Clustering Based User Allocation in 5G Networks*

1st Christos Bouras*
*Computer Technology Institute & Press "Diophantus"
†Computer Engineering and Informatics Dept.
Univ. of Patras, Greece
Patras, Greece
bouras@cti.gr

2nd Rafail Kalogeropoulos†
*Computer Technology Institute & Press "Diophantus"
†Computer Engineering and Informatics Dept.
Univ. of Patras, Greece
Patras, Greece
kaloger@ceid.upatras.gr

3rd Ioannis Sina†
*Computer Technology Institute & Press "Diophantus"
†Computer Engineering and Informatics Dept.
Univ. of Patras, Greece
Patras, Greece
st1059610@ceid.upatras.gr

***Abstract*—Machine learning is an extremely efficient technique for solving complex problems without the use of traditional programming but rather enabling machines to learn from an input of data and train them to cope with various problems. The rapid growth in the number of active mobile devices, mobile applications and services dictates an efficient utilization of mobile and wireless networking infrastructure. Communication networks need to evolve and valorize machine learning methods in order to process large volumes of data without introducing excessive time delay in these computations. Upcoming 5G systems are expected to be the first network infrastructure to support exploding mobile traffic volumes and machine learning techniques can be used in order to help manage the rise in data volumes. We present a mechanism for resource allocation in mobile and wireless networks, that effectively utilizes machine learning techniques.**

***Index Terms*—Machine Learning, 5G Networks, Mobile Networks, Deep Learning, Mechanism, Allocation, Resources**

I. Introduction

With the rise of Internet of Things (IoT) and the introduction of extremely capable mobile devices [2], all cellular networks face an exponential rise in the number of network connected devices that utilize the network's resources as well as the volume of data transferred through the network's infrastructure. All generations of networks share the same design foundation. They consist of a plethora of Base Stations (BSs) of different types. These BSs can be Macrocell Base Stations (McBSs), BSs that feature the same characteristics throughout the entire network and Smallcell Base Stations (ScBSs), BSs that can be further categorized based on their transmit power. In previous generations of networks, McBSs were the prominent BS type, because they feature high transmit power, which ensures high Signal to Interference plus Noise Ratio (SINR). Their similar characteristics also extend to the number of simultaneous users they can accommodate [4].

Older generations of networks, namely Homogeneous Networks, featured mainly McBSs featuring limited capabilities. This dated approach falls behind in coping with the massive data needs of current devices. Homogeneous networks have been replaced with heterogeneous networks (HetNets). The deployment of such networks is deemed necessary in order to adequately correspond to current expected user throughput. Heterogeneous networks consist of prominent McBSs and smaller BSs, such as Femtocell or nanocell BSs that are distributed along a Macrocell Base Station's vicinity [7].

The implementation of HetNets has been undergoing since past generations, but officially started with the introduction of 5G networks, effectively setting the foundations for further research to study technologies that can solidify its application advantages. Predictions suggest a new user-oriented approach on mobile network architecture is necessary to cope with the constant rise in feature-rich user demands and ensure accessibility for all connected devices while guaranteeing a respectable Quality of Service (QoS) in real-time [12]. This approach calls for new techniques to be adapted, to optimize the allocation of network resources and improve real time performance.

Some of the most promising techniques are Machine Learning (ML) based. The application of ML is blossoming, pervading all scientific fields and delivering solutions to more and more complex problems [1]. Some problems that ML is aiming to solve are spam detection, product recommendation and image recognition [10]. As for networks, in [11], self-organizing maps applied to cellular networks have been proposed to dynamically shape the connectivity of the network to the prospective demands. The survey in [13], discusses multiple open issues on ML applications for computer networks and potential problems that arise with them. Moreover, the authors in [17] suggest an application of ML techniques to explore unknown guest user dynamics for a network of users with different demands. These studies show that ML has increased its fields of influence with the addition of networks, aiming to more efficient and smart algorithms with the minimum number of resources.

Taking into consideration these studies, in this paper we will propose a mechanism to tackle the issue of Resource Allocation in networks. We propose utilizing ML techniques, more specifically a K-means based clustering method to improve real time user allocation on BSs while minimizing the volume of network resources necessary to do so. This algorithm assigns users into clusters that are assigned to their nearest Base Station, based on the calculated distance from the cluster's center to the BS in question, resulting in faster service with better data rates and lower energy consumption. Installing new Base Stations is a costly procedure and requires careful planning for the proposed positioning of these stations. The proposed mechanism aims to indicate the best pattern of allocating users to existing stations, minimizing operation costs for the network.

In the remaining sectors, we will complete our proposal. Specifically in Sector II we will present the simulation system model. In Sector III we present the ML based mechanism that we suggest for application. In Sectors IV and V we will analyse our simulation setup and the simulation results respectively. Finally, in Sector VI we draw our conclusions and we make our suggestions for future work.

II. System Model

With the emergence of new technologies, we see the development of new types of ScBSs, making Small Cells smaller improving their deployment capabilities. This fact, greatly increases the gap in transmit power between McBSs and ScBSs, but it also means that installing and utilizing more ScBSs is becoming easier and cost effective, making it critical to optimally match a User Equipment (UE) with the BS best suited to accommodate its UpLink/DownLink (UL/DL) needs. The increase of the volume of ScBSs also increases the amount of calculations needed for user allocation.

ML already applies to many scientific fields, delivering solutions to more and more areas of our daily lives [14]. Its integration in computer science is expected to help us address dynamic problems, such as real-time distribution of network resources, acceptable QoS and real time data based decisions. ML techniques applicable, vary according to the type of data available and the specifications of the problem addressed. They enable us to create models specifically designed for each network and its corresponding needs. With this approach ML based techniques can be developed to decide on how a UE can be matched to a BSs, in a network which constantly evolves based on the predicted user needs.

First we shall define the basic network layout. We consider a Heterogeneous Network, consisting of a set of McBSs, a set of ScMSs of the same type (S=1,...,$\mid S \mid$) and a set of UEs (U=1,....,$\mid U \mid$), which are the users that utilize the network's resources. Users expect to either transmit or receive data or both. In this regard, in HetNets, traffic can be split into two networks (UL Network and DL network) that are considered as two different channels in the network. All BSs can only serve simultaneously a limited volume of users quoted as n_i, i=1,...,$\mid U \mid$ and also feature a limited volume of available Resource Blocks (RBs). These characteristics are the same for all BSs belonging to the same type.

To calculate the number of RBs, a user needs, we use the following formula:

$$R_{j,i} = \left\lceil \frac{g_j}{B_{RB} * log_2[1 + SINR_{j,i}]} \right\rceil, \tag{1}$$

where g_j is the user's j throughput demands and $R_{j,i}$ is the number of RBs that user j needs from BS i.

To compute the siganl to interference and noise ratio (SINR) presented in the above formula, we can use the following one:

$$SINR_{\mathrm{i,j}} = \frac{P_j g_{\mathrm{i,j}} d_{\mathrm{i,j}}^{\text{-a}}}{\Sigma_{\mathrm{kB}}}, \tag{2}$$

The main focus of applying ML is to match users and BSs, with the minimum amount of calculations to reduce time complexity and enable better real time decision performance for the network. The load should be efficiently distributed among all BSs, retaining an acceptable QoS level for the users. In a decoupled model each user connects to its optimal BS for both DL and UL, based on a specific network metric. In our case the metric chosen for DL and UL connection between a user and a McBS is SINR. The goal is that the selected BS will be able to provide the desired Data Rate (DR) to the user utilizing the minimum amount of RBs.

In our simulation the Data Rate(DR) for each user can be calculated using the following formula:

$$DR = B_{RB} * log_2[1 + SINR_{j,i}], \tag{3}$$

where B_{RB} provides the bandwidth of a given RB.

In our research we will try to simulate a realistic network scenario. We assume that McBSs are mostly mounted above rooftop level, providing larger coverage than ScBSs. ScBSs are placed on ground level city-wide, all belong to the same size and their main purpose is to satisfy indoor users and cover up for Non-Line-Of-Sight (NLOS) scenarios, especially critical for stationary indoor users or others roaming the streets. A SBS-rich network should be able to provide high data throughput in conditions of high user density, by increasing spatial reuse and thus reducing the number of users per cell.

The capabilities of ML, such as classification and prediction, are expected to play an important role in network performance. They can be utilized on building accurate predictive models to represent complex system and user behaviors, covering a plethora of scenarios. Each network scenario may have different characteristics (e.g. current network state). Figuring out what factors can directly affect the network, its performance and its evolution is a challenging task, especially when trying to valorize its resources. ML can provide the tools to produce an estimation model that can be utilized in networks with acceptable accuracy and provide new possibilities in improving the network resource allocation.

A promising ML technique that can be used extensively in cellular networks is clustering. Clustering is the process of examining a set of points and grouping them into clusters according to some distance measure [16].

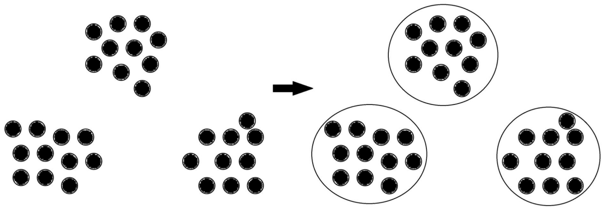

Fig. 1. Clustering network users into three clusters to be served by different base stations.

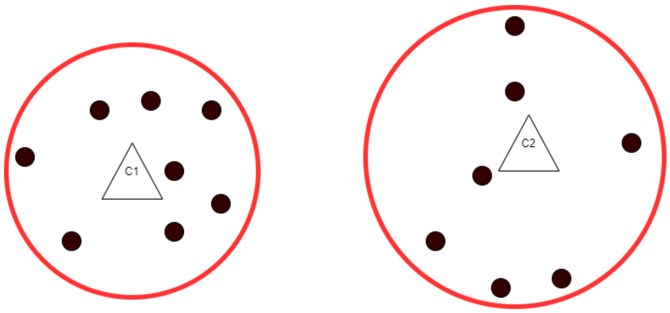

Fig. 2. Users assinged to McBSs after clustering.

III. The Mechanism

A. Clustering

ML enables us to classify objects into groups with similar characteristics, predict possible outcomes and many other possibilities. In wireless communications, a huge amount of unexploited data is produced every day, concerning user status or the communication infrastructure exploited. Utilizing this set of data can be proven extremely useful in improving network performance [15]. Classifying objects into groups, namely clusters is a procedure called Clustering, where objects are classified based on a metric, called distance. This metric can be the objects' actual spatial distance or a different metric, symbolizing their similarity. Objects with a small distance are placed in the same cluster, where objects with a big distance are placed on different clusters. The clusters can be of various shapes, that are dependent on the metric used. For example clusters, produced with the Euclidean distance are often shaped like circles. Clustering seems to be less effective in occasions where we have high-dimensional spaces, where most points seem to have similar distance between them. For our mechanism, we will utilize the K-means clustering algorithm, presented below.

B. K-means clustering

K-means clustering algorithm [9] starts by calculating the euclidean distance of all points (users in our case) from all initial clusters (McBSs in our case). Then a new centroid is calculated for every cluster and the same process is repeated. The algorithm terminates when there is no change between the penultimate and the last iteration. It is mainly used in fields such as Data analysis and Data mining but only for numeric values [8]. While there is a plethora of clustering algorithms, K-means has some advantages which are enough to make us choose it. The first advantage is that K-means guarantees convergence [5] which is essential in real time problems (our simulation). Moreover, the chosen algorithm scales to large data sets (users in our case) meaning that the number of users in a network does not affect the convergence of K-means. The last advantage is that the initial centroids' position can be set from the beginning which is crucial for our simulations as McBss(represented by centroids) cannot change position in the real world.

C. The algorithm

The proposed algorithm aims to produce optimized results and at the same time be stable, without taking into consideration the number of users in the network. Initially, there is no association between users and BSs. Furthermore, we consider that there is no upper bound to the number of assigned users (to a BS) and that the chance of connecting to all BSs for all of them is equal. Two lists consisting of BSs are created for each user (one for UL and one for DL direction). The preference lists are sorted using the SINR metric for both UL and DL direction. From now on we will refer to one list as both lists work in the same way. The first element in a user's preference list represents the most desired BS for UL and DL respectively, meaning that a possible connection with this particular BS will lead to optimal results. In cases which users try to connect to ScBSs in order to offload McBss, range extension is applied on the UL channel, resulting in users' greater demand for ScBSs than for McBSs. As for the selection, users will be able to connect to a BS whose RBs are enough in order to serve them based on their needs. Each user should let their desired BS know the number of RBs which are needed for the association.

We use UL_j for UL and DL_j for DL direction to represent a successful association between a user (user i) and a BS (BS j). In other words, each BS has a number of users to serve for both UL and DL direction. A list is created for every BS consisting of users who prefer it, stating possible connections. Initially we can assume that there is no reason for a user to connect to a BS or for a BS to serve a user.

In the main phase, a preference list has been created for each user based on the procedure described above. At this point, users have to submit their requests to the first BS of their preference list(for both directions) as it is the most desired. The optimal scenario for users is a successful connection with their most preferred BS for both directions. This is not always possible as the BS may not be able to handle a new association. As we mentioned above, a BS is able to reject a possible connection when there are not enough RBs to cover the user's needs. In this case, taking into consideration that the users' preference list is sorted in descending order, meaning that the first BS is the optimal choice and the last one is the worst choice, the rejected user tries to connect to the next BS until he finds the one to be served.

It is essential that every user will have an available BS to connect to, for both directions. Therefore, every user in our network should connect to at least one BS if the same BS serves both directions, or at least two if one BS covers UL and another one covers DL. Having provided their most desired BS with the required amount of RBs which they demand, users await response. After this process, a preference list which consists of all the possible connections between the BS and the user is built by every BS, based on the information they just received (needed RBs). Users are accepted by BSs until the number of RBs left is not enough to serve any of them which makes the number of connections limited. Before applying the clustering algorithm in our simulation, we have to describe our matching algorithm which checks if a connection between a BS and a user is possible.

Algorithm 1 Pseudocode for the Matching Algorithm

{Matching algorithm is executed for all users}
Preference list created for DL direction for U_i;
Preference list created for UL direction for U_i;
Calculate number of RBs to achieve wanted Rate for U_i;
Provide most preferred BS with number of,
RBs needed for UL and DL;
for BS in preference list of DL **do**
  {Iteration of preference list}
  if $BS's$ RBs $>$ $U_i's$ needed RBs **then**
    $BS's$ RBs = $BS's$ RBs — $U_i's$ needed RBs;
    break;
  end if
end for
for BS in preference list of UL **do**
  {Iteration of preference list}
  if $BS's$ RBs $>$ $U_i's$ needed RBs **then**
    $BS's$ RBs = $BS's$ RBs — $U_i's$ needed RBs;
    break;
  end if
end for

Having the matching algorithm ready, we can now apply the K-means clustering algorithm to our simulation. We have to mention here that each preference list consists of one McBS (centroid of K-means algorithm) and of ScBSs whose euclidean distance from the centroid is smaller than 50. The initial and final centroids of the algorithm are in the same position as we refer to McBSs which cannot change position in the real world. The only drawback in using K-means clustering algorithm is that we do not know the optimal number of clusters that have to be created [3]. After conducting tests we concluded that we had optimal results by setting the number of clusters to be equal to the number of McBSs. Applying the K-means algorithm to our network setup will create as many clusters as McBS and users will have been allocated to one of them according to their original position. Now we can treat the clusters as new independent networks and apply the above described matching algorithm to every one of them. Users will connect to their nearest McBS or ScBS resulting in more successful connections and better experience.

Let C_i, (i=1,....,| M |) be the clusters of our simulation after the K-means clustering whose centroids represent the McBSs [M, (M=1,....,| M |)] of our network setup.

Algorithm 2 Pseudocode for Final Algorithm

Applying K-means Clustering algorithm with,
our McBSs to be the initial centroids;
for $i = 1$ to C **do**
  Calculate DL demands for every user in C_i;
  Calculate UL demands for every user in C_i;
  Run the Matching Algorithm for every user in C_i
end for

IV. Simulation Setup

In this sector we will try to model a 5G enabled network, using a MATLAB based simulator to perform the simulations necessary for our proposal. For our ML simulations, MATLAB is proved a really useful tool, since it incorporates a set of standard ML functions that supplement our simulator . Matlab provides a GUI for the simulation of our network and the verification of our results. Matlab is widely preferred for network simulations, and its incorporation of ML functions dictates its use for the research provided in this paper.

TABLE I
SIMULATION PARAMETERS

Table	
Parameter	**Setting**
DL Bandwidth	100MHz
UL Bandwidth	100MHz
Network Deployment	29 McBSs and 45 ScBSs
Number of users	100/200/500/1000

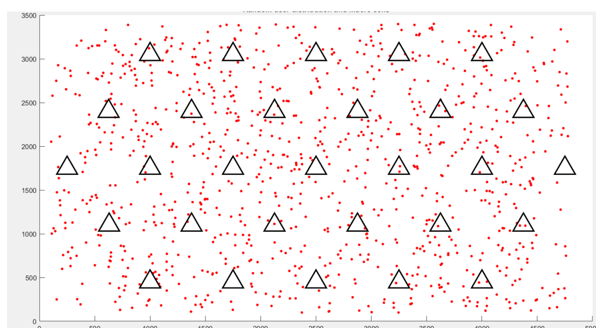

Fig. 3. Our simulation in MATLAB, consisting of 29 McBSs and 500 users.

MATLAB allows us to perform evaluations that are impossible to perform on real life systems and enables us to fully comprehend the performance of our mechanism in a highly controlled, reproducible environment.In figure 3 we see the network setup before the application of the clustering algorithm. The black triangles represent the McBSs and the red dots show the initial users' position. Our goal is to illustrate a real world network by setting users' and McBSs' initial position randomly using uniform distribution.

V. RESULTS

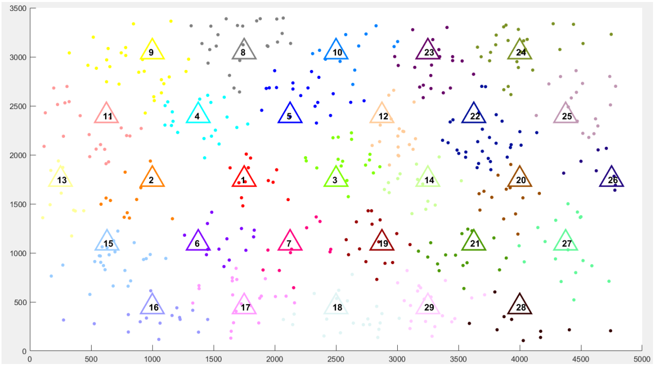

Fig. 4. Our simulation in MATLAB, consisting of 29 McBSs and 500 users, after K-means application.

The first figure presents the clustering results. As we can see one cluster is produced for every BS available in our network. Each item allocated in a cluster is expected to have an acceptable yet minimal distance from the cluster center. Examining the cluster centers produced in figure 4, and comparing them to the position of the network BSs, we can see that all cluster centers are located closely to the McBS related to the cluster. This proves that using clustering to allocate users produces very promising results, alleviating the need for extremely complex algorithms and techniques for user allocation, especially in occasions where small time complexity is of the essence.

Considering the shapes of the clusters, we see that they vary in shape, while always retaining a relatively small size. That means that the distance between each user and its respective BS is quite small, ensuring that all UEs connect to a nearby BS. This is extremely important, because a small distance ensures higher SINR. This in turn suggests that we can expect excellent data rates for all connected users. Smaller distances also yield promising results in energy consumption. UEs can limit their transmission power since they are connected to a nearby BS, drastically diminishing the energy wasted for signal reception and ensuring better battery life.

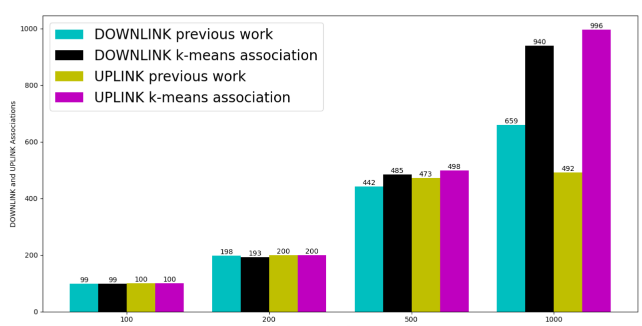

Fig. 5. Association for previous work over K-means association as DL and UL metric.

In figure 5 we can see the total number of user associations in the DL and UL direction for 100, 200, 500 and 1000 users. Clustering produced excellent results, always showcasing better performance than traditional methods based on network metrics like pathloss for associating users [6]. This metric is heavily affected by distance, meaning that a user connecting to a nearby BS will enjoy far greater performance than a user connected to a remote BS. Clustering takes distance into account in order to produce the cluster by choosing users with the smallest possible distance to add to the cluster. Therefore, ensuring a smaller distance enables better resource allocation that allows for more users to connect. Ensuring higher user association with acceptable data rates is the most effective way to combat network congestion and reduce the amount of users left with no coverage.

In the last two figures (6, 7) we can see the comparison of data rates between pathloss user association and our proposed association scheme. As shown below, the data rates we achieve after the clustering algorithm are higher in most cases, both for the DL and the UL direction. Using clustering shows vastly improved results, and manages to keep data rates steady across all possible congestion scenarios. On the other hand, using network metrics for user association produced data rates that work better in specific scenarios only. The difference between the values produced by our proposal is minor, but taking into consideration the improved number of associations we achieved for both UL and DL cases, our approach maintains and quite possibly increases the total network throughput utilizing the same amount of resources as previous works regardless of the network congestion.

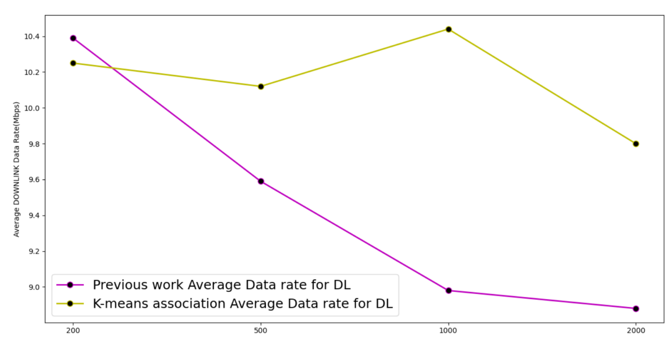

Fig. 6. Average DRs for previous work over K-means association as DL metric.

The produced results indicate that using clustering for user allocation is an approach that can drastically improve network performance. The produced results seem to be unaffected by the number of network users. Clustering assures allocation of multiple users on BSs with a minimal amount of calculations. Therefore the number of calculations for each McBS is reduced leading to less computational cost. As the number of network users rises, clustering is able to maintain steady data rates with the percentage of associated users remaining steady as well. This shows that in real life, this method can offer significantly improved yet stable quality of service to the users, provide a steady utilization of the network's resources and decrease network congestion. This in turn can drastically

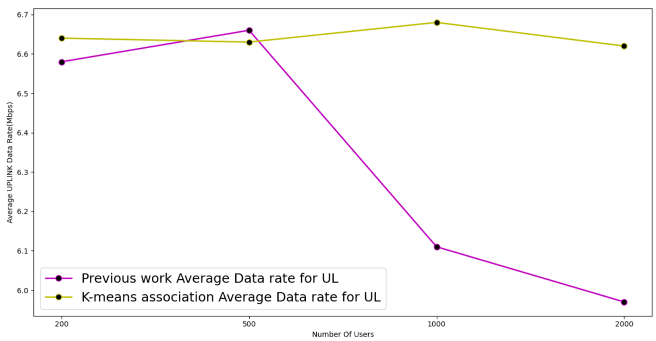

Fig. 7. Average DRs for previous work over K-means association as UL metric.

reduce energy consumption both for the UEs and for the network.

Better resource utilization can lead to more idle time for BSs which, combined with energy saving techniques (like BS sleep mode), can minimize energy waste. Taking into account multiple metrics for user association can lead to complex algorithms that use significant resources and have prohibitive time complexity. Clustering can ensure better real time performance for our network, which in turn is critical, especially in extreme congestion scenarios.

VI. Conclusions and future work

From the produced results, we see that using clustering to complement existing mechanisms in computer networks is not only important but it can actually improve overall network performance. Base Stations have problems coping with the rising amount of network users. Using clustering to substitute allocation with the use of network metrics, produced extremely positive results that are comparable with traditional approaches. We achieved homogeneous user distribution across the clusters and higher association rates. We conclude that our approach can play a key role to more stable, energy efficient and congestion-proof networks, especially in scenarios that demand real-time decision making.

Using clustering and other techniques based on machine learning is a matter of extreme scientific importance. Clustering can be a part of more research, not only limited to user association, but extending to better resource distribution as well. Applying other techniques should focus on improving real time decision making on networks, especially with the rise of Internet of Things, where more and more devices are expected to connect to the network. These techniques should be studied alongside current mechanisms and their time complexity and performance should be improved. Future work should take into account the massive increase in connected users and focus on ensuring quality of service before improving network performance.

References

[1] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[3] G Phanendra Babu and M Narasimha Murty. A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm. *Pattern recognition letters*, 14(10):763–769, 1993.

[4] Hamidreza Boostanimehr and Vijay Bhargava. Unified and distributed qos-driven cell association algorithms in heterogeneous networks. *Wireless Communications, IEEE Transactions on*, 14, 05 2014.

[5] Leon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. In *Advances in neural information processing systems*, pages 585–592, 1995.

[6] Christos Bouras, Georgios Diles, and Rafail Kalogeropoulos. An energy efficient mechanism for downlink and uplink decoupling in 5g networks. In *International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 241–252. Springer, 2019.

[7] Zebing Feng, Wei Li, and Wenping Chen. Downlink and uplink splitting user association in two-tier heterogeneous cellular networks. *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pages 4659–4664, 02 2015.

[8] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[9] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[10] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[11] J.M. Mom and Cosmas Ani. Application of self-organizing map to intelligent analysis of cellular networks. *ARPN Journal of Engineering and Applied Sciences*, 8:407–412, 06 2013.

[12] S. Sun, K. Adachi, P. H. Tan, Y. Zhou, J. Joung, and C. K. Ho. Heterogeneous network: An evolutionary path to 5g. In *2015 21st Asia-Pacific Conference on Communications (APCC)*, pages 174–178, 2015.

[13] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys Tutorials*, 21(4):3072–3108, 2019.

[14] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang. Machine learning for networking: Workflow, advances and opportunities. *IEEE Network*, 32(2):92–99, 2018.

[15] Francesc Wilhelmi, Sergio Barrachina-Muñoz, Boris Bellalta, Cristina Cano, Anders Jonsson, and Vishnu Ram. A flexible machine learning-aware architecture for future wlans, 2019.

[16] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[17] J. Yang, C. Wang, X. Wang, and C. Shen. A machine learning approach to user association in enterprise small cell networks. In *2018 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 850–854, 2018.