of Things **9**, 1–30 (2020). https://doi.org/10.1016/j.iot.2020.100161
3. Sun, Y., Peng, M., Zhou, Y., Huang, Y., Mao, S.: Application of machine learning in wireless networks: key techniques and open issues. IEEE Commun. Surv. Tutorials. 21(4), 3072–3108 (2019). https://doi.org/10.1109/COMST.2019.2924243
4. Mom, J.M., Ani, C.: Application of self-organizing map to intelligent analysis of cellular networks. ARPN J. Eng. Appl. Sci. **8**, 407–412 (2013)
5. Rokach, L., Maimon, O.: Data mining with decision trees. Theory and Applications (2008). https://doi.org/10.1142/9789812771728-0001
6. Shi, M., Yang, K., Xing, C., Fan, R.: Decoupled heterogeneous networks with millimeter wave small cells. IEEE Trans. Wireless Commun. 17(9), 5871–5884 (2017). https://doi.org/10.1109/TWC.2018.2850897
7. Elshaer, H., Boccardi, F., Dohler, M., Irmer, R.: Downlink and uplink decoupling: a disruptive architectural design for 5G networks (2014). https://doi.org/10.1109/GLOCOM.2014.7037069
8. Feng, Z., Li, W., Chen, W.: Downlink and uplink splitting user association in two-tier heterogeneous cellular networks. In: 2014 IEEE Global Communications Conference (GLOBECOM 2014), pp. 4659–4664 (2015). https://doi.org/10.1109/GLOCOM.2014.7037543
9. Ghosh, A., Raha, A., Mukherjee, A.: Energy-efficient IoT-health monitoring system using approximate computing. Internet of Things. **9**, 100166 (2020). https://doi.org/10.1016/j.iot.2020.100166
10. Sun, S., Adachi, K., Tan, P.H., Zhou, Y., Joung, J., Ho, C.K.: Heterogeneous network: an evolutionary path to 5G, pp. 174–178 (2015) https://doi.org/10.1109/APCC.2015.7412506
11. Cheung, Y.: K*-means: a new generalized k-means clustering algorithm. Pattern Recogn. Lett. **24**, 2883–2893 (2003). https://doi.org/10.1016/S0167-8655(03)00146-6
12. Boostanimehr, H., Bhargava, V.: Unified and distributed QoS-driven cell association algorithms in heterogeneous networks. In: IEEE Transactions on Wireless Communications. vol. 14 (2014). https://doi.org/10.1109/TWC.2014.2371465