applied sciences

MDPI

Article

Cooperation Based Proactive Caching in Multi-Tier Cellular Networks

Fawad Ahmad [1], Ayaz Ahmad [1], Irshad Hussain [2,*], Peerapong Uthansakul [3,*] and Suleman Khan [2]

[1] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Wah Campus, Wah Cantt 47040, Pakistan; fawadkhan@uetpeshawar.edu.pk (F.A.); ayaz.ahmad@ciitwah.edu.pk (A.A.)
[2] Faculty of Electrical & Computer Engineering, University of Engineering and Technology, Peshawar 25000, Pakistan; sulemankhan@uetpeshawar.edu.pk
[3] School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
* Correspondence: ee.irshad@gmail.com (I.H.); uthansakul@sut.ac.th (P.U.)

Received: 23 April 2020; Accepted: 11 August 2020; Published: 4 September 2020

Abstract: The limited caching capacity of the local cache enabled Base station (BS) decreases the cache hit ratio (CHR) and user satisfaction ratio (USR). However, Cache enabled multi-tier cellular networks have been presented as a promising candidate for fifth generation networks to achieve higher CHR and USR through densification of networks. In addition to this, the cooperation among the BSs of various tiers for cached data transfer, intensify its significance many folds. Therefore, in this paper, we consider maximization of CHR and USR in a multi-tier cellular network. We formulate a CHR and USR problem for multi-tier cellular networks while putting major constraints on caching space of BSs of each tier. The unsupervised learning algorithms such as K-mean clustering and collaborative filtering have been used for clustering the similar BSs in each tier and estimating the content popularity respectively. A novel scheme such as cluster average popularity based collaborative filtering (CAP-CF) algorithm is employed to cache popular data and hence maximizing the CHR in each tier. Similarly, two novel methods such as intra-tier and cross-tier cooperation (ITCTC) and modified ITCTC algorithms have been employed in order to optimize the USR. Simulations results witness, that the proposed schemes yield significant performance in terms of average cache hit ratio and user satisfaction ratio compared to other conventional approaches.

Keywords: caching; cooperative network; multi-tier cellular network; content popularity; machine learning

1. Introduction and Background

Recently, smart phone usage has enormously increased which results in exponential growth of mobile data traffic. This mobile data traffic poses an unbelievable challenge in the radio resource demand [1,2]. CISCO predicts that mobile traffic will increase eight times from 2015–2020 [3] and would reach 30.6 Exabyte per month by the year 2020 [4]. It is observed that a big chunk of increased data traffic is due to duplicate downloads of famous videos files from the remote server [5]. In the recent era, mobile media users are extensively sharing information and their thoughts via cross net. The mobile social media services such as Facebook and Twitter have also increased the data traffic exponentially [6]. This unbelievable mobile data traffic has brought a burden on the communication networks as these are limited in transmission rate and hardware capabilities. Also the rapid growth of the content on the Web, for example cossetting articles, games or other type of entertaining contents such as videos, audios, etc. have now started problems regarding quality and down loadings for the

users [7,8]. Thus, proactive caching performs a pivotal role in the upcoming 5G technology. Caching the popular files on a nearby base station and then serving the users from the local cache are now becoming advantageous due to reducing latency and the data traffic load, especially on the back haul.

On the other hand, machine learning tools bring a new research area for the modeling and prediction of the user behavior for proactive caching decisions [9]. Recently, the stupendous success of deep learning methods in pattern recognition, images, and natural language have attracted the operators to use them for network optimization [10,11]. Different machine learning tool can extract information in the data traffic and cache the contents at the local base station [10]. For example, extreme learning machine predicts the popularity of the contents based on the different meta features of the contents [12]. Unsupervised learning such as K-mean clustering and collaborative filtering methods are employed to cluster the users with similar crossest for the contents and then cache the popular contents accordingly [11,13]. Similarly adaptive caching scheme optimizes the network performance thereby decreasing the delay and back haul load [12,13]. In [14], the author suggested the multi-armed bandit algorithm to estimate content popularity and the authors in [15] proposed an extreme-learning machine algorithm for estimation of the future content popularity.

Similarly, reinforcement learning methods are employed in cache enabled networks [16–18]. For example, in [16], a multi-step return actor-critic architecture which is reinforcement learning agent is used for optimizing the cache hit ratio. The information content can be extracted from the data using a (LSTM) deep learning [19]. The contents are cached if the crossest of the data is known. This decreases the service latency. These contents are most likely requested by the users.

Deep CachNet is a proactive caching framework in cellular network using deep learning algorithm [20]. In this caching scheme, a huge amount of data is collected from the end mobile devices of users connected to SBSs. In [21], the author solved the problem of context-aware data caching in the heterogeneous small cell networks using deep learning algorithm. The author in [22], dealt with the problem of minimization average energy cost in cache enabled networks, especially in limited cache memory nodes, using reinforced learning algorithm. Similarly multi-agent multi-armed bandit can be applied to the caching problems regarding the device to device communication. Such design applied Q-learning in order to coordinate the caching decisions [23].

The author of paper [24], strived to jointly minimize the average transmission delay in cache enabled networks by jointly considering scheduling and the caching problem using reinforced leaning and deep learning. In [25], the author optimized multimedia service in 5G caching networks based on the semantic information of user. Content popularity is estimated by the singular value decomposition (RSVD) which is based on collaborative filtering (CF). In [26], the author optimized the cache hit ratio, and a decision matrix is formed from a huge available data. In paper [27], they proposed a caching scheme for the ultra-dense networks. The backhaul load minimization problem is solved, but not the k-Nearest Neighbors (k-NN) classification.

In this article, we proposed a framework for content caching and user satisfaction ratio using cross-tier and intra-tier cooperation among the base stations. Our prominent contributions are:

- Introducing a data caching system, where data is cached in every tier of the network, thereby, optimizing the CHR and USR utilizing classical method of collaborative filtering and K-mean clustering.
- The requests for different contents are received and a popularity matrix is formed in each tier, which is highly sparse in nature. Proactive caching is done by creating clusters of base stations using the unsupervised K-mean clustering in each tier and fill half of the cache memory with the most popular contents in the cluster and rest of vacant cache is occupied by the contents that will be requested in future predicted by collaborative filtering C.F.
- Presenting two novel methods for the user satisfaction rate. The first method deals with the clusters made by K-mean clustering, based on contents requests, in each tier and content caching is done by C.F while user satisfaction is achieved thorough cross-tier and intra cooperation among the different tiers.

- In the second method, initially contents are cached in all tier based on C.F algorithm. Then clusters are created using K-mean algorithm while taking all the base station, irrespective of which tier they belong to. The cluster formed may have base stations of different tiers. The second method is faster one as requests do not require intra-tier and cross tier flow for the fulfillment of their content satisfaction.
- Extensive simulations are conducted with different contents and their requests depends on the mode of the users to elaborate the effectiveness of the proposed algorithms. It is shown that by using learning algorithms, caching the data on base station in the different tier, considerably improved the cache ratio and the user satisfaction ratio.

2. System Model

Suppose a modern cache enabled multitier cellular network consisting of T tiers designated as $\mathcal{T} = \{1, \ldots, T\}$ of base stations of M number. The set of base stations could be denoted by set $\mathcal{M}$ which is equal to {1, … ,M}, and it could easily serve N number user terminals (UTs) from the set $\mathcal{N} = \{1, \ldots, N\}$ which are installed in a certain geographical area that consists of residential area, factories, colleges, schools, etc. As the network is multi-tier so it comprises of pico BSs, micro BSs, and a single macro BS to serve greater number of users. Each BS m of corresponding tier T is equipped with its limited physical storages S_{mt}. Then total the storage capacity of the network is denoted as $S = \{S_{11}, S_{21}, \ldots .. Sm_1, S_{12}, S_{22}, \ldots \ldots S_{13}, S_{23} \ldots ., S_{m3, \ldots \ldots \ldots .. SmT}\}$.

All the BSs are connected through limited capacity optical fiber links. When the users request the contents and if, the contents are cached in the associated base station then it is served otherwise it is routed to the other base station in the same tier or another tier or the contents are fetched from the content servers via limited capacity back haul. The users request a variety of contents such as music, movies, breaking news, face book, twitter, technical books, notes, etc. from a content library F = {1, … , … ..,F}, in this case each content f (of the library) are of L(f)Byte size whereas, bit rate requirement of B(f) M Byte/s. The content server caches F Bytes of data in its storage. In such a scenario, each base station caches the selected contents proactively from the library F and this happens during less (network) load or off-peak hours. Let $P_t(t') \in \mathbb{R}^{M \times F}$ is the content popularity matrix which indicates the history of frequently requested contents at certain time say time t', where each coefficient $p_{mt, f}$ (t') shows rate of requesting content f received by the BS_{mt} of tier t. In fact, the matrix P_t (t') is the local content popularity obtained at base station BS_{mt}, whereas the global popularity distribution of all the contents is described by the Zipf distribution PF(f), $\forall$ f ∈ F i.e., lesser contents have greater popularity and greater contents are less popular. Moreover, it is assumed that all Small Base Stations of all tiers have a content popularity matrix which spans over T' time slots (and is stationary by nature, hence we could easily represent P (t') by P. By utilizing the tools obtained from collaborative filtering and unsupervised machine learning, a proactive caching procedure is being proposed which employs *P* for determining the content which is supposed to be cached. To witness this relation, let us define the cache decision matrix of BSs as X_t (t')∈$\{0,1\}^{M \times F}$, where the entry $x_{mt,f}$ takes 0 if the f_{th} content is not cached at m_{th} BS at tier t in time t' and 1 otherwise. We suppose that content caching is carried out in off-peak hours, therefore X_t (t') is designated by a cache decision matrix X_t which does not vary in these peak hours. The second decision variable is Y which is user satisfaction ratio parameter whose value is one if request is served other wise 0.

2.1. Caching Model

In our model, every tier has its own users and they have their demands according to their preferences. In particular, we consider such caching system where users download different types of content, e.g., a university student will have to down load books, lecture notes, and video lecture relevant to their fields. People interested in current affairs will watch breaking news. We consider different types of files and the BS that receive request for similar data will be virtually clustered, which will discussed in detail later on. In our model the caching capacity of pico BS is lesser compared

to micro BS and macro BS. On the other hand, a macro base station will cache more files as compared to micros and picos. Similarly, the number of users for pico is lesser as compared to micro and macro. In the multi-tier system, there will be more pico then micro and macro. As the capacity is limited, only popular content can be stored. Thus, the base station will offload the traffic, thereby, reducing traffic on the back haul and improve the quality of services.

Let us consider a base station BS_m receives a set of requests $R(m)$ for certain data during time interval T' seconds. These requests are Zipf-like distribution in nature. Let X_t be the cache decision matrix that depends on the popularity matrix P of the contents and $C(m)$ is the set of cached contents. Then the cache hit ratio is given in Equation (1).

$$\mathrm{CHR} = \left| \frac{\sum_m C(m) \cap R(m)}{\sum_m R(m)} \right| \tag{1}$$

In a t-tier wireless communication system, each tier will require its cache to be filled by the requested contents. The overall or the average cache hit ratio of the multi-tier system can be modelled, as:

$$\alpha = \frac{\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left| C(mt) \cap R(mt) \right|}{\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left| R(mt) \right|}. \tag{2}$$

Each tier has its own users and cache decision matrix X_t which depends on the popularity matrix P_t (t') $\in \mathbb{R}^{M \times F}$:

$$X_{mt} = \begin{cases} 1 \text{ if content f is cached at } Bsm_t \text{ of tier t}, \forall m \in \mathcal{M} \\ 0 \text{ otherwise} \end{cases}.$$

2.2. User Satisfaction Ratio Model

In such a model, the user can get the data from: (1) locally associated BS cache storage unit, (2) remote BS cache storage unit but within the same tier, (3), remote BS cache storage unit but in another tier, and (4) content server. The remote content unit is the other BS that caches the user's required contents but is not associated to the user. It means there is cooperation among the BS not only within the tier but also across the tiers. Also, within the tier there will be intra-cluster and cross-cluster cooperation which will be discuss in Section 6 in detail.

The average user satisfaction can be modelled as:

$$N(y) = \frac{1}{R_\tau} \sum\nolimits_{r \in R_\tau} Y_r = \text{Average request satisfaction ratio} \tag{3}$$

where:

$Y = \{y_1, y_2, y_3, y_4 \cdots y_{RT}\}$
$R_\tau = \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left| R(m_t) \right|$
$y_r = \begin{cases} 1 \text{ if request } r \text{ is served} \\ o \text{ otherwise} \end{cases}$

In order to serve the user content request, two methods are proposed which will be discussed in Section 5.

3. Problem Formulation

Given this model, our prime goal is to develop a proficient caching approach to improve the cache hit ratio and the user satisfaction ratio based on the user content requests and prediction of those contents that will be requested in future. To achieve this goal, we formulate an average cache hit ratio and user satisfaction ratio optimization problem whose objective is to optimally cache the popular contents on BS of each tier in a such way that user find the intended data instantly and,

hence, increasing the cache hit ratio and user satisfaction ration in each tier of the cache enabled multi-tier network.

$$\max_{X_t, Y} \ (\alpha + \eta(y)) \tag{4}$$

Subject to constraints:

$$X_{mt,f} \in \{0,1\}, \ \forall_{mt} \ , \ t, \ f \tag{5}$$

$$Y_r \in \{0,1\}, \ \forall_r \tag{6}$$

$$\sum\nolimits_{f \in \mathcal{F}} L(f) X_{mt,f} \leq S_{mt}, \ \forall_{mt, f \in F} \tag{7}$$

The first term of the objective function accounts for the average cache hit ratio of the multitier cache enabled network. The second term of Equation (4) shows the average satisfaction ratio. In other words, the system is trying to optimize the usage of the limited cache of the BS in such a way that the user can easily find its requested content from its associated or remote BS, thereby, increasing the downloading speed and decreasing the burden on the backhaul. The caching capacity required for each BS is indicated by the system constraint (7).

This means that the cache entity capacity is bounded by corresponding capacity limit. Constraint (5) and Constraint (6) show that $X_{m,t,f}$ and y_r are binary variables, respectively. Therefore, the problem (1) is a constrained integer programming problem and is generally NP-hard which is very challenging task to solve. Also, storing the most popular data in the cache can increase the number of users which are recipient of the cached data. So caching the data that favours the users in limited cache is a challenging job. It means that such an intelligent system is required to predict the popularity of the data and cache them properly. To achieve this, machine learning plays a pivotal role. Within this frame work, based on past requests, we will first use K means clustering for clustering the BSs that receive similar requests and for predicting the future data, collaborative filtering is used. This will enhance the cache hit ratio. Similarly, for maximizing user satisfaction ratio, two types methods are employed which are based on mutual cooperation among the tier and will be discussed in Section 6.

4. Collaborative Filtering for Content Prediction

The optimization problem in (1) is a challenging issue to resolve because the cache hit ratio and user satisfaction ratio depend on the proactive caching which in turns depends on prediction of the contents to be requested in future. Since the BS need to be aware of the content popularity in advance. Hence, The optimization problem is very difficult to solve using conventional optimization algorithms as they are unable to predict the content popularity in advance especially for those BSs which are in dense environment where there are a lot number of users and limited cache size. This reason renders the optimization problem in (1) to be very challenging as the BSs got very limited information about the users' preferences. To address this challenge, we propose a K-means clustering algorithm for clustering the BSs which receive the similar request for the contents. After this, the contents, whose request are not received on any BS but it is expected that requests will be received in future, is predicted by CF.

4.1. K-Mean Clustering

In this subsection, the k-mean algorithm is applied to the past requests received on the BSs in each tier. K-mean clustering is an algorithm that tries to partition data set into K predefined distinct non-overlapping subgroups clusters where each data point belong to only one group. The content popularity matrix gives the frequency of the request on the BS. Hence, those BSs that receive similar requests for the contents, are grouped together to form a cluster. As micro BSs comprise of greater number of users as compared to pico BSs. Therefore, the request of the former will be greater as compared to the later. Also, there will be more pico BSs as compared to micro BSs and, hence, the cluster of pico cells will be more crowded compared with the micro cells. The similarity between the two base stations is based on the Euclidean distance. The algorithm has the following steps:

(1) First of all, specify the number of the cluster in each tier. Let us have K clusters $H_1, \ldots \ldots ., Hk$;

(2) Initialize centroids by first shuffling the data set which is the past requests on the base stations and then randomly selecting K data points for the centroids without replacement. Random Selection of K SBSs from the rows of P is carried out, which are represented as $c_1{}^{(1)}, \ldots \ldots .., c_K{}^{(1)}$;
(3) Keep iterating until there is no change of data points in the cluster;
(4) Compute the sum of the squared distance between data points and all centroids:

$$H_m^{t'} = \left\{ p_i : \|p_i - c_m^{t'}\|^2 \leq \|p_i - c_n^{t'}\|^2, \ \forall \mathrm{n}\ 1 \ \leq \ \mathrm{n} \ \leq \ \mathrm{k} \right\}$$

where p_i is the i-th row of p and is assigned to exactly one $H^{t'}$;
(5) Assign each data point to the closest cluster (centroid); and
(6) Compute the centroids for the cluster by taking the average of the all data points that belong to each cluster. So the new means to the centroid of the new cluster is $c_m^{(t+1)} = \frac{1}{\left|H_m^{(t)}\right|} \sum_{p_i \epsilon H_m^{(t)}} p_i$.

The new centroid is subtracted from the old one and the algorithm repeats until and unless this value becomes less than a threshold. Finally, we cluster K clusters. This process is taking place in each tier. The BS in each cluster has similar interest.

4.2. Content Prediction

In this subsection, we formulate the CF based content request distribution prediction algorithm. The prediction is based on the idea that the people who have agreed on something in the past may also agree in the future. It means that the people who have similar tastes may get best recommendations from one another. We can apply this idea to our multi-tier cellular caching network. If we find the similarity between the two BSs then, we can predict the popularity of the contents on the BSs that they have not been requested by now but will most probably be requested in the future.

Let us consider $A^t(mi)$ as the set of the content requested by the users of the $\mathrm{BS_{t,mi}}$ and $A(mj)$ be the set content requested by the $\mathrm{BS_{t,}mj}$. $A^t(f)$ be the set of the BSs on which file f is requested. Then to find the similarity between two BSs we have to find the weighted matrix $W_{i,j}$ which is given as:

$$\mathrm{W^T} = \begin{pmatrix} 1 & W_{12}^t & \cdots & W_{1n}^t \\ \vdots & & \ddots & \vdots \\ W_{1n}^t & & \cdots & 1 \end{pmatrix}$$

The matrix is obtained from the following Equation [26]:

$$W_{mi,mj}^t = \sum_{f^t \epsilon A^t(mi) \cap A^t(mj)} . \frac{\frac{1}{\log(1+\left|TA^t(f)\right|}}{\sqrt{\left|A^t(mi)\right|\left|A^t\ (mj)\right|}} \tag{8}$$

Now, the probability that the content f is requested by the $\mathrm{BS_{mi}}$ is obtained as:

$$\mathrm{P^t\ (mi_{,f})} \ = \sum\nolimits_{m_j^t \epsilon Z(mi,G) \cap A^t(f)} W_{mi,mj}^t P_{mj,f}^t \tag{9}$$

where, $Z(mi, G)$ contains G SBSs which are the most popular set with mi. $\mathrm{P_{mj,f}}$ is the entry of the matrix P. After calculating $\mathrm{P_{(mi,f)}}$ of each content for the SBS mi, we can cache the requested content on the BS based on p(mi,f) in decreasing order.

5. Proposed Framework

The prediction of the content requested in future must now leverage to determine which content is to be cache at the BS in each tier. Hence, cache hit ratio is maximized by not only placing the required content, requested by the users now but also it caches the content to be requested in future. If a user

finds the requested content in its associated base station then the cache is a hit or otherwise a miss. However, due to limited capacity of the base station, cache hit ratio cannot achieve its maximum value 1, but if the requested contents and of course, the future requested contents are placed then optimum value of the cache hit ratio can be obtained. The average cache hit ratio is maximize by taking all tier into account.

In this section, we formulate the proposed Algorithm 1 to the problem in (3). First of all we clustered the BSs in each tier based on past requests and initialize the capacity matrix S_{mt} and cache decision matrix X_t. Then in cluster each, we find the average request of the files and sort them in descending order. Now, fill half of the cache of each BS with the contents whose average requests is greater and set the entry in the Xt matrix equal to 1 and update the cache size and repeat it till size of the BS reaches 50% of the maximum size. The rest of the cache is filled by the contents that will be requested in future predicted by CF. Weights similarity are find out using Equation (8) and P^t ($m_{i,f)}$ using Equation (9) and are sorted in descending order. After doing this, if file f is not in the BSm and the BS current cache and the size of the file is less than maximum capacity than cache it and set the entry Xm,j of the cache decision Xt equal to 1 and update the current cache of the SBS m. Repeat this algorithm until it touches the SB maximum caching capacity. Repeat this algorithm for all tiers except Macro BS. Finally, cache decision matrix Xt is obtained. Subscript/superscript 1, 2, and 3 show tier 1, tier 2, and tier 3, respectively.

Algorithm 1 Contents Caching

Input: P^1, P^2, P^3, S_1, S_2, S_3, L^1, L^2, L^3, F^1, F^2, F^3
Output: X_t
1. Cluster the base stations k = 1, 2, 3 . . . .Hk based on past requests in each tier except Macro cell
2. **For** Pico BS **do**
3. Initialize the cache size {S^1} and caching matrix {X1}
4. [a,b] = **SORT** AR. *//where AR is the average popularity of the most requested file. Sort AR by descending order.*
5. **For** BSm in H_K **do**
6. **For** i = 1 . . . .F^1 **do**
7. Gets index of ith highest AR in cluster k,
8. j ←bi
9. **if** $L^1j + \hat{s}_{m1} \leq 0.5 * Sm1$ **then**
10. X m,j = 1
11. $\hat{s}_{m1} = L^1j + \hat{s}_{m1}$
12. **else**
13. Break
14. **End**
15. **End**
16. Calculate the similarity matrix i.e., Wi,j
17. Find the P(mit,f)
18. [a,b] ← **SORT** (P(mit,f)) *//Sorts it by descending order, returns a, b as ordered values and indices.*
19. **For** BSm in H_K **do**
20. **For** i = 1 . . . .F^1 **do**
21. **if** $L^1f + \hat{s}_{m1} \leq Sm_1$ **then**
22. **if** Xm,f ≠ 1 **then**
23. Make Xm,f = 1
24. $\hat{s}_{m1} = L^1f + \hat{s}_{m1}$
25. **else**
26. continue
27. **else**
28. break
29. return X_1
30. **End**
31. **For** Micro BS **do**
32. Repeat 3–30 and replacing all superscripts and subscripts 1 with 2 and return X_2
33. **For** Macro BS
34. Initialize S_3
35. [a,b] = **SORT** (CP) // Sort Content Popularity (CP) by descending order.
36. **For** i = 1 . . . . . F^3 **do**
37. get index of the highest CP
38. j ←bi
39. **if** Lj + S3 ≤ S3 **then**
40. X m,j = 1
41. S3 = $L^3j + S_3$
42. **else**,
43. Break
44. return X_3
45. **End**

6. User Satisfaction Ratio

The second goal of our research work is to optimize the user satisfaction ratio. If a user requests for a particular content then it is served by the local base station. If the associated base station does not have that content, then it is fetched from neighbouring base stations within the same cluster, but if it is

still not cached in the same cluster, then it is brought from another cluster in the same tier. Also, if it is not in the same tier then it is served from higher tiers, otherwise, it is fetched from the main serve. The above process is a time consuming and require energy. So besides this method, a faster and less energy hungry method is also presented.

6.1. Intra-Tier and Cross-Tier Cooperation (ITCTC)

The objective is to observe the user demand instantly. For this, Algorithm 2 is proposed which is based on cooperation among the BSs regarding the content sharing. This cooperation is not limited to the intra-tiers and, hence, associated BS of one tier can demand the requested content from the other tiers.

6.1.1. Intra-Tier Cooperation

This cooperation can be further divided into two types:

(1) Intra-cluster cooperation and
(2) Cross-cluster cooperation

Intra-Cluster Cooperation (ICC)

Clusters are formed in each tier and contents are cached accordingly. Figure 1 depicts clusters formation in tier 1 and tier 2. The user request may be fulfilled by the nearby associated base station or the contents may be fetched from other BSs within the cluster. The cluster comprises of similar contents on the BS, but probably some BSs have different contents within the cluster. The user will send a request to the local base station and if it is not satisfied then the request will be forwarded to the other BSs within the cluster.

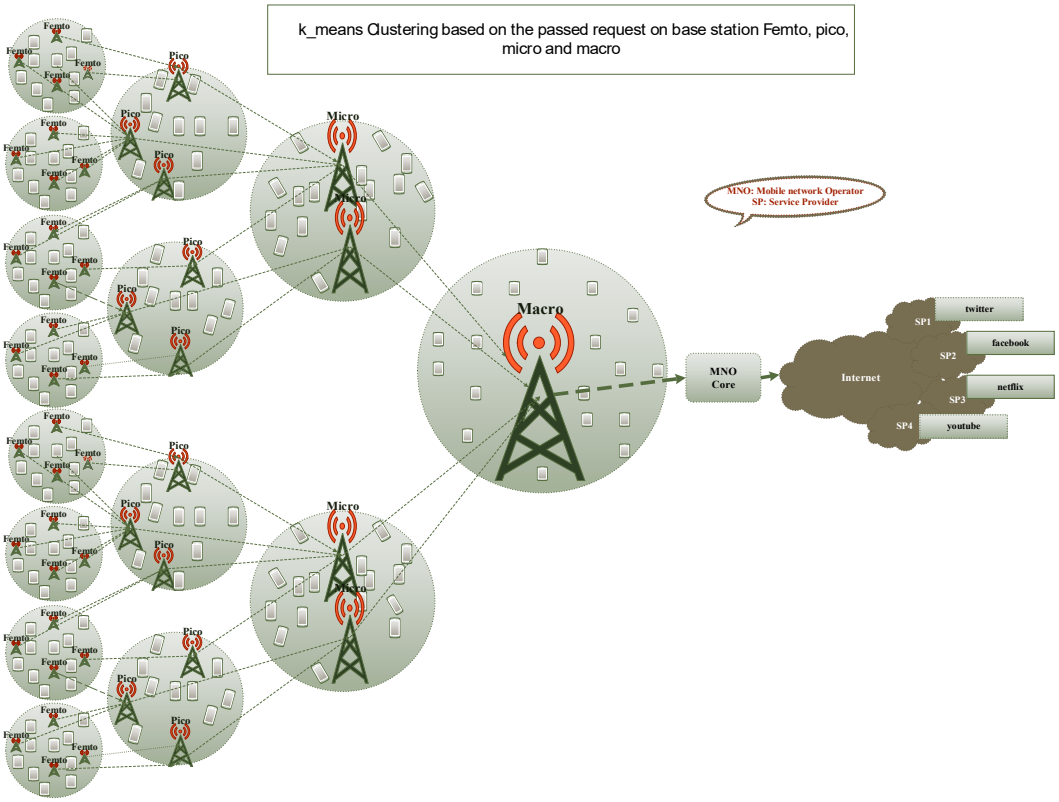


Figure 1. Cluster formation in each tier using K-mean clustering.

Cross-Cluster Cooperation (CCC)

If a user does not find its requested contents in the associated cluster then the request is forwarded to the other clusters of the same tier. This is cross-cluster cooperation.

6.1.2. Cross-Tier Cooperation (CTC)

The cross-tier cooperation means that if the requested contents is not found in the local tier then the contents should be forwarded to other tiers for example, if a user from Pico BS does not get the contents from any pico BS then its request will be forwarded to the micro BSs and so on. The micro BSs are also clustered and request is forwarded from one cluster to another cluster.

Algorithm 2 Intra-tier and Cross-tier Cooperation

```
Input: Xt
Output: Yr
1.  Cluster the base stations into k = 1, 2, 3 . . . .Hk based on past requests in each tier except macro cell
2.  Cache the data using CF in each cluster of each tier
3.  For pico BS do
4.      If request arrives at pico BS then
5.          Serve the request at associated pico BS
6.      Else if
7.          Send the request to neighboring BS but within the same cluster
8.      Else if
9.          Send the request to pico BS which is in any of the neighboring clusters (Cross- cluster
cooperation)
10.     Else if
11.         Send the request to 1st Cluster of micro BS
12.     Else if
13.         send it to neighboring clusters of micro BS
14.       Else if
15.         send it macro BS
16.       Else
17.         send it content server
18. End
19. For micro BS do
20.       If request is covered in micro BS then
21.         Serve the requests at associated micro BS
22.     Repeat steps 5–9
23.     Else if
24.         send it macro BS
25.     Else
26.         send it content server
27. End
28. Repeat for all requests
29. Return Y
```

6.2. Modifed Intra-Tier and Cross-Tier Cooperation (MITCTC)

The Algorithm 2 is time consuming and requires extensive signalling and more energy because unserved request has not only to be passed through all cluster of each tier but also from one tier to another tier on the other hand Algorithm is better one regarding energy overhead and E-E delay in executing the existing frame work. Therefore, Algorithm 3 is used which initially caches popular contents on each BS utilizing CF. Then clusters are formed based on the similar files cached in all tiers. Hence, a cluster may have BSs of Pico cells and micro cells if they have similar contents. Now, if a

request is received by a local BS and if it is not fulfilled, then it is moved to the other BSs within the cluster. There may be a fair chance that the requested contents may be found on the micro BS or Pico Base station of the cluster as the cluster now has micro BSs and pico BSs. These clusters are shown in Figure 2. The proposed Algorithm 3 is:

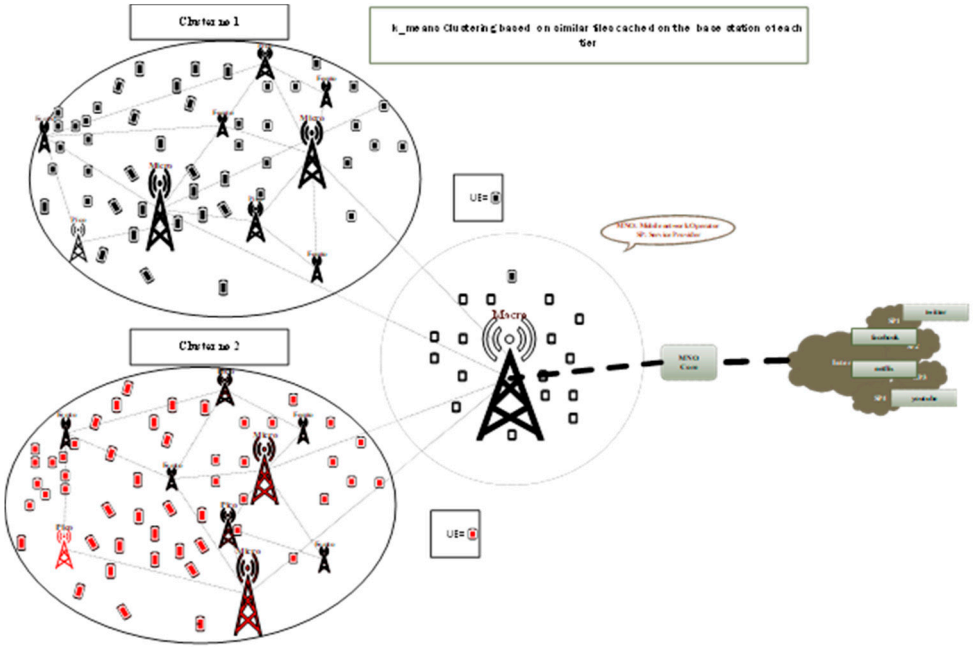


Figure 2. Cluster formation considering all tiers.

Algorithm 3 Modified Intra-Tier and Cross-Tier Cooperation

Input: X_t
Output: Y_r
1. Cache the data using CF in each tier based on past requests.
2. Cluster all BSs from all the tires(including pico, micro and macro) based on similar cached data files
3. **For pico BS do**
4. **If** request arrives at Pico BS then
5. Serve the request at associated pico BS
6. **Else if**
7. Send the request to neighboring BS (that may be of Pico, micro or macro) but within the same cluster
8. **Else if**
9. Send the request to another cluster
10. **Else**
11. Send it to content server
12. **End**
13. **For micro BS do**
14. **If** request arrives at micro BS then
15. Serve the request at associated micro BS
16. Repeat steps 6–12
17. **For macro BS do**
18. **If** request arrives at macro BS then
19. Serve the request at associated macro BS
20. Repeat steps 6–12
21. **Repeat for all requests**
22. **Return Y**

7. Simulations and Results

In this section, we simulated the cache hit ratio and the user satisfaction ratio of multi-tier caching systems. The popularity matrix for tier 1 and the list of parameter used in this simulation is given in Tables 1 and 2, respectively. The requests for the contents on BS on each tier is different, e.g., some people like one content while others are interested in some different content. Similarly, the number of BS in each tier is different. Each content size is considered to be same and is 1. Each tier has its own data set. The data set compromises of training data and testing data. First, the collaborative filtering is trained by using training data which is the content popularity. Testing data is utilized to predict the missing data in the data set. The popularity matrix of each tier is highly sparse matrix because users have different modes of likeness. Based on the similarities among the data, clusters are formed in each tier. Let the clusters in tier 1 and in tier2 is 2. We have single macro BS.

Table 1. Popularity Matrix.

BS	**Number of Requests**			
B1	F1	F2	F3	F4
B2	3	6	6	0
B3	3	0	6	8
B4	0	6	0	9
B5	5	5	5	5

Table 2. List of parameters.

Parameters	**Tier 1**	**Tier 2**	**Tier 3**
No. of BS	30	20	1
No. of past request	5563	7636	700
No. of future request	2000	2300	500
Caching capacity	30	40	50

Figure 3a–c shows how the cache hit ratio varies as the caching capacity varies in the tier 1, tier 2, and tier 3, respectively. In these figures, we showed that the cluster average popularity with CF (CAP-CF) is far better than other methods such as ground truth method [26,28] and pure CF. In ground truth methods, only popular contents are cached, based on past requests, but there is no future prediction of the contents. As no clustering is formed in the ground truth method, no similarity between the BSs is observed, which results in poor utilization of the cache memory. On the other hand, in Pure CF method the contents which are not yet requested but hopefully requested in future, are predicted and are cached in descending order based on future popularity predication, without taking into account the present popularity of the contents. The simulations reveal that our proposed algorithm, which is based on the highest average popularity of the contents in the cluster and popularity of the contents to be requested in future, perform well by optimizing the cache hit ratio for each tier and also the average cache hit ratio for the whole multitier caching network. The main achievement in the proactive caching network is to cache the frequent requested content in the limited cache.

Figure 4a,b shows the proposed method 1 for the USR based on intra-tier and cross cooperation for two cluster of tier 1. Similarly, Figure 4c,d depicts the same method for tier 2. This novel method is compared with other methods such as CF, intra-cluster cooperation, and cross-cluster cooperation methods. In CF there is no cooperation among the BSs for the data sharing, while in intra-cluster cooperation the data sharing take place among the BSs with in cluster and in cross-cluster cooperation the content exchange take place among the clusters with in the tier. On the hand, the proposed scheme performs both intra-tier and cross cooperation in order to maximize the USR. The simulations suggest that the proposed method performs well as compare to all other algorithms due to cooperation among

all of the tiers. Figure 5 show the second method which is MITCTC. Simulations show that the novel method MITCTC utilizing CF is better than the ground truth.

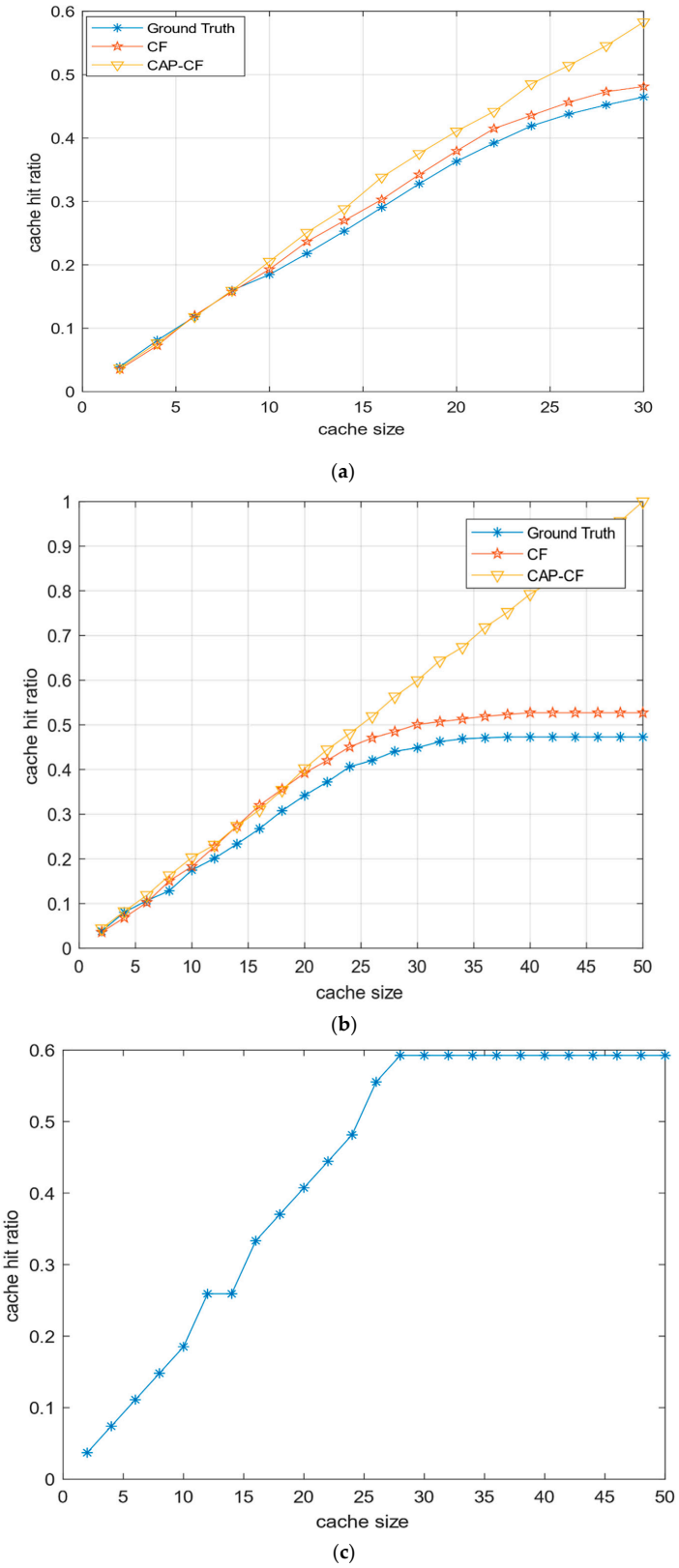
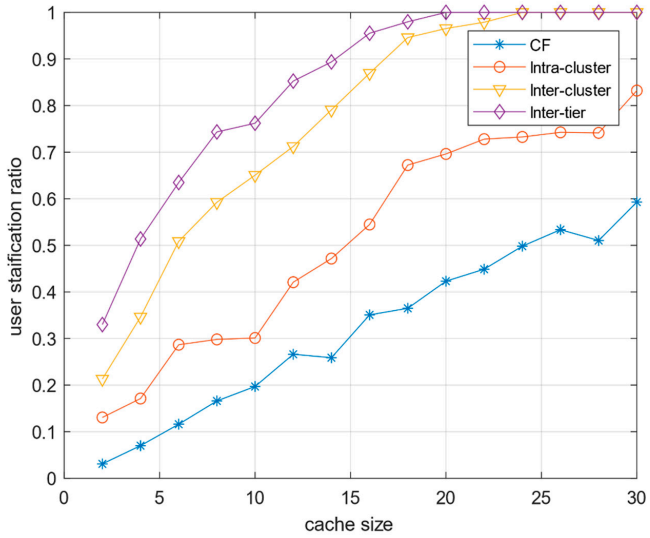
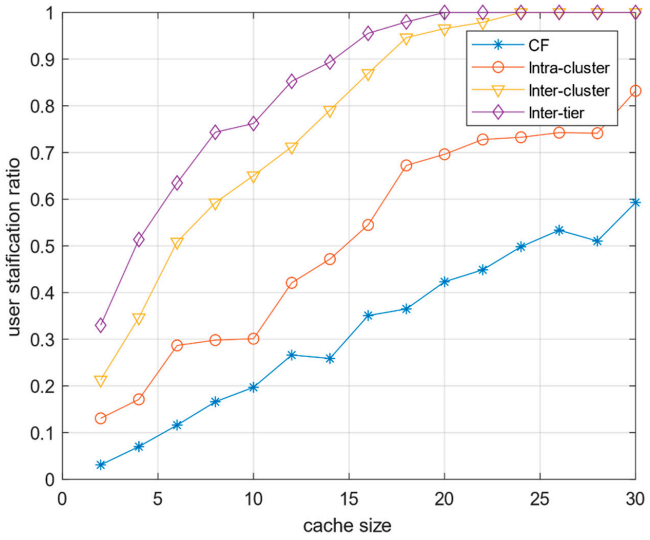


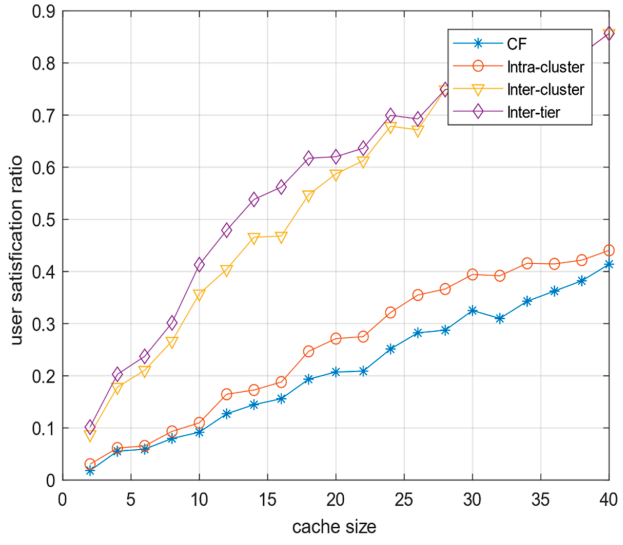
Figure 3. (**a**) Cache hit ratio versus cache capacity of pico cells. (**b**) Cache hit ratio versus cache capacity of micro cells. (**c**) Cache hit ratio versus cache capacity of macro cells.



(**a**)

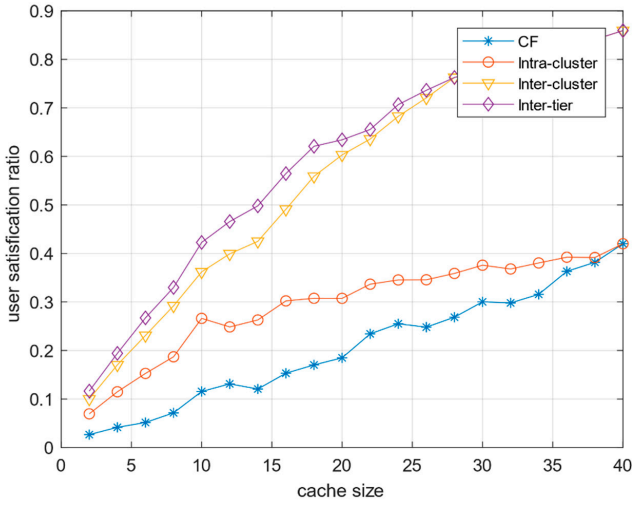


(**b**)



(**c**)

Figure 4. *Cont.*



(**d**)

Figure 4. (**a**): User satisfaction rate versus cache capacity in pico cell (cluster1 of tier 1). (**b**): User satisfaction rate versus cache capacity in pico cell (cluster 2 of tier 1). (**c**): User satisfaction rate versus cache capacity in micro cell (cluster 1 of tier 2). (**d**): User satisfaction rate versus cache capacity in micro cell (cluster 2 of tier 2).

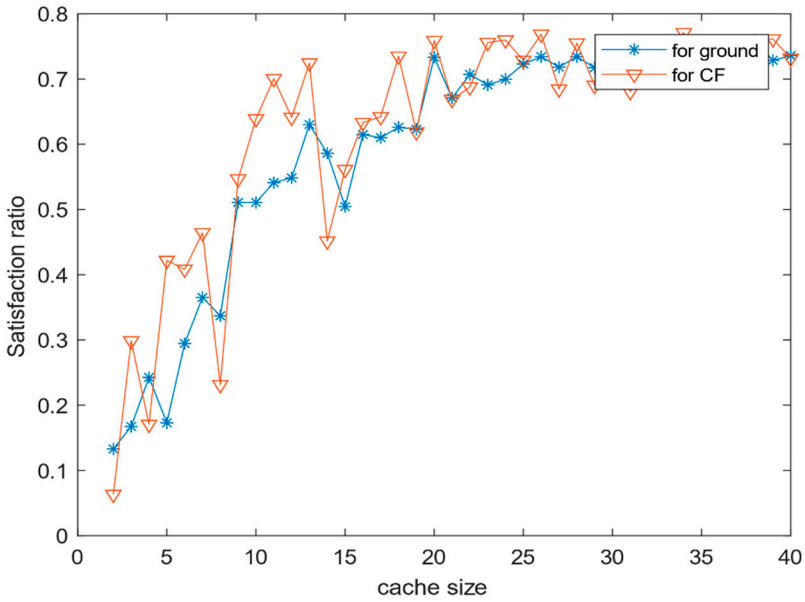


Figure 5. User satisfaction rate versus cache capacity in method 2.

8. Conclusions

In this paper, we presented popularity predicting caching policy for multitier cellular networks and two novel methods for user satisfaction rate based on intra-cluster, cross-cluster, cross-tier, and intra-tier cooperation among the BSs. We have formulated an optimization problem that seeks to maximize the average cache hit ratio and the user satisfaction rate. To solve this problem algorithms are developed which use machine learning tools such as unsupervised clustering and collaborative filtering and a heuristic approach for user satisfaction rate. Initially, clustered are formed in each tier based on past requests, received by each base station. Similarly, the contents which are not requested now, but most likely to be requested in future, are also predicted using the collaborative filtering and the average cache hit ratio is maximized. For maximization of user satisfaction rate, two methods are suggested. In the first method, we have ITCTC among the base stations. Intra-tier cooperation comprises of intra-cluster cooperation and cross-cluster cooperation in each tier. Similarly the second method, MITCTC is also

proposed. Simulation results have shown that the proposed schemes yields significant performance in terms of average cache hit ratio and user satisfaction ratio compared to conventional approaches.

Author Contributions: Conceptualization, F.A. and A.A.; methodology, F.A.; software, S.K.; validation, A.A.; formal analysis, F.A.; investigation, A.A. and F.A.; resources, I.H.; data duration, I.H.; writing—original draft preparation, F.A.; writing—review and editing, I.H., and F.A.; visualization, I.H.; supervision, P.U. and I.H.; project administration, I.H. and P.U.; funding acquisition, I.H. and P.U. All authors participated equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by SUT, Research and Development fund.

Acknowledgments: The authors are extremely thankful to Govt. of Pakistan and Thai.

Conflicts of Interest: The authors declare no conflict of crossest.

Abbreviations

The following abbreviation are used in this manuscript.

CHR	cache hit ratio
USR	user satisfaction ratio
BSs	Base stations
CAP-CF	cluster average popularity based collaborative filtering
ITCTC	intra-tier and Cross-tier cooperation
CF	collaborative filtering
UTs	user terminals
ICC	Intra-Cluster cooperation
CCC	Cross-Cluster Cooperation
CTC	Cross-Tier Cooperation
MITCTC	Modified Intra-tier and Cross-tier cooperation

References

1. Agiwal, M.; Roy, A.; Saxena, N. Next generation 5g wireless networks: A comprehensive survey. *IEEE Trans. Inform. Theory* **2016**, *18*, 1617–1655. [CrossRef]
2. Yu, F.; Krishnamurthy, V. Optimal joint session admission control in integrated WLAN and CDMA cellular networks with vertical handoff. *IEEE Trans. Mob. Comput.* **2007**, *6*, 126–139. [CrossRef]
3. Zhang, T.; Xingyan, C.; Xu, C. Intelligent Routing Algorithm Based on Deep Belief Network for Multimedia Service In Knowledge Centric Vanets. Available online: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visualnetworking-index-vni/complete-white-paper-c11-481360.html (accessed on 31 December 2019).
4. Zaman, S.; Hussain, I.; Singh, D. Fast Computation of Integrals with Fourier-Type Oscillator Involving Stationary Point. *Mathematics* **2019**, *7*, 1160. [CrossRef]
5. Breslau, L.; Cao, P.; Fan, L.; Phillips, G.; Shenker, S. Web caching and Zipf-like distributions: Evidence and implications. In Proceedings of the IEEE INFOCOM, New York, NY, USA, 21–25 March 1999.
6. Han, Z.; Hong, M.; Wang, D. *Signal Processing and Networking for Big Data Applications*; Cambridge University Press: Cambridge, UK, 2017.
7. Chen, Z.; Xu, J.; Tang, J.; Kwiat, K.; Kamhoua, C.; Wang, C. Gpu accelerated high-throughput online stream data processing. *IEEE Trans. Big Data* **2017**, *4*, 191–202. [CrossRef]
8. Dinc, E.; Ozger, M.; Ates, A.F.; Delibalta, I.; Akan, O.B. Crowdsourcing-based mobile network tomography for xg wireless systems. In Proceedings of the IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 346–351.
9. Cenk Gursoy, M.; Velipasalar, S. A deep reinforcement learning-based framework for content cachingchen zhong. In Proceedings of the 52nd Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 21–23 March 2018.
10. Gao, T.; Chen, M.; Gu, H.; Yin, C. Reinforcement learning based resource allocation in cache-enabled small cell networks with mobile users. In Proceedings of the 52nd Annual Conference on Information Sciences and Systems (CISS), Qingdao, China, 22–24 October 2017.

11. Yanxiang, J.; Miaoli, M.; Mehdi, B.; Fuchun, Z.; Xiaohu, Y. A novel caching policy with content popularity prediction and user preference learning in fog-ran. In Proceedings of the IEEE Conference, Singapore, 4–8 December 2017.
12. Yu, Y.; Zhang, Z.; Yang, G.; Xiao, M. Minimum cost based clustering scheme for cooperative wireless caching network with heterogeneous file preference. In Proceedings of the IEEE ICC Communications QoS, Reliability and Modeling Symposium, Paris, France, 21–25 May 2017.
13. Chen, B.; Yan, C. Caching policy optimization for D2D communications by learning user. Information Theory. *arXiv* **2017**, arXiv:1704.04860.
14. Song, J.; Sheng, M.; Quek, T.Q.; Xu, C.; Wang, X. Learning based content caching and sharing for wireless networks. *IEEE Trans. Commun.* **2017**, *65*, 4309–4324.
15. Tanzil, S.S.; Hoiles, W.; Krishnamurthy, V. Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach. *IEEE Access* **2017**, *5*, 5870–5881. [CrossRef]
16. Gruslys, A.; Azar, M.G.; Bellemare, M.G.; Munos, R. The reactor: A sample-efficient actor-critic architecture. *arXiv* **2017**, arXiv:1704.04651.
17. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
18. Dulac-Arnold, G.; Evans, R.; van Hasselt, H.; Sunehag, P.; Lillicrap, T.; Hunt, J.; Mann, T.; Weber, T.; Degris, T.; Coppin, B. Deep reinforcement learning in large discrete action spaces. *arXiv* **2015**, arXiv:1512.07679.
19. Tsai, K.C.; Wang, L.; Han, Z. Caching for mobile social networks with deep learning: Twitter analysis for 2016 U.S. election. *IEEE Trans. Netw. Sci. Eng.* **2017**, *7*, 193–204. [CrossRef]
20. Rathore, S.; Jung, H.R.; Kumar Sharma, P.; Hyuk Park, J. DeepCachNet: A proactive caching framework based on deep learning in cellular networks. *IEEE Netw.* **2019**, *33*, 130–138. [CrossRef]
21. Ben Hassine, N.; Milocco, R.; Minet, P. ARMA based popularity prediction for caching in content delivery networks. In Proceedings of the IEEE Wireless Days, Porto, Portugal, 29–31 March 2017; pp. 113–120.
22. Hussain, I.; Ullah, M.; Ullah, I.; Bibi, A.; Naeem, M.; Singh, M. Optimizing energy consumption in the home energy management system via a bio-inspired dragonfly algorithm and the genetic algorithm. *Electronics* **2020**, *9*, 406. [CrossRef]
23. Jiang, W.; Feng, G.; Qin, S.; Tak Shing, P.; Yum Guohong, C. Multi-agent reinforcement learning for efficient content caching in mobile D2D networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 1610–1622. [CrossRef]
24. Wei, Y.; Zhang, Z.; Yu, F.R.; Han, Z. Joint User Scheduling and Content Caching Strategy for Mobile Edge Networks Using Deep Reinforcement Learning. In Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018.
25. Hao, H.; Xu, C.; Wang, M.; Xie, H.; Yifeng, L.; Wu, D.O. Knowledge-centric proactive edge caching over mobile content distribution network. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018.
26. Yan, N.; Gao, S.; Nan, L.; Pan, Z.; Xiaohu, Y. Clustered small base stations for cache-enabled wireless networks. In Proceedings of the 9th Cross National Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017.
27. Gao, S.; Li, P.; Pan, Z.; Liu, N.; You, X. Machine learning based small cell cache strategy for ultra dense networks. In Proceedings of the 9th Cross national Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017.
28. Ejder, B.; Bennis, M.; Zeydan, E.; Kader, M.A.; Karatepe, I.A.; Er, A.S.; Debbah, M. Big data meets telcos: A proactive caching perspective. *J. Commun. Netw.* **2015**, *17*, 549–557.