Wireless Personal Communications (2022) 122:1455–1479
https://doi.org/10.1007/s11277-021-08957-4

Prediction Mechanisms to Improve 5G Network User Allocation and Resource Management

Christos Bouras[1,2] · **Rafail Kalogeropoulos**[2]

Accepted: 9 August 2021 / Published online: 17 August 2021

Abstract
As technology rapidly advances, the number of devices constantly communicating, transmitting and receiving data through the cellular networks keeps rising, posing an unprecedented load on them. Such an increase calls for establishing new methods to manage these devices as well as utilize the data produced by them to establish network architectures that can prevent cellular networks from overloading. To achieve the desired results, we need to optimally allocate network resources to existing users. Resource allocation has traditionally been considered an optimization problem where proposed solutions are hard to implement in real time, resulting in the use of inferior solutions with reduced produced performance. With the introduction of Machine Learning, we propose three mechanisms, intending to utilize network data to improve real time network performance. The first mechanism, a Decision Trees based mechanism aims to improve real time decision making by predicting the optimal matching of users and Base Stations. The second mechanism, a K-means based mechanism intends to tackle network congestion and ensure uninterrupted Quality of Service by predicting the optimal coordinates for placing Base Stations along the network based on traffic data. Finally, a Regression based mechanism manages to predict user movement along the network, resulting in improved resource management and reduced energy waste. These mechanisms can work cooperatively, essentially establishing a network architecture that utilizes prediction to efficiently allocate users and manage available resources.

Keywords Machine Learning · Internet of Things · Resource management · User allocation · Prediction · Efficiency

✉ Christos Bouras
bouras@cti.gr; bouras@upatras.gr

Rafail Kalogeropoulos
kaloger@ceid.upatras.gr

[1] Computer Technology Institute & Press "Diophantus", Patras, Greece

[2] Research Academic Computer Technology Institute and Computer Engineering and Informatics Department, University of Patras, Patras, Greece

1 Introduction

The prosperous emerge of Internet of Things (IoT) and other network innovations, has resulted in an immense increase in the number of devices, mobile or not, connected to the internet. These new devices include mainly computers, mobile devices such as smartphones and various sensors, eligible to be used in a variety of occasions such as medicine [1] or agriculture [2]. The computing power of these devices seems to increase exponentially over the years allowing them to be utilized for a plethora of applications, sometimes even creating the need for new ones. As a result, this massive increase in the volume of interconnected devices has created the need for a complete redesign of current cellular infrastructures to facilitate the introduction of new technologies and mechanisms.

Cellular networks need to evolve, so that they can sustain the introduction of such a multitude of interconnected sensors and devices. To cope with this unprecedented rise in the number of connected devices that seek to utilize their resources and the inevitable rise in the volume of data transferred through the network's infrastructure, cellular networks need to innovate both in terms of infrastructures but also in the techniques utilized by them.

The first step was to evolve from Homogenous Networks. These generations of networks, consisted of Base Stations (BSs) of the same type, namely Macro cell Base Stations (McBSs), stations that feature the same characteristics throughout the entire network, such as in the number of users they are able to support [3]. These stations feature high Transmit Power (TP), ensuring high Signal to Interference plus Noise Ratio (SINR), especially in the Downlink (DL) direction, where most of the data traffic was produced. The Uplink (UL) direction, produced little data volume, so most efforts were based on efficiently handling DL [4]. As a result, all User Equipment (UE) was allocated on the same BS for both directions, taking into consideration only the produced DL performance.

Lately, with the introduction of new UE devices, the volume of data produced in the UL direction, has increased dramatically, establishing the need for methods that are efficient both for UL and DL. To cope with the volume of connected UEs, Heterogeneous Networks (HetNets) emerged. These networks are extremely dense both in terms of the total volume of connected users and devices as well as the number of Access Points (AP) they encapsulate. These APs will feature multiple Radio Access Technologies (RATs), effectively multiplying the complexity of the network, but also increasing its capabilities in terms of the volume of applications it caters for [5]. The volume of such APs will need to be sufficient in order to care for the demands posed by users. In HetNets, different network components should work together to serve various types of traffic with different demands in QoS and coverage.

The fifth generation of networks (5G Networks), is in core a HetNet. These networks incorporate smaller BSs, Smallcell Base Stations (SBSs), stations that can be further categorized into different categories based on their TP. These different types of stations feature different coverage areas and they intend to help resolve the lack of available network resources especially in the sub-6 GHz access network [6]. 5G networks will incorporate BSs of various sizes. The prominent BS type remains McBSs, with SBSs being scattered in the network among the McBSs' vicinity [7]. In this generation, the technique of Downlink and Uplink Decoupling (DUD) ensures that users enjoy the highest possible QoS for both UL and DL. It separates both directions and allows users to associate with the best serving BS, either McBS or SBS for both [8]. Several approaches have been discussed. In some scenarios, SBSs are assigned to McBSs [9], meaning that selecting a SBS for user allocation is heavily governed by its associated McBS, while other approaches consider all BSs independent, regardless of their type.

The second approach is the one we follow in the research of this paper. This ability to connect users to various BSs has a massive impact on resource management, since we have to take into consideration the possibilities of hand-off in different cells [10]. So far, these networks have been proved quite efficient, being able to encapsulate any type of connected device and satisfy a respectable number of users with acceptable QoS.

Like all previous generations, with the increase in the volume of connected UE and the produced data traffic, 5G Networks will inevitably face various issues, such as real time decision making and security. Many of the newly developed applications that are served from 5G networks, impose severe requirements in terms of the required data rate as well as the latency required, especially in real time applications. In this aspect, Machine Learning (ML) can be utilized to complement existing network infrastructures and mechanisms and improve their efficiency in resource management and real time performance. Efficient real time decision making is extremely important since it guarantees that users are served according to their current state and needs, effectively minimizing resource waste. ML is already proven a very powerful tool to improve network performance and decision making, which in turn improves network scheduling [11].

The massive data volumes produced in 5G Networks, should be utilized to improve network performance. All ML based techniques are dependent on data to produce results. They have different demands in terms of the quantity of data necessary or even the type of data available. We should always take into account the application of the technique and the desired level of accuracy produced from the model. For example, when using sensor data for autonomous driving, we should prioritize the accuracy of the produced data and their instant transmission and processing. Here we have an application where accuracy and QoS are extremely critical. Other applications might be more resistant to lower accuracy data.

In this paper we will utilize ML to improve user allocation and resource management on 5G Networks. We will propose three ML based mechanisms, that can work individually or collectively to achieve this goal. The first proposed mechanism is based on Decision Trees. It can be trained on data produced by already deployed networks to predict the optimal user allocation on any network implementation, removing the need to constantly assess metrics, as it is dictated on traditional user allocation techniques. In the second proposal, we suggest a model to predict the optimal positioning of SBSs in the network. This mechanism utilizes the K-Means clustering algorithm to produce cluster centers, whose coordinates we then suggest as candidates for repositioning SBSs, considering that user distribution, varies from day to day but it usually follows the same patterns. For the third mechanism we utilize Regression to try and predict the path followed by a pack of users. Our proposals can be utilized on deployed networks for instant user allocation or be utilized for efficiently expanding already deployed network infrastructures.

In remaining sectors, we present our full proposal. Section 2 focuses on the related work. In Sect. 3 we present the system model of our research and in Sect. 4 we will present the ML mechanisms. Section 5 presents the simulation parameters while Sect. 6 presents the simulation results. In Sect. 7 we deliver our conclusions and in Sect. 8 we make suggestions on how ML can be introduced in future research.

2 Related Work

The application of ML in computer networks has been steadily increasing. So far network application of ML in computer network includes among others the research in [12], where the authors suggest exploiting the data stored in the cloud to search for optimal or near optimal solutions on historical scenarios. They suggest classifying these solutions, extracting their similarities and utilizing them to produce a ML based resource allocation scheme, running on BSs to ensure efficient allocation of radio resources. In the same concept the authors of [13] train two neural networks to come up with a near-optimal algorithm to redistribute corporate users among BSs. In [14], the authors apply ML to create a decentralized solution for the combinatorial network optimization problem. For their work they utilize Multi-class support vector machines and artificial neural networks to perform network offloading. Their goal is to minimize network resource consumption based on user allocation and range extension.

Other notable works include the work of [15], where the Serf-Organizing Map (SOM) algorithm was applied on cellular networks. Their method was heavily based on data, modelling network trends, with the purpose of optimizing the performance of cells in future network infrastructures. In [16], we see an extensive research on how several ML algorithms are suited for use in cyber-security. This paper discusses several algorithms, some of which are existent in our paper as well, further strengthening the notion that ML is a powerful tool with many applications in computer and cellular networks.

With IoT deployed in multitude of scientific areas, the authors in [17] aim to utilize various IoT resources dynamically to facilitate user demands. They propose a service resource allocation approach that minimizes Device to Device (2D) data transmissions for all network users while trying to cope with all the restraints set by the desired application. In their research, the resource allocation problem is considered to be a variant of the degree-constrained minimum spanning tree problem. In the end, the authors efficiently applied a genetic algorithm in an attempt to reduce the time necessary to produce a near-optimal solution. In a similar fashion the authors in [18] proposed a genetic algorithm for load balancing in fog IoT networks. Their proposal yields interesting results, and offers acceptable performance especially when the number of users per cell remains low. Research is deemed necessary to improve the produced results that as the number of users per cell increases.

The paper in [19] proposes a joint sub-channel and power allocation algorithm for D2D communication based on Non-Orthogonal Multiple Access (NOMA) trying to maximize energy efficiency in the UL direction and increase the total throughput of the established communications. The algorithm uses the Kuhn-Munkres (KM) criterion to allocate resources for each D2D communication. The produced simulations showcase impressive results with the proposed algorithm outperforming current in-use algorithms both in terms of energy efficiency as well as throughput under different network conditions.

The survey of [11] showcases the state-of-the-art applications of ML in wireless communication and discusses several unresolved problems and issues that are still in need of research. They study several topics, surveying ML based approaches proposed for insuring the efficient operation of wireless networks. Utilizing ML will not only improve network performance, but it will also help all research fields that want to, or already utilize it. As stated in [20], network optimization can be proven beneficial for ML workflow and bring performance gains in the deployment of ML techniques.

3 System Model

For our simulations we consider a HetNet, with multiple McBSs and SBSs. All BSs are allowed to serve any user in their coverage area as long as a successful association is possible. We utilize the Normal and Uniform distributions to calculate the spawn points of the users to create a viable and life-like environment, where users are realistically placed in the network. In future network deployments we expect a massive rise in the number of UEs trying to utilize the network's resources. To produce realistic results, we will test our simulations with a varying number of users, ranging from 10 to 1000 users in the three proposed mechanisms. Starting from a small number of users and steadily increasing it allows us to check the limits of our networks as well as pinpoint its weaknesses in various congestion scenarios.

To accurately depict user demands, we consider that all users have predefined demands, namely a desired Data Rate (DR) and an expected Quality of Service (QoS). To solidify a life like environment, 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 have developed a network that features more SBSs compared to McBSs. These SBSs aim to alleviate congestion in areas with high user density, or in areas where McBSs offer little to no coverage such as indoor underground areas.

We begin by presenting the network layout. In our simulations we have a number of McBSs, represented as M (M = 1, …, |M|) and SBSs, represented as S (S = 1, …, |S|). The number of both McBSs and SBSs remains the same across all simulations. For all mechanisms, the positions of McBSs also remain the same, while the coordinates of SBSs are re-evaluated for the second mechanism. Our network also features a number of users (UEs) represented as U (U = 1, ….,|U|). All network connected users have predefined DR demands for both the DL and UL direction. As a result, traffic is split into two networks, one for transmitting and one receiving data (UL Network and DL network), as DUD dictates. We consider that all BSs, either SBSs or McBSs are independent, each one able to satisfy a user as long as it is deemed as the optimal BS to do so. All defined BSs have limited resources and consequently they can only satisfy a limited amount of users at the same time. All BSs of the same type have the same resources available, but the number of users they can serve varies based on the individual user metrics.

To compute the required RBs required by any user (e.g. user j) from the associated BS, we use the following formula:

$$RB_{j,i} = \frac{T_j}{B_{RB} \log_2 \left(1 + SINR_{j,i}\right)}, \tag{1}$$

where T_j denotes the UE throughput demands, B_{RB} is the bandwidth of any RB and $SINR_{j,i}$, is the SINR between a BS and an associated user. We consider that RBs cannot be split, so the result of the above formula is instantly increased to the nearest bigger integral. For both UL and DL rate calculation is as follows:

$$\mathrm{R} = \mathrm{BW} \log_2 (1 + \mathrm{SINR}), \tag{2}$$

where R is the produced rate, SINR is the Signal and Intereference to Noise Ratio and BW is the bandwidth.

To calculate Pathloss (PL) for any type of BS we will use the following formulas, following the distance dependent Pathloss model, where PL is based on the user-BS distance

and the pathloss exponent. We present two equations, one for the Pathloss when connecting to a McBS and one for when connecting to a SBS. Pathloss expresses the signal loss experienced by any user, comparatively to how strong the signal was when it was emitted from the BS.

$$PL_M = 128.1 + 37.6 \log_{10} d, \tag{3}$$

$$PL_S = 140.1 + 36.7 \log_{10} d, \tag{4}$$

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

The following formulas calculate the Signal to Interference and Noise Ration (SINR) for the DL and UL directions. This metric represents a ratio, namely the strength of the signal received by the receiving antenna and can be calculated as:

$$SINR_{i,j}^{DL} = \frac{P_{BS}}{N + I}, \tag{5}$$

$$SINR_{i,j}^{UL} = \frac{P_{UE}}{N + I}, \tag{6}$$

Here, $SINR_{i,j}^{XX}$ corresponds to the SINR between a user j and its corresponding BS i, P_{UE} corresponds to the 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 [21]. To calculate noise, we use the following formula:

$$\mathrm{N} = \mathrm{N}D^a, \tag{7}$$

where N is the noise power. D corresponds to the distance between the UE and the BS in question, while a is the Pathloss exponent, set as 3.6 for SBSs and 4 for McBSs. In our simulations fading and shadowing are ignored. All antennas of the same type feature the same transmit power set as 50 dBm for McBSs, 24 dBM for SBSs and 20 dBm for UEs. For our simulation, by applying ML we expect to improve real time decision on associating users and BSs, by minimizing the amount of calculations necessary. We expect to minimize the distance of users and SBSs and be able to predict user movement in the network to minimize wasting of network resources. All these are only possible through ML. According to [22], ML is the process of building algorithms that not only work on data to produce results but can also be trained on them to make predictions and improve. ML models do not follow static instructions like regular algorithms but make data-driven decisions. For our research we will utilize the three following ML techniques:

3.1 Decision Trees

A decision tree is a structure that creates a tree like presentation to model decisions and estimate the possible outcomes. It creates a rooted tree. It has a node called “root” with zero incoming edges and nodes feature only one incoming edge. All these nodes represent a test on an attribute, based on a designated function. In fact, they split the instance space

into two or more sub-spaces, effectively producing branches that represent the outcome of the test, a class label. Nodes that have no outgoing edges are called leaves or decision nodes and are placed at the end of the tree. In a decision tree, based on a designated function each internal node.

In data mining a decision tree is a predictive model, that can be used both for classifiers and regression models, but they can also be used for classification tasks. In Decision tress, the leaves hold the information we seek from our model, and in order to receive answers, we need to follow the path from the tree root all the way down to a leaf [23]. In other words, by following this path we can view the decisions made by the Decision Tree leading to the leaf depicted result. Decision Trees in our simulations are used main.

3.2 K-Means Clustering

Clustering is a method that is widely used for data mining and pattern recognition. It is a process of separating a set of points into groups that are referred to as "clusters". All points that are placed in the same cluster are considered to be similar. The notion of similarity between points represents the distance between the points. To measure the distance between points we can choose the desirable metric based on the data we want to group or the expected result we want to achieve. In this regard, points that have a small distance are placed in the same cluster and points with a large distance are placed in different clusters.

The K-means clustering algorithm is an unsupervised ML method. It assumes a Euclidean space, so the distance metric can be spatial and more specifically the Euclidean distance. It also demands a predefined number of clusters. This is a major issue with the K-means clustering algorithm, considering that it is not always feasible to know the required number of clusters [24]. Using the Euclidean distance as the distance metric creates as a result, ball shaped clusters, shaped around the center of the produced cluster. Euclidean distance can be calculated as:

$$\mathrm{D} = ||\mathrm{X}-\mathrm{Z}|| = \sqrt{\sum_{i=1}^{n} \left(x_i - z_i\right)^2}, \tag{8}$$

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

At the beginning of the algorithm, we need to assume some cluster centers, the same number as the number of clusters we expect to be produced. These centers can be random points, or we can select points from the dataset that are placed as far away as possible from one another. All the points in the dataset will then be assigned to their nearest cluster, meaning the cluster with which they have the least Euclidean distance. The clusters grow as new points are added and as a result, with each addition their center is constantly re-evaluated. The process can stop when we have no more points to be clustered, as long as the clustering is reasonable. If necessary, the procedure can be repeated to improve the produced clusters by fixing the cluster centers and re-examine all data points.

3.3 Linear Regression

A method belonging to regression analysis, is Linear regression. It is a statistical tool that can be applied to a dataset aiming to define and quantify the relation between the considered dependent and independent variables. Using a linear regression model, is preferred because of two main advantages. Firstly, the model is descriptive. That means that it is helpful in analyzing the intensity of the produced association between the dependent variable and the independent one. The model also allows for adjustment, meaning that it can adjust for the effect of covariates or the confounders [25].

In general, the linear regression analysis uses a mathematical formula to produce the association or relationship we mentioned before, between the dependent variable y and the independent variable x. Due to the nature of the mathematical formula used, the relationship is always presented as a line. The utilized formula is the following:

$$\mathrm{y} = \mathrm{bx} + \mathrm{c}, \tag{9}$$

where b denotes the regression coefficient and c is a constant.

3.4 Polynomial Regression

Regression analysis is widely used to identify the relationship between a dependent variable and one or more independent variables. It is a very powerful statistical tool that is extensively utilized in various scientific sectors. Polynomial Regression is preferred in occasions where have reason to believe that the relationship between the two variable is not linear but curvilinear. It is a special case of multiple regression, with only one independent variable X.

One-variable polynomial regression model can be expressed as polynomial regression model by the following formula [26]:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_k \chi_i^k + e_i, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n} \tag{10}$$

where k is the degree of the polynomial that is equal to the order of the model.

4 Proposed Mechanisms

In this sector we will present the three proposed mechanisms. Each mechanism targets to provide a solution for a different networking issue. The first mechanism utilizes ML to predict user allocation based on previous network data, the second mechanism can be utilized in predicting the best coordinates for placing network infrastructure while the third can be implemented to predict user movement in the network, when necessary.

4.1 Mechanism for Predicting User Allocation on BSs

The first mechanism is based on Decision Trees. This mechanism aims to predict the best allocation scheme for users on the available BSs. The mechanism requires a training dataset to be able to perform. The dataset should be produced by a deployed network, associating users to BSs, based on the desired metric. In this regard, we begin by creating a network and scatter users using the uniform distribution. As the preferred metric for association, we will use SINR in both UL and DL.

As a result, the association produced by our prediction model will match UEs to BSs based on SINR performance. When the network model has completed associating users and BSs, the produced allocation results will be saved on a dataset. The dataset includes the coordinates of each user in the network, as well as their matching BS for both the DL and UL direction. This dataset is then used to train our ML based model, as well as testing its prediction accuracy.

The size of the dataset is dependent on the number of users deployed in the network. We simulate our network with 200, 500 and 1000 users, and on each occasion a different dataset is created with the same size as the number of users in the simulation. The produced dataset will be split into two different datasets. The first one will be used for training the model while the second will be used to test its accuracy. We will test the mechanism with three ratios of training/test dataset relatively to the original dataset. This is a necessary step to examine issues of overfitting and underfitting for future deployments. After the training is

Pseudo code for the first mechanism

```
% U: Denoting the number of users
For i=1 to U do
    Calculate SINR for all BSs on DL, UL;
    Create BS preference list for DL, UL over SINR;
     Associate to optimal BS for DL, UL;
end for
% Produce dataset with user coordinates and DL, UL associated BS
% S: Denoting size of training dataset relevant to produced dataset
INPUT: Dataset with associations
For i=200, 500, 1000 do
     For size=0.1,0.2,0.4 do
          Train model
          Test model -Calculate precision
    end for
end for
OUTPUT: Predicted associations, train model precision
```

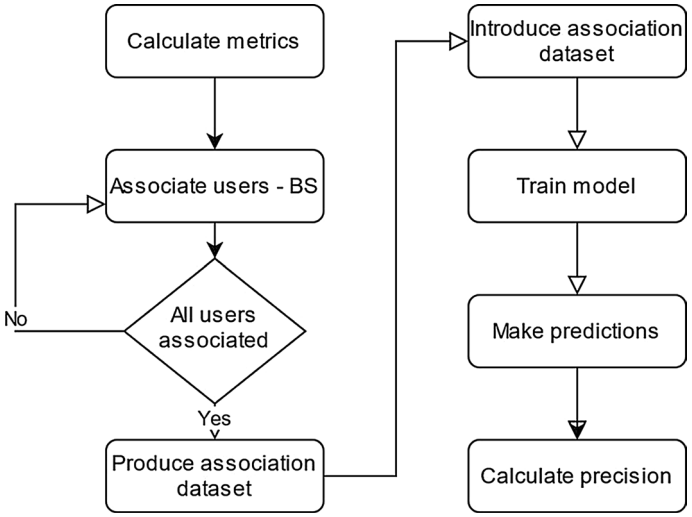

Fig. 1 Flow chart for first mechanism

completed, we use the test dataset to calculate its produced precision for all dataset sizes and assess its performance. In essence, we propose a mechanism that given historical data about the optimal associations between BSs and users, is able to predict future matchings without the need of performing the same number of calculations using network parameters (Fig. 1).

4.2 Mechanism for Efficient Placement of SBSs

The second proposed mechanism is based on the K-means Clustering algorithm. This technique can be used to identify the optimal position for placing (or replacing) SBSs across the network, to maximize the amount of users satisfied by our network and avoid network congestion. All users are distributed uniformly across the network, following a realistic deployment scenario and placing similar load on BSs on the entirety of the network.

All users deployed are then classified into groups called clusters, based on their coordinates. Users that are closer together, will be placed in the same cluster. To define the spatial distance between users we use the Euclidean distance. To cluster users we use the K-Means clustering algorithm. The algorithm begins with a set of empty clusters. By assumption, we consider the number of clusters to be the same as the number of BSs in the network. For our simulation we consider 42 BSs so we have 42 clusters.

The algorithm classifies users into clusters based on their spatial distance from the cluster center. When a new user is allocated into a cluster, the cluster center changes so its coordinates should be calculated again, taking into account the new user. The algorithm will not stop unless all users are allocated into a cluster. We can consider that each user

carries a weight, so all users should be considered in order to produce cluster centers that can be utilized in the prediction.

After the successful completion of the algorithm for each of these final cluster centers, we calculate their distance from all BSs to identify the closest BS. Considering that McBSs are stationary and cannot be moved, the coordinates of cluster centers that are close to McBSs are ignored, while the coordinates of cluster centers that are placed near SBSs, are considered to be the new coordinates for placing the SBSs. For example, in our simulations we can produce the optimal coordinates for 29 out of the 42 total BSs in the network. Finally, we restart the simulation with the new SBS coordinates. We will then compare the network performance using "random" coordinates for SBS positioning (the first case) and the K-Means based positioning scheme we proposed (Fig. 2).

4.3 Mechanism for Estimating User Movement in the Network

The third proposed mechanism is based on Regression, both linear and polynomial and it aims to define the most accurate method to predict how users move across the network. To showcase our mechanism, we only study a limited number of users, more specifically a set of 10 users, that want to move across the network together following a path, with no users stranding from the group. We consider that our users have a beginning position in the network and a destination.

Considering that these users are not stationary, we need to study the network across ten instances, a number enough to showcase the path they want to follow. To model the users' movement across the network, we used the Normal distribution. We decided to use that

Pseudo code for the second mechanism

```
% U: Denoting the number of users
% We consider 42 clusters
For i=1 to U do
    Associate user with a specific cluster;
    Calculate new cluster center;
end for
INPUT: New SBS coordinates
Calculate cluster centers distance with all BSs
Delete cluster centers closer to McBSs
For cluster centers closer to SBSs do
    Use remaining cluster centers as the new SBS coordinates
end for
Run model again with new SBS coordinates
OUTPUT: Association results with new centers
```

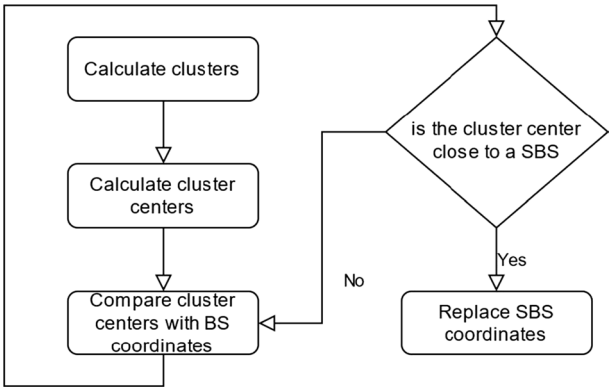

Fig. 2 Flow chart for second mechanism

distribution, firstly because we wanted to ensure that the users' movement was not random, ensuring this way that their path can be predicted. Secondly, we wanted to make sure that the set of users would move across the network in a single direction to avoid cases where users would just move about in a specific area.

In every instance of the network, this group of ten users can be considered as a cluster of users and is depicted as such. In more detail, we calculate the coordinates for all ten users, across all ten instances of the network. For each instance we cluster them and represent them by the cluster center which is used to showcase the group's path across the network. The users start from a specific location in the network. They want to move along the network. For the sake of our mechanism, we consider that our users, only move in a single direction and that they all move together. We study our users on ten instances of the network, starting from their beginning to their destination.

Our mechanism then uses both Linear and Polynomial regression to estimate the path that these users follow. After the path is estimated we will them compare the produced path from both models to the actual path, that the users followed. If we can successfully predict the path followed by the users, we can utilize this information to utilize the network's resources accordingly. For example, in our simulations we expect to produce a good approximation of the cluster centers for all ten time instances, so we can we can shut down all SBSs that are far away from the path followed (Fig. 3).

5 Simulation Setup

This section contains the parameters we used to model the 5G network. To simulate the network and implement our proposed mechanisms, we chose the Python programming language since it incorporates predefined functions and models for a plethora of ML techniques, making it a powerful and respectable tool for ML centric simulations. The produced network for the first two mechanisms can be seen on Fig. 4. This network consists

Pseudo code for the third mechanism

```
% U: Denoting the number of users (10)
For i=1 to U do
    Calculate path using Normal Distribution;
end for
% I: Denoting network instances (10)
INPUT: Actual user path
For i=1 to I do
    Cluster users in a cluster;
    Produce cluster center;
end for
User Linear, Polynomial Regression to predict user paths
OUTPUT: Predicted user path
```

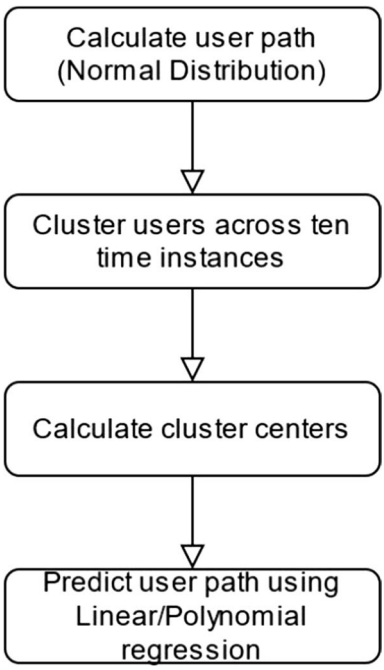

Fig. 3 Flow chart for third mechanism

of 13 McBS, 29 SBSs and indicatively 200 UEs. McBS are depicted by big triangles in the middle of each hexagon, SBSs are depicted as "Y" figures scattered along the McBSs' vicinity and UEs are depicted as colored bullets. Users depicted by the same color of bullets can be considered a cluster.

For the first two mechanisms we have the same number of users. The users are distributed in the network using the uniform distribution. This ensures similar load on all BSs across the network. Clustering of users is produced using the K-Means clustering algorithm, and user distance is counted using the Euclidean distance. For this simulation we consider the same number of clusters and BSs. This way we expect to extract conclusions about the optimal position for all BSs. In the same way if we want to minimize the number of BSs in the network (available only by reducing the number of SBSs) or increase it, we can use the same mechanism, with a smaller/larger number of clusters.

For the third mechanism, we intent to create a predictive model for the movement of users. In order to be able to follow their movement, we limit the number of users in the network to 10 users. These users are considered as a pack, meaning they all want to move from the same place to another place, following a similar path, with no user stranding from the others. The beginning position of the pack of users can be seen in Fig. 5 and their final position can be seen in Fig. 6.

To simulate their movement, we use the normal distribution. Our distribution choice enables us to simulate a single-direction path, that is not random, so that we can actually try to simulate and predict it. Users move along the x-axis of the network. To study their movement, we examine the network on ten time instances. On each instance, all the users have a distinct position in the network, that is different from their position in previous and next instances. For each instance, the pack of ten users, is considered to create a cluster, with a distinct cluster center that represents it. All 10 cluster centers indicate the full path followed by the users. Representing all users with a single point, allows us to use the Regression model to try and predict their path along the network. For all our simulations, the parameters can be seen on Table 1.

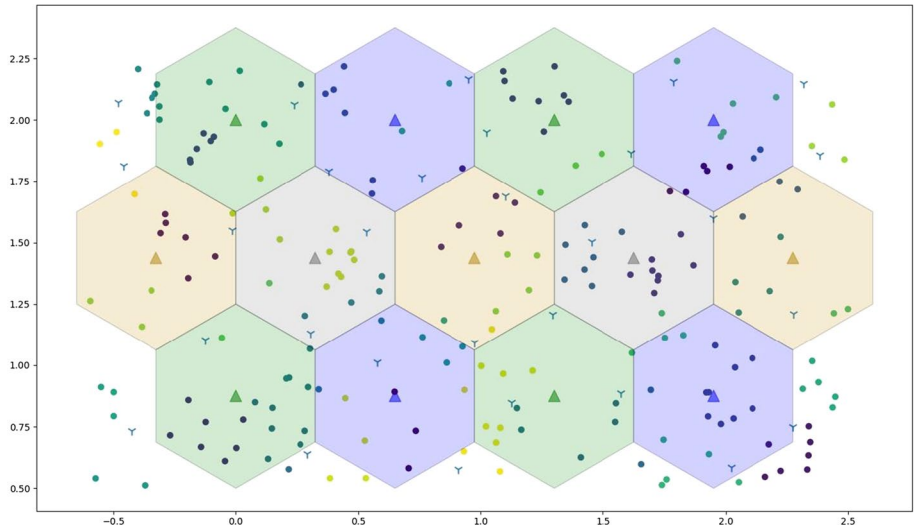

Fig. 4 Network deployment for the first two mechanisms

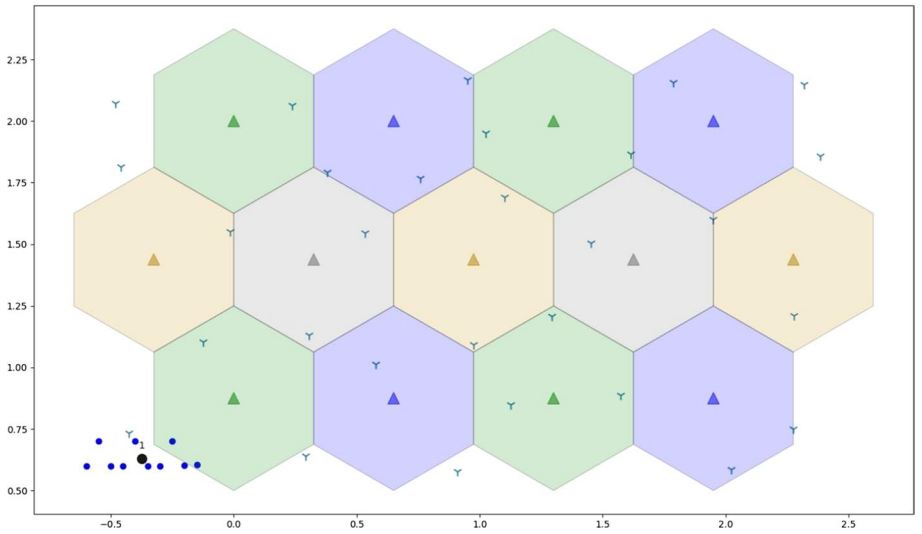

Fig. 5 Starting position for all users

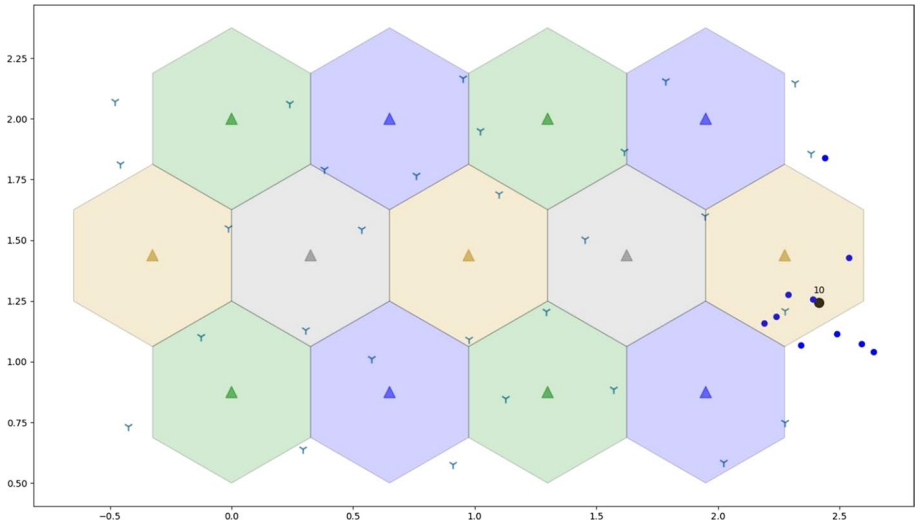

Fig. 6 All users reaching their destination

6 Results

In this section we will present the results produced from the three mechanisms. In all cases, the network topology and infrastructure remain the same. For the first two scenarios, we consider the network in a single instance, effectively meaning that users are static. The first mechanism is making predictions for the optimal user allocation, so

Table 1 Simulation parameters

Parameter	Setting
Macro cell Transmission Power	50 dbm
Smallcell Transmission Power	24 dbm
User Equipment Transmission Power	20 dbm
McBS Pathloss exponent	4
SBS Pathloss exponent	3.6
Network Deployment	13 Macro cells, 29 Smallcells
Number of users	10, 200, 500, 1000
Stationary User Distribution	Uniform Distribution
Moving User Distribution	Normal Distribution

we will study its prediction accuracy. In Fig. 7, we can see and compare the produced results for a dataset of 200 and 500 and 1000 users.

The dataset produced by the conventional networking model is split on two parts, a training set, used for training the model and a test set, used for testing it. In the results depicted, we consider a fixed relation between the size of the training dataset and the test dataset, with the first being 80 percent of the size of the original dataset and the second being 20 percent. Starting from 200 users, in the UL direction we get a precision value of 0.625 while DL direction gets a value of 0.85. Increasing the number of users (and the size of the dataset), we see a significant increase in the produced accuracy, with UL direction getting a value of 0.73 and DL direction getting a value of 0.92. Finally, with a dataset of 1000 users, we see another improvement, with the produced accuracy rising to 0.85 for the UL direction and to 0.935 for the DL direction.

Next, we want to understand how the size of the training dataset and test dataset (in relation to the original dataset) affects the produced accuracy. Starting with the DL direction, on Fig. 8 we see that the higher the size of the training dataset in relation the original dataset, the better the prediction accuracy is. For the cases of 200 users we see a massive improvement as the size of the dataset increases. After 500 users, we see that the

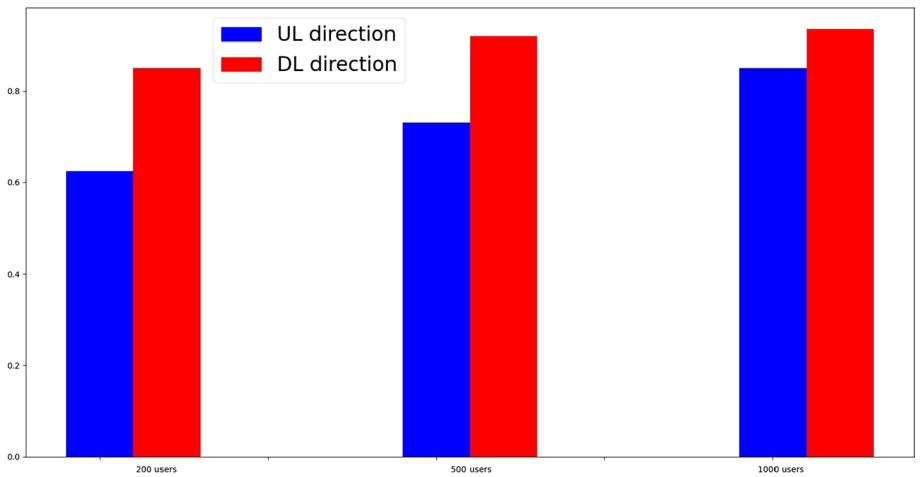

Fig. 7 Accuracy of predictions for DL, UL (test dataset of size = 20% of the original dataset)

differences begin to be limited. That means that with 500 users we have reached a point where, the original dataset is big enough to accommodate for higher splitting ratios, between training and test dataset.

In Fig. 9, we can see the produced results for the UL direction. Following the same trend as the DL direction, for 200 users we see a massive improvement in the produced accuracy when the size of the training dataset is 90% of the original dataset. For 500 users we see an anomaly as the accuracy is better when the test dataset is 40% of the original dataset rather than when it is 20%. Results like that are expected considering the complexity of such predictions, especially for the UL direction. For 1000 users, the model works as expected, with the accuracy being better as the size of the training dataset increases. The above results indicate that the dataset of 1000 users is a minimum requirement to avoid issues of underfitting in our model and our network.

Taking into account the size and complexity of our simulation environment, it is safe to say that for larger simulation environments a larger dataset is necessary to produce accurate results. The produced results showcase that using ML based techniques can be quite accurate for allocating users in the network. For associating users and BSs, many metrics are taken into account, making it quite a complicated procedure. In the results presented above we see that the predictions for the DL direction are far superior to the ones for the UL direction. Considering that most users are associated with McBSs for the DL direction and that McBSs are a minority in our network, we expect the model to be more precise in this case, since it has a small pool of BSs to choose from, for associating users. For the UL direction, we see that indeed an increase in the dataset size massively improves the prediction accuracy. That is to be expected, since a larger pool of available BSs, significantly complicates the decision process, and a larger dataset size enables our model to pinpoint all connections between the data parameters and understand the patterns on allocating users to BSs.

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. In real world

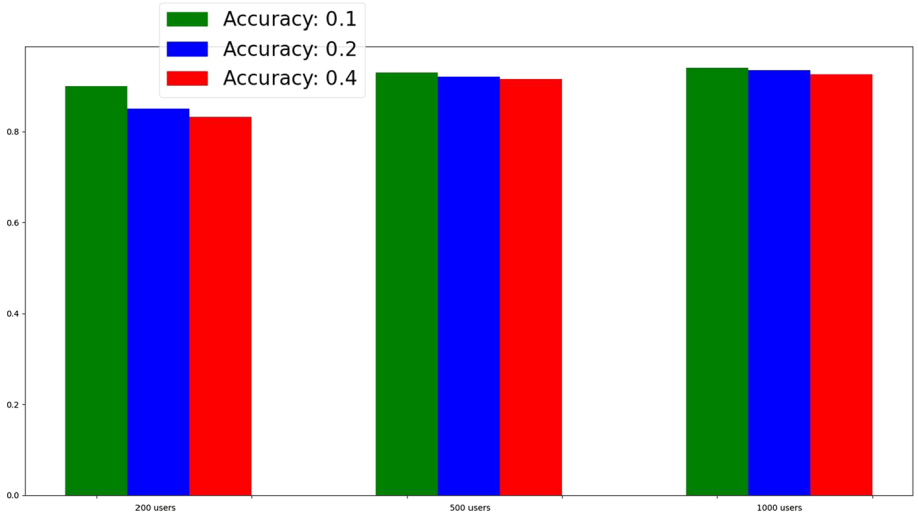

Fig. 8 Accuracy for the DL direction (test dataset of size = 10, 20, 40% of the original dataset)

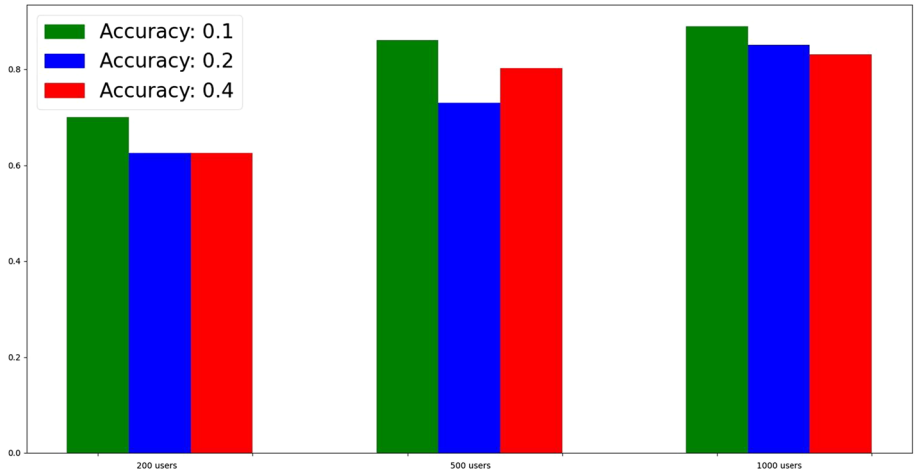

Fig. 9 Accuracy for the UL direction (test dataset of size = 10, 20, 40% of the original dataset)

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.

The produced clusters and their respective cluster centers produced from the second mechanism can be seen on Fig. 10. They are depicted as black bullets in the center of the clusters and all users belonging to the same cluster are depicted as bullets of the same color. Repositioning of McBSs is not possible so we only care about the cluster centers near SBSs. So we will ran our simulating using the cluster center coordinates as the SBSs' coordinates.

After running the simulation again, the proposal produces smaller average distances between cluster centers and the closest available SBS. Such a result guarantees less pathloss and signal deterioration for all allocated users. This along with the improved SINR values, since SINR is dependent on the distance between users and BSs, will result in increased DRs and total network throughput. The above ensure better overall usage of the available network resources. Users will fulfill their DR demands, requesting less RBs from their matching BSs, meaning that more users will be able to connect with each BS. Since cluster centers that are closer to McBSs, have not been considered, the average distance between users and McBSs remains the same, resulting in minimal changes in the DL direction, where most users are satisfied by McBSs.

Next, we study the number of successful user associations after using the coordinates produced by the second mechanism. In Fig. 11, we see that the number of associated users in the UL direction is improved over the previous simulation. Considering the size of the network and that the new positions are not very far away from the previous iteration, it is crucial that we still see an improvement over the random positioning of SBSs. These results are in order with the results in [21], where the authors attempt to better utilize SBSs

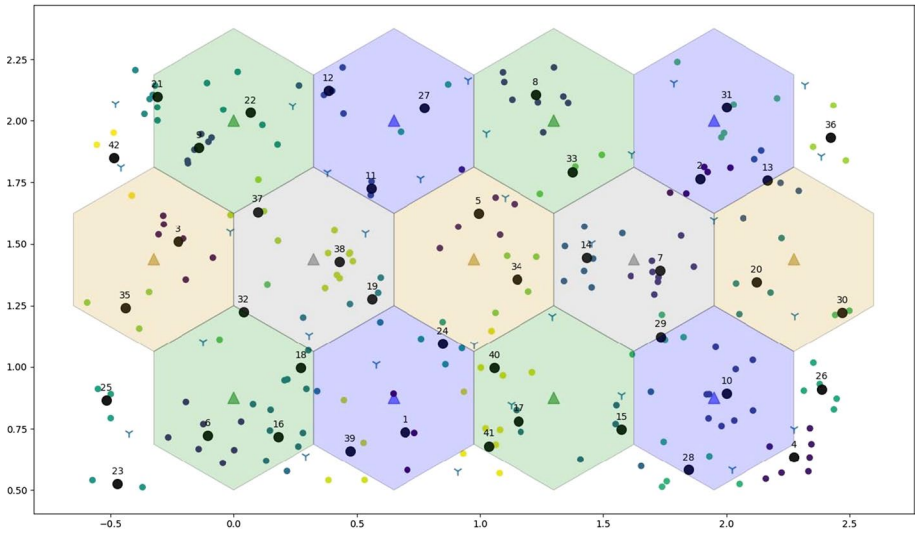

Fig. 10 200 users, split on 42 clusters

in the UL direction. Similarly with our proposal, SBSs yield better metrics for users, resulting in higher association levels in the UL direction where they are prominent. In a larger network, where SBSs will be placed in entirely different locations we expect the number of associations to see massive improvements. This is of course subject to small changes, since we can never fully model the spawn points of users in any network, at least until now.

Since the basic allocation mechanism remains the same in both cases, the users are matched with their optimal BS, meaning they enjoy the higher SINR available and QoS possible. The results indicate that our suggestion can indeed improve achieved performance, but mostly they indicate that our proposal can be utilized for refining or expanding

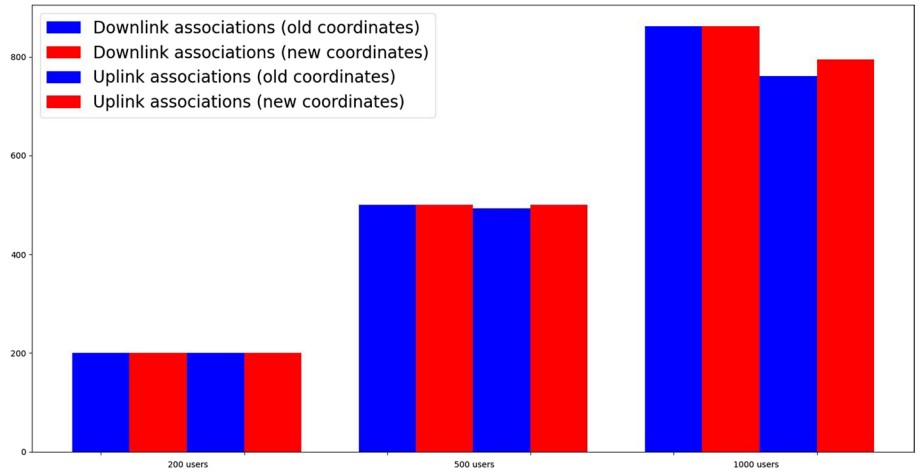

Fig. 11 Successful associations for various volumes of users

the existent network infrastructures, where SBSs are concerned. While the DL direction, seems relatively unaffected, our proposal will improve both UL and DL, since better SBSs placement can reduce occasions of overloading BSs. That will result in a higher offered QoS for both directions and an improved total network throughput.

For the third mechanism, we use two models. The first one is based on Linear Regression, while the second one is based on Polynomial Regression. We want to compare their performance and establish if there is a model best suited for our mechanisms' goals. In Fig. 12, we can see the users' entire path along the network. Our users set on a path that spans the entirety of the network. While our users constantly move until they reach their destination, we only depict 10 of their instances, that cover the entire path.

At first, we try to simulate their movement, using Linear regression. The results can be seen in Fig. 13. Linear regression can only predict values along a straight line. As we can see this model's performance is very poor for our desirable goal. The predicted path does not accurately predict the coordinates, for any cluster center, making this model unacceptable in occasions, where we need precision for the clusters' coordinates, such as occasions where we need to pinpoint users' location, e.g. outdoor localization for safety purposes.

Next, we study the results of Polynomial Regression as seen in Fig. 14. Polynomial Regression is able to predict with very high accuracy the path followed by the users. Its performance is far superior to linear regression. That is to be expected since the movement of the users, is never really linear. This method can not only depict the path with extreme accuracy but it is also able to pinpoint the exact location of the group at any instance. As a result, it is a very reliable method to use for safety applications where we want to locate users for safety reasons. Our results are similar to the research provided in [27], where the authors examined the accuracy of Polynomial regression and produced similar results.

In terms of network performance, such a result enables us to achieve massive energy gains. For example, if we can predict the path for all users in the network, we can pinpoint

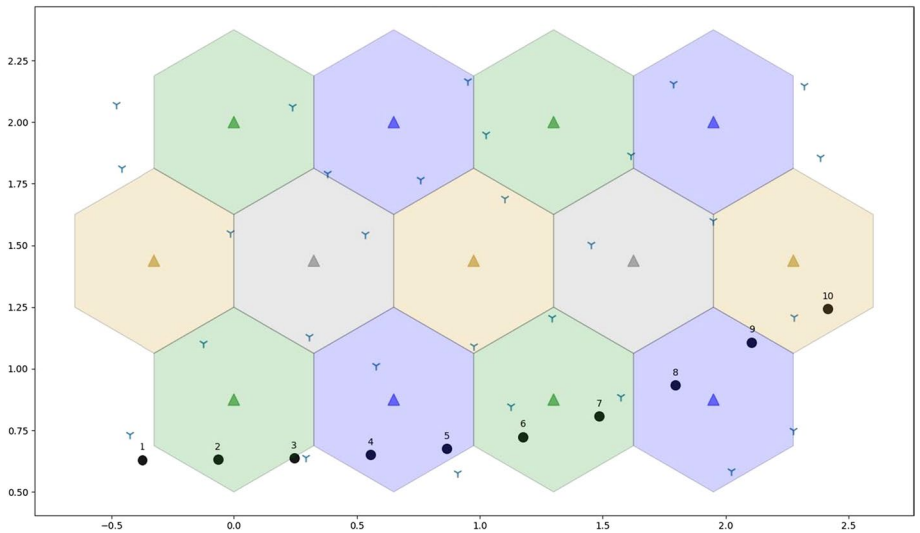

Fig. 12 Presentation of the path followed by the set of ten users

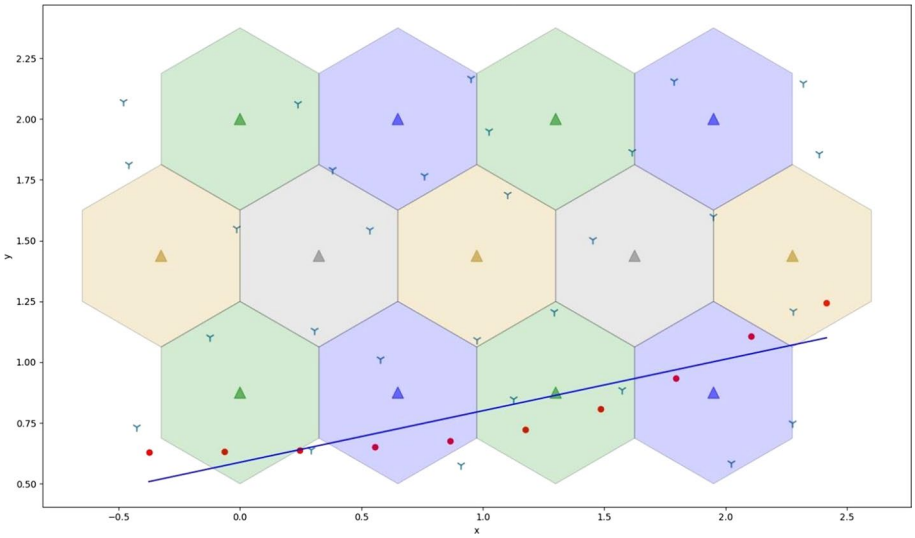

Fig. 13 Prediction with Linear Regression

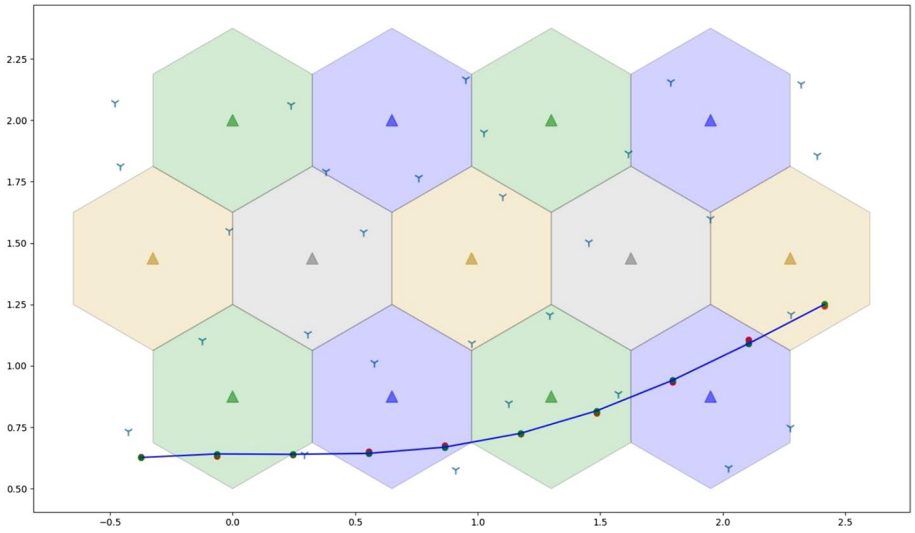

Fig. 14 Predictions using Polynomial Regression

all the BSs in the network that are so far away from the predicted path and shut them down to minimize energy waste. In our case we see that more than 16 SBSs can be shut down.

A comparison between the results produced from Linear and Polynomial Regression can be seen in Fig. 15. The instance predictions from Linear prediction are shown as black bullets, the ones from Polynomial Regression are shown as green bullets, while the predicted path is shown as red bullets. We can see how close Polynomial Regression is to accurately predict the path.

Our proposals show promising results for ML. It is therefore inevitable to integrate it in cellular networks. By doing so we can begin to see important benefits on both user allocation as well as resource management. These methods represent only a small fraction of the applicable ML mechanisms. Utilizing them can result in immediate user allocation, minimizing the usage of network resources, ensuring better decision making and providing high QoS for all devices in the network. The importance of predictions is immense and is dictated as we move towards more user-centric and smart network architectures.

7 Conclusions

Concluding our research, we are certain that with IoT emerging, the integration of ML in computer networks is undoubtfully going to increase network performance. The volume of interconnected devices is massively increasing, posing unprecedented burden in the network infrastructures and dictating a fruitful utilization of the available resources. Such a burden can significantly affect real time performance considering the amount of calculations needed for establishing the association between users and BSs and sharing of the network's resources. On the other hand, if optimal associations cannot be met, then we risk an inefficient utilization of the available resources, all of which massively impact the network's performance and the perceived QoS for the users.

Presenting our three mechanisms we expect to successfully prepare computer networks for the introduction of such a volume of devices, provide the means to efficiently manage the network's resources and improve real time performance. Our goal is to provide users with the experience they expect from the next generation of networks. With the models we presented above, we proved that ML models can be used to predict optimal user association with BS's in cellular networks, provided we have enough data to train the model on the preferred method of association. We provided a mechanism to optimally place SBSs in the network to complement the performance of the first mechanism and reduce waste of network resources. Finally, we implemented a regression mechanism to predict users' movement along the network, a method with massive energy gains. We proved that ML can be a significant tool in improving network performance with a wide variety of applications.

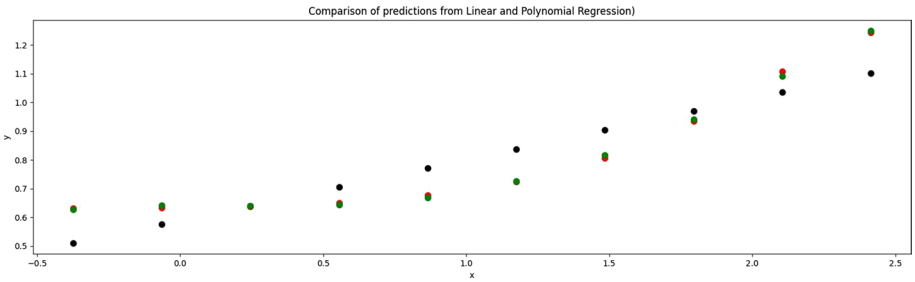

Fig. 15 Comparison of linear and Polynomial Regression performance

8 Future Work

Since ML techniques have been successfully applied to a plethora of other scientific sectors, introducing it to networking is inevitable. Its utilization will grow more with the rise in the capabilities of all network connected devices. This creates an opportunity to enhance existing mechanisms, either by improving their performance or by enriching them with new characteristics. Future implementation of ML techniques in cellular networks should focus in improving resource management (especially in the frequency domain), and decision making. Especially the last, covers multiple issues, like network security or user perceived QoS. ML can be implemented to instantaneously pinpoint malicious users that seek to sabotage the network, or users that bottleneck its resources.

Effectively managing such issues can have an exceptional result on network performance. While ML can be a useful tool, when choosing a ML technique for implementation, we should always take into consideration the application targeted, the nature of the data provided and how critical is the achieved accuracy for our model. When the above have been carefully established, ML will undoubtfully skyrocket network performance and unlock new possibilities for network managers and users.

Funding None reported.

Availability of data and material The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Code availability The authors produced custom code for the simulations.

Declarations

Conflict of interest The authors declare no potential conflict of interests.

References

1. Ghosh, A., Raha, A., & Mukherjee, A. (2020). Energy-efficient IoT-health monitoring system using approximate computing. *Internet of Things, 9*, 100166. https://doi.org/10.1016/j.iot.2020.100166
2. Mahbub, M. (2020). A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. *Internet of Things, 9*, 1–30. https://doi.org/10.1016/j.iot.2020.100161
3. Boostanimehr, H., & Bhargava, V. (2014). Unified and distributed QoS-driven cell association algorithms in heterogeneous networks. *Wireless Communications, IEEE Transactions on*. https://doi.org/10.1109/TWC.2014.2371465
4. Sun, S., Adachi, K., Tan, P. H., Zhou, Y., Joung, J., & Ho, C. K. (2015). Heterogeneous network: An evolutionary path to 5G. 174–178. https://doi.org/10.1109/APCC.2015.7412506.
5. Akyildiz, I., Nie, S., Lin, S.-C., & Chandrasekaran, M. (2016). 5G roadmap: 10 Key enabling technologies. *Computer Networks*. https://doi.org/10.1016/j.comnet.2016.06.010
6. Jain, A., López-Aguilera, E., & Demirkol, I. (2021). User association and resource allocation in 5G (AURA-5G): A joint optimization framework. *Computer Networks, 192*, 108063.
7. Feng, Z., Li, W., & Chen, W. (2015). Downlink and uplink splitting user association in two-tier heterogeneous cellular networks. *2014 IEEE global communications conference, GLOBECOM 2014*. 4659–4664https://doi.org/10.1109/GLOCOM.2014.7037543.
8. Shi, M., Yang, K., Xing, C., & Fan, R. (2017). Decoupled heterogeneous networks with millimeter wave small cells. IEEE Transactions on Wireless Communications. PP. https://doi.org/10.1109/TWC.2018.2850897.

9. Prasad, N., & Rangarajan, S. (2017). Exploiting dual connectivity in heterogeneous cellular networks. *2017 15th International symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WiOpt), 2017*, pp. 1–8. https://doi.org/10.23919/WIOPT.2017.7959889.
10. Hussain, F., Hassan, S. A., Hussain, R., & Hossain, E. (2020). Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. In *IEEE communications surveys & tutorials*, Vol. 22, No. 2, pp. 1251–1275, Secondquarter 2020. https://doi.org/10.1109/COMST.2020.2964534.
11. Sun, Y., Peng, M., Zhou, Y., Huang, Y., & Mao, S. (2019). Application of machine learning in wireless networks: Key techniques and open issues. *IEEE communications surveys and tutorials*. PP. 1–1. https://doi.org/10.1109/COMST.2019.2924243.
12. Wang, J.-B., Wang, J., Wu, Y., Wang, J.-Y., Zhu, H., Lin, M., & Wang, J. (2017). A Machine learning framework for resource allocation assisted by cloud computing. *IEEE Network*. https://doi.org/10.1109/MNET.2018.1700293
13. Yang, J., Wang, C., Wang, X., & Shen, C. (2018). A machine learning approach to user association in enterprise small cell networks. 850–854. https://doi.org/10.1109/ICCChina.2018.8641148.
14. Fa-Long, L. (2020). Machine learning for optimal resource allocation. In Machine learning for future wireless communications, IEEE, 2020, pp. 85–103. https://doi.org/10.1002/9781119562306.ch5.
15. Mom, J. M., & Ani, C. (2013). Application of self-organizing map to intelligent analysis of cellular networks. *ARPN Journal of Engineering and Applied Sciences., 8*, 407–412.
16. Das, R., & Morris, T. (2017). Machine learning and cyber security. 1–7. https://doi.org/10.1109/ICCECE.2017.8526232.
17. Kim, M., & Ko, I. (2015). An efficient resource allocation approach based on a genetic algorithm for composite services in IoT environments. *2015 IEEE international conference on web services, 2015*, pp. 543-550.https://doi.org/10.1109/ICWS.2015.78.
18. Oueis, J., Strinati, E. C., & Barbarossa, S. (2015). The fog balancing: Load distribution for small cell cloud computing. *2015 IEEE 81st vehicular technology conference (VTC Spring), 2015*, pp. 1–6. https://doi.org/10.1109/VTCSpring.2015.7146129.
19. Alemaishat, S., Saraereh, O. A., Khan, I., & Choi, B. J. (2019). An efficient resource allocation algorithm for D2D communications based on NOMA. *IEEE Access, 7*, 120238–120247. https://doi.org/10.1109/ACCESS.2019.2937401
20. Cheng, Y., Geng, J., Wang, Y., Li, J., Li, D., & Wu, J. (2018). Bridging machine learning and computer network research: A survey. *CCF Transactions on Networking*. https://doi.org/10.1007/s42045-018-0009-7
21. Elshaer, H., Boccardi, F., Dohler, M., & Irmer, R. (2014). Downlink and uplink decoupling: A disruptive architectural design for 5G networks. https://doi.org/10.1109/GLOCOM.2014.7037069.
22. Simon, A., Deo, M., Selvam, V., & Babu, R. (2016). An overview of machine learning and its applications. *International Journal of Electrical Sciences and Engineering,* Volume. 22–24.
23. Rokach, L., & Maimon, O. (2008). Data mining with decision trees. *Theory and Applications*. https://doi.org/10.1142/9789812771728-0001
24. Cheung, Y.-M. (2003). K*-means: A new generalized k-means clustering algorithm. *Pattern Recognition Letters., 24*, 2883–2893. https://doi.org/10.1016/S0167-8655(03)00146-6
25. Kumari, K., & Yadav, S. (2018). Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences,* 4. 33. 10.4103.
26. Ostertagova, E. (2012). Modelling using polynomial regression. *Procedia Engineering., 48*, 500–506. https://doi.org/10.1016/j.proeng.2012.09.545
27. Al-Hattab, M., & Agbinya, J. (2010). Trajectory estimation for wireless mobile networks using polynomial regression. *International Journal of Electronics and Telecommunications., 56*, 451–456. https://doi.org/10.2478/v10177-010-0061-9

Christos Bouras is Professor in the University of Patras, Department of Computer Engineering and Informatics. Also he is a scientific advisor of Research Unit 6 in Computer Technology Institute and Press—Diophantus, Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Mobile and Wireless Communications, Telematics and New Services, QoS and Pricing for Networks and Services, e-learning, Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published more than 450 papers in various well-known refereed books, conferences and journals. He is a co-author of 9 books in Greek and editor of 2 in English. He has been member of editorial board for international journals and PC member and referee in various international journals and conferences. He has participated in R&D projects.

Rafail Kalogeropoulos was born in 1995 in Patras, Greece. He speaks English fluently and he obtained the Michigan Certificate of Proficiency in 2009. In 2013 he entered the Computer Engineering and Informatics Department in Patras and received his diploma in 2018. Currently he is a post graduate student in the same department. His field of interest consists of Computer Networks and Protocols, design and analysis of Algorithms and Discrete Mathematics. Since 2018 he is a member of the RU6 unit, studying next generation mobile communications Networks.